

Related titles

Aircraft System Safety: Military and Civil Aeronautical Applications

(ISBN 978-1-84569-136-3)

Airworthiness, Second Edition: An Introduction to Aircraft Certification

(ISBN 978-0-08-096802-5)

Commercial Airplane Design Principles

(ISBN 978-0-12-419953-8)

**Woodhead Publishing in Mechanical
Engineering**

Aircraft System Safety

**Assessments for Initial Airworthiness
Certification**

Duane Kritzing



AMSTERDAM • BOSTON • CAMBRIDGE • HEIDELBERG
LONDON • NEW YORK • OXFORD • PARIS • SAN DIEGO
SAN FRANCISCO • SINGAPORE • SYDNEY • TOKYO
Woodhead Publishing is an imprint of Elsevier



Woodhead Publishing is an imprint of Elsevier
The Officers' Mess Business Centre, Royston Road, Duxford, CB22 4QH, United Kingdom
50 Hampshire Street, 5th Floor, Cambridge, MA 02139, United States
The Boulevard, Langford Lane, Kidlington, OX5 1GB, United Kingdom

Copyright © 2017 Duane Kritzinger. Published by Elsevier Ltd. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording, or any information storage and retrieval system, without permission in writing from the publisher. Details on how to seek permission, further information about the Publisher's permissions policies and our arrangements with organizations such as the Copyright Clearance Center and the Copyright Licensing Agency, can be found at our website: www.elsevier.com/permissions.

This book and the individual contributions contained in it are protected under copyright by the Publisher (other than as may be noted herein).

Notices

Knowledge and best practice in this field are constantly changing. As new research and experience broaden our understanding, changes in research methods, professional practices, or medical treatment may become necessary.

Practitioners and researchers must always rely on their own experience and knowledge in evaluating and using any information, methods, compounds, or experiments described herein. In using such information or methods they should be mindful of their own safety and the safety of others, including parties for whom they have a professional responsibility.

To the fullest extent of the law, neither the Publisher nor the authors, contributors, or editors, assume any liability for any injury and/or damage to persons or property as a matter of products liability, negligence or otherwise, or from any use or operation of any methods, products, instructions, or ideas contained in the material herein.

Library of Congress Cataloging-in-Publication Data

A catalog record for this book is available from the Library of Congress

British Library Cataloguing-in-Publication Data

A catalogue record for this book is available from the British Library

ISBN: 978-0-08-100889-8 (print)

ISBN: 978-0-08-100932-1 (online)

For information on all Woodhead Publishing publications
visit our website at <https://www.elsevier.com/>



Working together
to grow libraries in
developing countries

www.elsevier.com • www.bookaid.org

Publisher: Joe Hayton

Acquisition Editor: Carrie Bolger

Editorial Project Manager: Carrie Bolger

Production Project Manager: Poulouse Joseph

Designer: Greg Harris

Typeset by TNQ Books and Journals

Preface

The Safety Assessments required to demonstrate compliance to FAA and/or EASA requirements are challenging to compile and often difficult to programme manage. In the author's experience, while individual aspects within the complexity of a comprehensive System Safety Assessment (SSA) are well understood within their respective functional/discipline silos, managing the programme to compile and collate a meaningful end result proves difficult and challenging.

The demonstration of safety for the initial application of increasingly complex technologies has always been a formidable and iterative task. Prior to the 1960s, this was primarily accomplished by the widespread use of the fail-safe design concept [Kritzinger (2006), Chapter 7], and the safety assessment tool of choice was often a Failure Modes & Effect Analysis (FMEA), which is largely focussed on single point failures. As later-generation aeroplanes developed (and the Concorde programme was a significant catalyst), more functionally interrelated safety-critical features were required. The resulting increase in system complexity highlighted the shortcomings of relying on the single fault criteria only.

Although the cornerstone in any system design is indeed founded on the correct application of all fail-safe design principles, a more comprehensive approach was needed – and this approach would need to examine and evaluate the malfunctioning of multiple aircraft components, their subsequent interactions and effect on various aircraft systems as well as the ongoing ability to ensure the aircrafts continued safe flight and landing. This led [Kritzinger (2006), Chapters 5 and 8] to the principle that an inverse relationship (see Fig. 2.4) should exist between the probability of a combination of failures and the severity of their effect, which the civil aviation regulatory authorities promulgate in their regulatory requirements (e.g. under FAR/CS25.1309, FAR/CS23.1309, FAR/CS29.1309). These requirements have a SSA as the Acceptable Means of Compliance (AMC), which has the typical (CS25.1309) objective (or design safety target) to prove that:

“The aeroplane systems and associated components, considered separately and in relation to other systems, must be designed so that:

- 1. Any CATASTROPHIC failure condition**
 - a. is EXTREMELY IMPROBABLE; and**
 - b. does not result from a single failure; and**
- 2. Any HAZARDOUS failure condition is EXTRMELY REMOTE; and**
- 3. Any MAJOR failure condition is REMOTE”**

SSA techniques are well established and are used extensively during the design of aircraft systems. Despite this, most of the techniques are highly subjective and dependent on the skill of the practitioner. The iterative nature (ARP 4754A, Fig. 6) of the SSA process also makes it difficult to programme manage. Furthermore, the complexity of modern technology invariably means that few, if any, systems can be viewed in isolation within the SSA – a particular system often has significant interactions with other systems, each of which may have been developed by separate individuals, groups or organisations (ARP 4754A, paragraph 1.1). To satisfy the typical SSA objectives of FAR/CSxx.1309 therefore requires a top down¹ and iterative² approach, which requires the careful management and consolidation of multiple inputs. For efficient business management purposes, this complexity is thus of academic and professional interest.

This book follows on from the author's previous work,³ which set the broad scene of the regulatory environment, the approach, tools and techniques used in the assessment of aircraft system safety. In this book the author presents a practical user guide (for both the novice safety practitioner and their managers) in the more specific area of conducting and managing SSA's to satisfy the requirements of FAR/CS25.1309. This book is written to supplement (not substitute) the content of advisory material (e.g. AMC25.1309) or their principal supporting reference standards (i.e. SAE ARP 4761, SAE ARP 4754A, RTCA/DO-178, RTCA/DO-154). In summary, this book strives to amalgamate these documents into a consolidated strategy, supported by simple process maps to aid the user in their understanding and subsequently to allow the more experienced to optimise their efficient use.

[Chapter 1](#) provides an introduction to the SSA concept and overall approach. A case study is defined therein, for which a safety strategy is derived in [Chapter 2](#) to support the Safety Programme Plan. [Chapters 3–11](#) then employ this strategy (see Fig. 2.5) to explore the theory of the most commonly used safety assessment techniques required to execute the Safety Assessment. Each chapter then concludes by applying the theory to the case study. This case study thus provides continuity throughout each chapter and enables the reader to bring the Safety Assessment together in a logical and efficient manner. The three last chapters of this book are of specific interest to those wishing to understand some of the potentially more esoteric disciplines used in assessing system safety:

- [Chapter 9](#) addresses the systematic causes of failures or unanticipated system behaviours and provides guidance on Development Assurance Levels (DALs) to the Safety Assessor who may not be expert in the fields of software(S/W) or complex hardware (H/W).
- [Chapter 10](#) addresses the assessment of the ability flight crew to cope with unsafe system failure conditions identified during the SSA process. As with [Chapter 9](#), this chapter is written to assist the Safety Assessor in understanding system approaches to minimising crew error, the role of the operator, and how or when specialist Human Factors (HF) input and engagement might be required.

¹ 'top down' means from aircraft level down to material level (refer Fig. 1.1).

² 'iterative' because, not only do required aircraft-level functions drive the design, but component and system architecture have derived functions and/or impact on aircraft-level failure severity allocations.

³ Kritzinger, D.E., 2006. Aircraft System Safety: Military and Civil Aeronautical Applications. Woodhead Publishing Ltd.

- [Chapter 11](#) looks beyond the ‘Certification Phase’ and provides a high-level discussion of the SSA interface with the Safety Case and/or Safety Management System (SMS) in the ‘Continuing Airworthiness Phase’. Note: The scope of this chapter is restricted to the Initial and Continued Airworthiness obligations of the relevant approval holders only, and not specific Continuing Airworthiness activities.

The reader will note that there is no chapter dedicated to the topic ‘Human Factors’ in the maintenance or production environments. In this book we (1) concentrated on proving compliance to CS25.1309 only and (2) have followed a more integrated approach by embedding HF considerations (e.g. see Table 6.1 as well as [Chapter 10](#)) as and when needed. The ‘Human Factors’ subject is vast, often requiring the input of specialists in areas such as cognitive psychology, human performance, physiology, visual perception, ergonomics and man–machine interface design. Although these specialists should form key members of the design and certification teams, the objective of this book is not to teach someone to be an HF specialist. Rather, the objective is to provide sufficient information so that an average Safety Assessment Engineer can either:

- conduct or coordinate a very basic HF Assessment on his/her own, or
- identify the correct disciplines and expertise to be brought together for a more complex HF assessment to be conducted, or
- to understand when the input of one or more HF specialists is required.

It must be noted that the scope of this book does not include any Occupational Health and Safety (OH&S) approaches. Although the intent of the civil aviation authorities is to prevent death and injury, their regulatory remit is to ensure airworthiness as explained below:

- Aircraft safety regulation is legally prescribed by national approval authorities (e.g. CAA, FAA, etc.), which operate under the guiding terms of International Civil Aviation Organization (ICAO). These SSA-related regulations (e.g. CS25.1309) are largely ‘goal/failure-based’ (Kritzinger (2006), Chapter 5), where each function of the aircraft is allocated a Failure Probability Target or a Development Assurance Level, which is set (under ICAO) as a universally accepted goal to achieve.
- OH&S regulations are prescribed and enforced by different national and/or federal authorities.⁴ The Safety Criteria are largely ‘risk/accident based’ (Kritzinger (2006), Chapter 4), where the assessor evaluates the risk of an accident which any hazard may present. The risk criteria used is often company specific and often even differs between different projects in the same company.

Although commonality does exist in their intent (largely limited to preventing harm), the criteria and terminology used (as well as the techniques, approaches and skill required) differ sufficiently (i.e. Airworthiness hazards vs. OH&S hazards) to justify two different safety reports – each aimed at meeting specific regulatory requirements. This, however, should not preclude the informed specialist from subsequently taking a holistic approach and reviewing hazards from such reports to ensure all possibilities have been considered. This book, however, focuses on safety from an Airworthiness perspective only (i.e. hazards to airworthiness).

⁴ Matters are therefore complicated when the two regulatory regimes are mixed, which is the case in military standards such as MIL-STD 882 and DEF-STAN-0056 (it can be argued that the perspective from which these standards were initiated was influenced by the military being both the operator and the regulator).

Acknowledgements

I cannot begin to thank adequately those who helped in the preparation of this book:

- I would like to thank Stephen Goldsmith ([Chapter 4](#)) and Derek Reinhardt ([Chapter 9](#)) whose help, insight and experience have proven invaluable to me. It is my sincere hope that the summary and simplification in [Chapters 4 and 9](#) do not in any way take away from the complexity of your input. I am especially grateful for the detail you provided for the Annexes of [Chapters 4 and 9](#).
- I am also grateful to Vahid Norouzalibeik, who reviewed [Chapter 10](#) and offered his paper on the ‘Human-Machine Interface in Aerospace’ as a supporting Annex to [Chapter 10](#).

A special thank you to my friends and colleagues who kindly volunteered to review and provide valuable new insights for the theories I was trying to convey:

- Scott Vaughan for his incisive comments and meritorious suggestions to all chapters of this book;
- Andy ‘Evsco’ Evans (for reviewing the [Preface](#) and [Chapter 11](#));
- Grant Findlay (for reviewing [Chapters 6 and 10](#));
- James Hayton (for reviewing [Chapter 10](#));
- Ian Roberts for providing the Case Study in [Chapter 1](#) and for reviewing its application in [Chapter 4](#).

An additional note of gratitude goes to Stephen Goldsmith for all his help in maintaining www.aircraftsystemsafety.com in support of this book, as well as his input to clarifying the AD ‘rectification campaign’ in [Chapter 11](#).

It is also a pleasure to express gratitude to Carrie Bolger at Elsevier Group, for her support and enthusiasm in developing this book.

And, of course, a very special thank you to my loving wife, Nicole, for keeping me on task, who has been a careful reader of the manuscript, and who patiently supported all those 3-am wake-ups when my brain decided to kick into gear again.

Introduction

1

If we slide into one of those rare moments of military honesty, we realize that the technical demands of modern warfare are so complex a considerable percentage of our material is bound to malfunction even before it is deployed against a foe. We no longer waste manpower by carrying the flag into battle. Instead we need battalions of electronic engineers to keep the terrible machinery grinding.

Ernest K. Gann, *The Black Watch*

1.1 Introduction to System Safety Assessments

1.1.1 Background

It is broadly accepted that the prime causal factors of an aircraft accidents are either:

- Operational (such as pilot error, weather and operating procedures) or
- Technical (such as design errors, manufacturing errors, maintenance errors and component failures).

When certifying a new (or modified) system, designers conduct a thorough assessment of potential failures to demonstrate an inverse relationship exists between the probability of occurrence and the severity of consequence inherent in its effect (e.g. see Fig. 2.4). The designer must also consider whether the design presents qualities that might lead to errors during manufacture, maintenance or operation, or whether the system is vulnerable to foreseeable variations¹ in the operating environment.

The collated documents required to demonstrate the above are often collectively referred to as a System Safety Assessment (SSA).²

1.1.2 Aim of a System Safety Assessment

For a new (or modified) system, the SSA typically (Kritzinger (2006), Chapter 8) aims to ensure that:

- safety is designed into the system in a timely and cost-effective manner;
- hazards associated with each aircraft subsystem are identified, tracked, evaluated and eliminated or communicated (e.g. via warnings in the flight manual) to those likely to experience the hazard(s) during operation.

¹ This could involve large variations in atmospheric temperature, pressure, acceleration (e.g. due to gusts), vibration and other hostile events such as lightning strikes and icing.

² For more information on how the System Safety Assessment came about (and a comparison with Safety Case), see Kritzinger, D.E., 2005. *Aircraft System Safety: Military and Civil Aeronautical Applications*. Woodhead Publishing Ltd, Cambridge, CB1 6AH.

- Historical safety data, including lessons learned from other systems, are considered and applied where appropriate.
- Minimum risk is pursued in the use of novel technology, materials, or designs; and in any production, test and operational techniques.
- Those actions taken to eliminate hazards or reduce risk to an acceptable level are appropriately documented to ensure this is maintained in the Continuing Airworthiness phase.
- Any retrofit actions required to improve safety are minimised through the timely inclusion of appropriate additional safety features that are implemented when necessary.
- Procedural and Training requirements are identified to support and maintain safety assumptions and assertions.
- The program team is made aware of system safety and how the design can be used to mitigate certification risks.

Within the scope of this book, the SSA is generated as the primary means of compliance to design codes such as CS/FAR25.1309 (for large aircraft), CS/FAR23.1309 (for commuter aircraft), etc. The SSA is therefore defined as:

a pro-active opportunity to optimise the design and one which provides a structured body of objective evidence that the system, if used in accordance with the listed recommendations and limitations, can be certified as being “safe enough” to be released into a defined service environment.

1.1.3 Objectives of a System Safety Assessment

For a new (or modified) system, the SSA’s objectives are typically to:

- demonstrate that an inverse relationship exists between the probability of an undesired occurrence and the degree of severity inherent in its effect;
- demonstrate that the design is such that it cannot lead unnecessarily to errors during manufacture, maintenance or operation by the crew;
- demonstrate that the systems are suitable for the environment that the systems would be exposed to.

1.1.4 Scope of a System Safety Assessment

Scoping a SSA is a vital (but often largely neglected) part of the successful start, conduct and completion of the SSA report.

The following two interrelated factors play a significant part in defining the scope of an SSA:

- Safety Criteria: The safety criteria used are often influenced by the regulations applicable to the contract and/or category of aircraft (e.g. see CS23.1309 vs CS25.1309). There are two fundamentally different approaches:
 - The accident- or risk-based approach ([Kritzinger \(2006\)](#), Chapter 4) found in standards such as MIL-STD-882 or Def Stan 00-56, or
 - The failure target or goal-based approach ([Kritzinger \(2006\)](#), Chapter 6) found in standards such as FAR25.1309.

- **System Level:** Safety requirements exist at the aircraft, system, item, component and material level (refer, inter alia, [ARP4754A](#) para 4.1.3). The contracting agency and the assessor therefore must consider the system level of the design and scope the assessment accordingly. The illustration in [Fig. 1.1](#) shows a hierarchy which distinguishes between the different system levels. This illustration, or a derivative thereof,³ can be applied to the SSA as it helps us define what the system under consideration is, and demonstrates that each level is supported by the key (i.e. summarised) results of the lower level assessments. For instance:
 - A Level 1 material assessment will largely be scoped towards identifying any hazardous material (e.g. flammable, toxic, etc.) and making safety recommendations accordingly (e.g. instructions for handling, machining, disposal, etc.).
 - A Level 2 component (or article/part item) Safety Assessment will incorporate the findings of Level 1 and would, for instance, add Failure Mode Effects Analyses (FMEA), predicted failure rates, Hardware and Software Development Assurance Levels,⁴ etc. Ideally, for optimised system architecture, the appropriate H/W or S/W DALs should be flowed down from the Level 4 design team. While referring to DAL's, it is of note that Design Organisations may elect to add another system Level between the currently illustrated Level 1 and Level 2, which more clearly differentiates between items and components to align with [SAE ARP4754A](#) (Fig. 5), where:
 - An **Item** is '*A Hardware of Software element having bounded and well-defined interfaces*' (ARP5754A, p12). Think of it as any part that is allocated a configuration identifier. From both a certification and maintenance perspective, items can be further separated into:
 - Black box/LRU level (for instance for TSO items), and
 - SRU Level (for items inside the black box).
 - A **Component** is '*Any self-contained part, combination of parts, subassemblies or units that perform a distinctive function necessary to the operation of the system*' (ARP5754A, p11). Think of it as any self-contained part that performs a distinctive function necessary to the operation of the item.
 - At Level 3 the designer will be faced with integrating various components into a subsystem. This is the first level at which a SSA is required, and the scope will not include any aircraft interface considerations other than what is flowed down contractually from the Level 4 assessment. Note that [Fig. 1.1](#) again has a limitation here in that is often useful to insert another level between the current Level 2 and Level 3, where:
 - A **System** is '*A combination of inter-related items arranged to perform a specific function*' (ARP5754A, p13). An example would be the Altitude display system (refer Section 4).
 - A **Subsystem** is subordinate to a System, for instance, in Section 4 the systems consist out of a Primary Attitude Display Subsystem and a Standby Attitude Display subsystem.

³ Although the philosophy of system hierarchy is not new, there is no common industry standard. Design Organisations are thus encouraged to clearly define and agree a hierarchy with all stakeholder involved in the System Safety deliverables.

⁴ Ideally, for an optimised system architecture, the appropriate H/W or S/W DALs should be flowed down from the Level 4 design team.



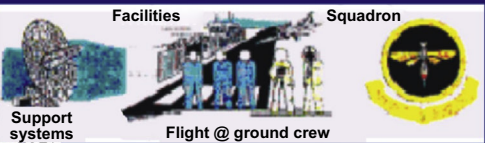


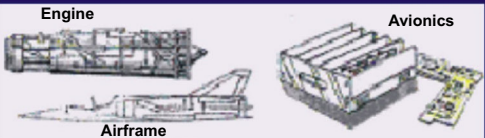

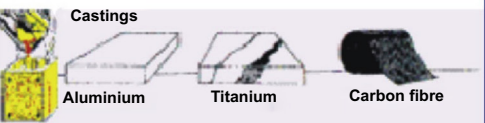
The system hierarchy		
System names	Level	Examples - configuration
Operational forces	8	 National defence
Combined combat forces	7	 Combined combat forces
User systems	6	 Facilities Squadron Support systems Flight @ ground crew
Product systems	5	 Aircraft Weapons Simulator Logistic support
Product	4	
Product subsystems	3	 Engine Avionics Airframe
Component	2	 Instruments Undercarriage Turbine blades
Characteristics of material/process	1	 Castings Aluminium Titanium Carbon fibre

Figure 1.1 System hierarchy.⁵

⁵ Reproduced with kind permission from the Armaments Corporation of South Africa (ARSMCOR). Note these system levels are not definitive (e.g. the safety assessor might elect to insert a few more levels). For Levels 1–4, see comparative hierarchy in Fig. 5 of ARP4754A.

- The Level 4 SSA is at the aircraft level and is the responsibility of the aircraft integrator. For a modification (e.g. STC), it is scoped to consider the performance of the new system as well as the interaction between all affected aircraft systems. Safety requirements are functionally decomposed in a hierarchical structure from product (i.e. aircraft) level to subsystem (e.g. altitude display system) to components (e.g. Altitude Display Unit). At Level 4 the safety requirements are those requirements generated from the aircraft Functional Hazard Analysis (FHA) based on required aircraft functions

Note: Up to this level, the ‘goal-based’ safety criteria (either via regulations such as CS/FAR25.1309, or flowed down from the System Level 5 accident sequence models, refer Chapter 11) is the most appropriate method to evaluate an acceptable level of safety.

- Levels 5 and 6 fall within the remit of the system operator (not the designer) in the format of a Safety Case (Kritzinger (2006), Chapter 9). The Safety Case is a live document, which manages the operational risk once the system enters service.

Note: At this Level 5 the ‘risk-based criteria’ (e.g. of an Organisational SMS, MIL-STD-882, or the Health & Safety Regulations) is typically used.

1.2 Conducting a System Safety Assessment

1.2.1 Modelling the process

With reference to Fig. 1.1, the illustration in Fig. 1.2 provides a simple illustration of a suggested Level 4 and Level 5 SSA process. Each step will be briefly explored in the sections below:

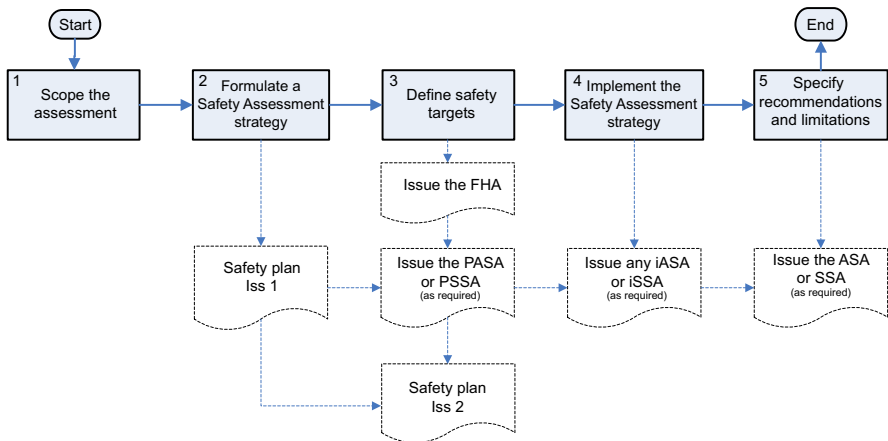


Figure 1.2 System Safety Assessment process.

1.2.2 Step 1: scope the assessment

Scoping of the assessment is discussed in Section 1.1.4 and looks to outline stakeholder responsibilities (e.g. at each system level supporting the Safety assessment);

the safety standards/criteria applicable and the physical and functional boundaries/interfaces of the system under consideration.

Should the boundaries not be clear to all parties involved in the assessment, it is highly likely some vital part may either be overlooked, fail to integrate appropriately with another part of the assessment, or simply be omitted altogether. Furthermore, boundaries aid with responsibility allocation, especially when integrating products from subcontractors (who also have safety deliverables) into a higher system.

1.2.3 Step 2: formulate the safety assessment strategy

Within the context of the scope, we next need to plan *how* we are going to do a systematic assessment, *who* is going to contribute to each part, and *when* each part of the Safety Assessment needs to be completed.

See [Chapter 2](#) for more information on planning the Safety Assessment.

The output of this phase can then be captured in the first issue of the Safety Plan (or Safety Program Plan). See [ARP4754A](#) para 5.1.5 (and its Appendix B) for example contents of such a Plan.

1.2.4 Step 3: define the safety targets

It is important to set safety objectives for the system and its functions before any specific architecture or technology is finally decided upon. Not only is this good Systems Engineering practice (i.e. Requirements Management as per [SAE ARP4754A](#) para 5.3), but, considering that these targets have significant impact on the costs associated with the proposed solution,⁶ it is important they are established at the earliest feasible opportunity.

The Verification and Validation (V&V model) of Systems Engineering (refer, inter alia, [ARP4754A](#) Fig. 5) can be usefully adapted (refer [Fig. 1.3](#)) to illustrate how such targets are captured during the requirements management process.

When setting safety targets/requirements, designers also need to be aware that there are internationally accepted safety/reliability targets for key aircraft functions. For example, TCAS safety targets as described in the FAA's [AC20-131B](#), which stipulated, inter alia:

*“Unannunciated failures of the TCAS II equipment or its associated transponder, sensors or displays that generate resolution advisories must be **Improbable**.*

⁶ For instance, if the Level 3 or 4 Safety Assessment requires a function with software developed to Level A, then there will be a significant impact on cost and schedule if a lower software development assurance level was assumed during the bid phase.

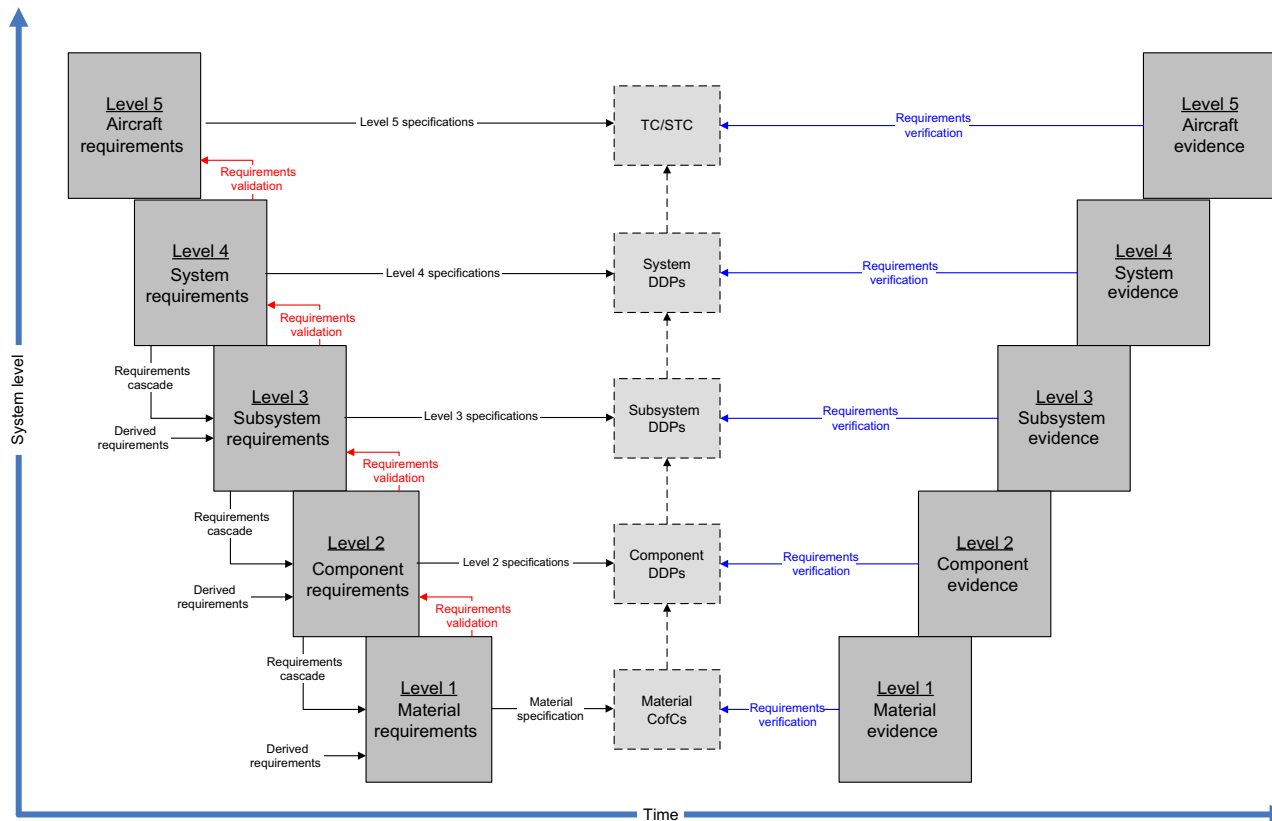


Figure 1.3 The V&V of system safety requirements.⁷

⁷ Note, in this illustration we have tailored the system level concepts of Fig. 1.1 by inserting an extra subsystem level. It is recommended that each organisation define these system levels accurately for their need and then use it consistently throughout all their V&V approaches (as well as their subcontracting models and delivery frameworks).

*Unannounced failures of the TCAS II equipment or its associated transponder, sensors or displays that generate resolution advisories that would cause midair collisions or other catastrophes must be **Extremely Improbable**.*"

See [Chapter 3](#) for more information on defining the safety target in the FHA. The FHA is the step to determine, in simple terms, what can go wrong at the functional level. The process begins with the aircraft-level FHA to assess the significant failure conditions attached to given aircraft functions. After functions are assigned to specific aircraft systems, the FHA is then repeated at the system level. These FHA safety requirements should be fed back into the next revision of the Safety Program Plan ([ARP4754A](#) App B para 3.3.3), which captured the functional safety targets for the evolving design solution and the methodology (e.g. see [Fig. 2.5](#)) by which it will be accomplished.

The output of this step is the issue of a Preliminary Aircraft Safety Assessment (PASA) and/or a Preliminary System Safety Assessment (PSSA), which:

- allows the regulatory authority and/or the aircraft integrator the opportunity to agree the safety strategy (e.g. protective strategies, fail safe concepts, architectural attributes and safety tools/techniques) which may be needed to meet the defined safety targets.⁸ If possible, this agreement must be obtained in a timely fashion so as to minimise any costly changes which might result if additional work, or even a redesign if more stringent targets are required.
- takes the Failure Conditions that have been identified and using techniques, such as Fault Trees Analyses (see [Chapter 4](#)), coupled with data about component reliability, seeks to determine if selected probability targets for given Failure Conditions will be met (i.e. requirements validation). After the PSSA has been issued, equipment suppliers are given specifications (see [Fig. 1.3](#)) to meet for the reliability of their particular equipment or components and are subsequently mandated to produce their own lower level Safety Plans which will feed into the final SSA as the means of compliance to CSxx.1309 (see Step 5).

1.2.5 Step 4: implement the safety assessment strategy

The agreed Safety Assessment strategy now needs rigorous implementation to ensure that the aims (refer [Section 1.1.2](#)) and objectives (refer [Section 1.1.3](#)) are successfully achieved. Further guidance and examples on how this implementation can be accomplished are provided in subsequent chapters.

The output of this step could be various Interim issues of the Aircraft Safety Assessment (iASA) or of the Aircraft Safety Assessment (iSSA) in the SSA's journey to completion.

⁸ The PSSA is the top-down part of the safety process – this is where architecture is established based on the FHA knowledge; FDALs are decomposed to the point they allocated to items, at which point they become IDAL targets. There may be several layers of architectural decomposition here. In simple terms the PSSA proposes what we are going to do about the things that can go wrong.

1.2.6 Step 5: specify recommendations and limitations

The output of this step is a Safety Assessment⁹ for each level of system integration. The objective is to prove the safety integrity of that level and summarise the key aspects which need to be considered at the next level of integration.

For each level of integration, a clear set of recommendations should be made stating whether the system or equipment should either be accepted into service, allowed to proceed to the next project phase, or describe any further work required to overcome any shortcomings that have been identified prior to continuing to the next project phase.

Any limitations or safety-related restrictions on the use of the system should be stipulated, which at platform level might include:

- Flight Restrictions (e.g. no flying in IFR conditions below a certain altitude)
- Flight Profile Limitations (e.g. maximum rate of turns or bank angles)
- A list of hazards which need to be managed

For more information, see Sections 11.2.2 and 11.2.3.

1.3 Regulatory context of CS/FAR2X.1309

Fig. 1.4 illustrates where design codes such as FAR/CS25.1309 are positioned in the regulatory regime and some of the advisory materials available to support the generation of a compliant Safety Assessment.

SAE ARP4754A is of particular note here, as it provides recommended practices for the development of aircraft systems taking into account the overall aircraft operating environment and functions. This includes:

- Safety Assessment processes
- Development Assurance Level (DAL) assignment (see below)
- Requirement capture
- Requirement validation
- Implementation verification
- Configuration management
- Process assurance
- Certification and regulatory authority coordination

Table 1.1 maps the SAE ARP4754A Appendix A safety objectives (see columns 1 to 4) with the content of this book (see column 5).

⁹ The SSA is the bottom-up part of the safety process – where effectively the evidence is built up to prove that safety targets have been met and which is the ultimate means of compliance (MoC) against requirements such as FAR/CS25.1309.

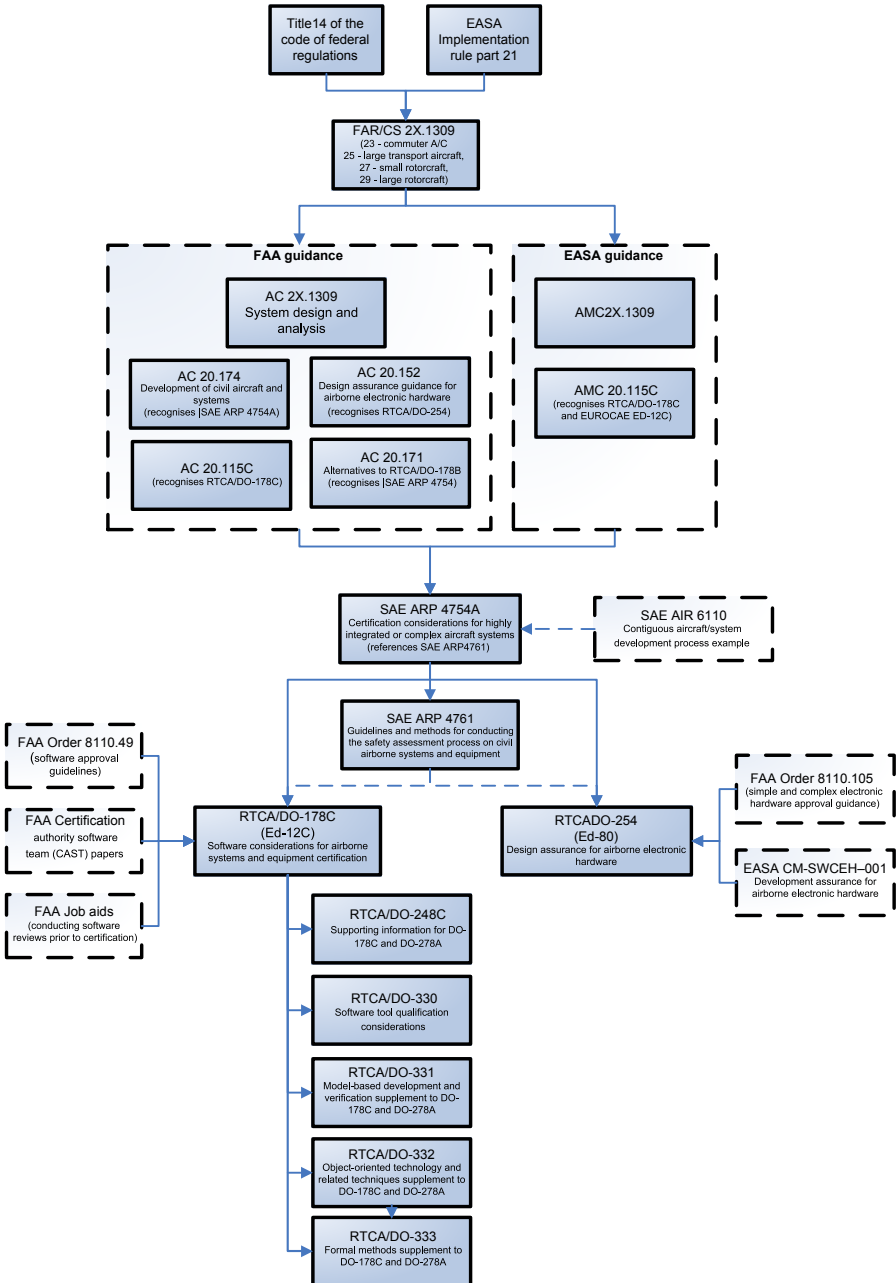


Figure 1.4 Civil regulations for system safety.¹¹

¹¹ For FAA Job Aids, see https://www.faa.gov/aircraft/air_cert/design_approvals/air_software/guide_jobaid/ For FAA CAST Papers, see https://www.faa.gov/aircraft/air_cert/design_approvals/air_software/cast/ Note that both ARP4761 and ARP4754A are expected to be revised during 2018.

Table 1.1 **ARP4754A** Safety Assessment process

No.	Objective description	ARP4754A section	Outputs	Where in this book?
3.1	The aircraft/system functional hazard assessment is performed	5.1.1 5.2.3 5.2.4	Aircraft FHA System FHA	Chapter 3
3.2	The preliminary aircraft safety assessment is performed	5.1.2 5.2.3 5.2.4	PASA	Sections 1.1.4 and 1.2
3.3	The preliminary system safety assessment is performed	5.1.2 5.1.6 5.2.3 5.2.4	PSSA	Sections 1.1.4 and 1.2
3.4	The common cause analyses are performed	5.1.4	PRA	Chapter 7
			CMA	Chapter 6
			ZSA	Chapter 8
3.5	The aircraft safety assessment is performed	5.1.3 5.1.6	ASA	Sections 1.1.4 and 1.2
3.6	The system safety assessment is performed	5.1.3 5.1.6	SSA	Sections 1.1.4 and 1.2
3.7	Independence requirements in functions, systems and items are captured	5.3.2	System, H/W, S/W requirements	Section 9.2.1.2
		5.2.3	PASA	Fig. 1.2, with input from Chapter 4 (FTA) and Chapter 9 (DAL)
		5.1.2	PSSA	

1.4 The Case Study

1.4.1 Introduction

To provide a comprehensive demonstration of the concepts discussed in this book, each chapter in this book will apply the theory discussed to the following case study: let us assume we have a customer who has a requirement to upgrade the avionics on a large military transport aircraft by replacing the old, now unreliable, analogue displays (see Section 1.4.2) with the current state-of-the-art in solid-state digital technology (see Section 1.4.3). We will also assume that the customer has agreed to use the safety criteria of CS25.1309.¹⁰

¹⁰ The reasons for using a civil standard on a military aircraft may include:

- needing to operate in civil air space under the latest operating rules (e.g. vertical separation minima);
- not wanting to dedicate resource in maintaining unique Military Regulation for functional integrity accepted in the civil transport industry;
- encouraging industry to develop equipment/systems which meet international standards;
- not needing to flow down unique safety targets from the Level 5 Safety Case, which would need a complete Hazard Log with supporting accident sequence (and or Loss Model) to get to system safety targets.



Figure 1.5 Legacy avionic display.

To demonstrate the concepts discussed in this book, [Chapters 2–11](#) will each apply the theory to the Altitude and/or the Attitude Display systems (primary and standby) within the avionic upgrade.

1.4.2 Legacy system description

1.4.2.1 General description

For the purposes of this case study, we shall assume that the legacy system comprised a number of totally independent Line Replaceable Units (LRUs), with the attitude and altitude information being presented on different displays as shown in [Fig. 1.5](#).

1.4.2.2 Altitude

The mechanical altimeter measured atmospheric pressure by utilising an internal bellows which were connected directly to the aircraft's pitot-static system. The bellows were attached to a needle which moves as the bellows expanded or contracted dependent on atmospheric pressure and therefore altitude.

Because atmospheric pressure can change with changes in weather patterns,¹² mechanical pressure compensation was provided through the use of a variable millibar scale which could be set to a level determined by Air Traffic Control (who provides the pressure elevation of the local area).

The altimeter had a lighting input and 28V DC for an internal vibrator to prevent the needle from sticking particularly when there are only small changes in altitude (an effect known as 'stiction').

¹² With an approaching cold front, air pressure could change by say 5 mbar, which in turn could result in a skewed altitude reading of up to 130 feet.

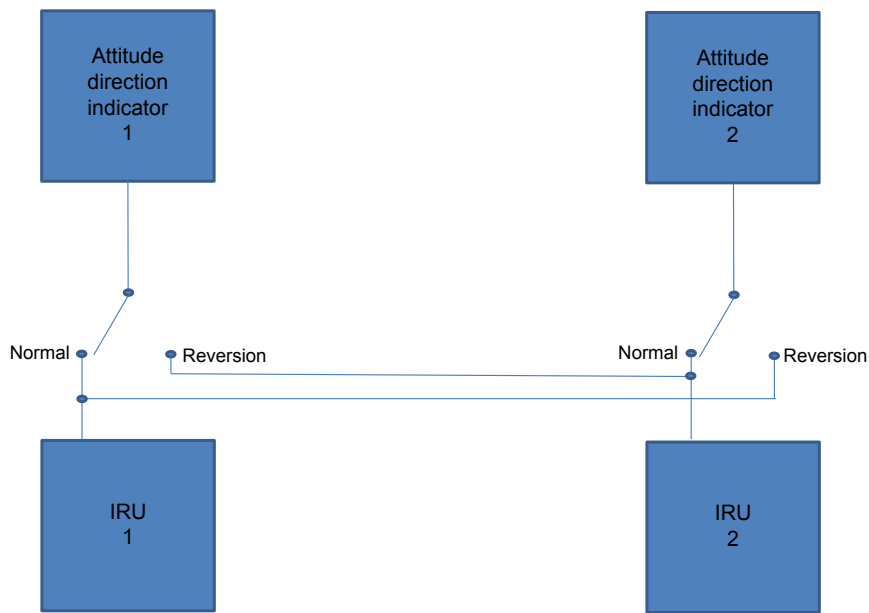


Figure 1.6 ADI reversion.

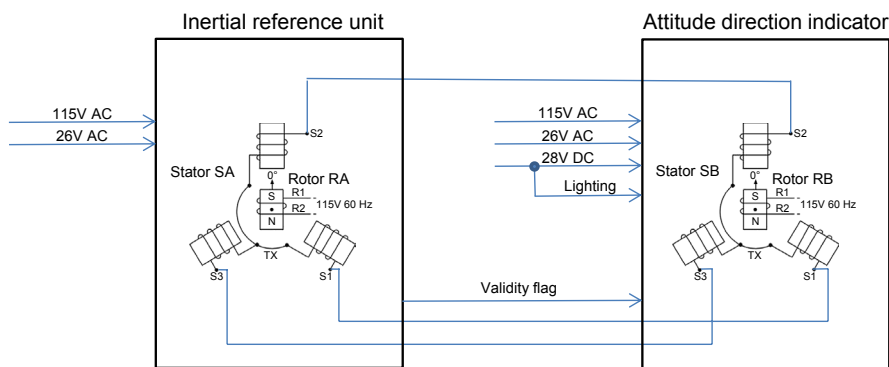


Figure 1.7 Synchro interface.

1.4.2.3 Attitude

Attitude information was presented on an Attitude Director Indicator (ADI) which received its signal from an Inertial Reference Unit (IRU) or Vertical Gyro, see Fig. 1.6. The IRU used a synchro system to transmit attitude information based on displacement from the horizontal (see Fig. 1.7).

A syncho consisted of a static element (the Stator) and a rotating element within it (the rotor). Within the IRU, a field is set up in each of the three legs of the stator. This is as a result of the relative angle (pitch) of the Gyro connected to the Rotor

(RA). This relative angle will cause each leg to have a different field induced within it. These different fields will be replicated in the corresponding stator leg of the stator (SB) within the ADI. This drives Rotor (RB) to a position which corresponds to the position of IRU rotor (RA) which will also represent the aircraft's pitch angle.

Separate synchro signals were provided for each of the pitch and roll channels.

The attitude reference system included a reversion capability which allowed the off-side IRU to be connected to the onside ADI in the event of an onside IRU failure.

The other inputs to the Attitude reference system were 115V AC Power, 26V AC synchro reference and 28V DC power for processing circuitry and lighting for the ADI.

There was also a validity flag line between the IRU and the ADI: if the IRU considered its output to be invalid, a flag was set and a corresponding warning was displayed on the ADI. This would not have detected a break in one of the Synchro lines between the IRU and ADI, but would only have been flagged if there was a detector on the Synchro input lines of the ADI.

1.4.3 Upgraded system description

1.4.3.1 General description

For the purposes of this case study, we shall assume that the upgraded avionics utilises the same principles, though looks significantly different (as can be seen in [Fig. 1.8](#)) due to its more integrated use of an ARINC 429 data bus.¹³

The display system presents the crew with key flight information (including attitude and altitude) on an integrated display as can be seen in [Fig. 1.8](#). The display system for each pilot comprises three different key LRUs:

- The Flight Displays, which are large format Liquid Crystal Displays (LCDs). Each pilot has one configured as a Primary Flight Display (PFD) and one as a Navigation Display (ND). In the event of a PFD failure, the same side ND will reformat as a PFD. In the event of an ND failure, the PFD cannot be reconfigured as an ND, but the flight plan will be available on the lower half of the PFD as shown in [Fig. 1.8](#)

¹³ ARINC 429 is the technical standard for the predominant avionics data bus used on modern commercial and transport aircraft. It defines the physical and electrical interfaces of a two-wire data bus and a data protocol to support an aircraft's avionics local area network. Label guidelines are provided as part of the ARINC 429 specification, for various equipment types. Each aircraft will contain a number of different systems, such as flight management computers, inertial reference systems, air data computers, radar altimeters, radios, and GPS sensors. For each type of equipment, a set of standard parameters is defined, which is common across all manufacturers and models. For example, any air data computer will provide the barometric altitude of the aircraft as label 203. This allows some degree of interchangeability of parts, as all air data computers behave, for the most part, in the same way. There are only a limited number of labels, though, and so label 203 may have some completely different meaning if sent by a GPS sensor, for example. Very commonly needed aircraft parameters, however, use the same label regardless of source. However, some manufacturers use variations on this standard to produce bespoke interfaces.

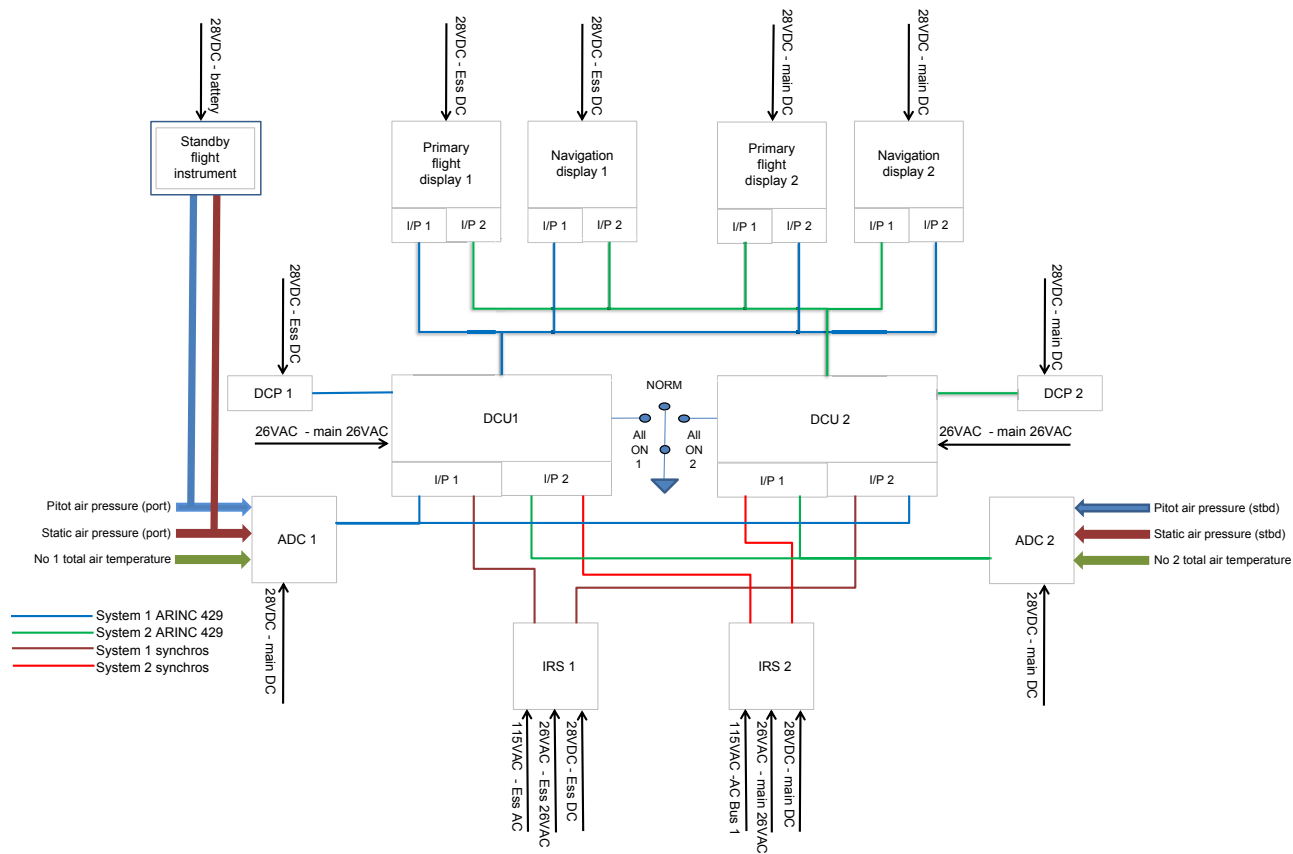


Figure 1.8 Case study system architecture.

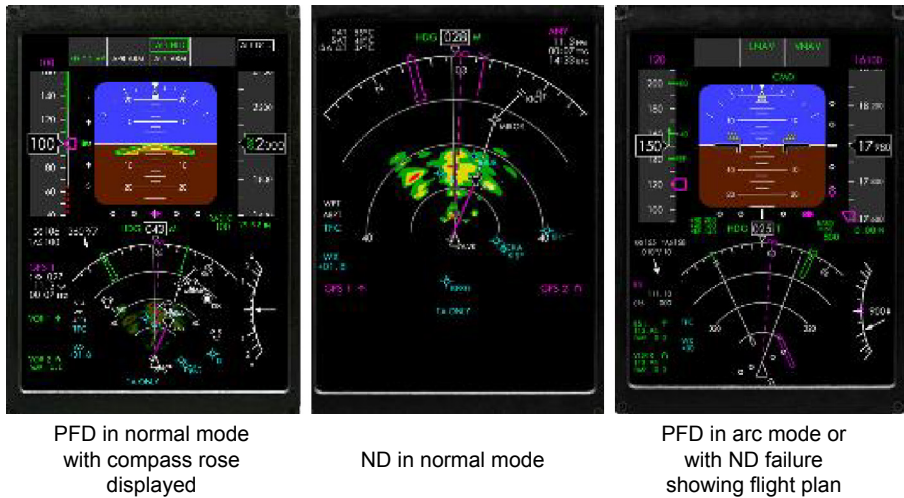


Figure 1.9 Primary flight display modes.

- A separate¹⁴ Display Concentrator Unit (DCU), which provides an interface to all associated sensors. The DCUs include comparators which determine if there is a discrepancy between the No. 1 and No. 2 system information being fed to them. The case study parameters being compared are:
 - Altitude
 - Airspeed
 - Attitude
 - Heading
- In the event of a DCU failure, a discrete input is provided from the Reversion switch to each of the DCUs. When the switch is selected to one of the DCUs then that DCU transmits its output to the displays on both sides.
- A Display Control Panel (DCP) to set the PFD mode as shown in Fig. 1.9 and allow input of other variables (such as Barometric setting for the altimeter).

1.4.3.2 Altitude

Altitude information is determined within an Air Data Computer (ADC) using the principles of the mechanical altimeter, with the resultant altitude transmitted to the DCU on an ARINC 429 data bus (see Section 3.5). The ADC is connected to the pitot-static system, with both the No. 1 ADC and Standby Instrument being fed by the port side system and the No. 2 ADC being connected to the starboard system. Each ADC is also connected to the outside Total Air Temperature (TAT) probe.

The TAT probe compresses the impacting air to zero speed, and the resulting temperature causes a change in the resistance of the sensing element. The air data then convert this resistance to temperature. The air temperature is used to calibrate the impact pressure as well as in determining air density. These inputs provide the following

¹⁴ The use of a separate DCU, which is mounted in the avionics bay, reduces the PFD depth making it easier to install in the instrument panel.

computed air data output signals which supply primary flight displays: Pressure Altitude, Baro-Corrected Altitude, Vertical Speed, Mach Number, Total Air Temperature, Calibrated Airspeed, True Airspeed, Altitude Hold, Airspeed Hold.

1.4.3.3 Attitude

Attitude reference data are determined within the IRUs with each utilising a ring laser gyro and accelerometers to determine pitch and roll information. As the case study is based on a retrofit, it has been assumed that existing IRUs have been retained. As a result the Attitude information is transmitted to the DCUs as synchro signals.



Figure 1.10 Standby flight display.

Each DCU receives data from the onside¹⁵ sensor (ADC/IRU) as its primary input. It also receives the offside sensor data as its secondary input. This means that instead of the reversion mode physically switching signals between units, as is the case on the legacy system, it is only necessary to provide the DCU with a reversionary signal to instruct it to look at the secondary input.

1.4.3.4 Standby display

To support the main installation, a Standby Flight Display (Fig. 1.10) has been included. The Standby Instrument provides an additional source of Heading, Attitude, Airspeed and Altitude information. The inclusion of the Standby instruments provides:

- a source of data to compare with the PFDs in the event of a disparity between the information presented on the two PFDs;
- a reversionary capability in the event of loss of all Primary Flight Information, or an element of the Primary Flight Information (such as Attitude or Airspeed) in the event of an unexpected common mode failure;
- support to the Safety Assessment in meeting safety targets and mitigating common mode failures.

¹⁵ 'On side' is the side that provides the data under normal circumstances (i.e for the Pilot the No. 1 systems and No. 2 systems for the Co-Pilot). 'Cross side' would be the other side which is normally connected after a failure.

The Standby instrument contains its own Gyro, accelerometers and Air Data Sensors to ensure independence from the primary displays. The only common connection is to the pitot-static system.

1.4.3.5 Electrical interface

The system also has inputs for power (28V DC) and lighting control; the electrical connectivity is detailed in Table 1.2.

The electrical generation system (see Fig. 1.11) provides AC power to four independent MAIN power busses (MAIN AC BUS 1, 2, 3 and 4) and an ESSENTIAL AC BUS (ESS AC). The Essential Bus can be connected to either AC BUS 3 or 4 to provide a reversion capability in the event of a Number 3 or 4 generator failure. If both these generators should fail, then the ESSENTIAL Bus can be powered from the ESSENTIAL DC (ESS DC) Bus via the Emergency Inverter (EMER INV).

26V AC for the Synchros is provided by the 115–26V Transformers (115/26V TFMR) connected to the MAIN AC BUS 1 and ESS AC BUS. 26V AC is provided to all equipment receiving or transmitting Syncho Signals.

DC power is provided from four Transformer Rectifier Units (TRUs 1, 2, 3 and 4) which convert the power from the AC buses, into stabilised 28V DC. The TRUs work in pairs to provide redundancy. TRUs 1 and 2 supply the MAIN DC BUS, and TRUs 3 and 4 supply the ESSENTIAL DC BUS (ESS AC BUS). The loading on the MAIN and ESSENTIAL Busses is such that a single TRU has the capacity to support the electrical load on the connected bus. The MAIN and ESSENTIAL Busses and the

Table 1.2 Equipment power supply distribution

	Busbars	Equipment connected
AC BUSBARS	MAIN AC BUS 1	IRU 2 DCU 2
	ESSENTIAL AC BUS	IRU 1 DCU 1
	MAIN 26 V AC BUS	IRU 2 SYNCHRO DCU 2 SYNCHRO
	ESSENTIAL 26 V AC BUS	IRU 1 SYNCHRO DCU 1 SYNCHRO
DC BUSBARS	MAIN DC BUS	PFD 2, ND 2, DCP 2, ADC 2, IRU 2
	ESSENTIAL DC BUS	PFD 1, ND 1, DCP 1, ADC 1, IRS 1
	BATTERY BUS	Standby flight instrument

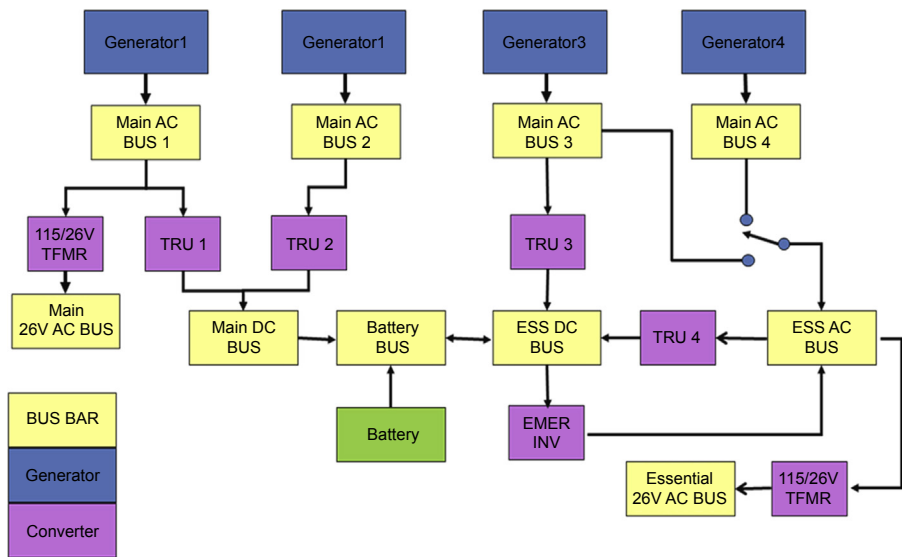


Figure 1.11 Electrical generation schematic.

battery are all connected to the BATTERY BUS to provide a high level of integrity. In the event of a cascading electrical failure, the BATTERY Bus will be the last bus operating.

Table 1.2 provides the details of which bus each equipment is connected to.

1.4.3.6 ARINC 429 data bus

ARINC data words are always 32 bits and typically use the format shown in Fig. 1.12.

32	31	30	29											11											10	9	8											1										
P	SSM		Data											Pad											Discretes											SDI		Label										
			MSB																						LSB																							

Figure 1.12 ARINC data word composition.

The data word descriptions are:

- Parity (P): The Most Significant Bit (MSB) is always the parity bit for ARINC 429. Parity is normally set to odd. Odd parity means that there must be an odd number of ‘1’ bits in the 32-bit word that is ensured by either setting or clearing the parity bit.
- SSM: Bits 31 and 30 contain the Sign/Status Matrix or SSM. This field contains hardware equipment condition, operational mode, or validity of data content. This includes the equivalent of the analogue validity flag. In this case, if the connecting line is broken, the valid bit will not be received.

- Data: Bits 29 to 11 contain the data. This means all the data from a given LRU are on the same signal line. If the line is broken, the data and validity signal are not received overcoming the need for the equivalent of a synchro signal detector as described for the analogue system.
- SDI: Bits 10 and 9 provide a Source/Destination Identifier or SDI. This is used for multiple receivers to identify the receiver for which the data are destined. It can also be used in the case of multiple systems to identify the source of the transmission. This is used in the case study where a line feeds more than one LRU as is the case with the ADCs.
- Label: Bits 8 to 1 contain a label identifying the data type and the parameters associated with it. It is used to determine the data type of the remainder of the word and, therefore, the method of data translation to use.

References

- AC20.115c, 2013. Airborne Software Assurance. U.S. Department of Transportation, Federal Aviation Administration.
- AC20-131B, March 1993. Airworthiness Approval of Traffic Alert and Collision Avoidance Systems (TCAS II) and MODE S Transponders. FAA, Washington.
- AC20-152, 2005. Design Assurance Guidance for Airborne Electronic Hardware. U.S. Department of Transportation, Federal Aviation Administration.
- AC20.171, 2011. Alternatives to RTCA/DO-178B for Software in Airborne Systems and Equipment. U.S. Department of Transportation, Federal Aviation Administration.
- AC20.174, 2011. Development of Civil Aircraft and Systems. U.S. Department of Transportation, Federal Aviation Administration.
- AMC20-115B, 2013. Software Considerations for Certification of Airborne Systems and Equipment. EASA. ED Decision 2013/026/R.
- CAST-27, June 2006. Clarification on the Use of RTCA Document DO-254 and EUROCEA Document ED-80, Design Assurance Guidance for Airborne Electronic Hardware. FAA Position Paper.
- CAST-28, December 2006. Frequently Asked Question on the Use of RTCA Document DO-254 and EUROCAE Document ED-80, Design Assurance Guidance for Airborne Electronic Hardware. FAA Position Paper.
- CAST-29, February 2007. Use of COST Graphical Processors (CPG) in Airborne Display Systems. FAA Position Paper.
- CAST-30, August 2007. Simple Electronic Hardware and RTCA Document DO-254 and EUROCAE Document ED-80, Design Assurance Guidance for Airborne Electronic Hardware. FAA Position Paper.
- CM-SWCEH-001, 2011. Development Assurance of Airborne Electronic Hardware, Certification Memorandum. EASA.
- FAA Order 8110.49, 2003. Software Approval Guidelines. U.S. Department of Transportation, Federal Aviation Administration.
- FAA Order 8110.105, U.S. Department of Transportation, Federal Aviation Administration.
- Kritzinger, D., 2006. Aircraft System Safety: Civil and Military Aeronautical Applications. Woodhead Publishing.
- RTCA/DO-178C, 2011. Software Considerations in Airborne Systems and Equipment Certification. RTCA Inc., Washington, DC.

-
- RTCA/DO-254, 2000. Design Assurance Guidance for Airborne Electronic Hardware. RTCA Inc., Washington, DC.
- RTCA/DO-330, 2011. Software Tool Qualification Considerations. RTCA Inc., Washington, DC.
- RTCA/DO-331, 2011. Model-based Development and Verification Supplement to DO-178C and DO-278A. RTCA Inc., Washington, DC.
- RTCA/DO-332, 2011. Object-oriented Technology and Related Techniques Supplement to DO-178C and DO-278A. RTCA Inc, Washington, DC.
- RTCA/DO-333, 2011. Formal Methods Supplement to DO-178C and DO-278A. RTCA Inc, Washington, DC.
- SAE AIR 6110, 2011. Contiguous Aircraft/System Development Process Example. SAE International.
- SAE ARP4754A, Guidelines for Development of Civil Aircraft and Systems, SAE Aerospace, <http://www.sae.org>.

Safety assessment strategy (with Goal Structuring Notation)

2

Strategy is a style of thinking, a conscious and deliberate process, an intensive implementation system, the science of insuring future success.

P. Johnson

2.1 Introduction

Many traditionally compiled Safety Assessments/Cases are inadequately planned, resulting in a report which is often disjointed, poorly expressed, and generally not easily understood by those who subsequently have to read or use it. Many readers of a comprehensive report are often overwhelmed by the bulk of effort, but left with a lingering feeling of ‘what has been missed?’

Goal Structuring Notation (GSN) provides a useful tool to plan and scope a safety assessment strategy. GSN explicitly represents the individual elements of any safety argument (requirements, claims, evidence and context) and, perhaps more significantly, the relationships that exist between these elements (i.e. how individual requirements are supported by specific claims, how claims are supported by evidence and the assumed context that is defined for the argument). When the elements of the GSN are linked together in a network they are described as a ‘goal structure’. This chapter provides some background information on GSN and looks to provide an example of how it can be used to formulate and guide the content of a Safety Assessment.

2.1.1 Background¹

Any convincing argument or report requires three elements:

- A distinct objective(s) or goal
- Supporting evidence
- A clearly discernable ‘thread’ or argument, which communicates the relationship between the evidence and objectives

This is illustrated as shown in [Fig. 2.1](#).

¹ Note: Section 2.1 in this chapter duplicates the content of Kritzinger (2006) Annex C, but this is necessary in order to support the Case Study in Section 2.3.

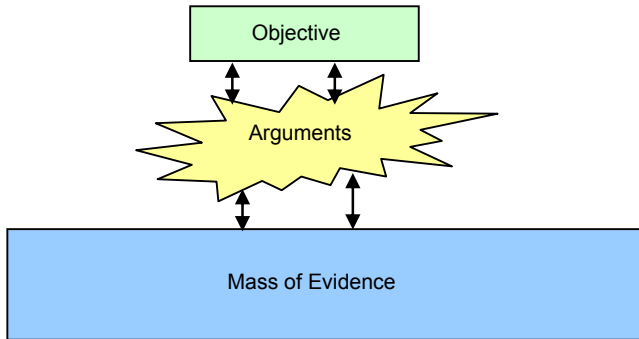


Figure 2.1 The three elements in an argument.

A Safety Assessment (or Safety Case Report) is no different: a ‘mass of evidence’ is generated during:

- system development and certification (e.g. stress analysis, electrical load analysis, fatigue test, flight tests, performance verification, FHA, ZHA, FTA, Regulatory Compliance Check-lists, etc.) and
- service experience (e.g. user confidence, training, etc.)

This ‘mass of evidence’ attempts to provide substantiation of our confidence in the safety of the system. The challenge is to tie all this evidence to our safety objective via a logical, systematic and complete safety argument. This argument can be provided in textual format – but is likely to be cumbersome and, for complex arguments, the ‘devil may get lost in the details’. Argument without supporting evidence is unfounded, and therefore unconvincing. Equally, evidence without argument is unexplained – it can be unclear what (or how) safety objectives have been satisfied ([Kelly & Weaver](#)). Complex arguments therefore often lend themselves to being presented graphically – especially if the picture ‘carries’ the reader through the argument with sufficient, judiciously placed ‘stepping stones’ (i.e. subgoals and subarguments down to an inevitable solution). GSN is logic-based methodology that can represent all aspects of the safety argument (i.e. requirements, claims, evidence and context) in an elegant and logical manner.

GSN was developed by the University of York over 10 years ago to support the development and presentation of safety arguments within Safety Cases. Since then, it has been adopted by a growing number of organisations within safety-critical industries (such as aerospace, railways and defence). The key benefit experienced by those adopting GSN is the improvement in comprehension of the safety argument afforded by the method, among all of the key project stakeholders (i.e. system developers, safety engineers, independent assessors and certification authorities). In turn, this has improved the quality of debate and discussion among stakeholders, which serves to reduce the time taken to reach agreement on the argument approaches being adopted.

The objective of this chapter is to provide a summary of how GSN methodology may be employed to define a Safety Strategy which provides a systematic (or methodical) way of assessing safety.

2.1.2 Aim of the Goal Structuring Notation argument

The aims of any GSN argument are:

- to plan the safety assessment strategy (i.e. how are we going to prove that the system is acceptably safe);
- to show how goals (claims about the system) are successively decomposed into subgoals until a point is reached where claims can be supported by direct reference to available evidence (solutions).

2.1.3 Scope of the Goal Structuring Notation argument

Any safety argument must be made with reference to the system level under consideration. Therefore, by using the system levels set out in the example Fig. 1.1:

- a Level 2 safety argument will be geared towards proving that a Line Replaceable Unit (LRU) contains no unsafe features and functions reliably;
- a Level 3 safety argument proves that an LRU can be safely integrated into a system;
- a Level 4 safety argument will be oriented towards proving the ‘as-designed’ safety baseline of the product (e.g. the aircraft);
- a Level 5 or 6 safety argument will be focussed on how the product is operated, maintained and monitored to ensure that the ‘as-designed’ level of safety is maintained, or even improved upon.

It is thus important to agree the target system level, expectations and requirements of the safety argument with the user/customer/regulatory authority before it is executed. This will scope the assessment.

2.2 Conducting a Goal Structuring Notation safety argument²

2.2.1 Goal Structuring Notation symbols











Most logical arguments are naturally hierarchical so that each argument can be broken down hierarchically into claims, arguments, subclaims, subarguments, and eventually evidence. This suits the application of GSN, which illustrates how goals are broken into subgoals, and eventually supported by evidence (solutions) while making clear the strategies adopted, the rationale for the approach (assumptions, justifications) and the context in which goals are stated.

The GSN argument thus follows the following process:

- Claim/objective (i.e. what we want to show)
- Argument/premise (i.e. why we believe, subject to any assumptions/justifications, the claims met), based on
- Evidence/solutions (e.g. tests, analyses, etc.)

² Section 2.2 duplicates the content of Kritzinger (2006) Annex C, but this is necessary in order to support the Case Study in Section 2.3.

The following symbols are often used in a GSN argument:

	Goals (claims, premises and conclusions are represented as goals)
	Strategies [an inference from one or more premises (also known as propositions or grounds) to a conclusion]
	Solutions (or evidence)
	Assumption/justification (i.e. the rationale for the approach)
	Context (in which goals are stated)
	Goal to be supported
	Goal to be instantiated (i.e. to be replaced with something ‘real’ at a later date)
	Solved by/supported by (links ‘goals’ to ‘arguments’ or subgoals or ‘solutions’)
	In context of (links to ‘assumptions’ and/or ‘justifications’)
	Model (i.e. leads to further information outside this GSN structure)

2.2.2 Modelling the process

The GSN safety argument can be formulated by following the process summarised in Fig. 2.2 (tailored from the six-step method (Kelly, 1999)). Each step is explained in Sections 2.2.3–2.2.9.

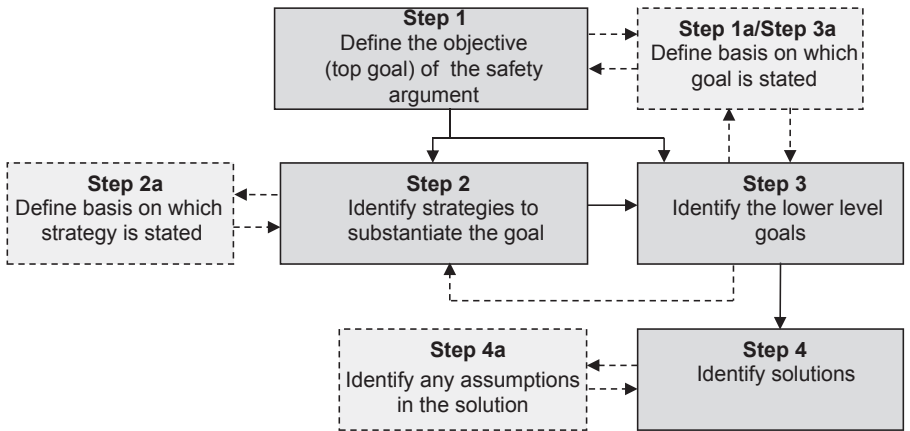


Figure 2.2 Formulating a Goal Structuring Notation safety argument.

2.2.3 Step 1: defining the objective

This top goal is the seed from which the argument can develop and represents the ultimate aim of the System Safety Assessment or Safety Case. In setting the goal, all stakeholders need to be in agreement regarding the scope (see Section 2.1.3).

When formulating any goal (see also [Section 2.2.7](#)), the following guidelines are useful:

- Goals should be phrased as positive propositions (i.e. a statement which can be said to be either true or false). [Kelly \(1999\)](#) advises that goals are best expressed in a '<Noun-phrase><Verb-phrase>' format (i.e. Noun-phrase identifies the subject of the goals, and the verb-phrase defines the predicate over the subject) (e.g. '*The sky is blue*').
- Be careful of oversimplification (e.g. '*System X is safe*' vs '*System X is acceptably safe within context Y*').

2.2.4 Step 1a (and Step 3a): contextualise the objective (if required)

Having presented a goal, make clear the basis on which that goal is stated. This is done by inserting information on Context, Assumptions, Justifications, and/or Models which ensure that the Goal is unambiguous.

2.2.5 Step 2: strategy to accomplish the objective

Work out how to substantiate the goals (i.e. what reasons are there for saying the goal is 'True'). This may require that you break the argument down into a number of smaller goals.

Two options are available:

- If the argument is implicit (i.e. suggested though not directly expressed), go to Step 3 (i.e. straight from goal to subgoal).
- If an explicit (i.e. stated clearly and in detail, leaving no room for confusion or doubt) argument is required, then insert it between the goal and the subgoal.

[Kelly \(1999\)](#) advises that strategies are best expressed in a in the Noun-phrase form: 'Argument by...<approach>' (e.g. '*Argument by consideration of historical data*').

2.2.6 Step 2a: contextualise the strategy (if required)

As per Step 1a and 3a, ask yourself what information is required in order to expand/fulfil the strategy outlined.

2.2.7 Step 3: define distinct objectives

Having identified an approach, it is necessary to set out the goals that fulfil that approach (i.e. by going into subgoals). Here it is important not to lose the argument by making to big a leap. As soon as the question '*why*' is raised, then consideration should be given to going 'up' a level to provide another 'stepping stone' to the argument.

[Kelly \(1999\)](#) advises to concentrate on the breadth of the argument first, before getting wrapped-up in the depth of it.

2.2.8 Step 4: identify solutions

Eventually, faced with a goal that does not need further expansion/refinement/explanation, add (or reference) the solution.

Ideally solutions should be Noun-phrases (e.g. ‘*Software Tests Result XYZ*’) (Kelly (1999)), but is it often useful to refer to reports/assessment where the solutions can be found (e.g. an FHA need not be taken from tabular format into individual GSN arguments for each functional failure mode).

2.2.9 Step 4a: contextualise the solution (if required)

Declare (or reference) any assumptions needed in the development of the solution. A solution might be ‘*Not applicable*’, in which case a ‘*justification*’ will be required.

2.3 The Case Study

2.3.1 Case study using CS25.1309

Consider the case study in Section 1.4. Let us assume the objective of the Safety Assessment is to prove compliance to CS25.1309:

CS25.1309 (Amendment 17): Equipment, systems and installations

‘The requirements of this paragraph, except as identified below, are applicable, in addition to specific design requirements of CS-25, to any equipment or system as installed in the aeroplane. Although this paragraph does not apply to the performance and flight characteristic requirements of Subpart B and the structural requirements of Subparts C and D, it does apply to any system on which compliance with any of those requirements is dependent. Certain single failures or jams covered by CS 25.671(c)(1) and CS 25.671(c)(3) are excepted from the requirements of CS 25.1309(b)(1)(ii). Certain single failures covered by CS 25.735(b) are excepted from the requirements of CS 25.1309(b). The failure effects covered by CS 25.810(a)(1)(v) and CS 25.812 are excepted from the requirements of CS 25.1309(b). The requirements of CS 25.1309(b) apply to powerplant installations as specified in CS 25.901(c).

- (a) The aeroplane equipment and systems must be designed and installed so that:
 - (1) Those required for type certification or by operating rules, or whose improper functioning would reduce safety, perform as intended under the aeroplane operating and environmental conditions.
 - (2) Other equipment and systems are not a source of danger in themselves and do not adversely affect the proper functioning of those covered by sub-paragraph (a)(1) of this paragraph.

CS25.1309 (Amendment 17): Equipment, systems and installations—cont'd

- (b) The aeroplane systems and associated components, considered separately and in relation to other systems, must be designed so that:
 - (1) Any catastrophic failure condition
 - (i) is extremely improbable; and
 - (ii) does not result from a single failure;and
 - (2) Any hazardous failure condition is extremely remote; and
 - (3) Any major failure condition is remote
- (c) Information concerning unsafe system operating conditions must be provided to the crew to enable them to take appropriate corrective action. A warning indication must be provided if immediate corrective action is required. Systems and controls, including indications and annunciations must be designed to minimise crew errors, which could create additional hazards.
- (d) Electrical wiring interconnection systems must be assessed in accordance with the requirements of CS 25.1709.

And where CS25.1709 states:

CS25.1709 (Amm 17): System Safety; EWIS

EWIS must be designed and installed so that:

- (a) Each catastrophic failure condition
 - (1) is extremely improbable; and
 - (2) does not result from a single failure;
- and
- (b) Each hazardous failure condition is extremely remote.

With due notice of the GSN challenges referred to in [Section 2.5](#), [Fig. 2.4](#) is intended to provide a starting point for anyone trying to compile a GSN argument for a [CS25.1309](#) compliant System Safety Assessment. This argument is by no means definitive and appropriate in all circumstances, but should provide a stimulus for debate.

2.3.2 Safety criteria for [CS25.1309](#)

The objective of [CS25.1309](#) is to ensure an acceptable safety level for equipment and systems as installed on the aeroplane. A logical and acceptable inverse relationship must exist between the Average Probability per Flight Hour and the severity of failure condition effects as shown in [Fig. 2.3](#), such that:

1. Failure conditions with No Safety Effect have no probability requirement.
2. Minor failure conditions may be Probable.
3. Major failure condition must be no more frequent than Remote.
4. Hazardous failure condition must be no more frequent than Extremely Remote.
5. Catastrophic failure condition must be Extremely Improbable.

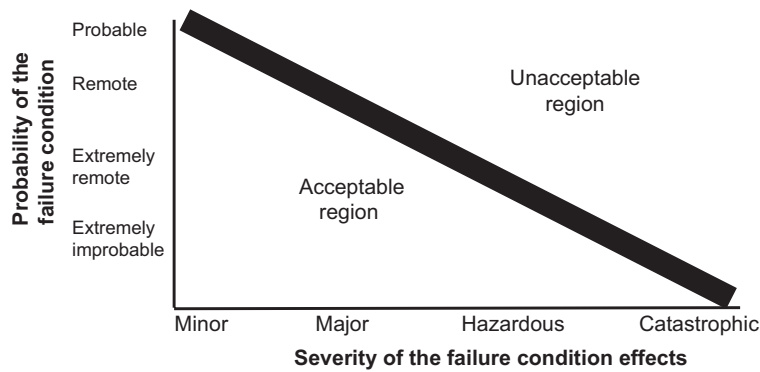


Figure 2.3 Probability and severity of failure condition effects.
AMC25.1309 Amm16.

The safety objectives associated with failure condition are described in [Table 2.1](#).

2.3.3 Case study if using FAR25.1309

The following table provides a checklist against the equivalent requirement of [FAR25.1309](#) (Amendment 25.123):

FAR 25.1309	Requirement	CS25.1309 equivalent	GSN reference
(a)	The equipment, systems, and installations whose functioning is required by this subchapter, must be designed to ensure that they perform their intended functions under any foreseeable operating condition	(a)	Goal 2 Goal 3
(b)(1)	The airplane systems and associated components, considered separately and in relation to other systems, must be designed so that— (1) The occurrence of any failure condition which would prevent the continued safe flight and landing of the airplane is extremely improbable, and	(b)(3)	Strategy 3
(b)(2)	(2) The occurrence of any other failure conditions which would reduce the capability of the airplane or the ability of the crew to cope with adverse operating conditions is improbable	b(4)	Strategy 3

Continued

FAR 25.1309	Requirement	CS25.1309 equivalent	GSN reference
(c)	Warning information must be provided to alert the crew to unsafe system operating conditions, and to enable them to take appropriate corrective action. Systems, controls, and associated monitoring and warning means must be designed to minimise crew errors which could create additional hazards.	(c)	Strategy 4
(d)(1)	Compliance with the requirements of section (b) of this table must be shown by analysis, and where necessary, by appropriate ground, flight, or simulator tests. The analysis must consider— (1) Possible modes of failure, including malfunctions and damage from external sources.	(b)	Goal 1 Goal 3
(d)(2)	(2) The probability of multiple failures and undetected failures.	(b)	Strategy 12
(d)(3)	(3) The resulting effects on the airplane and occupants, considering the stage of flight and operating conditions, and	(b)	Strategy 6
(d)(4)	(4) The crew warning cues, corrective action required, and the capability of detecting faults	(c)	Strategy 4
(e)	In showing compliance with sections (a) and (b) of this table with regard to the electrical system and equipment design and installation, critical environmental conditions must be considered. For electrical generation, distribution, and utilisation equipment required by or used in complying with this chapter, except equipment covered by Technical Standard Orders containing environmental test procedures, the ability to provide continuous, safe service under foreseeable environmental conditions may be shown by environmental tests, design analysis, or reference to previous comparable service experience on other aircraft	(a)(1)	Strategy 5
(f)	EWIS must be assessed in accordance with the requirements of §25.1709	(d)	Strategy 3

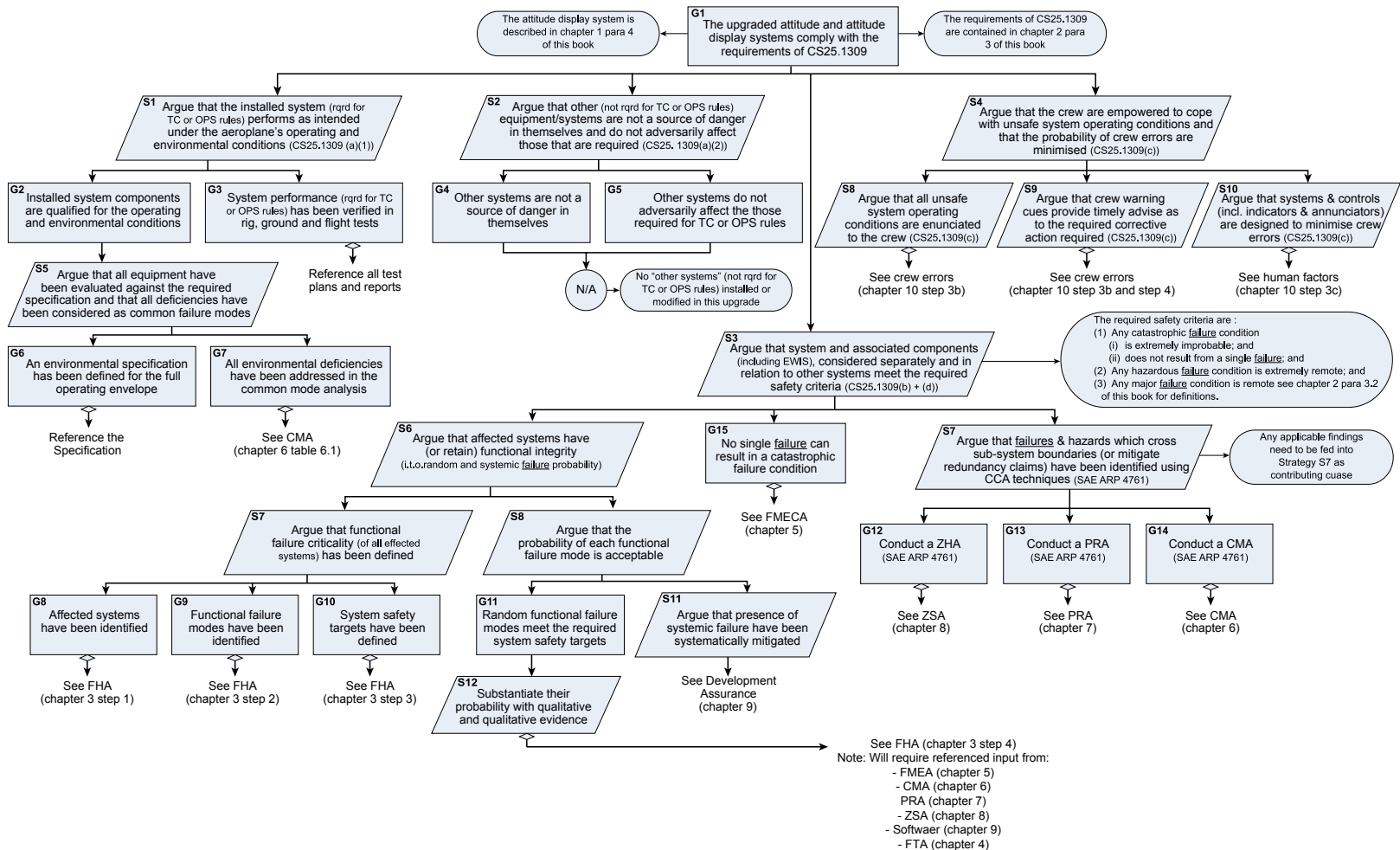


Table 2.1 Safety criteria for **CS25.1309** (Amendment 17)

Failure classification	No safety effect	Minor	Major	Hazardous	Catastrophic
Effect on aeroplane	No effect on operations capabilities or safety	Slight reduction in functional capabilities or safety margins	Significant reduction in functional capabilities or safety margins in adverse operating conditions	Large reduction in functional capabilities or safety margins in adverse operating conditions	Normally with hull loss
Effect on occupants (excl flight crew)	Inconvenience	Some physical discomfort	Physical distress, possibly including injuries	Serious or fatal injury to a small number of passengers or cabin crew	Multiple fatalities
Effect on flight crew	No increase in flight crew workload	Slight increase in workload, well within their capabilities	Physical discomfort or a significant increase in workload or impairing crew efficiency	Physical distress or excessive workload impairs ability to perform tasks accurately or completely	Fatalities or incapacitation
Allowable qualitative probability	No probability requirement	Probable	Remote	Extremely remote	Extremely improbable
Allowable quantitative probability (average probability per flight hour)	No probability requirement	In the order of 10^{-3} to 10^{-5a}	In the order of 10^{-5} to 10^{-7}	In the order of 10^{-7} to 10^{-9}	In the order of $<10^{-9}$

^aA numerical probability range is provided here as a reference. The applicant is not required to perform a quantitative analysis, nor substantiate by such an analysis, that this numerical criteria has been met for Minor failure condition. Current transport category aeroplane products are regarded as meeting this standard simply by using current commonly accepted industry practice. Tailored from AMC25.1309 para 7 and Fig. 2.

2.4 Discussion

The high-level safety argument must be developed as early as possible as it provides a clear picture of the methodology by which safety will be substantiated. If agreed with the applicable customer/authority, this argument will scope all future safety-related activities, e.g.:

- generating lower level arguments and solutions or,
- if the high-level argument shows (with justification) that a particular solution is not applicable, then no further action is required for that solution.

Possible approaches to inclusion of GSN argument in Safety Assessment include:

- In full as an appendix/annex to the report.
- As a chapter/paragraph in a report, which guides the reader through a potentially confusing and intimidating report.
- As an ‘Executive Summary’ at beginning of the report.
- As separate, stand-alone Index Document (i.e. to link separate documents together).

However, one should remember that:

However beautiful the strategy, you should occasionally look at the results.

Winston Churchill

2.5 Conclusions

GSN is a coherent method for the expression of claims, supporting strategy and sub-claims, assumptions, justification, and evidence. By using GSN in developing a Safety Assessment, you will be introducing a confidence in the stated claims that is hard to establish by other means. GSN has proven to be an excellent tool for the development of strategy, but, as with all methodologies it is useful to be cognisant of both its associated advantages and limitations:

GSN is most useful wherever:

- there is most uncertainty about the argument (i.e. key claims and evidence);
- the argument is currently confused or is overcomplex;
- there is disagreement about the approach taken to the Safety Assessment and how the variety of evidence all relate to support compliance to the regulatory requirement;
- the consequences of having a wrong argument are high.

Advantages of using a GSN safety argument include:

- improved comprehension of existing arguments;
- useful way to define Safety Assessment/Case strategy;
- easy to read (even to a novice);
- forces a logical argument (i.e. from a goal to its solution(s), identifying holes in an argument);
- positively identifies assumptions;

- removes ambiguity (i.e. you have to define measurable goals);
- assist in managing programme risk (i.e. solution planning and prioritising);
- ease to audit;
- prevent duplication of solutions;
- prevents unnecessary work (e.g. if not required by a goal);
- defines scope of work, so assist in planning and budgeting;
- arguments can be reused in another project.

However, GSN is not without its limitations:

- Arguments are always subjective, so every person will compile a GSN differently. Significant amounts of time can be expended debating an argument instead of getting on with the required solutions.
- It takes a lot of effort to develop the arguments, and there is no substitute for experienced skill to do so efficiently.
- The inexperienced practitioner can easily enter into too much detail too soon (e.g. it is often more efficient to stop the argument at a solution, which contains a Compliance Matrix or an FTA, than to try to replicate these within the GSN). Therefore, the assessor may elect to restrain GSN to a top-level argument only and not to repeat each finding which exist in tabular format (e.g. such as in a Functional Hazard Assessment)
- GSN is not as user-friendly in hard copy format, because a complex and large GSN argument can stretch over many dozens of pages. Many users thus prefer an 'e-Safety Case' which allows for quick and easy referencing via hyperlinks to appropriate branched of the GSN argument. However, the e-Safety Case comes with its own challenges, not least of which is getting non-GSN practitioners to understand it enough to be able to approve/authorise its release.

References

- CS25, July 2015. Certification Specifications and Acceptable Means of Compliance for Large Aeroplanes. Amendment 17. European Aviation Safety Agency, Cologne.
- FAR25, November 2007. Airworthiness Standards: Transport Category Airplanes. Amendment 25-123. Federal Aviation Authority, Washington.
- Kelly, T., Weaver R. The Goal Structuring Notation – a Safety Argument Notation, Department of Computer Science and Department of Management Studies, University of York, YO10 5DD UK. <https://www-users.cs.york.ac.uk/tpk/dsn2004.pdf>
- Kelly, T., September 1999. A Six-Step Method for the Development of Goal Structures. https://www.researchgate.net/publication/245072237_A_Six-Step_Method_for_the_Development_of_Goal_Structures.
- Kritzinger, D., *Aircraft System Safety: Civil and Military Aeronautical Applications*, Woodhead Publishing, 2006

Further reading

The application of GSN in the safety environment has been developed and refined by Dr Tim Kelly, whose doctoral research at the University of York focused upon safety argument presentation, maintenance and reuse. For more information, see: <http://safetyengineering.wordpress.com/2008/04/04/the-goal-structuring-notation-gsn/>.

Functional Hazard Analysis

3

Our will is a function regulated by reflection; hence it is dependent on the quality of that reflection.

Author unknown

3.1 Introduction

3.1.1 Background

Each aircraft system has an overall goal to achieve. This is the function of the system, i.e. the action which exchanges input for output as illustrated in Fig. 3.1.

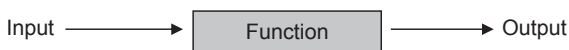


Figure 3.1 Functional exchange.

The functional approach stems from the fact that any system (or item) is merely the embodiment of a set of functions. A Functional Hazard Analysis (FHA) is a systematic, comprehensive top-down examination of each function of the system to consider the effects and probability of a functional failure, malfunction and/or normal response to unusual or abnormal external factors [AMC25.1309 para 10b(1)].

The FHA proves¹ the functional integrity of a particular system and is often generated to support compliance statements against regulations such as the EASA Certification Specification 25.1309 (or the FAA equivalent FAR Part 25.1309). AMC25.1309 [Amm16 para 10b] advises that

‘before a detailed safety assessment is proceeded with a Functional Hazard Assessment (FHA) of the aeroplane and system functions to determine the need for and scope of subsequent analysis should be prepared’.

3.1.2 Aim of the Functional Hazard Analysis

It can be contended that the aim of a:

- preliminary FHA is to determine how safe a particular system should be by specifying the safety objectives² (i.e. specifying the minimum requirements) of the system relative to the identified functional failure modes;

¹ Each functional failure condition is allocated an allowable probability target in accordance with the Safety Criteria. This probability target is the design objective for the completed system.

² The preliminary FHA is normally conducted before the system exists; that is during the early design phase, where it is concerned [AMC25.1309 Amm16 para 8a] with the operational vulnerabilities of the system rather than with the detailed hardware analysis.

- *final* FHA is to prove that the minimum safety targets have indeed been accomplished in the implemented (i.e. designed and installed) system.

3.1.3 Objectives of the Functional Hazard Analysis

Similarly, it can be argued that the objectives³ of the FHA are to:

- identify the required functions of the system under consideration,
- assess the significant failure conditions associated with loss/malfunction of this functionality,
- classify failure conditions according to the severity of their consequence,
- allocate minimum safety targets (e.g. failure probabilities) in accordance with the agreed safety criteria,⁴
- reference (or propose) the verification action which (will) show that the safety targets are accomplished.

3.1.4 Scope of the Functional Hazard Analysis

The scope of any particular FHA is dependent on the system level (refer Fig. 1.1) under consideration, as discussed in [Section 3.2.1](#).

The scope of the FHA extends throughout the development life cycle:

- For the purposes of the Preliminary System Safety Assessment (PSSA), the FHA considers functional failure modes only (i.e. not their probability of occurrence). Failure conditions identified at this level are not dependent on the way the functions are implemented or the system architecture.
- As the design process matures, the system architecture is incorporated into the occurrence probability declaration to complete the FHA in the final SSA.

3.2 Conducting the Functional Hazard Analysis

The FHA is a ‘top-down’ process⁵ and can be illustrated as in [Fig. 3.2](#). Each step is explained in [Sections 3.2.1–3.2.4](#). Some assessors prefer to have issue a stand-alone FHA (i.e. not part of the SSA). With reference to the safety assessment strategy in [Fig. 2.4](#), this process assumes that the FHA is integrated into the SSA.

Note that the process is iterative as it is conducted at each system development level (refer [Figs 1.1 and 1.3](#)) above that of component/item (for which an FMEA is a more appropriate safety assessment tool).

3.2.1 Step 1: define the scope of the Functional Hazard Analysis

Initially, we need to define the scope of aircraft functions that will be subject to the FHA. To commence this process, it is important to define the target ‘system level’. So,

³ Objectives are measurable results (in contrast to aims which are general statement of intent).

⁴ Thus, in this context, safety objectives specify the maximum tolerable probability for the occurrence of a hazard of a given severity.

⁵ ‘Top-down’, i.e. the FHA is often first carried out for the whole aircraft – working from a description of aircraft functions. Then, following allocation of functions to aircraft systems, the FHA is then performed again for each subsystem (Wilkinson and Kelly, 2005). A ‘bottom-up’ approach would typically be in the format of an FMECA.

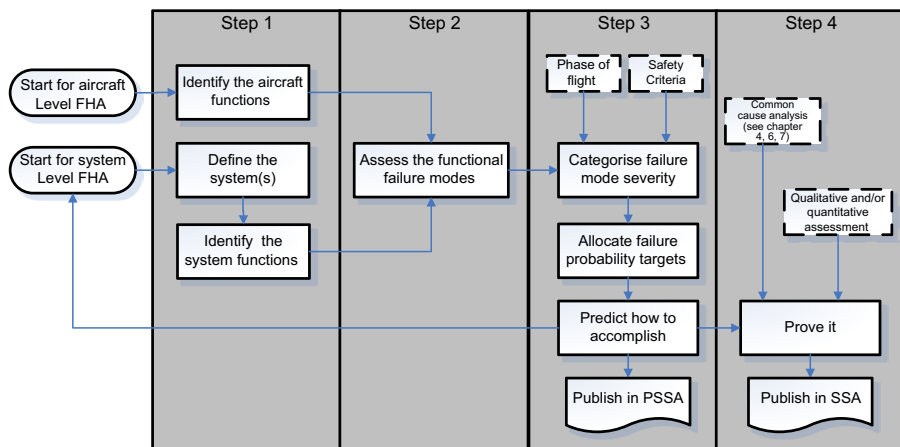


Figure 3.2 An FHA process flow.

to use the example in Fig. 1.1, the FHA's top-down approach can start at either the product (e.g. aircraft) level, or the product subsystem level (e.g. the avionic system)⁶:

- The aircraft-level FHA identifies and classifies failure conditions associated with aircraft functions (i.e. Level 4 functions in the example Fig. 1.1) and is used to support identification of possible multiple system failure conditions.⁷ The clarifications of these failure conditions subsequently establish the safety requirements (or derived safety targets) that the aircraft must meet.
- The system/subsystem-level FHA (i.e. Level 3 in the example Fig. 1.1) considers failure or combinations of system failures that affect that system's function. The clarifications of these failure conditions establish the safety requirements that the system must meet.

⁶ AMC25.1309 [Amm5 para 10] advises that 'Depending on the extent of functions to be examined and the relationship between functions and systems, different approaches to Functional Hazard Assessment may be taken. Where there is a clear correlation between functions and systems, and where system, and hence function, interrelationships are relatively simple, it may be feasible to conduct separate Functional Hazard Assessments for each system, providing any interface aspects are properly considered and are easily understood. However, where system and function interrelationships are more complex, a top-down approach, from an aeroplane-level perspective, should be taken in planning and conducting Functional Hazard Assessments'. The assessment of any particular hardware or software item is not the goal of the system-level FHA [SAE ARP4761 para 3.2]. If a particular new item/subsystem is added to an existing system, the system-level FHA considers the implications on that system. The effects of failure conditions on the system level leads to the allocation of safety objectives to the item/subsystem. The FHA thus shows the effect of configuration changes to the integrated system.

⁷ Because the aircraft-level FHA is an analysis of the aircraft functions at the highest level, most of these functions will be the same for most aircraft types and may include providing altitude control; providing ground directional control; providing adequate lift; providing adequate thrust; providing flight critical information (which includes fault monitoring); providing fire and explosion protection; maintaining a habitable environment; maintaining a safe external environment on the ground; providing adequate crash survivability; maintaining structural integrity; providing adequate stowage of cargo; providing adequate range (which includes in-flight refuelling); providing for a safe landing; providing adequate self-defence mechanisms; providing adequate offensive mechanism; etc. See SAE ARP4754A (pages 26–27) for more examples of aircraft-level functions.

Both use the same principles (as illustrated in [Fig. 3.2](#)). With due consideration of the functional issues discussed below, it is for the assessor(s) to negotiate with the regulator (or the customer who is doing the aircraft integration) as to whether they only use the system/subsystem-level FHA to prove system integrity, or both to prove the system's integrity as integrated onto the aircraft (e.g. when upgrading an aircraft system, an aircraft-level FHA might not be needed if the system-level FHA can derive its safety targets without it).

The FHA should identify all functions associated with the system under study. The equipment's functionality, capability, and limitations must be deliberately incorporated (i.e. there must be no hidden functionality, as a function not assessed might have undesired failure modes).

Each system function should also be examined with respect to functions performed by other aeroplane systems, because the loss of different but related functions provided by separate systems may affect the severity of failure conditions postulated for a particular system. Consideration must be given to:

- Principal, subsidiary and nonobvious functions: In conducting an FHA, one must resist restricting its scope to whether a system or function is required by any specific regulation. Some systems required by specific regulations (e.g. transponders, position lights, and public address systems) may, in Steps 2 and 3, have the potential for only Minor failure conditions. Conversely, other systems not required by any specific regulation (such as Flight Management Systems and Automatic Landing Systems) may have potentially Major, Hazardous or Catastrophic failure conditions [refer FAA AC23.1309–1C page 20];
- Functions internal to the considered level (internal functions): At the aircraft level, these are main functions of the aircraft and those functions exchanged between the internal systems of the aircraft. At the system level, these are functions of the specific system under consideration and those functions exchanged between the internal equipment of the system [[SAE ARP4761](#) para A3.1.2];
- Functions external to the considered level (i.e. exchanged functions): At the aircraft level, these are functions that interface with other aircraft and/or ground systems. At the system level, in all cases, these are the functions which are either provided by or to other systems, including:
 - warning functions or functions that provide operator indications and control;
 - functions that protect against hazards;
 - functions provided by human operators;
 - functions that moderate the effects of failure of other functions.

To facilitate the FHA (in terms of system hierarchy and interrelated functionality), it is recommended that functional trees (see examples in [Fig. 3.5](#) and [3.6](#)) are constructed to show the tiered relationship between aircraft functions and/or system component functions. Functional trees, while deceptively simple to look at, are in fact quite tricky⁸ to compile, and for complex systems, such trees may fail to adequately communicate or portray complex functional interaction. In some cases the assessor may elect to supplement these trees with additional data such as the example matrix in [Table 3.1](#).

⁸ Identification of functions (and the compilation of the functional tree) is further complicated by the fact that the preliminary FHA considers only functional hazards and, when issued, often receives criticism for hazards not yet identified (but which will be once hazard identification techniques such as the ZSA, CCA and PRA are employed) to the maturing design.

Table 3.1 Complex functional interaction

Systems	Functions				
	Function 1	Function 2	Function 3	Function 4	Function 5
System 1	X	X		X	X
System 2	X			X	X
System 3			X		
System 4		X		X	X
System 5				X	
System 6		X			X

3.2.2 Step 2: assess the functional failure modes

Identify and describe the possible failure conditions associated with these functions. These failure modes should include availability (i.e. continuity) and integrity (i.e. correctness) of function. These failure modes can be user-defined at will, although standardising on terminology (i.e. providing consistence of terms and expression) will help with the FHA’s compilation and review. To this purpose, the following standardised terms may be useful:

- Loss of Function (Annunciated and Unannunciated)⁹ such as landing gear failing to extend;
- Partial Loss of Function (Annunciated and Unannunciated) such as low hydraulic pressure or reduced electrical power supply;
- Function provided when not required (Annunciated and Unannunciated) such as uncommanded ejection or uncommanded flight control movement;
- Incorrect function (Annunciated and Unannunciated) such as incorrect altitude or heading display.

3.2.3 Step 3: derive safety targets

3.2.3.1 Classify the failure effects

The assessor must then consider the worst-case effects (i.e. those occurring prior to any mitigation) of the failure mode as experienced in the specific flight phase and with due regard to environmental conditions.

The following factors should be considered (and appropriately declared if used) when determining the severity of a failure condition:

- time to detection (i.e. when detected);
- failure recognition provided (i.e. how detected);
- how would the pilot react (i.e. what to do) to cope with the failure and the timeliness thereof, with due consideration of crew workload and relative priority to other tasks (see Chapter 10 for more information);

⁹ An annunciated failure condition is one which fails ‘actively’ (i.e. in such a manner as to inform crew of the failure, either by virtue of indicators or via aircraft behaviour obviously attributable to it). In contrast, an unannunciated failure is potentially a latent or passive failure condition, or one that is misleading. AMC25.1309 [para 5] states that a failure is latent until it is made known to the flight crew or maintenance personnel.

- flight phase in which the failure occurs (e.g. take-off, cruise, descent, landing, etc.) and operational conditions (e.g. IFR conditions); in most cases, it may only be necessary to apply the failure mode to the most critical flight phase as effort will soon become obviously duplicated in the FHA;
- which severity conditions require further validation and/or verification (see Fig. 1.3), especially with the flight crews who will actually be exposed to those failure conditions. Failures may need to be simulated to justify the assumptions made in the FHA.

These effects can then be compared to the severity category defined in the applicable safety criteria. Table 3.2 shows the goal/failure-based safety criteria typically used by EASA for large civil transport aircraft.

When considering the worst-case effect of a failure condition, it is recommended that the assessor bears the following in mind:

1. The FHA should, up to this point (Step 3), not consider the system architecture. One must remember that the system architecture influences aircraft- or system-level failure probability and not its failure severity.
2. Consider factors which might alleviate or intensify the direct effects of the initial failure condition. Include consequent or related conditions (e.g. the presence of smoke, acceleration vectors, interruption of communication, interference with cabin pressurisation, etc.) which may affect the ability of the crew to deal with direct effects.
3. When assessing the consequences of a given failure condition, account should be taken of the warnings given, the complexity of the crew action (see Fig. 10.1) and the relevant crew training required to respond to the failure condition. It is hopefully obvious to the reader that operators (e.g. pilots and cabin crew) should form an integral part of such discussions as many Safety Assessors have little to no operational experience.

3.2.3.2 *Allocate safety targets based on the severity*

In accordance with the applicable safety criteria, each failure mode classification can now be allocated a quantitative and/or a qualitative safety target based on its level of criticality. Table 3.3 shows the safety criteria typically used for large civil transport aircraft. These safety targets will then become a design objective (or safety goal) for the system architects to achieve.

3.2.3.3 *Predict how the targets will be accomplished*

The targets set above need to be accomplished within the actual system architecture and should thus form part of the relevant aircraft/system/item specifications (see Fig. 1.3). These specifications can subsequently be referenced from the FHA as evidence of intent.

The FHA then needs to identify the qualitative and/or quantitative methods that will be used to verify compliance with the failure condition requirements and (if necessary) allocate responsibility to each outstanding verification (see Fig. 1.3) action. The level of detail needed for the various safety assessment activities is dependent on the aircraft-level condition classification, the degree of integration, and the complexity of the system implementation. Some authorities provide useful guidance in this regard; see the decision tree in Fig. 3.3.

Table 3.2 Typical failure severity criteria

Failure classification	No safety effect	Minor	Major	Hazardous	Catastrophic
Effect on aeroplane	No effect on operations capabilities or safety	Slight reduction in functional capabilities or safety margins	Significant reduction in functional capabilities or safety margins in adverse operating conditions	Large reduction in functional capabilities or safety margins in adverse operating conditions	Normally with hull loss
Effect on occupants (excluding flight crew)	Inconvenience	Some physical discomfort	Physical distress, possibly including injuries	Serious or fatal injury to a small number of passengers or cabin crew	Multiple fatalities
Effect on flight crew	No increase in-flight crew workload	Slight increase in workload, well within their capabilities	Physical discomfort or a significant increase in workload or impairing crew efficiency	Physical distress or excessive workload impairs ability to perform tasks accurately or completely	Fatalities or incapacitation

Note: These criteria are based on *failure* severity (not *accident* severity). There are many functional failures which will not necessarily result in an accident. The risk/accident-based criteria (e.g. from Def Stan 00-56 or MIL-STD-882) considers the severity of various types of accidents and cannot directly allocate a severity to a failure mode which, for instance, leads to a 'significant increase in crew workload or conditions that impair crew efficiency'. For more information, see Kritzinger (2006), Chapters 4 and 5. This table is tailored from AMC25.1309, Amendment 17, para 7 and Fig. 2.

Table 3.3 Typical safety objectives

Failure classification	No safety effect	Minor	Major	Hazardous	Catastrophic
Allowable qualitative probability	No probability requirement	Probable	Remote	Extremely remote	Extremely improbable
Qualitative probability definition		Failure conditions anticipated to occur one or more times during the entire operational life of each aeroplane	Failure conditions unlikely to occur to each aeroplane during its total life, but which may occur several times when considering the total operational life of a number of aeroplanes of the type	Failure conditions anticipated to occur to each aeroplane during its total life but which may occur a few times when considering the total operational life of all aeroplanes of the type	Failure conditions so unlikely that they are not anticipated to occur during the entire operational life of all aeroplanes of one type
Allowable quantitative probability (average probability per flight hour)	No probability requirement	In the order ^a of 10^{-3} to 10^{-5}	In the order of 10^{-5} to 10^{-7}	In the order of 10^{-7} to 10^{-9}	In the order of $<10^{-9}$
Required Functional Development Assurance Level ^b	FDAL E	FDAL D	FDAL C	FDAL B	FDAL A

^aA numerical probability range is provided here as a reference. The applicant is not required to perform a quantitative analysis, nor substantiate by such an analysis, that this numerical criteria has been met for Minor failure conditions. Current transport category aeroplane products are regarded as meeting this standard simply by using current commonly accepted industry practice.

^bWith reference to Fig. 1.3: FDAL allocation is the left side of the V&V model. FDAL satisfaction is the right side of the V&V model. See [Chapter 9](#) for more information on Development Assurance. This table is tailored from AMC25.1309, Amendment 17, para 7 and Fig. 2.

Note: These criteria are based on *failure* probability (not *accident* probability). The accident/risk-based criteria (e.g. from Def Stan 00-56 or MIL-STD-882) consider the probability of the *consequence* (i.e. the accident) of various types of hazards. It is in this aspect that the accident/risk-based approach is difficult to apply, because a system target can only be set after an accident risk is defined and the accident sequence is fully populated with the probability of each contributing cause/event. For more information, see Kritzinger (2005), Chapters 4 and 5. When considering the use of risk (i.e. the product of accident probability and accident severity), please do keep in mind:

- that the functional failure mode is only one link in the accident chain (see [Chapter 11](#) para 2.2 and Fig. 11.3);
- that the challenge faced by designers of a system to set functional integrity targets (which will be internationally recognised) during the bid stages of a programme.

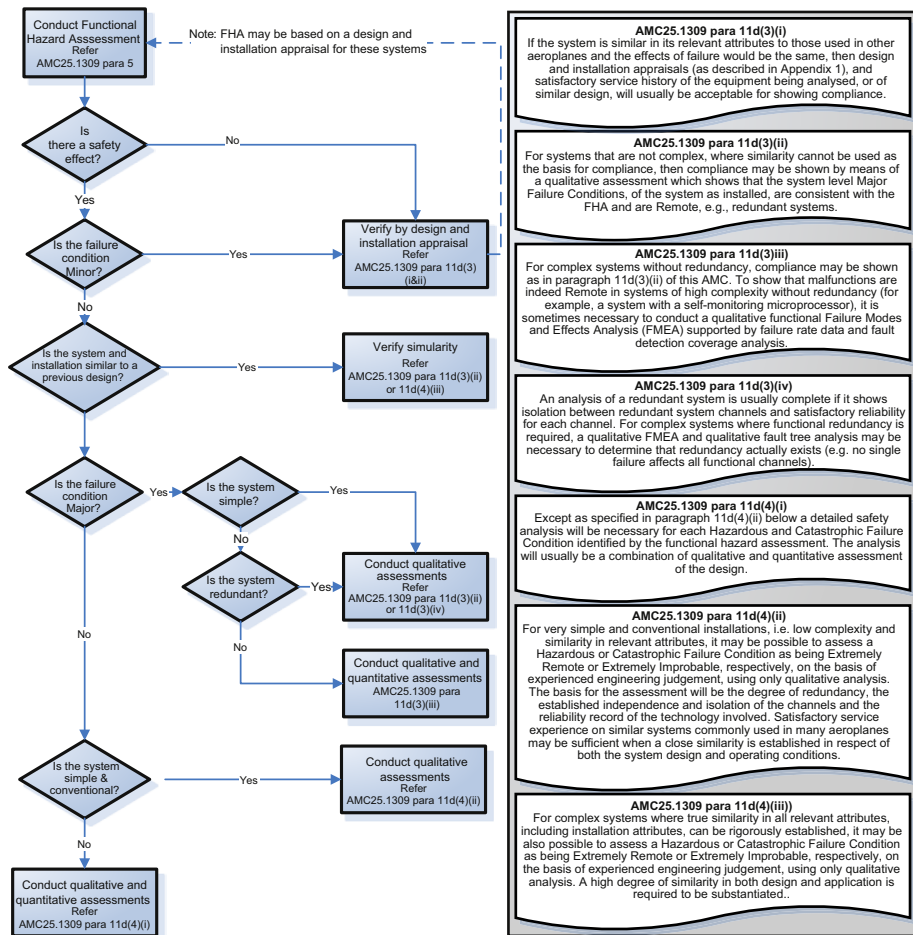


Figure 3.3 Depth of analysis.

Tailored from AMC25.1309 Amendment 17 Annex A. For a larger print of this illustration, please see www.aircraftsystemsafety.com.

3.2.3.4 Publish Preliminary System Safety Assessment

The objective of the PSSA is to list all the failure conditions from the FHA and to demonstrate how the intended system will meet the qualitative and quantitative objectives allocated to each functional failure mode. Accordingly, the PSSA should typically [refer, inter alia, SAE ARP4745A para 5.1.2] contain the following information:

- derived system safety requirements such as redundancy requirements, Development Assurance Levels, etc. (see Fig. 1.3);
- the need for alternative protective strategies (e.g. partitioning, built-in-test, dissimilarity, monitoring, safety maintenance task intervals, etc.);
- the Safety Strategy (Chapter 2);

- the preliminary FHA (Chapter 3);
- Design and Installation Guidelines from the ZSA (Chapter 8).

3.2.4 Step 4: prove safety target accomplishment

Implement the system’s design by:

- defining its architecture;
- ensuring that all contributing causes are identified and no common cause failure modes are introduced during the actual implementation (see Fig. 3.2); if any are found, it may be necessary to redefine the system’s architecture or logic;
- validate the architecture by determining, via qualitative/quantitative assessment (see Fig. 3.4), the actual probability of the top failure condition failure occurrence. These data become the evidence that the safety targets have been met and should include (or reference) any appropriate recommendations¹⁰ and limitations.

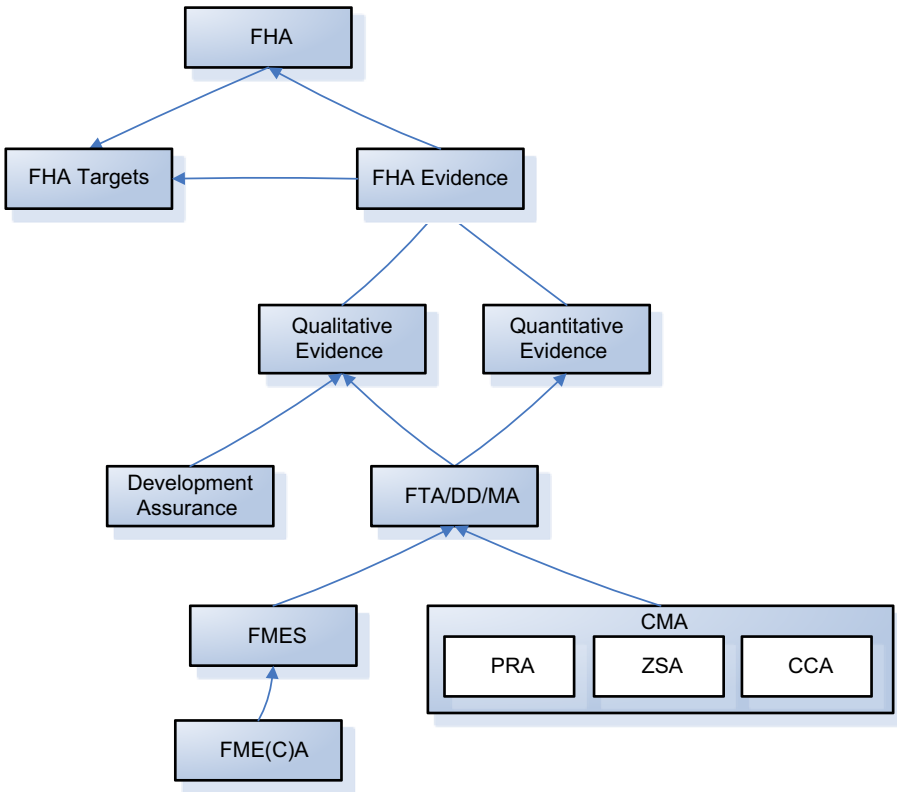


Figure 3.4 Relating the FHA to other assessment tools.

¹⁰ Failure Conditions involving other than instinctive crew actions may influence the flight crew performance. Training requirements may need to be specified in some cases. See Chapter 11 for more information.

Fig. 3.4 is not definitive, but shows how some of the assessment tools (in [SAE ARP4761](#) and the remaining chapters of this book) interrelate. For instance, an item-level Failure Modes and Effects Analysis (FMEA) is performed and is summarized into the Failure Modes and Effects Summary (FMES) to support the failure rates of the failure modes considered in the item FTA. The system FMEA is summarized into the system FMES to support the failure rates of the failure modes considered in the system FTA. The system is reviewed via FTA to identify the failure modes and probabilities used in the aircraft FTA. The aircraft FTA is used to establish compliance with the aircraft-level failure conditions and probabilities described by the aircraft FHA.

3.3 The Case Study

All safety requirements should be traceable through each system level, and all safety probability declarations should be validated against the requirements (see Fig. 1.3). We will now consider the case study from [Chapter 1](#) to demonstrate the process flow from the aircraft-level FHA to the system-level FHA.

For the auspices of this example, we will concentrate on the aircraft function ‘Display Aircraft Altitude’ (which is a subfunction of ‘Provide Spatial Orientation’), assuming that the customer has asked us to replace the primary displays, but retain the existing analogue standby display, which has an MTBF of 1000 hours.

3.3.1 Aircraft level Step 1: define the scope of the FHA

Consider the example functional tree in [Fig. 3.5](#), where we have elected to break certain aircraft-level functions down into three tiers. The functional tree can have as many tiers deemed necessary. For the aircraft-level FHA, a useful guide would be to go down as low as required until just before you actually identify an aircraft system (in which case you are moving towards a system-level FHA). A starting point for the aircraft level Functional Tree could be the Air Transport Association’s (ATA) Chapter. The ATA Chapter numbers provide a common referencing standard for all commercial aircraft documentation. This commonality permits greater ease of learning and understanding for pilots and engineers alike. The unique aspect of the chapter numbers is its relevance for all aircraft so, for instance, ATA 26 addresses Fire Protection in any civil transport aircraft type.

3.3.2 Aircraft level Step 2: assess the functional failure modes

Function failure modes can now be assessed in normal paragraph format (using service experience, engineering and operational judgement) or a top-down qualitative examination of each function performed by the system.

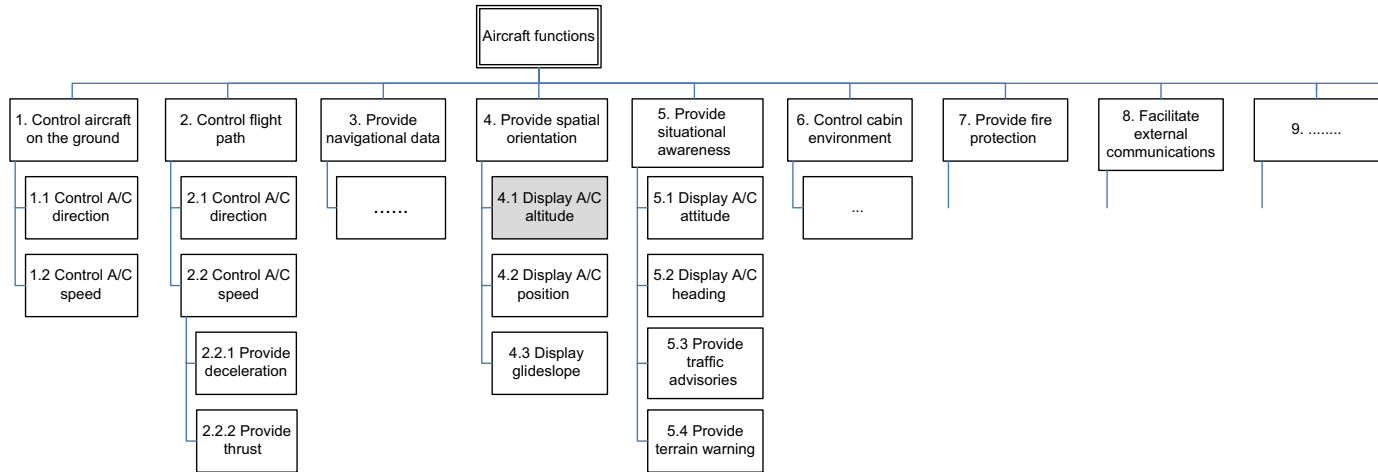


Figure 3.5 Aircraft-level function tree. A starting point for the aircraft-level Functional Tree could be the Air Transport Association's (ATA) chapter. The ATA chapter numbers provide a common referencing standard for all commercial aircraft documentation. This commonality permits greater ease of learning and understanding for pilots and engineers alike. The unique aspect of the chapter numbers is its relevance for all aircraft so, for instance, ATA 26 addresses Fire Protection in any civil transport aircraft type.

For large FHAs, a tabular format¹¹ such as that in [Table 3.6](#) is often most useful, with columns 1–6 being applicable to Steps 1 and 2 of the FHA process.

3.3.3 Aircraft level Step 3: derive safety targets

Apply the worst-case flight phases to the functional failure mode in column 4 of [Table 3.6](#) and allocate qualitative safety objectives ([Table 3.2](#)) in column 9 (i.e. minimum probability of occurrence) to each failure condition based on the worst potential consequence of the failure.

3.3.4 Aircraft level Step 4: prove safety target accomplishment

Use the selected safety criteria to substantiate how the safety targets are going to be shown to be accomplished:

- For Qualitative Assessments the assertions/claims in [Table 3.4](#) (extracted from [Table 3.3](#)) may satisfy the Qualitative Objectives if properly substantiated.¹²
- For Quantitative Assessments the numerical probability targets in [Table 3.5](#) (extracted from [Table 3.3](#)) are commonly accepted as aids to engineering judgement:

3.3.5 System level Step 1: define the aircraft functions

Our starting point for the system-level functional tree is the lowest branch in the aircraft-level tree. In [Fig. 3.6](#) we branch the tree down to a level where a specific system can be identified and assessed for its failure conditions.

3.3.6 System level Step 2: assess the functional failure modes

Conduct the system-level FHA for those systems which are affected by the modification. As we are not changing the Standby Display in this Case Study, this FHA will be limited to the primary displays only.¹³ See columns 1 to 6 in [Table 3.7](#).

3.3.7 System level Step 3: derive safety targets

Use the appropriate safety criteria (e.g. see [Table 3.2](#)) to allocate qualitative safety objectives (i.e. minimum probability of occurrence) to each failure condition based on the worst potential consequence of the failure. See columns 7 to 9 in [Table 3.7](#).

¹¹ Note that some people prefer using Goal Structuring Notation (GSN) for FHA, but the author prefers to reserve GSN for high-level arguments only, and to retain ‘old faithful’ tools such as the FHA as a discreet deliverable set of evidence.

¹² Use as many Fail Safe Principles [Kritzinger (2006) Chapter 7 para 3] as possible to help provide substantiation of qualitative probability declarations.

¹³ Actually, through the pitot-static system the standby display is impacted by the modification. However, this will be accounted for during the probability estimation (see Chapter 4 where we conduct the FTA for this example).

Table 3.4 Qualitative safety objectives

Frequent	Probable	Remote	Extremely remote	Extremely improbable
Conditions anticipated to occur several times	Conditions anticipated to occur one or more times during the entire operational life of each aeroplane	Conditions unlikely to occur to each aeroplane during its entire life but which may occur several times when considering the total operational life of a number of aeroplanes of this type	Conditions unlikely to occur when considering the total operational life of all aeroplanes of the type, but nevertheless has to be considered as being possible	Conditions so unlikely to occur that they are not anticipated to occur during the entire operational life of all aeroplanes of the type ^a

^aExperienced engineering judgement may enable an assessment that such a failure is not foreseeable. The assessment logic and rationale should be readily obvious that a knowledgeable, experienced person would unequivocally conclude that the failure condition simply would not occur. When making such an assessment, all possible and relevant considerations should be taken into account, including all relevant attributes of the design. Extensive service experience alone showing that the failure condition has not yet occurred is not sufficient reason to indicate that a single failure condition cannot exist.

Table 3.5 Quantitative safety objectives

Frequent	Probable	Remote	Extremely remote	Extremely improbable
No requirement	$p \leq 1.0E-3$ per flight hour	$p \leq 1.0E-5$ per flight hour	$p \leq 1.0E-7$ per flight hour	$p \leq 1.0E-9$ per flight hour

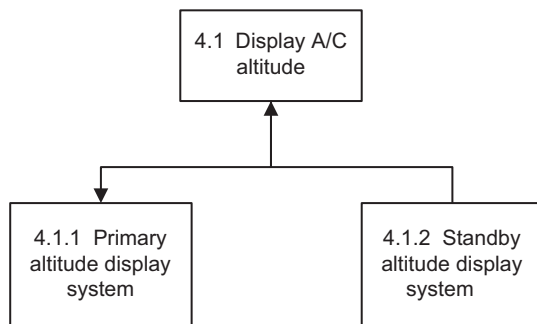


Figure 3.6 System-level functional tree.

Note: The system-level FHA should not go down to LRU level (for this, tools such as the FMEA may be more appropriate). The LRU's form part of the unique system architecture and will be accounted for in the estimation of the system failure probability.

3.3.8 System level Step 4: prove safety target accomplishment

For the purposes of the Preliminary SSA, provide indication of how these safety objectives are going to be met (see last two columns in [Table 3.7](#)).

All safety objectives will need to be shown to be accomplished in the final SSA.

3.4 Discussion

The FHA is undertaken at the beginning of the aircraft/system development life cycle, it is the first step in a safety assessment process that is performed on both new and modified aircraft programs.

As illustrated in [Fig. 3.2](#), before proceeding with a detailed System Safety Assessment (SSA), the FHA is often used to determine the need for and scope of any subsequent analysis. An FHA may contain a high level of detail in some cases (such as for a Flight Guidance and Control System with many functional modes), but many installations may need only a simple review of the system design [AMC25.1309 para 10b(3)]. If further safety analysis is not required, then the FHA could itself be used as a complete safety assessment.

The FHA may be conducted using service experience, engineering and operational judgement, or a top-down deductive qualitative examination of each function performed by the system. When compiling the FHA Report, it may be useful to declare the list of FHA participants and the manner in which they were engaged. It may also be prudent to state that:

- the failure modes and severity allocation in the assessment are subjective (i.e. based on the combined experience of the FHA participants) and
- the actual failure probability statements are based on engineering judgement, back up by appropriate qualitative and/or quantitative assessments.

Table 3.6 Aircraft-level FHA (after Steps 3 and 4)

ID	Function	Failure condition/ mode (hazard description)	Phase	Effect of failure condition ^a	Consequence ^b	Severity	Justification	Qualitative objective	Predicted failure probability	Verification planned/ achieved
4.1.a	Display Aircraft Altitude	Loss of all barometric Altitude Display (annunciated)	IFR conditions	Pilots immediately aware of malfunction (either through failure flag or totally 'off-line') and will need to contact ATC ASAP in order to maintain altitude	Failure conditions which would prevent Continued Safe Flight and Landing	Catastrophic	Further substantiated by AC25-11B Table 4.3 and CS25 [Amm17] AMC Appendices Chapter 3 Table 5	Extremely improbable	TBD	Conduct FTA (#4.1.A) to show that loss of all Altitude displays has $p < 1 \times 10^{-9}$ per flight hour Prove Development Assurance Level A
4.1.b	Display Aircraft Altitude	Loss of all barometric Altitude Display (unannunciated)	IFR conditions	See 4.1A above	Failure conditions which would prevent Continued Safe Flight and Landing	Catastrophic	Further substantiated by AC25-11B Table 4.3 and CS25 [Amm17] AMC Appendices Chapter 3 Table 5	Extremely improbable	TBD	As per 4.1.1.a above, no further verification planned, as this will never be a passive (i.e. unannunciated) failure condition) Prove Development Assurance Level A

4.1.c	Display Aircraft Altitude	Incorrect barometric Altitude Display (annunciated)	IFR conditions	This is essentially the same failure condition as 4.1A above; however, be further aware that the transponder could send same incorrect data to ATC	Failure conditions which would prevent Continued Safe Flight and Landing	Catastrophic	Further substantiated by AC25-11B Table 4.3 and CS25 [Amm17] AMC Appendices Chapter 3 Table 5	Extremely improbable	TBD	Conduct FTA (#4.1.B) to show that loss of all Altitude displays has $p < 1 \times 10^{-9}$ per flight hour Prove Development Assurance Level A
4.1.d	Display Aircraft Altitude	Incorrect barometric Altitude Display (unannunciated)	IFR conditions	Misleading Altitude display would result in spatial disorientation in IFR conditions	Failure conditions which would prevent Continued Safe Flight and Landing	Catastrophic	Further substantiated by AC25-11B Table 4.3 and CS25 [Amm17] AMC Appendices Chapter 3 Table 5	Extremely improbable	TBD	Conduct FTA (#4.1.D) to show that misleading Altitude displays has $p < 1 \times 10^{-9}$ per flight hour Prove Development Assurance Level A

^aDescribe “*Effect of the failure condition*” in own words. AMJ25.1309 para 10c advises that “*In assessing the effects of a failure condition factors which might alleviate or intensify the direct effects of the initial failure condition should be considered, including consequent or related conditions existing within the aeroplane which may affect the ability of the crew to deal with direct effects, such as the presence of smoke, acceleration vectors, interruption of communication, interference with cabin pressurisation, etc.*”

^bDescribe ‘*Consequence*’ using words from the Safety Criteria ([Table 3.2](#)) so that appropriate Severity can be selected in the next column. When assessing the ‘*consequences*’ of a given failure condition, account should be taken of the warnings given, the complexity of the crew action and the relevant crew training. The number of overall failure conditions involving other than instinctive crew actions may influence the flight crew performance that can be expected. Training requirements may need to be specified in some cases.

Table 3.7 System-level FHA

ID	System	Function	Failure condition/ mode (hazard description)	Phase	Effect of failure condition ^a	Consequence ^b	Severity	Justification	Qualitative objective	Actual failure probability	Verification planned/ achieved
4.1.1.a	Primary Barometric Altitude Display System	Display Aircraft Altitude	Loss of Primary Barometric Altitude Display (annunciated)	IFR conditions	Pilots immediately aware of malfunction (either through failure flag or totally 'off-line') and will revert to use of standby display	Large reduction in safety margins or functional capabilities	Hazardous	Further substantiated by AC25-11B Table 4.3, which set the range from 'Remote' to 'Hazardous', and states 'System' architecture and functional integration should be considered in determining the classification within this range. This failure may result in a sufficiently large reduction in safety margins to warrant a hazardous classification.	Extremely remote	TBD	Conduct FTA (#4.1.1.a.1) to show $1 \times 10^{-7} < p < 1 \times 10^{-9}$ per flight hour Prove Development Assurance Level B. Conduct FTA (#4.1.1.a.2) to allocate IDAL and FDAL
4.1.1.b	Primary Barometric Altitude Display System	Display Aircraft Altitude	Incorrect functioning (annunciated)	IFR conditions	Pilots immediately aware of malfunction (either through failure flag or totally 'off-line') and will cross refer to standby display for fault isolation	Large reduction in safety margins or functional capabilities	Hazardous	See 4.1.1.a worse case would be if both primaries disagree with the standby, with the standby being correct	Extremely remote	TBD	Conduct FTA (#4.1.1.b.1) to show $1 \times 10^{-7} < p < 1 \times 10^{-9}$ per flight hour Prove Development Assurance Level B. Conduct FTA (#4.1.1.b.2) to allocate IDAL and FDAL

4.1.1.c	Primary Barometric Altitude Display System	Display Aircraft Altitude	Incorrect functioning (unannunciated)	IFR conditions	Pilots may believe misleading instrument	Failure conditions which would prevent Continued Safe Flight and Landing	Catastrophic	Substantiated by AC25-11B Table 4.3	Extremely improbable	TBD	Conduct FTA (#4.1.1.c.1) to show $1 \times 10^{-7} < p < 1 \times 10^{-9}$ per flight hour Prove Development Assurance Level A. Conduct FTA (#4.1.1.c.2 to allocate IDAL and FDAL
4.1.2a	Standby Barometric Altitude Display System	Display Aircraft Altitude	Loss of Standby Altitude Display (annunciated)	IFR conditions	Pilots immediately aware of malfunction (either through failure flag or totally 'off-line') and will cross refer to primary display for fault isolation	Slight reduction in functional capabilities or safety margins	Minor		Probable		
4.1.2a	Standby Barometric Altitude Display System	Display Aircraft Altitude	Incorrect functioning (annunciated)	IFR conditions	Pilots immediately aware of malfunction (through failure flag or totally obvious failure) and will cross refer to primary display for fault isolation	Slight reduction in functional capabilities or safety margins	Minor		Probable		
4.1.2a	Standby Barometric Altitude Display System	Display Aircraft Altitude	Incorrect functioning (unannunciated)	IFR conditions	Pilots will need to diagnose which barometric altitude display is indeed correct	Significant reduction in safety margins and increase in pilot workload	Major	Substantiated by AC25.11A, which assumes Primary Barometric Altitude is still available	Remote		

^aDescribe 'Effect of the failure condition' in own words. AMJ25.1309 para 8a advises that *'In assessing the effects of a failure condition factors which might alleviate or intensify the direct effects of the initial failure condition should be considered, including consequent or related conditions existing within the aeroplane which may affect the ability of the crew to deal with direct effects, such as the presence of smoke, acceleration vectors, interruption of communication, interference with cabin pressurisation, etc.'*

^bDescribe 'Consequence' using words from the Safety Criteria (Table 3.2) so that appropriate Severity can be selected in the next column. When assessing the consequences of a given failure condition, account should be taken of the warnings given, the complexity of the crew action and the relevant crew training. The number of overall failure conditions involving other than instinctive crew actions may influence the flight crew performance that can be expected. Training requirements may need to be specified in some cases.

3.5 Conclusions

The FHA is a systematic, comprehensive examination of a system's functions to identify potential failure conditions which the system can either cause or contribute to. It should therefore be performed early in the design and updated as required until all functional integrity targets are accomplished.

The functional approach simplifies traceability since a clear reference frame is created:

- It is a top-down approach, extending downwards in progressively expanding stages (like a pyramid), because each function needs to be implemented by the systems contribution to that function.
- These implementations form requirements for lower layer functions. The various levels of required functionality aid us in:
 - allocating responsibility (for the function's implementation) to the various engineers (and to the various subcontractors);
 - flowing the safety objectives (or Development Assurance Levels) down the hierarchy of systems and subsystems to the design authority responsible for its accomplishment.

The FHA technique can be difficult to apply well. It is all too easy for the inexperienced assessor to simply generate reams of meaningless tables. A well-designed FHA will lead to a better understanding of the effect of failures ([Wilkinson and Kelly, 2005](#)); clearly defined safety targets for the design (hence it requires early coordination between the applicant and the certification authority) and auditable evidence of their accomplishment.

3.5.1 Advantages

As a safety assessment technique, the FHA has a number of advantages which include:

- Provides a systematic approach for the identification of critical functional failure conditions.
- Determines the scope and depth of further safety assessments (i.e. assists in bounding the scope of the safety assessment by determining the safety assessment requirements of the system).
- Determines the integrity requirements of the function. Useful as primary mechanism in the identification of safety critical and safety involved failures of a system.
- It is predictive and target setting (i.e. it determines the system's safety objectives without any architectural limitations). It should be used to identify design precautions necessary to ensure independence, to determine the required software level and to avoid common mode and cascade failures (see [Chapters 6–8](#)).
- Highlights functional failures that affect another aircraft system (through interfaces/dependencies/boundaries).
- Improves understanding of how the design relates to safety.
- Provides the FTA top events (see [Chapter 4](#)).

3.5.2 Limitations

The FHA also has its limitations:

- It addresses only functional hazards. The determination of the hazard severity level does not attempt to account for the system failures necessary for its occurrence; it only seeks to determine the appropriate limits for probability of occurrence for a given hazard.
- It may be disproportionately time-consuming. The ‘law of diminished returns’ applies (i.e. refrain from taking the analysis too far by selecting the appropriate system level and assess the worst-case conditions only).

References

- AC25-11B, July 10, 2014. Electronic Flight Displays. U.S. Department, of Transportation, Federal Aviation, Administration.
- CS25, July, 2015. Certification Specifications and Acceptable Means of Compliance for Large Aeroplanes, Amendment 17. European Aviation Safety Agency, Cologne.
- Kritzinger, D., Aircraft System Safety: Civil and Military Aeronautical Applications, Woodhead Publishing, 2006.
- SAE ARP4754A, Guidelines for Development of Civil Aircraft and Systems, SAE Aerospace, <http://www.sae.org>.
- SAE ARP4761 Aerospace Recommended Practise, 1996. Guidelines and Methods for Conducting the Safety Assessment on Civil Airborne Systems and Equipment. The Engineering Society for Advanced Mobility Land Sea Air and Space, Warrendale, USA.
- Wilkinson, P.J., Kelly, T.P., 2005. Functional Hazard Analysis for Highly Integrated Systems, (A Paper on FHA Application on an Engine Controller). University of York.

Further reading

- SAE ARP4761 (Paragraph 3.2 and Appendix A).
- Scharl, A., Stottlar, K., Kady, R. Functional Hazard Analysis (FHA Methodological Tutorial. In: International System Safety Training Symposium (August 4–8, 2014), NSWCCD-MP-14-00380, St. Louis, Missouri, http://issc2014.system-safety.org/83_Functional_Hazard_Analysis_Common%20Process.pdf.

Fault tree analysis

4

The greatest of faults, I should say, is to be conscious of none.

Thomas Carlyle (1795–1881)

4.1 Introduction

A system is a collection of components in a defined architecture with the sole purpose of accomplishing that system's function (refer to Fig. 3.1). The functional failure probability of that function is determined by the integrity of the constituent components as well as the logic of the systems' architecture. The more complex the system, the more there is a need for an in-depth analysis technique to identify all possible combinations of failure that could result in loss of the system's integrity. The Fault Tree Analysis (FTA) is such a technique. A fault tree¹ shows graphically, by means of a specified notation, the logical relationship between a particular system failure and all its contributing causes.

This chapter considers the manner in which an FTA is used to show how an undesirable top-level failure (or event) may occur via the combination(s) of individual contributing failures, events and/or errors. In doing so, this chapter provides a simple process (in Fig. 4.1) on how to approach and manage the FTA process. The reader is encouraged to review the reference material for more specialist details on the intricacies of drawing an actual fault tree.

4.1.1 Background

The FTA is a diagrammatic² analytical technique that is used for Reliability, Maintainability and Safety Analysis. It is a top-down³ (deductive) analysis, proceeding through successively more detailed (i.e. lower) levels of the design until the probability of occurrence of the top event (the feared event) can be predicted in the context of its environment and operation.

¹ The term 'tree' is used because the diagrammatic representation of the analysis has a branching structure which increases in size as various levels of details are considered. In fact the structure is more analogous to the roots of a tree, since the normal convention for constructing a fault tree is to start at the top of the page with the 'consequence' or system failure mode being considered, then represent underneath the causes which could lead to the 'consequence', in increasing details as one progresses down the page.

² It is a graphic 'model' (consisting of gates and events) of the pathways within a system that can lead to a foreseeable event. The causes of the top event are 'connected' through logic gates and modelling of the corresponding system.

³ A fault tree is a cause-and-effect network. It starts by assuming a system failure mode (the top event) and works backwards (i.e. the opposite to FMECA) to identify the possible causes of this.

According to Clemens (2002) and Javadi et al. (2011), the FTA was initially used in 1962 for the US Air Force by Bell Telephone Laboratories on the Minuteman Weapon System⁴ (Eckberg, 1964). Since then, the technique has been adopted and adapted by many companies who are interested in reliability engineering. FTA received extensive coverage at a 1965 System Safety Symposium in Seattle sponsored by Boeing and the University of Washington. Boeing began using FTA for civil aircraft design around 1966 (Hixebbauch, 1968).

Subsequently within the US military, application of FTA for use with fuses was explored by Picatinny Arsenal⁵ in the 1960s and 1970s (Larsen, 1974). In 1976, the US Army Material Command incorporated FTA into the *Engineering Design Handbook, Design for Reliability* (Evans, 1976). The Reliability Analysis Center at Rome Laboratory, and its successor organisations (now the Defense Systems Information Analysis Center), has published documents on FTA and reliability block diagrams since the 1960s [Chapter 6 (FTA) in MIL-HDBK-338B (Electronic Reliability Design Handbook)].

In 1970, the US Federal Aviation Administration (FAA) published a change to 14 CFR25.1309 airworthiness regulations for transport category aircraft in the Federal Register [FR 5665 (1970-04-08)]. This change adopted failure probability criteria for aircraft systems and equipment and led to widespread use of FTA in civil aviation (Haroonbadi and Haghifam, 2009).

In 1998, the FAA published Order 8040.4 establishing risk management policy and hazard analysis in a range of critical activities beyond aircraft certification, including air traffic control and modernisation of the US National Airspace System. This led to the publication of the FAA System Safety Handbook,⁶ which describes the use of FTA in various types of formal hazard analysis.

Today the FTA methodology is widely used in system safety and reliability engineering, and in all major fields of engineering. It is described in several industry and government standards, including NUREG-0492, SAE ARP4761, MIL-HDBK-338B, and IEC 61025.

4.1.2 Aim of the Fault Tree Analysis

Any sufficiently complex system is subject to failure as a result of one or more subsystems or components failing. The aim of the FTA is to use deductive⁷ logic to understand all the underlying causes of a particular failure in a sufficiently complex system so that the likelihood of failure can be reduced through improved system design (i.e. different component selection, more stringent development assurance levels and/or via system architectural improvements).

⁴ Minuteman was a revolutionary concept and an extraordinary technical achievement. Both the missile and basing components incorporated significant advances beyond the relatively slow-reacting, liquid-fuelled, remotely controlled intercontinental ballistic missiles of the previous generation.

⁵ See <http://www.pica.army.mil/Picatinny/>.

⁶ System Safety Handbook. Federal Aviation Administration, 30 December 2000.

⁷ FTA is a top-down, deductive failure analysis in which an undesired state of a system is analysed using Boolean logic to combine a series of lower-level events.

4.1.3 Objectives of the Fault Tree Analysis

An FTA is conducted to satisfy any of the following objectives:

- Improve understanding of system characteristics by diagrammatically representing the system architecture. This then:
 - Assists the safety assessor in identifying the logical combination of events that must first happen for an undesirable outcome to occur.
 - Facilitates the optimising of maintenance effort (as fault diagnostics should benefit from the logic of the FTA).
- Prove the accomplishment of Functional Hazard Analysis (FHA) safety targets:
 - Allows for quantitative evaluation of a probability for the undesirable outcome, so evaluating the ability of a chosen architecture to meet its safety/reliability requirements.
 - Allocate the Development Assurance Level (DAL)⁸ to determine the rigour necessary when demonstrating compliance using [RTCA/DO-178](#), [RTCA/DO-254](#) and [SAE ARP4754A](#).

4.1.4 Scope of the Fault Tree Analysis

The FTA is initiated because of a concerned top-level event (e.g. originating from an FHA) and goes down through a succession of logic gates to basic events (i.e. an event which does not need to be broken down any further).

An FTA can be conducted for both positive and negative events:

- The logic tree segments leading to a Negative Event, such as an accident, defines all of the things that could go wrong to cause the negative event. Logic tree segments for negative events usually use more OR gates than AND gates, except for redundant safeguards.
- The logic tree segment leading to a positive event defines all of the things that must work together for the machine to operate or to complete a successful mission. Logic trees for positive events generally use more AND gates than OR gates, except for redundancy. Maintenance troubleshooting trees are a good example of logic trees for positive events. Inverting the output of a positive event converts it into a negative event.

4.2 Conducting the Fault Tree Analysis

[Fig. 4.1](#) provides a simple illustration of a typical FTA process.

Note that the process is iterative, as it is repeated whenever the system architecture changes and/or when a new contributory cause is identified (e.g. via the Common Cause Analyses of [Chapters 7 and 8](#)).

4.2.1 Step 1: scope the analysis

The first step for a successful FTA is to define the objective of the FTA. The resulting scope of the FTAs will depend on the exact phrasing of the top-level event as well as the scope of the controlling System Safety Assessment (e.g. see [Fig. 2.5](#)). Careful

⁸ See Chapter 9 for more information on the DAL approach.

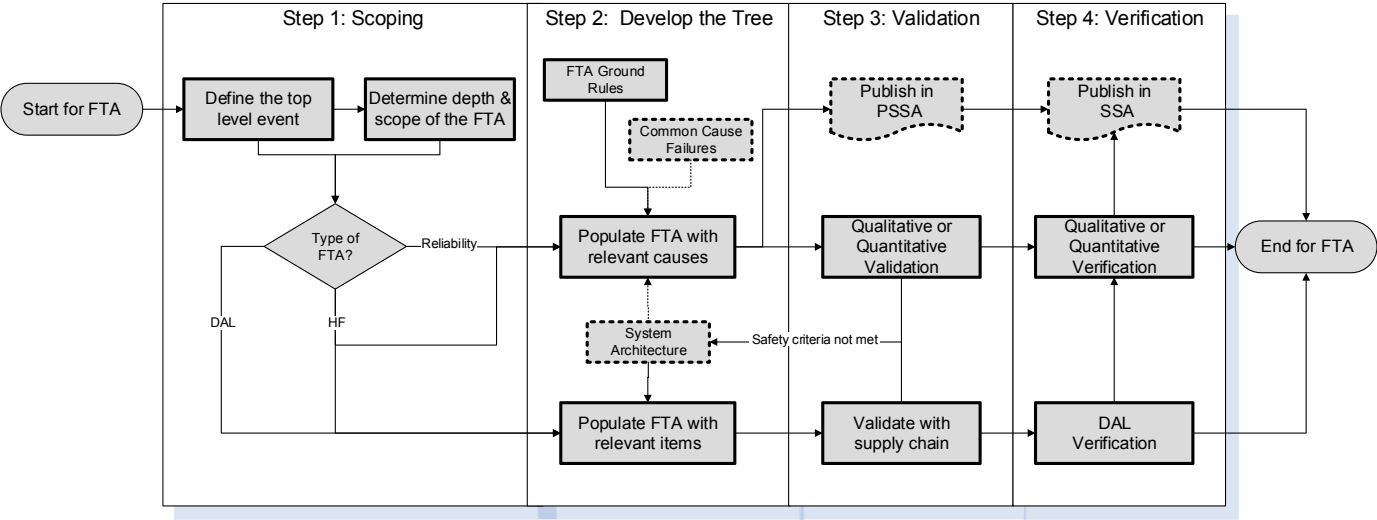


Figure 4.1 An FTA process flow.

definition of the top event is thus necessary: if too general, the analysis becomes unmanageable while, if too specific, the analysis does not provide a sufficiently broad view of the system. If the top event is poorly defined, then the entire assessment will become unfocused. In support of a CS/FAR2x.1309 Safety Assessment, the top event should be a functional failure description (i.e. what occurred or did not occur), not a description of the end result. Typically, the top events are defined/identified via a higher level analysis, such as the FHA, see Step 2 in Fig. 3.2.

The assessor then needs to clarify what the aim of the FTA is, and it could be for either (or all) of the reasons below:

- A ‘Reliability FTA’ is aimed at assessing random failures (see Section 1.3) against which probabilistic safety targets have been set (typically generated for complex systems with failure conditions with a high severity, see Fig. 3.3 for guidance in this regard). With reference to Section 2.3.1, this type of FTA is typically aimed at proving compliance to CS25.1309(b).
- A ‘DAL FTA’ is aimed at assessing systemic failures (see Section 1.3). It is always generated from an FHA so that Functional DALs (FDAL) and Item DALs (IDAL) (see Chapter 9) can be allocated to the engineering development process. When developing the DAL FTA we restrict its content to the boundaries of system architecture and the process which created it [i.e. we do not incorporate operational inputs such as crew or maintenance error or any Particular Risk Analysis (PRA) events]. With reference to Section 2.3.1, this FTA is typically aimed at proving compliance to CS25.1309(a)(1).
- A ‘Human Factors FTA’ is aimed at modelling all potential Human Errors (refer Table 10.1 and Table 6.1) to determine the likely human error contributions to the top event. With reference to Section 2.3.1, this type of FTA is typically aimed at proving compliance to CS25.1309(c), where flight crew warnings are used to mitigate failure conditions. Depending on the top-level failure condition, some of these Human Factors FTAs might need to be incorporated (or transferred) into the Reliability and/or DAL FTAs discussed above. For instance, a Human Factors FTA might identify a condition where a warning system can be used to mitigate a Catastrophic top event. In this case, failure of the warning system must be included within the Reliability FTA for the related failure condition.

The depth (or resolution) of the FTA will depend on the system level under consideration (refer to the example in Fig. 1.1) as well as the boundaries of the system:

- Most system integrators (i.e. System Level 3 and 4 in Fig. 1.1) would not go below black box [i.e. Line Replaceable Unit (LRU)] level, expecting the component design authority (i.e. Level 2 in Fig. 1.1) to provide the relevant failure modes and substantiated probability of each of those failure modes.⁹ The NASA Fault Tree Handbook (paragraph 5.7) advises to model to the highest level for which data exist and for which there are no common hardware interfaces with other contributors.¹⁰ It is, however, important to show the supporting interfaces (such as supply of power or cooling air). These interfaces are what determine whether there are any hardwired or functional dependencies among the components [NASA Fault Tree Handbook, paragraph 4.7].

⁹ It is better to detail the particular failure mode than to generically state failure of a particular unit, as the way the unit fails will likely change the effect on the system. For instance, a power failure would not result in power being applied to a particular signal when not required. However, as MTBF covers the failure of any component within the unit, it can serve as a worst-case figure for probability of failure for any interested failure modes.

¹⁰ Modelling to a lower level will not only be a waste of time but will often provide erroneous probabilities or probabilities with much larger uncertainties. This is an example of the fault tree maxim – ‘too much detail, too much uncertainty’.

- Component designers (i.e. System Level 2 in Fig. 1.1) may be required by the system integrator to develop a piece-part FTA with a top-level event for particular failure modes of a unit. The piece-part FTA would then develop through layers of logic gates until individual component failures (resistors, capacitors, etc.) are identified. This is often supported by a Failure Modes and Effects Analysis (FMEA) from which a Failure Modes and Effects Summary (FMES) (see [Chapter 5](#)) can be generated for the individual next or end effects.
- For a subsystem (i.e. Level 3 in Fig. 1.1) FTA, where the supplier of that subsystem is not the integrator, a decision would also need to be made if the scope includes failures from events outside the system boundary (e.g. the probability of power supply failure to the LRU or the probability of maintenance error).
- When it comes to wiring between components, the NASA Fault Tree Handbook (paragraph 5.7.2) advises not to model wiring faults between components¹¹ unless there are (1) no significantly higher contributors or (2) if the wiring can be impacted by other failures (e.g. a fire) or (3) if the objective¹² includes (e.g. see FAR/CS25.1709) the modelling of wiring faults.

Finally, as with any other modelling technique, the boundaries of the FTA must be defined. Paragraph 3.3 of the NASA Fault Tree Handbook advises that ‘If system failure is analysed as the undesired event, then defining the boundary of the analysis involves defining the boundary of the system that will be analysed. Interfaces to the system such as power sources or water supplies are typically included in an analysis and are therefore within the analysis boundary. If they are excluded from the analysis, then their states need to be defined to define the inputs to the components that are analysed.’

4.2.2 Step 2: develop the fault tree

4.2.2.1 Ground rules

Before the FTA is started, it is important to define the FTA ground rules (refer to Fig. 3-1 in the NASA Fault Tree Handbook). These ground rules¹³ include:

- The procedure and nomenclature¹⁴ by which events and gates are named in the FT, as this (1) is very important in creating an understandable FTA, (2) ensures that correct cut sets¹⁵ and probabilities are calculated if gate or basic events occur more than once in

¹¹ Generally, wiring faults, such as shorts to ground and shorts to power, have very low probabilities compared to probabilities of major components failing.

¹² An example of such an objective is in CS25.1309(d), which states that ‘*Electrical wiring interconnection systems must be assessed in accordance with the requirements of CS 25.1709*’. CS25.1709 states that ‘*EWIS must be designed and installed so that:*

- (a) *Each catastrophic failure condition*
 - (1) *is extremely improbable; and*
 - (2) *does not result from a single failure; and*
- (b) *Each hazardous failure condition is extremely remote*’.

¹³ The construction of fault trees is a process that has evolved gradually over a period of about 50 years. In the beginning it was thought of as an art, but it was soon realised that successful trees were best drawn in accordance with a set of basic rules. Observance of these rules helps to ensure successful fault trees so that the process is now less of an art and more of a science. For a lot more details on suggested ground rules, see paragraph 4.5 in the NASA Fault Tree Handbook.

¹⁴ Establish a naming (or labelling) convention for the FTA and stick to it. Avoid using words such as ‘fail’ as it may not be descriptive enough (e.g. ‘*power supply fails*’ versus ‘*power supply does not provide +5VDC*’).

¹⁵ The cut sets are the combination of failure events that can cause the top event to occur. They reveal the critical and weak links in a system design.

the model, and (3) provides consistency among different FTs especially when different individuals are developing them. For more information on good nomenclature practice, see the NASA Fault Tree Handbook paragraph 5.6.

- The manner in which to model repeated events and Common Cause Factors (CCF). See the Annex to this chapter for more detail.
- The manner in which to model human errors. For instance, the NASA Fault Tree Handbook (paragraph 5.7) advises not to model human errors of commission.¹⁶
- The discipline to not including ‘success states’: As the failure probability on a ‘no-failure’ of an item should normally approximate 1, the exclusion of success states will not affect quantification of the tree but will simplify its construction.

For organisations operating under a Design Organisation Approval (DOA) scheme (such as EASA Part 21 Subpart J), it is anticipated that these ground rules would be defined in company processes.

4.2.2.2 *Populate the Fault Tree Analysis*

Using these ground rules, each fault tree should be developed from the top-level event down to its basic events through a successive number of logic gates.

Start by putting the event under consideration¹⁷ at the top of the page. Appropriately, this event is referred to as the ‘*top event*’, and it is the objective of this particular tree. All immediate possible causes of the top event should be identified and placed below it on the tree.

Work your way down¹⁸ by examining the system schematics (e.g. see Fig. 1.8) and considering all credible component failures (and combinations of faults or failures¹⁹) that could lead to the top-level event. These contributory failure conditions are called Intermediate Events. There are a number of structured techniques that can be used to help with the iterative process of defining all intermediate and contributing events to the undesired top-level event [Ericson, paragraph 11.5.3]:

- The ‘*Immediate-Necessary-Sufficient*’ (I-N-S) concept is a question the analyst should ask themselves when defining inputs to any particular gate. Have most immediate causes been identified, have all necessary causes been identified, but only those necessary and sufficient to lead to the event.

¹⁶ The NASA Fault Tree Handbook (paragraph 5.7) defines human errors of commission as ‘*those involving the human committing an unforeseen action. The reason human errors of commission are not modelled is that current modelling approaches would require a consideration of an almost unlimited scope of actions*’.

¹⁷ In most civil aviation System Safety Assessments, this event originates from a Function Hazard Analysis (FHA, see Chapter 3), but it can also come from any other hazard identification technique (e.g. ZSA or PRA).

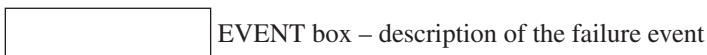
¹⁸ An FTA is a deductive approach (i.e. top down) that determines how a given state (i.e. the undesired event) can occur. It does not identify all failures in a system in a way that inductive approaches (such as an FMEA) would.

¹⁹ A distinction is made here between the rather specific word ‘*failure*’ and the more general word ‘*fault*’. The NASA Fault Tree Handbook (paragraph 3.5) provides an example of the distinction: ‘*If a relay closes properly when a voltage is applied across its terminals, this is a relay “success.” If, however, the relay fails to close under these circumstances, this is a relay “failure.” Another possibility is that the relay closes at the wrong time due to the improper functioning of some upstream component. This is clearly not a relay failure; however, untimely relay operation may well cause the entire circuit to enter into an unsatisfactory state. An occurrence like this is referred to here as a “fault” so that, generally speaking, all failures are faults but not all faults are failures. Failures are basic abnormal occurrences, whereas faults are “higher order” or more general events*’.

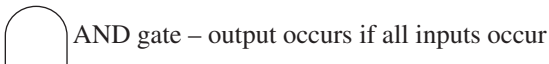
- The ‘*Primary-Secondary-Command*’ (P-S-C) concept is to concentrate the analyst on specific causal factors. This concept is based on components having three ways of failing, the primary failure mode (inherent failure), secondary failure mode (external influence) or a command path fault (function provided when not required). All of these failure modes should be considered to ensure nothing is overlooked.
- The ‘*State-of-the-System*’ and ‘*State-of-the-Component*’ (SS-SC) concept is used to identify whether the I-N-S or P-S-C concept should be used. If the indicated fault is a system failure (SS), then the I-N-S concept is best used. If the indicated fault is a component failure (SC), then the event will have an OR gate with P-S-C inputs.

These concepts, which are described in detail by Ericson in ‘*Hazard Analysis Techniques for System Safety*’, help prevent the analyst from jumping ahead and missing the required detail to methodically develop the tree.

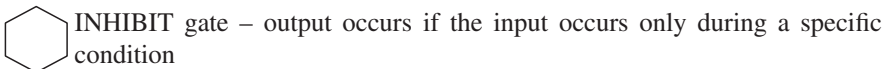
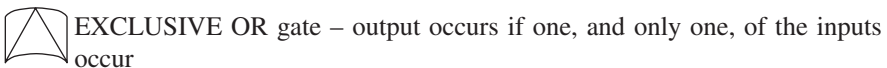
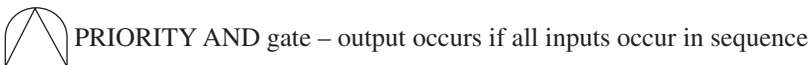
An event description box is used to describe a particular state of the system at any particular level of the FT. Typically, an event description is provided for each gate, starting with the top-level event.



Each intermediate event is linked by Boolean operators²⁰ (or ‘gates’) connecting them together. Any Boolean operator can be used in a FT; however, the vast majority of gates²¹ used will be either an AND gate²² or an OR gate²³:



The following gates are all special derivatives of the above and are called Conditioning Events²⁴:



²⁰ The majority of fault tree computer programs support a number of other Boolean operators, such as Exclusive OR (XOR), Voting and Priority AND gates; however, this is outside of the scope of this chapter (for more information, see the references in paragraph 6).

²¹ The five gates discussed in Section 4.2.2.2 represent only the most commonly used types. Further detail and examples of the less used gates can be found in NUREG-0492.

²² An AND gate allows progression only if ALL the contributing events occur simultaneously.

²³ An OR gate allows progression whenever ANY one or ANY combination (i.e. at least one) of the contributing events occur.

²⁴ Many of these special derivatives can be replaced by a construction of OR and AND gates that represent the same logic (for example, see Bossche, page 21).

Note: If a combination of gates appears to be required at any point, then you may be progressing too quickly and a suitable intermediate stage may be required. The NASA Fault Tree Handbook (paragraph 4.5) advises that *‘The “gate-to-gate” connection is indicative of sloppy analysis. When a FT is being constructed, the gate-to-gate shortcuts may lead to confusion and may demonstrate that the analyst has an incomplete understanding of the system. A FT can be successful only if the analyst has a clear and complete understanding of the system to be modelled’*.

To ensure the tree is complete, develop the FT in small steps without jumping to basic events too early. At each level of the FT, consider all possible influences and create gates for them all before developing one in too much depth (i.e. concentrating on *‘breadth before depth’* will help develop a more balanced FT that represents all contributions rather than only those for which basic events are obvious).

Remember to consider Common Cause Factors (CCF). The NASA Fault Tree Handbook (paragraph 5.2) advises *‘Neglecting these CCF contributions can result in a significant underestimate of the probability of the top event. The key in successfully including CCFs in a fault tree is to identify the components that are susceptible to CCFs and then properly model them in the fault tree’*. CCF can be included in the FTA in two ways:

- Explicitly, by adding as a specific failure mode (e.g. maintenance error) into the FT. The CCF typically originate from the Common Mode Analysis (CMA) (see [Chapter 6](#)), the Particular Risk Analysis (PRA) (see [Chapter 7](#)) or the Zonal Safety Analysis (ZSA) (see [Chapter 8](#)).
- For redundant components, CCF as modelled as a separate contributor via an OR gate, where the one branch would consider the independent failure probability and the other considers the CCF failure probability. The NASA Fault Tree Handbook (paragraph 5.2) provides a good rule of thumb: *‘Include CCF contributions for any redundancy of identical, active components’*. See the Annex to this chapter for more detail.

Continue down the FT until you reach a fundamental triggering event that (a) cannot be usefully deconstructed any further, (b) at which the failure probability can be justified or (c) where you have reached the scope (see [Section 4.2.1](#)) of the FTA. These events are at the bottom of the FT and are referred to as either Basic Events or Undeveloped Events:

- Basic events represent a fundamental triggering event that cannot be usefully deconstructed any further. The level is largely dictated by the system level of the analysis. For a system integrator or system designer, this will typically be a failure mode of an LRU.
- An undeveloped event represents an event where further development is either not possible or would serve no benefit. This event type is mostly used when the probability is already orders of magnitude smaller than other contributors to a gate, such that the gate probability is dominated by other inputs. It is therefore of no benefit to expend further effort in developing this part of the FT.



BASIC event – an initiating event

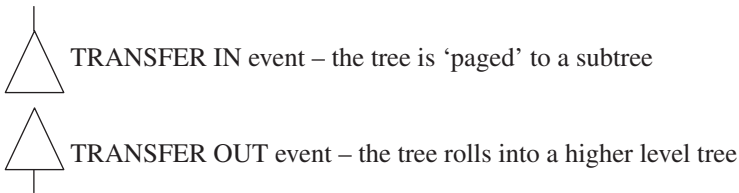


UNDEVELOPED event – an event that has not been developed either because it is developed in a separate assessment, the contribution is negligible or there is insufficient information to develop further

Within the scope of the analysis, extend the contributing causes identification process by evaluating other failure source data such as:

- Single point failures from the FMEA/FMECA (see [Chapter 5](#)).
- System history/experience (e.g. accident/incident reports and significant regulatory changes such as SFAR88 which altered fuel tank ignition prevention guidance).
- Operating procedures of the system, both during normal and any abnormal or emergency operation.
- Any Common Cause Factors, which can be sourced from:
 - The ZSA will highlight any influences from surrounding systems (see [Chapter 8](#)).
 - The PRA will highlight any external influences (see [Chapter 7](#)).
 - The CMA will highlight common cause factors which could negate any redundancy (i.e. the AND-gates) claims (see [Chapter 6](#)).

Transfer symbols are used for convenience, e.g. to avoid a large tree on a single page or to include a common set of failures in numerous places in the same tree.



At Step 2, it is worth remembering to leave room for future tree expansion (i.e. allow for possible future changes in the tree) as the design matures and more contributory failure modes become known.

4.2.3 Step 3: validation²⁵

With reference to Fig. 1.3, Step 3 is all about validating that the proposed system architecture will indeed meet the safety requirements and to consider which actions are needed to define more robust and/or fail safe system architecture.

4.2.3.1 Validating the Reliability Fault Tree Analysis

The emphasis here is to ensure that the FHA safety targets can be accomplished from two perspectives:

- Is the proposed architecture suitable? We need to check if it is correct (i.e. does it reflect the actual design), appropriate (i.e. challenge all AND gates to ensure independence, see Step 1a in [Chapter 6](#)) and optimised (i.e. by exploring the weakest links via cut sets²⁶).

²⁵ Validation is the determination that the requirements for a product are correct and complete. It can be summarised in the question: ‘Are we building the right thing?’

²⁶ A cut set is a set of basic events that causes the top event to occur. Cut sets are referred to as ‘first order’, ‘second order’, etc. First-order cut sets are items that cause the top event directly (It may be a design target to have no first-order cut sets). Second-order cut sets require two states to exist concurrently or states to exist when an event occurs. Higher order cut sets follow the same pattern. The FTA provides a technique for the verification of such requirements. A minimal cut set is the cut set with the minimum number of events that can still cause the top event. A Critical Path is the highest probability cut set which drives the top event probability. Cut sets may often be determined by inspection of the fault tree. However, more formal and sophisticated procedures are usually necessary as the tree increases in size and complexity.

- Have we selected the appropriate subsystems and components? These often form the Basic Events in the FTA, and any shortfalls here would need to be compensated for in the system architecture.

This is often achieved using a checklist, the system schematic and a thorough examination of the logic of the FTA to ensure at each level that all permutations have been included.

In support of the SSA, validation is accomplished by populating the FTA with failure probability data (see [Annex A1](#)) to determine if the FHA safety targets (see [Chapter 3](#)) will be accomplished. However, as a fault tree represents events of only two states (either TRUE or FALSE), the rules of Boolean algebra²⁷ must first be applied (see NASA Fault Tree Handbook paragraph 6.1 and its Appendix A). These rules are complex, but in the case of FTs, there are three important considerations as shown in [Table 4.1](#).

Table 4.1 Laws of Boolean algebra

Law name	Examples	What this means
Associative law	$A \times (B \times C) = (A \times B) \times C$ $A + (B + C) = (A + B) + C$	Normal algebra can be used to expand terms, and order is not important
Distributive law	$A \times (B + C) = A \times B + A \times C$	
Commutative law	$A \times B = B \times A$ $A + B = B + A$	
Idempotent law	$A \times A = A$ $A + A = A$	An event combined (through either AND or OR) with itself is simply the event itself
Absorption law	$A \times (A + B) = A$ $A + (A \times B) = A$	If an event is combined with a combination of itself and another independent event, it is irrelevant whether the other event occurs since the first event would already result in the next higher event

Failing to apply the idempotent and absorption law can result in a lower probability estimate than is true (see example in [Annex A2](#)).

Once the rules of Boolean algebra have been applied to the FTA, the failure probabilities can be entered into the basic events. The top event can then be calculated by successively determining the probability of each gate until the probability of the top event is known. Providing all basic events are small (in the order of 1E-3 or lower),

²⁷ Note: The FTA is primarily a graphical method using logic gates and fault events to model the cause-effect relationship in causing an undesired event. This graphical method can be translated into a mathematical model to compute failure probabilities and system importance measures [Ericson, 2005, Chapter 11]. This quantitative approach provides more useful results, but requires more time (e.g. gathering of component failure rate data) and experienced personnel.

the probability can be determined simply by multiplying inputs into an AND gate and summing inputs into an OR gate.

Probability targets are normally expressed as a failure rate as opposed to an absolute probability of failure. For instance, CS25.1309 expresses probability targets based on the severity of the failure condition in '*Average Probability per Flight Hour*'. This is easy to accomplish when all of the basic events are also expressed as a failure rate; however, confusion can arise when some of the basic events are affected by time at risk²⁸ or are dormant events²⁹ that may lay in a failed state for many flights and only result in an issue when another failure or a specific circumstance occurs.

In actuality, this can be handled relatively easily by calculating the FT on a per flight basis and then converting the probability of the top-level event back to a failure rate per flight hour. This can then be compared to the '*Average Probability Per Flight Hour*' of the original probability target.

Each basic event should be converted into a per flight probability (P), using the following [Kritzinger, 2006, Chapter 10]):

- Failure rate (basic event), $P = pT$, where T = average flight time
- Time at risk, $P = pt$, where t = time at risk
- Dormant failure, $P = p(\tau/2)$, where τ = check interval

The probability *per flight* can then be calculated for the top-level event. To convert back to a failure rate, this number can simply be divided through by the average flight time to get back to an average probability per flight hour.

See Section 4.3.3.1 of this chapter (as well as paragraph 5.8 in the NASA Fault Tree Handbook) for more information on how to do this validation.

4.2.3.2 Validating the Development Assurance Level Fault Tree Analysis

The DAL FTA allocates specific safety design requirements (i.e. FDALs, see Section 9.2.1.2) onto subsystems and items which now needs to be validated with the supply chain.

Each FHA failure condition is assigned an FDAL based on the FHA severity (see Table 3.3), after which the DAL FTA is then used to allocate IDAL (see Table 4.2) to

²⁸ Time at risk is applied when the failure can only lead to the next higher event if it occurs over a specific portion of the flight cycle. This would only be applicable if the failure is not dormant and is immediately known, and if the failure is known, an alternative system can be used.

²⁹ Dormant failures relate to nonactive failures, where the failure is not detectable. In this case, the probability for a flight cycle is the probability per flight hour multiplied by the check interval rather than the average flight time. The accepted equation actually uses half of the check interval, on the basis that the furthest you can be from knowing the state of the item is either half the check interval from the last checked state, or half the check interval from the future checked state.

individual equipment, software and Airborne Electronic Hardware which make up that function (with due consideration of any common mode³⁰ failures).

Table 4.2 summarises the guidance provided in SAE ARP4754A (paragraph 5 and Appendix C)³¹:

Table 4.2 **FDAL^a and IDAL^b assignment**

If combination of 2 or more development errors between 2 or more independently developed systems ^c	
FHA failure condition	DAL assignment
Catastrophic	Any one system/function/item at Level A or two at Level B All others at level commensurate with most severe individual effect but no lower than Level C Level A for process to ensure that the 2 (or more) functions are indeed independent
Hazardous	Either one system/function/item at Level B or two at Level C All others at level commensurate with most severe individual effect but no lower than Level D Level B for process to ensure that the 2 (or more) functions are indeed independent
Major	Any one system/function/item at Level C or two at Level D All others at level commensurate with most severe individual effect Level C for process to ensure that the 2 (or more) functions are indeed independent
Minor	Any one system/function/item at Level D All others at level commensurate with most severe individual effect
No safety effect	Level E

^aDuring the Functional Development Phase, requirements for functions are developed and allocated to each level in the system hierarchy (down to items). These requirements need to be validated at each level, and the rigour is defined via the FDAL approach in ARP4754A Appendix A. FDAL validates the functional requirements.

^bDuring the Item Development Phase, the rigour of the development process is validated via the IDAL approach in RTCA/DO-178C for Software and RTCA/DO-254 for Complex Electronic Hardware. IDAL validates the specific item's development processes.

^cIf a function (or item) contains functions that individually have different allocated DALs, then one of the following approaches may be taken. The entire item may be assured at the highest DAL. If partitioning between subcomponents cannot be demonstrated, the subcomponents should be viewed as a single component when assigning DALs (that is, all components are assigned the DAL associated with the most severe failure condition to which the item can contribute).The individual functions may be assured separately at their respective DALs, if and only if their function, interfaces and shared resources can be protected from adverse effects of functions of lower Development Assurance levels (i.e. using some form of partition). Only partitioned components can be assigned individual DALs by the system safety assessment process.

³⁰ Common modes are common characteristics or potential failures (random or systemic) that effect multiple items which should be independent but are not. See Chapter 6 for a common mode checklist. When looking for common modes in architecture, it is useful to adopt the philosophy of 'guilty until proven innocent' – i.e. the common mode exists unless it can be shown not to [Spitzer, Ferrel, *Digital Avionics Handbook*, third ed., ISBN:9781439868980].

³¹ For Part 25 aircraft, use SAE ARP4754A (paragraph 5) for guidance on the DAL allocation. However, use AC23.1309-1E for Part 23 aircraft.

The emphasis here is to ensure that:

- Those who are to develop (or supply) the items required by the proposed architecture are informed of the FDAL and IDAL V&V (see Fig. 1.3) obligations and are committed (contractually if needed) to achieving them.
- The proposed functional architecture has no common mode vulnerabilities. As can be seen in Table 3.1, independence attributes are key to protect against common mode errors. There are two factors to consider here:
 - Functional Independence, which minimises the likelihood of common requirements error (including their interpretation) [SAE ARP4754A, paragraph 5.2.3.2.1.1]. Functional Independence is substantiated when the common source of error between multiples requirement sets have been minimised at the appropriate FDAL at all levels of requirements decomposition.
 - Item Development Independence, which minimised the likelihood of common mode errors in the development process. [SAE ARP4754A, paragraph 5.2.3.2.1.2]. Examples include S/W or H/W design error (e.g. errors in requirements, architecture, etc.) and S/W or H/W development error (e.g. development process, configuration management, etc.). Example ways to achieve Item Development Independence include using different technologies, operating systems, S/W languages, teams, processes, etc. Item Development Independence is substantiated when the common source of error between multiples items have been minimised at the appropriate IDAL (at all levels of requirements decomposition).

4.2.3.3 Validating the Human Hazard Fault Tree Analysis

The emphasis here is to ensure/validate that the crew can interact with the equipment during normal operation and during responses to failures. We thus need to validate that the crew can mitigate failures (through recovery and control actions) and understand how the system responds if the crew cause an initiating event.³² Examples of human errors that can be modelled include [NASA Fault Tree Handbook, paragraph 5.3]:

- Test- and maintenance-related errors
- Errors causing initiating events
- Procedural errors during an incident or accident
- Errors leading to inappropriate actions
- Detection and Recovery errors

The use of probability numbers for human error within FTA is discussed in more detail in [Annex A4](#) as well as [Chapter 10](#) (see Table 10.1).

The validation results are then published in the PSSA (refer Step 3 in Fig. 1.2) and may need to be repeated as it is an iterative process.

Note: Typically, FTs will be produced to support key milestones in the design process (such as PDR or CDR with a validated FTA), but also as evidence to support flight

³² NASA FTA Handbook (paragraph 5.7) advises not to model human errors of commission: ‘Human errors of commission are those involving the human committing an unforeseen action. The reason human errors of commission are not modelled is that current modelling approaches would require a consideration of an almost unlimited scope of actions’.

trials and finally certification via a verified FTA. Accordingly, the fault tree report must clearly identify what build standard it represents. As the design matures, these changes must be incorporated and the fault trees regenerated. This should be completed in tandem with updates to the FMECAs (see [Chapter 5](#)), from which the fault trees will draw heavily on for the basic events.

4.2.4 Step 4: verification³³

Verification of the FTA involves ensuring that the model reflects the final design solution's configuration and behaviours. Verification methods [SAE ARP4754A, paragraph 5.5.5] applicable to the FTA include:

- Inspection and Review of drawings and built artefacts;
- Analysis of the FTA versus the configuration baseline it is reflecting;
- Test or Demonstration of simulated failure modes, etc.

Once fully verified the up-to-date and fully populated fault trees should then be written up either as an annex of the System Safety Assessment or as a separate report. The annex or report should include an index of the failure conditions considered, including the probability targets which were set. For each fault tree, include the following:

- Describe the fault tree, including a detailed description of what the top-level event is interpreted to mean, and any scope considerations.
- Detail any assumptions or caveats relied upon during the construction of the fault tree, such as interface elements or system behaviour that has been assumed.
- Include a table of basic events and the probabilities assigned, with references to where they have been obtained from.
- Include the fault tree diagrams, split over multiple pages using transfer gates if needed so that it is readable in its printed form.
- Finally, add cut set diagrams, particular first order (where a single basic event chases up to the top-level event) as these represent single point failures in the system.

Finally, an annex of the SSA (or a separate FTA report) should include a summary of the fault tree results, and if they are compliant to the numerical targets and, for top events with a catastrophic severity, whether any single point failures have been identified. Additionally, the report should detail any maintenance actions that are necessary to alleviate the effects of dormant failures.

4.3 The Case Study

The objective of the FTA is to prove the functional integrity of a system. In the following sections we will apply the theory of [Section 4.2](#) to the case study from [Section 1.4](#) to demonstrate what a typical FTA would look like in support of the functional integrity requirements required in the FHA (see [Table 3.7](#)).

³³ Verification is the evaluation of an implementation of requirements to determine that they have been met. It can be summarised in the question: 'Did we build the thing right?'

This case study will consider only the functional requirements for providing barometric altitude readings in the upgraded system. Barometric (or pressure) altitude is used to determine Flight Level and is based on a standard air-pressure datum. As detailed in [Chapter 1](#), altitude is displayed on the pilot and co-pilot Primary Flight Display (PFD) and on the standby flight instrument. We will consider here only those requirements for display of primary barometric altitude display on each PFD.

4.3.1 Step 1: scope the analyses

For this case study, we will concentrate on the following two system-level functional failures taken from Table 3.7 (chosen because these failure conditions are influenced by pitot-static effects and software causes):

Table 4.3 FHA targets for FTA

ID	System	Function	Failure condition/ mode	Severity	Qualitative objective	Verification planned
4.1.1a	Primary Barometric Altitude Display System	Display Aircraft Altitude	Loss of Primary Barometric Altitude Display (annunciated)	Hazardous	Extremely Remote	Conduct FTA (#4.1.1.a.1) to show $1 \times 10^{-7} < p < 1 \times 10^{-9}$ per flight hour Conduct FTA (#4.1.1.a.2) to allocate IDAL and FDAL
4.1.1.c	Primary Barometric Altitude Display System	Display Aircraft Altitude	Incorrect functioning (unannunciated)	Catastrophic	Extremely Improbable	Conduct FTA (#4.1.1.c.1) to show $p < 1 \times 10^{-9}$ per flight hour Conduct FTA (#4.1.1.c.2) to allocate IDAL and FDAL

Now that we have the objectives and purpose of these FTAs, we need to determine:

- The depth (or resolution) of the FTA: To demonstrate the concepts of this chapter, the FTAs drawn will go no further than the black box (i.e. LRU) level. For the items incorporated as part of the modification, basic events will be claimed at the top-level failure modes of each LRU (such as a PFD unit or an Air Data Unit). This level would typically be

selected by a system integrator, where equipment failure modes are defined by equipment suppliers in an FMES.

- The boundaries of the FTA: To demonstrate the concepts of this chapter, the FTAs drawn will not explore the interfaces to the original aircraft systems further than top-level failure modes of the inputs. Take power, for instance, where basic events will be defined for loss of power from the specific aircraft power bus where the power is sought (such as loss of Main DC). These data are typically available from aircraft FMEA or can also be obtained from FRACAS data for in-service aircraft with a statistically significant amount of in-service reliability data.

An understanding of the system architecture is needed to develop each FT, and for this case study the following assumptions³⁴ have been used to generate the trees in Step 2:

- As can be seen in Fig. 1.8, each PFD can be supplied with altitude data from either Display Concentrator Unit (DCU), and in this case study it is assumed to be controlled purely by the Reversion switch (NORM, All on 1, or All on 2). It is assumed that the PFD displays the source of the data (i.e. DCU1 or DCU2) to allow the flight crew to verify operation of the Reversion switch.
- While Section 1.4.3.1 states that if a PFD fails the flight crew can display altitude on the equivalent Navigation Display (ND), this has not been taken into consideration to simplify the resulting trees for this case study.
- Each DCU is continuously supplied with altitude data from both Air Data Computers (ADCs) and will compare the readings to alert the flight crew if there is a loss of one data source or a large discrepancy between readings.
- In this case study, the ADC determines altitude by reading static air pressure from its own dedicated port and does not use any other air data for correction or comparison purposes (therefore the pitot air pressure and total air temperature have no bearing on barometric altitude reading).
- The barometric setting provided by the flight crew on the Display Control Panel is not relevant for pressure altitude, and it is therefore not considered further in this case study. Other elements of the upgraded system, such as the Standby Flight Instrument and Inertial Reference System (IRS), are also not considered in this case study.
- It is assumed in this case study that the integrity of the data bus connections (ARINC 429) and of the Electrical Wiring Interconnection System (EWIS) looms are separately analysed and shown appropriate for the purposes used (such as through a combination of reliability and development assurance). As detailed previously, power sources are not developed further than the bus providing the power; the power sources for each unit are Essential DC (PFD1, ADC1), Main DC (PFD2, ADC2), Essential AC (DCU1) and Main AC 1 (DCU2).

Now that we have the purpose and the scope of the FTAs and we can continue to Step 2.

4.3.2 Step 2: develop the fault tree (i.e. the fault tree logic)

For the purposes of this case study, we will be developing four FTAs in support of Table 4.3, although they are only based on two functional failures. The objective here

³⁴ These assumptions have been defined for the purposes of this case study and have been developed either to 'fill in' missing information from Chapter 1 or to remove complexity that was deemed not pertinent to explaining the use of fault trees. In reality, the analyst should use hard facts to create the fault tree rather than assumptions. If assumptions are used during early development of fault trees, these must either be replaced or validated (such as with the customer) as the design matures.

is to demonstrate/explore the difference approach taken between Reliability FTA versus a DAL FTA.

4.3.2.1 *Reliability Fault Tree Analysis for loss of primary barometric altitude display (annunciated)*

To have an annunciated (known) loss of barometric altitude from the primary displays, at least one of the following events must have occurred to the architecture of Fig. 1.8:

- Event 1: As both PFD display barometric altitude, and altitude from either DCU can be displayed on each PFD, a simultaneous loss of both PFD must occur.
- Event 2: As either DCU can feed data to each PFD, a simultaneous loss of both DCU outputs must occur. Note, the Reversion switch is needed to switch DCU inputs to the PFD. This has not been reflected in the FT below as it would add significant complication to an otherwise manually calculable tree. For completeness, and assuming the normal position of the Reversion switch is NORM, the actual events would be both DCU simultaneously fail (as is currently shown), or PFD1 fails thereby losing DCU1 display and DCU2 fails thereby losing DCU2 data on PFD2 and the Reversion switch fails preventing PFD2 displaying DCU1 data, or the reverse (PFD2, DCU1 and Reversion switch fail simultaneously). It is safe to omit this complexity as, even if assuming a relatively poor reliability Reversion switch, the impact on the calculated probability would be minimal. However, this complexity should be included when analysing real systems as it may have implications on development assurance and future changes to the system might otherwise impact the functionality. Additionally, fully representing all functionality provides useful information to help understand system operation and for fault finding.
- Event 3: As each DCU receives air data from both ADC, a simultaneous loss of both ADC outputs must occur. In this case study, ADC1 determines barometric pressure data from the port static port, while ADC2 uses the starboard static port (see Fig. 1.7 in [Chapter 1](#)). Problems with these inputs have not been considered to result in an annunciated loss of altitude display, as it is more likely that misleading data would be fed to the PFDs (unless data are out of bounds resulting in an error being raised). Hence, these inputs are considered only in the FTA for #4.1.1.c.1.

Each of these events represents a separate OR branch for loss of primary barometric altitude display. The FTA in [Fig. 4.2](#) is proposed³⁵ to represent the way these three events contribute to the top-level functional failure condition. Within the scope (see Step 1) of this assessment, each event was developed further by considering the LRUs involved and their data and power inputs, where applicable.

[Table 4.4](#) provides a list of the basic events for FTA #4.1.1.a.1, with their failure probability and reference to the source of this data. This will be used during validation of the FT to determine the probability of the top event.

While the basic events have been developed only as far as major aircraft systems in this case study, it is important to ensure that the supporting evidence for the basic events include all aspects that must be covered under the scope of the change (i.e.

³⁵ Proposed, as subject to validation (step 3) and verification (step4). Note the action required in [Table 6.2](#).

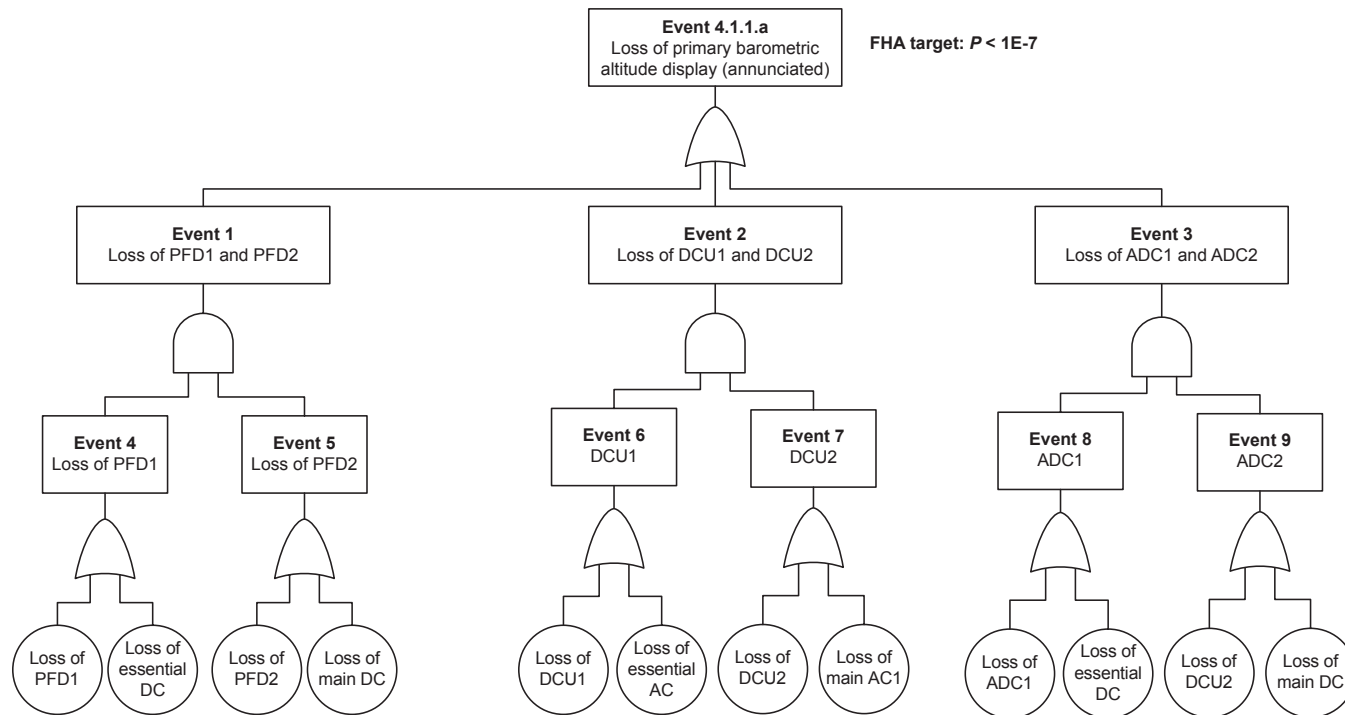


Figure 4.2 Reliability FTA #4.1.1.a.1.

Table 4.4 Basic events for FTA #4.1.1.a.1

ID	Description	Failure probability	Example source/ comments
PFD1	Loss of PFD1	2E-4/flight hour	Supplier FMES (refer #...) for PFD
EssDC	Loss of Essential DC	1E-7/flight hour	Aircraft FRACAS (refer #...)
PFD2	Loss of PFD2	2E-4/flight hour	Supplier FMES (refer #...) for PFD
MainDC	Loss of Main DC	1E-6/flight hour	Aircraft FRACAS (refer #...)
DCU1	Loss of DCU1	5E-5/flight hour	Supplier DDP (refer #...) for DCU, where MTBF = 20,000 h
DCU2	Loss of DCU 2	5E-5/flight hour	Supplier DDP (refer #...) for DCU, where MTBF = 20,000 h
ADC1	Loss of ADC 1	1E-4/flight hour	Supplier FMES (refer #...) for ADC
ADC2	Loss of ADC 2	1E-4/flight hour	Supplier FMES (refer #...) for ADC
EssAC	Loss of Essential AC	1E-7/flight hour	Aircraft FRACAS (refer #...)
MainAC1	Loss of Main AC 1	1E-4/flight hour	Aircraft FRACAS (refer #...)

update Step 1). For instance, loss of the power systems will involve a large number of subcomponents from the existing aircraft systems, including engine generators, transformer rectifier units, junctions boxes, and just as importantly the wiring looms (EWIS) that connect them all together. These would all need to be reflected in the source data (i.e. aircraft FRACAS) for the above basic event to be valid. Similarly, the integrity of the wiring looms between the main LRUs of the modification (PFD, DCU, ADC, etc.) must also be ensured either by including within the FTs as additional basic events or separately analysing the wiring looms to ensure they meet the required integrity, and predicted failure rates required to ensure their contribution to the FT is negligible.

4.3.2.2 Development Assurance Level Fault Tree Analysis for loss of primary barometric altitude display (annunciated)

With reference to Fig. 1.8, the DAL FTA in Fig. 4.2 is proposed³⁶ to represent this functional failure condition, with the assumption that Functional Independence is claimed but not Item Development Independence [SAE ARP4754A, paragraph 5.2.3.2.3.3].

³⁶ Proposed, as subject to validation (step 3) and verification (step 4).

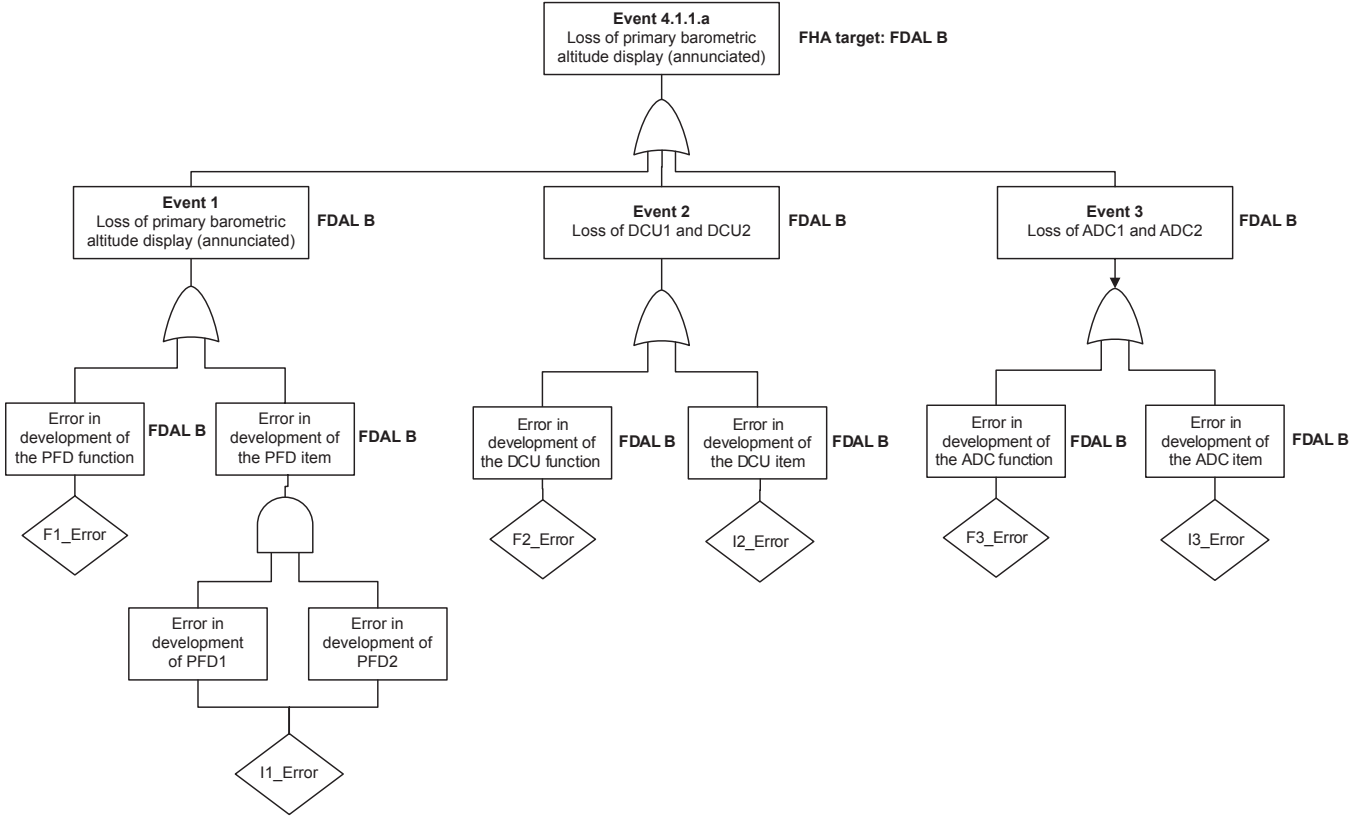


Figure 4.3 DAL FTA #4.1.1.a.2.

4.3.2.3 *Reliability Fault Tree Analysis for incorrect functioning of primary barometric altitude display (unannunciated)*

To have an unannunciated (misleading) display of barometric altitude on the PFD, at least one of the primary displays must show a misleading reading to the flight crew. Due to the cross-connected inputs (each DCU reads both ADC) and data checking between the DCU, the possibilities for erroneous but believable data are dramatically reduced. The DCU will typically raise a crew alert if data are missing, out of range, changes unreasonably fast or the readings from ADC1 and ADC2 are unreasonably different.

- Event 1: To display misleading and similar barometric altitude on both altitude displays, the same misleading data must be fed to both DCU, otherwise cross-check between them would identify a fault to the flight crew making it no longer misleading. Simultaneous blockage of both the port and starboard static ports is considered to be the only means this could occur. Note, there may be a common cause for both ports being blocked. This is not shown in this initial FT, but would be analysed later in the CMA which may result in a required update to the tree (see [Chapter 6](#) for more information).
- Event 2: Due to the cross-check between the two DCU, the only further means to display misleading data is if one of the ADC is no longer providing any output combined with the remaining ADC producing incorrect data (again due to blockage of its static port). Event 2 considers the case of loss of ADC1 combined with the static port input to ADC2 being blocked.
- Event 3: This event is the opposite of Event 2 and considers the case of loss of ADC2 combined with static port input to ADC1 being blocked.

Due to the cross-check, failures of the DCU do not feature within the misleading FT. Additionally, the Reversion switch does not play a part as the flight crew would not have a need to use the Reversion switch if the display is not known to be wrong. It is also assumed that the PFD displays the name of the input source, so failure of the Reversion switch if used would be obvious to the flight crew. This assumption would need to be validated against crew procedures.

In constructing this fault tree, it was assumed that the flight crew would still be misled if one ADC is no longer providing data and resulting in both DCU using the same data source which simultaneously is erroneous. In reality, a crew alert will be raised for missing data and the flight crew would cross-check with the standby flight instrument.

During verification (checking) of the fault trees, it is crucial that fault trees involving crew operations are verified by comparison with flight reference cards and operating manuals, and wherever possible by consulting flight test pilots who have experience on the system. The same is true for verification of fault trees with maintainer operations, where the advice of experienced maintainers is invaluable.

The FTA in [Fig. 4.4](#) is proposed³⁷ to represent this functional failure condition.

³⁷ Proposed, as subject to validation (step 3) and verification (step 4).

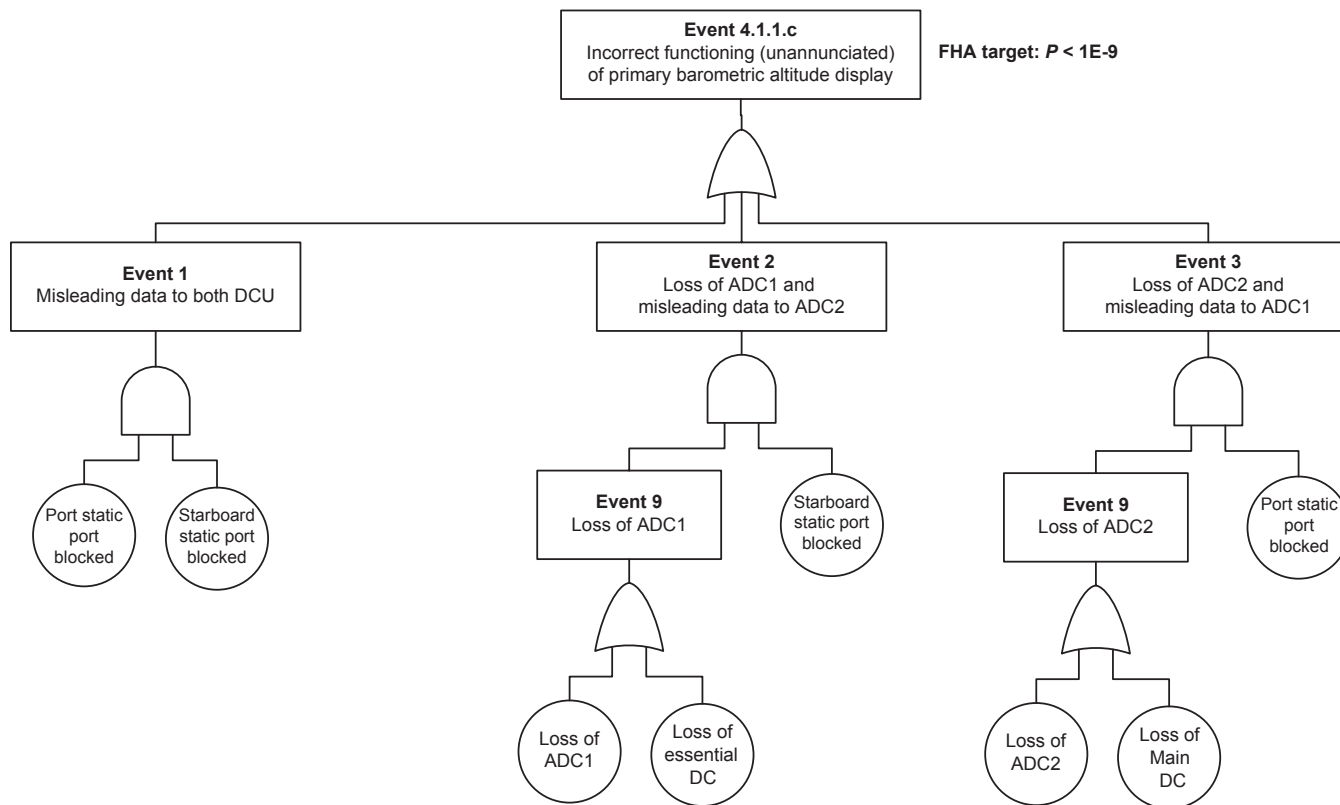


Figure 4.4 Reliability FTA #4.1.1.c.1.

Table 4.5 provides a list of the basic events for FTA #4.1.1.c.1, with their failure probability and reference to the source of these data. This will be used during validation of the FT to determine the probability of the top event.

Table 4.5 Basic events for FTA #4.1.1.c.1

ID	Description	Failure probability	Source/comments
PortStatic	Port static port blocked	1E-4/flight hour	Aircraft FRACAS (refer #...)
StbdStatic	Starboard static port blocked	1E-4/flight hour	Aircraft FRACAS (refer #...)
ADC1	Loss of ADC 1	1E-4/flight hour	Supplier FMES (refer #...) for ADC
ADC2	Loss of ADC 2	1E-4/flight hour	Supplier FMES (refer #...) for ADC
EssDC	Loss of Essential DC	1E-7/flight hour	Aircraft FRACAS (refer #...)
MainDC	Loss of Main DC	1E-6/flight hour	Aircraft FRACAS (refer #...)

4.3.2.4 Development Assurance Level Fault Tree Analysis for incorrect functioning of primary barometric altitude display (annunciated)

With reference to Fig. 1.8, the FTA in Fig. 4.5 is proposed³⁸ to represent this functional failure condition.

4.3.3 Step 3: validation

The four fault trees generated in Section 4.3.2 must be validated uses the steps outlined in Section 4.2.3. The purpose of the validation is to ensure that the architecture is correct and that the targets will be satisfied by the proposed design solution.

4.3.3.1 Reliability Fault Tree Analysis for loss of primary barometric altitude display (annunciated)

With reference to Section 4.2.3.1, the first step to validate the reliability FT is to ensure that the proposed architecture is suitable and that the appropriate subsystems and components have been included.

Once this is done, the next step to validate the tree is to populate the FTA with failure probability data and calculate the resultant probability for the top-level gate.

³⁸ Proposed, as subject to validation (step 3) and verification (step 4).

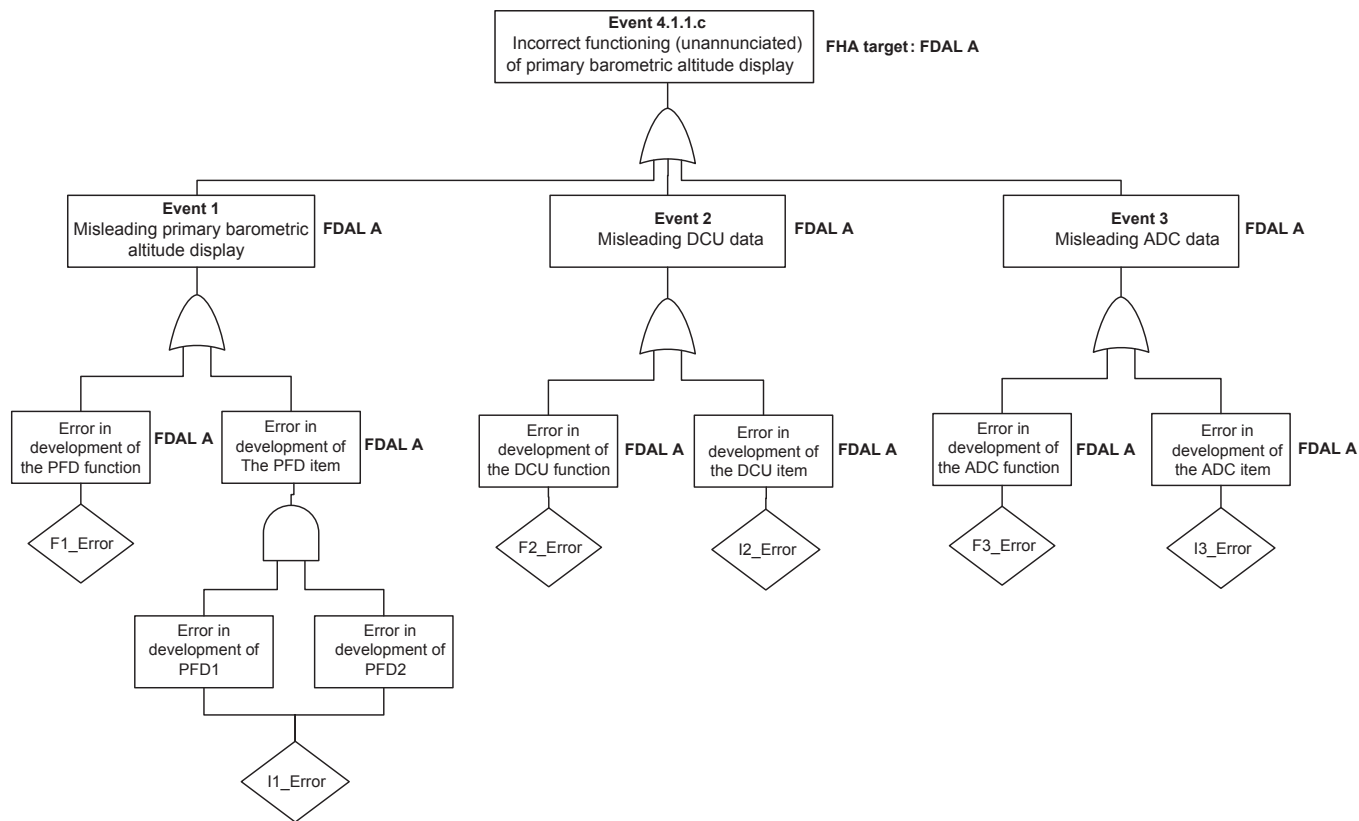


Figure 4.5 DAL FTA #4.1.1.c.2.

The fault tree of Fig. 4.2 is written out and simplified by expanding all multiplied (i.e. AND) terms using the rules detailed in Section 4.2.3.1:

$$\begin{aligned}
 P_{\text{Loss of All Display}} &= (\text{PFD1 OR EssDC}) \text{ AND } (\text{PFD2 OR MainDC}) \\
 &= (\text{PFD1} + \text{EssDC}) \times (\text{PFD2} + \text{MainDC}) \\
 &= (\text{PFD1} \times \text{PFD2}) + (\text{PFD1} \times \text{MainDC}) + (\text{EssDC} \times \text{PFD2}) \\
 &\quad + (\text{EssDC} \times \text{MainDC}) \\
 P_{\text{Loss of Both DCU}} &= (\text{DCU1 OR EssAC}) \text{ AND } (\text{DCU2 OR MainAC1}) \\
 &= (\text{DCU1} + \text{EssAC}) \times (\text{DCU2} + \text{MainAC1}) \\
 &= (\text{DCU1} \times \text{DCU2}) + (\text{DCU1} \times \text{MainAC1}) + (\text{EssAC} \times \text{DCU2}) \\
 &\quad + (\text{EssAC} \times \text{MainAC1}) \\
 P_{\text{Loss of Altitude Data}} &= (\text{ADC1 OR EssDC}) \text{ AND } (\text{ADC2 OR MainDC}) \\
 &= (\text{ADC1} + \text{EssDC}) \times (\text{ADC2} + \text{MainDC}) \\
 &= (\text{ADC1} \times \text{ADC2}) + (\text{ADC1} \times \text{MainDC}) + (\text{EssDC} \times \text{ADC2}) \\
 &\quad + (\text{EssDC} \times \text{MainDC}) \\
 P_{\text{Top}} &= P_{\text{Loss of All Display}} \text{ OR } P_{\text{Loss of Both DCU}} \text{ OR } P_{\text{Loss of Altitude Data}} \\
 &= (\text{PFD1} \times \text{PFD2}) + (\text{PFD1} \times \text{MainDC}) + (\text{EssDC} \times \text{PFD2}) \\
 &\quad + (\text{EssDC} \times \text{MainDC}) + (\text{DCU1} \times \text{DCU2}) + (\text{DCU1} \times \text{MainAC1}) \\
 &\quad + (\text{EssAC} \times \text{DCU2}) + (\text{EssAC} \times \text{MainAC1}) + (\text{ADC1} \times \text{ADC2}) \\
 &\quad + (\text{ADC1} \times \text{MainDC}) + (\text{EssDC} \times \text{ADC2}) + (\text{EssDC} \times \text{MainDC})
 \end{aligned}$$

Applying the idempotent law $A \times A = A$ and $A + A = A$ means that any duplicate terms should be removed, which in this case is the repeated ‘(EssDC \times MainDC)’. As there are no single terms, applying the absorption law $A \times (A + B) = A$ and $A + (A \times B) = A$ has no effect. The final equation is therefore:

$$\begin{aligned}
 P_{\text{Top}} &= (\text{PFD1} \times \text{PFD2}) + (\text{PFD1} \times \text{MainDC}) + (\text{EssDC} \times \text{PFD2}) \\
 &\quad + (\text{EssDC} \times \text{MainDC}) + (\text{DCU1} \times \text{DCU2}) + (\text{DCU1} \times \text{MainAC1}) \\
 &\quad + (\text{EssAC} \times \text{DCU2}) + (\text{EssAC} \times \text{MainAC1}) + (\text{ADC1} \times \text{ADC2}) \\
 &\quad + (\text{ADC1} \times \text{MainDC}) + (\text{EssDC} \times \text{ADC2}) \\
 &= (\text{PFD1} \times \text{PFD2}) + (\text{PFD1} \times \text{MainDC}) + (\text{EssDC} \times \text{PFD2}) \\
 &\quad + (\text{EssDC} \times \text{MainDC}) + (\text{DCU1} \times \text{DCU2}) + (\text{DCU1} \times \text{MainAC1}) \\
 &\quad + (\text{EssAC} \times \text{DCU2}) + (\text{EssAC} \times \text{MainAC1}) + (\text{ADC1} \times \text{ADC2}) \\
 &\quad + (\text{ADC1} \times \text{MainDC}) + (\text{EssDC} \times \text{ADC2})
 \end{aligned}$$

The next step to validate the reliability FT is to test the independence of events within the tree. This is not simply ensuring that repeated events are not present (which have already been resolved by applying Boolean algebra), but to consider if there are common causes that can cause otherwise independent components to fail simultaneously (such as use of common components of the same batch, common environmental factors or common interfaces like power sources). Of particular importance is to ensure that the events on either side of an AND gate are truly independent, as common causes (like repeated events) can result in a lower probability estimate than is true by several

orders of magnitude. This independence assessment is typically achieved by carrying out a CMA, see [Chapter 6](#) for more information. While it is advisable to redraw the tree to represent identified common causes, there are methods to apply Common Cause Factors (CCF) to events in a fault tree such as the Beta model (see [Annex A3](#)).

The final step to validate the reliability fault tree is to input the basic event failure rates and calculate the probability of the top-level gate.³⁹ The basic event failure rates for the case study are taken from [Table 4.4](#) and entered into the equation derived above:

$$\begin{aligned}
 P_{\text{Top}} &= P_{\text{Loss of All Display}} \text{ OR } P_{\text{Loss of Both DCU}} \text{ OR } P_{\text{Loss of Altitude Data}} \\
 &= (PFD1 \times PFD2) + (PFD1 \times \text{MainDC}) + (\text{EssDC} \times PFD2) + (\text{EssDC} \times \text{MainDC}) \\
 &\quad + (\text{DCU1} \times \text{DCU2}) + (\text{DCU1} \times \text{MainAC1}) + (\text{EssAC} \times \text{DCU2}) \\
 &\quad + (\text{EssAC} \times \text{MainAC1}) + (\text{ADC1} \times \text{ADC2}) + (\text{ADC1} \times \text{MainDC}) + (\text{EssDC} \times \text{ADC2}) \\
 &= (2\text{E-}4 \times 2\text{E-}4) + (2\text{E-}4 \times 1\text{E-}6) + (1\text{E-}7 \times 2\text{E-}4) + (1\text{E-}7 \times 1\text{E-}6) + (5\text{E-}5 \times 5\text{E-}5) \\
 &\quad + (5\text{E-}5 \times 1\text{E-}4) + (1\text{E-}7 \times 5\text{E-}5) + (1\text{E-}7 \times 1\text{E-}4) + (1\text{E-}4 \times 1\text{E-}4) + (1\text{E-}4 \times 1\text{E-}6) \\
 &\quad + (1\text{E-}7 \times 1\text{E-}4) \\
 &= (4\text{E-}8) + (2\text{E-}10) + (2\text{E-}11) + (1\text{E-}13) + (1\text{E-}9) + (5\text{E-}9) + (5\text{E-}12) + (1\text{E-}11) \\
 &\quad + (1\text{E-}8) + (1\text{E-}10) + (1\text{E-}11) \\
 &= 5.6\text{E-}8/\text{flight hour}
 \end{aligned}$$

The above probability prediction is now compared against the target provided in the FHA of 1E-7/flight hour. As can be seen, the proposed design configuration provides adequate reliability to meet the required target. This result can be fed back into the FHA as evidence to demonstrate the robustness of the design. However, no matter the probability estimate, the analyst should still use the FTA to determine if there are any inherent weaknesses in the design that could be improved.

It is worth noting at this point that, once Boolean algebra has been applied to the fault tree, the remaining added parts represent the cut sets of the fault tree. As shown in [Table 4.6](#), there are 11 cut sets in FTA #4.1.1.a.1, which have been ranked in the order of decreasing probability.

The cut sets can be drawn as individual FT diagrams and can be used by the safety working group to focus on improving the design. Normally only those cut sets featuring a single point failure or within a few orders of magnitude of the target probability need to be focused on.

4.3.3.2 Development Assurance Level Fault Tree Analysis for loss of primary barometric altitude display (annunciated)

With reference to [Section 4.2.3.2](#), FDALs are allocated from the top down using the OR gate logic (as indicated in bold font in [Fig. 4.3](#)).

³⁹ Simply summing the OR's events is a convenient approximation that only holds when all inputs are small (of the order of 1E-3 or smaller). This is known as 'rare event approximation', see NASA Fault Tree Handbook paragraph 7.1 for more information.

Table 4.6 Cut sets for FTA #4.1.1.a.1

Cut set	Contributing probability	Contributing events
Cut set 1:	4E-8/flight hour	PFD1 AND PFD2
Cut set 2:	1E-8/flight hour	ADC1 AND ADC2
Cut set 3:	5E-9/flight hour	DCU1 AND MainAC1
Cut set 4:	1E-9/flight hour	DCU1 AND DCU2
Cut set 5:	2E-10/flight hour	PFD1 AND MainDC
Cut set 6:	1E-10/flight hour	ADC1 AND MainDC
Cut set 7:	2E-11/flight hour	EssDC AND PFD2
Cut set 8:	1E-11/flight hour	EssAC AND MainAC1
Cut set 9:	1E-11/flight hour	EssDC AND ADC2
Cut set 10:	5E-12/flight hour	EssAC AND DCU2
Cut set 11:	1E-13/flight hour	EssDC AND MainDC

In Fig. 4.3, independence is indicated via an AND gate and the assessor then has some options on how to flow the DAL down:

- Fig. 4.3 assumes that each of the duplicate/redundant items is developed by the same supplier (which minimised the spare part burden), so the IDAL for each of these function is set at Level B
- If any of the duplicate/redundant items were to be developed independently, then the assessor would have had two options (refer Table 4.2) for the Item Development Assurance of each as provided below for the PFD:
 - One PFD at Level B and the other at Level C
 - Both PFDs at Level C

4.3.3.3 Reliability Fault Tree Analysis for incorrect functioning (unannunciated)

The FT in Fig. 4.3 (RELIABILITY FTA #4.1.1.c.1) can be validated using exactly the same method as used for FTA #4.1.1.a.1 as described in Section 4.3.3.1:

$$\begin{aligned}
 P_{\text{Event 1}} &= (\text{PortStatic AND StbdStatic}) \\
 &= (\text{PortStatic} \times \text{StbdStatic}) \\
 P_{\text{Event 2}} &= (\text{ADC1 OR EssDC}) \text{ AND } (\text{StbdStatic}) \\
 &= (\text{ADC1} + \text{EssDC}) \times (\text{StbdStatic}) \\
 &= (\text{ADC1} \times \text{StbdStatic}) + (\text{EssDC} \times \text{StbdStatic}) \\
 P_{\text{Event 3}} &= (\text{ADC2 OR MainDC}) \text{ AND } (\text{PortStatic}) \\
 &= (\text{ADC2} + \text{MainDC}) \times (\text{PortStatic}) \\
 &= (\text{ADC2} \times \text{PortStatic}) + (\text{MainDC} \times \text{PortStatic})
 \end{aligned}$$

$$\begin{aligned}
 P_{\text{Top}} &= P_{\text{Event 1}} \text{ OR } P_{\text{Event 2}} \text{ OR } P_{\text{Event 3}} \\
 &= (\text{PortStatic} \times \text{StbdStatic}) + (\text{ADC1} \times \text{StbdStatic}) + (\text{EssDC} \times \text{StbdStatic}) + \\
 &\quad (\text{ADC2} \times \text{PortStatic}) + (\text{MainDC} \times \text{PortStatic})
 \end{aligned}$$

Applying the idempotent law $A \times A = A$ and $A + A = A$ has no effect as there are no duplicate terms. Also as there are no single terms, applying the absorption law $A \times (A + B) = A$ and $A + (A \times B) = A$ has no effect. The final equation is therefore unchanged.

The probability can be calculated by entering the probability of each basic event (from [Table 4.5](#)) into the above equation, as follows:

$$\begin{aligned}
 P_{\text{Top}} &= (\text{PortStatic} \times \text{StbdStatic}) + (\text{ADC1} \times \text{StbdStatic}) + (\text{EssDC} \times \text{StbdStatic}) \\
 &\quad + (\text{ADC2} \times \text{PortStatic}) + (\text{MainDC} \times \text{PortStatic}) \\
 &= (1\text{E-}4 \times 1\text{E-}4) + (1\text{E-}4 \times 1\text{E-}4) + (1\text{E-}7 \times 1\text{E-}4) + (1\text{E-}4 \times 1\text{E-}4) + (1\text{E-}6 \times 1\text{E-}4) \\
 &= (1\text{E-}8) + (1\text{E-}8) + (1\text{E-}11) + (1\text{E-}8) + (1\text{E-}10) \\
 &= (1\text{E-}8) + (1\text{E-}8) + (1\text{E-}11) + (1\text{E-}8) + (1\text{E-}10) \\
 &= 3.1\text{E-}8/\text{flight hour}
 \end{aligned}$$

The above probability prediction is now compared against the target provided in the FHA of $1\text{E-}9/\text{flight hour}$. In this instance, the proposed design architecture does not provide sufficient resilience to displaying misleading barometric altitude data to the flight crew on the PFDs. The analyst must convene a safety working group with the design team to enact a change to the proposed design or to understand in more detail the maturity of the reliability data used and whether any of the basic events can be developed in more detail. As shown in the [Table 4.7](#), there are five cut sets in FTA #4.1.1.c.1, which have been ranked in the order of decreasing probability.

Table 4.7 Cut sets for FTA #4.1.1.c.1

Cut set	Contributing probability	Contributing events
Cut set 1:	$1\text{E-}8/\text{flight hour}$	PortStatic AND StbdStatic
Cut set 2:	$1\text{E-}8/\text{flight hour}$	ADC1 AND StbdStatic
Cut set 3:	$1\text{E-}8/\text{flight hour}$	ADC2 AND PortStatic
Cut set 4:	$1\text{E-}10/\text{flight hour}$	MainDC AND PortStatic
Cut set 5:	$1\text{E-}11/\text{flight hour}$	EssDC AND StbdStatic

The safety working group should therefore focus on blockages of the port and starboard static ports either together, or with the associated ADC.

4.3.3.4 Development Assurance Level Fault Tree Analysis for incorrect functioning (unannunciated)

With reference to [Section 4.2.3.2](#), FDALs are allocated from the top down using the OR gate logic (as indicated in bold font in [Fig. 4.5](#)).

In [Fig. 4.5](#), independence is indicated via an AND gate, and the assessor then has some options on how to flow the DAL down:

- [Fig. 4.5](#) assumes that each of the duplicate/redundant items are developed by the same supplier (which minimised the spare part burden), so the IDAL for each of these function are set at Level A.
- If any of the duplicate/redundant items were to be developed independently, then the assessor would have had two options (refer [Table 4.2](#)) for the Item Development Assurance of each as provided below for the PFD:
 - One PFD at Level A and the other at Level C.
 - Both PFDs at Level B.

4.3.4 Step 4: verification

The FTA needs to be verified prior to being finalised for publication (or reference) in the System Safety Assessment. The validated FTA might have been completed in the early stages of the design to help substantiate that we have a suitable architecture in the proposed design. However:

- Up to CDR (and hopefully not beyond), constant tweaks are made to the solution as the design matures.
- After CDR (during the build phase) the design team may have to deal with change requests and concessions or waivers.

It is thus important the FTA is verified prior to publication to ensure that all these changes are accounted for and that nothing has been missed.

With reference to [Section 4.2.4](#), verification activities might include:

- ZSA inspections (see [Chapter 8](#) Step 3) of the installed assembly.
- Review of all drawing revisions and change requests.
- Independent analysis of the FTA, which might include contracting someone to draw an FTA for the same scenario without seeing the in-house produced FTA.
- Inducing failure modes on the actual system to see whether the FTA can be used to usefully diagnose causes and direct corrective action.

Once verified, the FTA can be authorised (i.e. fully signed off) for publication in support of certification.

4.4 Discussion

The FTA is the most commonly used technique for causal analysis in risk and reliability studies ([Nighot, 2003](#)). It should be undertaken as soon as engineers start defining system architecture as it provides the mechanism for them to evaluate the integrity of that architecture. As illustrated in [Fig. 4.1](#), it should be a live document reflecting the evolving system architecture at all times, with an emphasis on ensuring that no common mode vulnerabilities are introduced or missed.

However, the FTA is not the only tool which is available to use in this manner, and other options include:

- Dependence Diagrams (DD): These are similar to the FTA, but replace the logic gates by paths to show the relationship of the failures. A DD analysis is success oriented and is conducted from the perspective of which failures must not occur to preclude a defined Failure Condition. For more information, see Appendix E in SAE ARP4761.
- Markov Analyses (MA): Similar to the DD and FTA, but additionally calculates the probability of the system being in various states as a function of time. Here airworthiness is not a simple mathematical calculation, but depends on relative states of part of the system. However, MA has limitations that an FTA does not (e.g. it is difficult to model large complex systems and visualise fault paths in an MA model). For more information, see Appendix F in SAE ARP4761 (App F) and Clifton (2005, Ch18).

The FTA calculation methodology described in this chapter represents only simple calculation methods, which are an approximation that can be used when all inputs are constant and small (less than $1E-3$). If probabilities greater than this need to be used (especially when setting certain branches to a probability of one during sensitivity assessment of failures), a more accurate calculation method must be used. Refer to either NUREG-0492 or the NASA Fault Tree Handbook for more details, but it is worth noting that using a computer software tool for quantitative calculation of FTA can be important in minimising the potential for error.

In addition to the constant probability calculations described in this chapter, it is possible, although not common, to use FTA to perform some advanced failure analysis, such as:

- Time-dependent assessment. This allows for assessing the probability of failure as a function of time and for representing basic events which have a variable probability rate [see NASA Fault Tree Handbook, paragraph 7.7].
- Causal analysis. Intricate modelling of system components and states, and the impact of their failure. Causal analysis can be used to build up an FTA for different outcomes [Bossche, paragraph 3]. Algorithms to calculate component models can be used to build up reusable models for basic components.

The complexity of the mathematics for time-dependent assessment and component models mean this type of modelling can only realistically be achieved using computer software, and its usefulness is limited to very specific scenarios where accepted rate-based probabilities cannot be applied or approximated.

These complexities aside, the primary requirement in the construction of an accurate FT is an excellent working knowledge of the system. Accordingly, the emphasis (at least initially) should be on the process (i.e. the acquisition of information) and not the product (i.e. the FTA model). This is an important distinction [NUREG-0492, paragraph 2] because, in the absence of a directed, manageable and disciplined process, the resulting model will be poor. The nature of the decision-making process is shown in Fig. 4.6, where:

- block A represents the actual reality (e.g. the system as implemented on the aircraft);
- block B represents our perception of reality (e.g. our FTA);
- block C represents the conclusions we draw from our perceptions of reality.

Clearly the emphasise must therefore be on the accuracy of our model, hence the importance of the verification and validation activities.

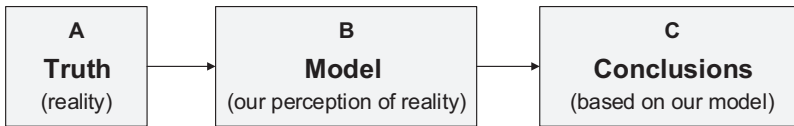


Figure 4.6 Model and reality tailored from [NUREG 0492 (Fig I-3)].

What is clear is that, while FTA is mostly used to provide a quantitative assessment of a failure condition, it remains fundamentally a qualitative analysis method due to the means that the FTA is developed [see NASA Fault Tree Handbook paragraph 1.2]. Nevertheless, the discipline the analyst goes through to consider each failure path methodically provides an excellent deductive method to provide a reasoned estimate of failure probability. Additionally, the FTA provides more information than simply probability of the top event and can be used even without probability calculation to understand weaknesses in the system design (such as single point failures) and to conduct sensitivity analysis to determine which parts of the system may drive the overall probability of particular failure modes.

4.5 Conclusions

The FTA is a systematic method for acquiring information about a system which drives informed decision-making. By its nature, it needs to be repetitive, structured and methodical.

4.5.1 Advantages

As a deductive analysis tool, FTA is useful in determining the combinations of failures that must occur for an undesired event to materialise. The foremost strength of the FTA is that it is a decision support tool [see paragraph 1.5 in the NASA Fault Tree Handbook]. The methodology helps managers and engineers find design and operational weaknesses in complex systems so as to systematically and efficiently uncover and prioritise safety improvements. It is therefore the correct tool to use when system failure states and associated probability targets have been identified (e.g. via the FHA) and must be analysed (e.g. via the FTA) to determine whether the system will meet (in the PSSA), or has met (in the SSA) these targets. It provides a visual representation of:

- the system architecture, so ensuring that all stakeholders understand the functional dependencies;
- the combination of failures which most probably lead to a functional failure, which can then be used to optimise system improvements.

The FTA is also a great tool for prioritising effort as it focuses on only those causes leading to a specific failure condition. This reduces costs of the Safety Assessment

(during the IAW phase) as well as costs of fault diagnostics (during the CAW phase). To best benefit from FTA in management decisions, the NASA Fault Tree Handbook advises that it is important that managers and their support staffs be familiar with the value and application of this method. In addition, there should be a small but robust group of in-house technical experts that understand the methods and can explain its meaning and applicability to given problems to management and serve as in-house technical advisers to the management decision process for safety improvement.

4.5.2 Limitations

The FTA does, however, have its limitations, of which some are listed below:

- The top event of the FT directs all of the rest of the analysis. If the top event is incorrectly defined (and this happens a surprising number of times), then the FTA will be incorrect, which can result in wrong decisions being made [NASA Fault Tree Handbook, paragraph 3.4].
- An FTA tree is not a model of all possible system failures or all possible causes for system failure. Instead, the tree is tailored to the top event that corresponds to some particular system failure mode, and the FT thus includes only those faults that contribute to this top event. Moreover, these faults are not exhaustive – they cover only the faults that are assessed to be realistic by the analyst.
- FTA and its Boolean nature is limited to considering components with only two states, either good or failed [Bossche, 1988, paragraph 2]. In reality, components may have multiple degraded states that have different impacts on system behaviour, and further variations depending on flight phase or time variables. It is very difficult if not impossible to consider this type of complexity in FTA, and other tools (such as Markov) may be useful.
- Can be costly (time-consuming) and cumbersome, so best suited at System Level 3 (refer Fig. 1.1).
- Requires thorough understanding of the design, which might not be mature enough at the early stages (e.g. bid phase of the design). It is terror iterative to reflect the evolving system architecture.
- It is easy for failure paths to be missed and is open to some interpretation (i.e. there is more than one way to draw a valid tree for the same system).
- Poor treatment of explicit time dependence.
- If it is assumed that the alternative channels of the systems are completely independent, and no allowance is made for common mode failure (refer Chapter 6), this leads to very low and optimistic values of failure probability being predicted.
- The FTA does not have 100% fidelity. It is a model/estimate/perception of reality.
- The FTA is very dependent on accurate failure probability data. It is therefore important to conduct ‘*sensitivity analyses*’ and ‘*uncertainty analyses*’ on the FTA [see paragraph 7.7 in the NASA Fault Tree Handbook for more information].

References

- Bossche, A., 14 January 1988. Fault Tree Analysis and Synthesis. A Ph.D. thesis submitted to the University of Delft.
- Clemens, P.L., February 2002. Fault Tree Analysis, a Jacobs Sverdrup Presentation.
- CS25, Certification Specifications and Acceptable Means of Compliance for Large Aeroplanes, Amendment. European Aviation Safety Agency, Cologne.
- Eckberg, C.R., 1964. WS-133B Fault Tree Analysis Program Plan, (Rev B). The Boeing Company, Seattle, WA. D2-30207-1.

- Ericson, C.A., 2005. Hazards Analysis Techniques for System Safety. Wiley-Interscience.
- Evans, R.A., 05 January 1976. Engineering Design Handbook, Design for Fault Tree Handbook with Aerospace Applications, Version 1.1, Prepared for NASA Office of Safety and Mission Assurance. NASA Headquarters, Washington, DC. 20546. August 2002.
- Hixenbaugh, A.F., 1968. Fault Tree for Safety. The Boeing Company, Seattle, WA. D6-53604. Retrieved 2014-05-18.
- Haroonabadi, H., Haghifam, M., 2009. Generation reliability evaluation in power markets using Monte Carlo simulation and neural networks. In: 15th International Conf. on Intelligent Systems Applications to Power Systems, Curitiba.
- IEC 61025. Fault Tree Analysis (FTA). International Electrotechnical Commission.
- Javadi, M., Nobakht, A., Meskabashee, A., September 2011. Fault tree analysis approach in reliability assessment of power system. International Journal of Multidisciplinary Sciences and Engineering. 2 (6). <http://www.ijmse.org/Volume2/Issue6/paper9.pdf>.
- Kritzinger, D., 2006. Aircraft System Safety: Civil and Military Aeronautical Applications. Woodhead Publishing.
- Larsen, W., January 1974. Fault Tree Analysis. Picatinny Arsenal. Technical Report 4556. Retrieved 2014-05-17.
- MIL-HDBK-338B. Electronic Reliability Design Handbook. U.S. Department of Defense.
- Nighot, R., 2003. Incorporation Substation and Switching Station Related Outages in Composite System Reliability Evaluation (M.Sc. thesis). College of Graduate Studies and Research, University of Saskatchewan, Canada.
- NUREG-0492, 1981. Fault Tree Handbook. Office of Nuclear Regulatory Research, U.S. Nuclear Regulatory Common, Washington, USA.
- RTCA DO-178. Software Considerations in Airborne Systems and Equipment Certification. RTCA, Inc.
- RTCA DO-254. Design Assurance Guidance for Airborne Electronic Hardware. RTCA, Inc.
- SAE ARP4754A, 2010. Guidelines for Development of Civil Aircraft and Systems, the Engineering Society for Advanced Mobility Land Sea Air and Space. Aerospace Recommended Practise, Warrendale, USA.
- SAE ARP4761, 1996. Guidelines and Methods for Conducting the Safety Assessment on Civil Airborne Systems and Equipment, the Engineering Society for Advanced Mobility Land Sea Air and Space. Aerospace Recommended Practise, Warrendale, USA.

Further reading

- Andrews, J., Moss, T.Y., 1993. Reliability and Risk Assessment. Longman Scientific & Technical.
- Barlow, R., Fussel, J., Singpurwalla, N., 1970. Reliability and Fault Tree Analysis, Conference on Reliability and Fault Tree Analysis. US Berkeley, SIAM Pub.
- Henley, E., Kumamoto, H., 1996. Probabilistic Risk Assessment and Management for Engineers, second ed. IEEE Press.
- Rauzy, A., 1993. New Algorithms for Fault Tree Analysis, Reliability Engineering and System Safety, vol. 40.
- Sinnamon, R., Andreas, J., January 1996. Fault tree analysis and binary decision diagrams. In: Proceedings of the Reliability and Maintainability Symposium.
- Vesely, W., et al., 2002. Fault Tree Handbook with Aerospace Applications. NASA Office of Safety and Mission Assurance. Version 1.1.
- System Safety Handbook. 30 December 2000. Federal Aviation Administration. http://www.faa.gov/regulations_policies/handbooks_manuals/aviation/risk_management/ss_handbook/.

Annex A to Chapter 4

Calculation Methods and Boolean Algebra in Fault Tree Analysis

A1. Basic event failure rates

The greater adoption of reliability engineering during and post the Second World War has put an emphasis on the need for accurate failure data. Many engineering sectors have now collected a vast amount of statistical data on product usage and failure. These sectors include, notably, the military, energy sectors (oil, gas and nuclear) and automobiles. This can be of benefit to the safety assessor when attempting to quantitatively evaluate a fault tree.

There are a number of sources of data for failure rates, including Failure Modes and Effects Analysis (FMEA)/Failure Modes and Effects Summary (FMES), product data sheets, failure tests data and accelerated life tests. However, these sources ultimately stem from the following data types:

- Similarity: This is simply the approach of reasoning a similarity between two components or systems and adopting the reliability of one to the other. This can be made due to legacy, heritage or similarity of internal components. However, it is extremely important to take into consideration the operational environment and the application, as this can have a large effect on the achieved reliability.
- Prediction: This is the use of detailed parts data to predict a failure rate for the unit. This is mostly used for electronic items, using standardised lookup data for part components such as that described in MIL-STD-217F and RIAC 217Plus. The assessor should be warned that the predicted failure rate for any failure mode from a tool such as MIL-STD-217F is an idealised number, useful for comparison between varying designs. Its correlation with true MTBF is dependent as much on manufacturing variability as operational usage, and therefore applying compensating factors should be considered.
- Actual: When using equipment that is already in use in the field or that has undergone reliability testing, actual failure data⁴⁰ can be used to estimate an MTBF. Large generic data sets have already been collated by organisations such as the Reliability Information Analysis Center (RIAC) on the failure rates of both electronic and nonelectronic parts. However, when using this data, it is important to select the closest environment to the intended application so as to ensure that effects from environmental factors (vibration, altitude, temperature, etc.) are taken into consideration. Alternatively, if using directly collected failure data, it is important to understand the confidence in the data to ensure that the result is statistically significant. While an MTBF can be calculated using a simple formula (i.e. number of failures divided by total number of operations/hours),⁴¹ the result may not be significant – particularly if no failures have occurred, little operational time has been recorded, or the failures that have occurred do not represent a significant proportion of the equipment in service. In these circumstances, the chi-squared distribution can be used as follows:

n = number of failures

T = cumulative hours of operation

⁴⁰ For more on component failure rate data, see NASA's FTA Handbook (paragraph 7.2) and Kritzinger (2006, paragraph 10.6).

⁴¹ For more on how to use failure rate data to calculate MTBF, see Kritzinger (2006, paragraph 10.2.4).

$k = \text{degrees of freedom} = 2 \times (n)$

Note: Use $n + 1$ instead of n if the test/data was stopped after a certain number of hours (time truncated) to represent the worst case of a failure occurring immediately after the test was stopped. If the test was stopped after a certain number of units failed (failure truncated), simply use n .

Look up the chi-squared value (χ^2) from distribution tables, or in spreadsheet applications, use the CHIINV function (if available):

$$\chi^2 = \text{CHIINV}(1 - \text{Confidence}, k)$$

where *Confidence* is the certainty needed in the result. Typically, a confidence of 95–99% is selected, with at least 95% required for a statistically significant result. Note that the value to be entered into CHIINV is *1-Confidence*, and therefore 95% confidence is entered as $1 - 0.95 = 0.05$. The resulting probability in failures per operating hour can then be calculated using the following formula: $p = \chi^2 / 2T$.

Fundamentally, all of the above are predictions. While past performance can be a good indication of future performance, there are too many variables to provide a truly accurate prediction of technical failure probabilities (let alone operational failure probabilities where human factors play a large role). However, this does serve as a good basis for comparing different designs, or in this case to support a quantitative calculation of an FTA to supplement a qualitative argument that the relative probability of a particular failure condition is suitably low. Note, AMC25.1309 paragraph 6c(2) states ‘*The analytical tools used in determining numerical values are intended to supplement, but not replace, qualitative methods based on engineering and operational judgement.*’

A2. Repeated events

The previous description of Boolean algebra is important when dealing with repeated events. Due to the idempotent and absorption laws (described in [Section 4.2.3.1](#)), basic events must not be repeated on both sides of a gate which requires independent events. In an AND gate, failing to apply the idempotent and absorption law can result in a lower probability estimate than is true. It is therefore essential that AND gates are subject to close scrutiny. While the idempotent and absorption laws also apply to OR gates, the probability predictions are worse if not applied, and therefore a mistake here would not falsely show that a safety target has been met.

Consider the following AND gate example in [Fig. 4A-1](#):

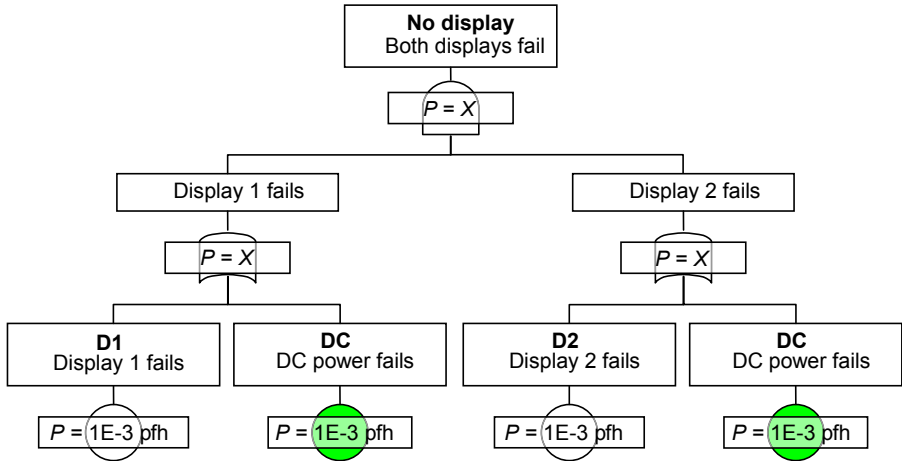


Figure 4A-1 FTA before applying Boolean Logic.

For comparison purposes, let us manually calculate the probability of the top-level event using an assumption that all basic events have a probability of 1E-3 per flight hour (pfh).

$$P = (D1 + DC) \times (D2 + DC) = (1E-3 + 1E-3) \times (1E-3 + 1E-3) = 4E-6 \text{ pfh}$$

If, however, we apply Boolean algebra to the tree:

$$P = (D1 + DC) \times (D2 + DC)$$

Expanding noting the associative, distributive and commutative laws,

$$P = (D1 \times D2) + (D1 \times DC) + (D2 \times DC) + (DC \times DC)$$

Applying the idempotent law, $A \times A = A$, for the DC event,

$$P = (D1 \times D2) + (D1 \times DC) + (D2 \times DC) + DC$$

Applying absorption law, $A + (A \times B) = A$, on all events with the DC basic event in it results in the following $(D1 \times DC) + (D2 \times DC) + DC = DC$, therefore,

$$P = (D1 \times D2) + DC$$

Always check that there are no more terms to apply the idempotent and absorption laws to.

We can now redraw the tree as shown in [Fig. 4A-2](#).

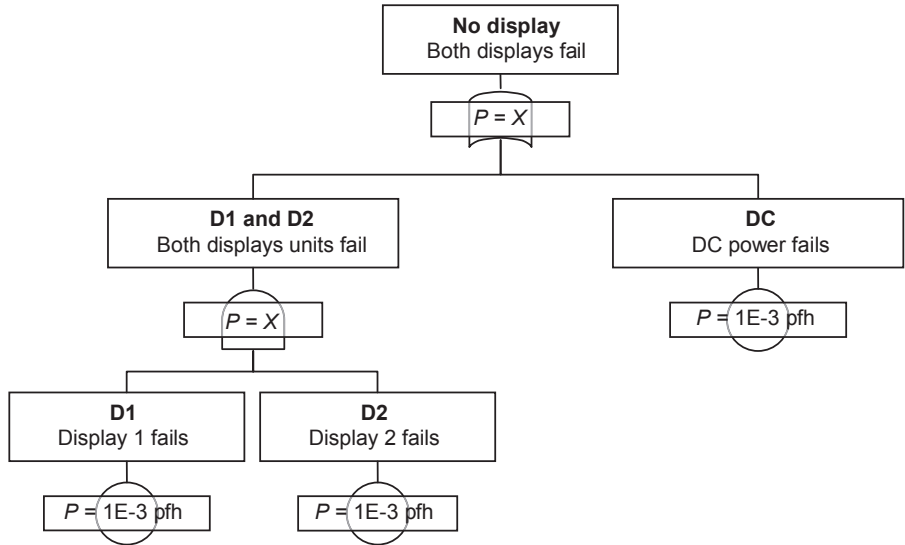


Figure 4A-2 FTA after applying Boolean Logic.

Again, if we calculate using 1E-3 pfh for all basic events, we get:

$$P = (D1 \times D2) + DC = (1E-3 \times 1E-3) + 1E-3 = 1E-3 \text{ pfh}$$

This is a significantly higher probability than that calculated using the tree with the event repeated on both sides of the fault tree. This illustrates that this type of mistake can make the probability of the top-level event seem smaller than it really is.

The above example may seem obvious, but this effect can easily be hidden when working with large fault trees across multiple pages. Most fault tree programs can handle repeated events across a fault tree and calculate the correct probability. However, the software can only apply this if the basic event entered on both sides of the tree is entered as the same event. If they have different event identifiers, the software will treat them as different events and allow the incorrect calculation of probability for the gate and the top-level event.

A3. Common cause factors

The Boolean logic of a fault tree requires that each basic event is independent. While we have tackled repeated events, and how they can be reduced using Boolean logic, there is also the case where a common cause can affect multiple seemingly independent components simultaneously. The common causes being referred to are either an external influence that may cause multiple otherwise independent parts to fail simultaneously (such as particular risks, or manufacturing defects for common batch components), or an implicit dependency in the failures where the failure of

one item influences the failure of others [see NASA Fault Tree Handbook paragraph 5.2]. It is important to consider these common cause factors where these events form inputs to an AND gate, and hence the common cause can violate the independence requirements between the contributing events. This can be accounted for in a fault tree by applying a Common Cause Factor (CCF) to all of the effected basic events under an AND gate.

There are many different models that can be used to apply common causes, but the most common (and the one preferred by IEC 61508) is the Beta factor (β) model. This model applies a β factor between 0 and 1 representing the fraction of the failure of all affected inputs resulting from the common cause. For instance, a β value of 0.1 implies that 10% of failures where all inputs fail were in fact the result of a common cause. There are some specialised resources for appropriate CCF values that can be applied, but fundamentally a sensitivity analysis should be performed to determine how much an effect the CCF has on the top event probability. A large influence would indicate the need for further analysis [see NASA Fault Tree Handbook paragraph 7.2].

An example of a common cause is where two components are identical and may be subject to batch-related quality issues. The Quality Management System (QMS) of suppliers in use today (such as ISO 9001 and AS 9100) has reduced the likelihood of quality-related failures; however, it is not possible to completely discount them.

Consider the simple case of two components A and B, which must both fail for the top event to occur. This can be represented as two inputs to an AND gate if the events are truly independent. However, if there is a potential common cause that could result in both components A and B failing together, this should be represented using the fault tree logic shown in [Fig. 4A-3](#).

The Beta factor model allows for the probability of occurrence to be calculated by adjusting the probability of the basic events using the Beta factor. These adjustments are shown in the fault tree above for the basic events.

The above layout would be cumbersome to construct for each fault tree generated, so many fault tree analysis programs allow CCF groups to be created. Basic events are added to the group and a Beta factor between 0 and 1 specified. This is normally represented visually on the fault tree by displaying a β symbol next to the included events, as shown in [Fig. 4A-4A_4](#) below.

The program will then calculate the probability of failure for the AND gate using the same formulas presented in the full layout of [Fig. 4A-3](#), namely:

$$P(\text{AND}) = (P(A) \times (1 - \beta)) \times (P(B) \times (1 - \beta)) \times (\dots) + (\text{MAX}(A, B, \dots) \times \beta)$$

While the common cause factor used above uses the maximum probability of the contributing events, the Beta factor model could equally apply minimum or mean average. While using the maximum probability is the most conservative, it should be noted that the Beta factor selected is often also conservative and it may be more appropriate to

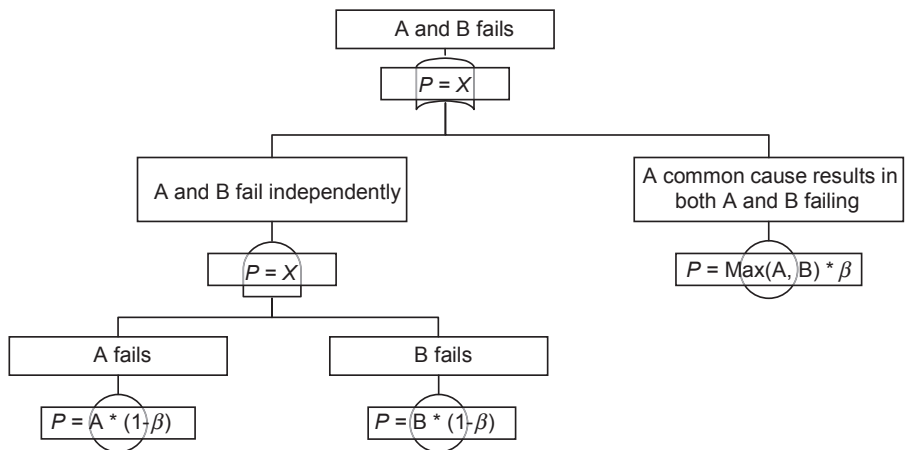


Figure 4A-3 Common Cause Factor using Beta model.

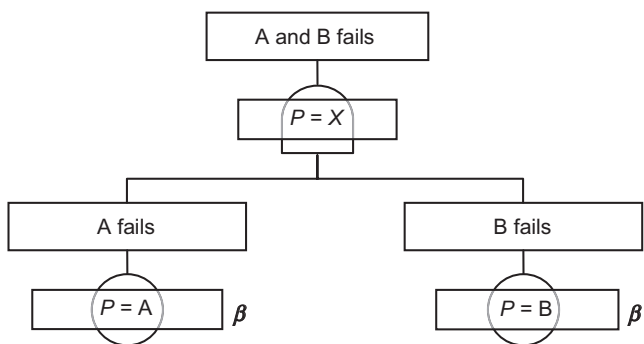


Figure 4A-4 Shorthand notation for Beta factor model.

use the mean average or minimum. When using a software tool to apply the Beta factor model, care should be taken to understand which model is used and this can often be selected in the calculation options. In addition, care should be taken when creating CCF groups as the maximum, average or minimum of ALL events in the created CCF group will normally be taken, rather than limiting only to those contributing to any particular AND gate.

A4. Human Error data

The NASA Fault Tree Handbook [paragraph 7.2] advises that Human Error quantification and Human Error reliability are different from Human Factors analysis:

- Human Factors Analysis is a psychological assessment of the factors affecting human behaviour. It is a qualitative analysis.
- Human Reliability Analysis quantifies the probability of different types of human actions. It is human reliability analysis that is used in FTA.

Human error rates are notoriously difficult to quantify from a reliability perspective and depend heavily on the situation, training, distractions, tiredness, etc. While human error databases have been compiled, the human error being modelled should be matched as closely as possible to the same type of human error in the database. It is advisable to consult expert opinion when estimating human error rate due to these factors, and the rate selected should factor in possible variations in human performance and conditions as well as statistical estimation error.

However, it can be useful to assign conservatively high human error rates to determine how sensitive the top event probability is to human error. Where a design is particularly sensitive to human error, a more detailed assessment is warranted to identify either the need for a redesign, compiling detailed instructions for use, or prescribing the need for enhanced levels of training. For more information, see [Chapter 10](#).

Failure Modes and Effects Analysis

5

Failure is the rule rather than the exception, and every failure contains information.

Steve Wozniak (Apple co-founder)

5.1 Introduction

In this chapter we explore the Failure Modes & Effects Analysis (FMEA) and the Failure Modes Effects & Criticality Analysis (FMECA) which, as the name suggests, simply is an extension of the FMEA.¹ This chapter will use the acronym FMEA and highlight those instances where is extended to include FMECA.

5.1.1 Background

An FMEA is a systematic ‘bottom-up’ method of:

- identifying single failure modes and failure probabilities of a system, item, function or piece-part (i.e. smallest individual part or component);
- determining the effects of a failure mode on the next higher level of the design (if available to the assessor as an LRU supplier will not necessarily know how much redundancy the system integrator is going to build into his system);
- classifying failure modes according to the worst case severity of the end effect (FMECA only).

The FMEA approach has been around for a very long time. Before any documented format was developed, inventors and process experts would try to anticipate what could go wrong with a design or process before it was developed or tried. The FMEA discipline was first formalised in the late 1940s by the US Military (refer [MIL-P-1629A](#), page 2) where it used as a reliability evaluation technique to determine the effect of system and equipment failures. Failures were classified according to their impact on mission success and personnel/equipment safety.

Later it was used for aerospace/rocket development (including the Apollo space program) to avoid errors in the relatively small sample sizes associated with costly rocket technology.

¹ The FMECA requires more information be obtained than an FMEA, particularly information dealing with the criticality and detection of the potential failure modes.

In the late 1970s the Ford Motor Company introduced FMEA to the automotive industry for safety and regulatory consideration after the Pinto scandal.² Ford also used it to improve the quality and efficiency of the production and design process.

The FMEA methodology is now extensively used in a variety of industries including semiconductor processing, food service, plastics, software and health care, where it is sometimes integrated into Advanced Product Quality Planning (APQP) to provide a primary design and process risk mitigation tool.

The Automotive Industry Action Group (AIAG), for example, requires the use of FMEA in the automotive APQP process and publishes a detailed manual on how to apply the method.³ Each potential cause must be considered for its effect on the product or process and, based on the risk, actions are determined and risks revisited post-completion of such actions. Toyota has advanced this even further with its Design Review Based on Failure Mode (DRBFM⁴) approach.

5.1.2 Aim of the Failure Modes and Effects Analysis

An FMEA is an analytical tool used to evaluate the impact that any single failure may have on the system under consideration. It answers the question ‘if this part fails, what will be the subsequent result?’

The aim of the FMEA is to identify actions to control, eliminate or reduce the highest priority⁵ failures.

5.1.3 Objectives of the Failure Modes and Effects Analysis

An FMEA is usually conducted for any one, or all, of the following objectives:

- To support the System Safety Assessment (SSA) by:
 - identifying failure modes which need highlighting (e.g. a failure mode that could cause an ignition source in a fuel tank), or

² Through early production of the Ford Pinto model, it became a focus of a major scandal when it was alleged that the car’s design allowed its fuel tank to be easily damaged in the event of a rear-end collision which sometimes resulted in deadly fires and explosions. Critics argued that the vehicle’s lack of a true rear bumper as well as any reinforcing structure between the rear panel and the tank meant that in certain collisions, the tank would be thrust forward into the differential, which had a number of protruding bolts that could puncture the tank. This and the fact that the doors could potentially jam during an accident (due to poor reinforcing) made the car a potential deathtrap.

³ The Automotive Industry Action Group (AIAG) and the American Society for Quality Control (ASQC) copyrighted industry-wide FMEA standards in February of 1993, the technical equivalent of the Society of Automotive Engineers procedure SAE J-1739. The standards are presented in an FMEA Manual approved and supported by all three automakers. It provides general guidelines for preparing an FMEA.

⁴ DRBFM moves the user through the FMEA process by considering all intentional and incidental changes and their effects on the performance of a product or process. These changes drive potential causes which require follow-up action to resolve the risk. Design reviews are the primary place to review progress and address these risks.

⁵ Priority can be allocated to those that are more probable (in an FMEA) or those that are more severe (in an FMECA).

- by adding (refer ARP4761 para G.1) basic events in top-down techniques such as FTA, DD or MA
- supporting the estimated probability of a specific basic failure event
- To support the Maintainability Analysis by providing failure detection and/or diagnostic procedures. This is usually conducted under the Integrated Logistic Support (ILS) umbrella and forms a key part of any MSG-3 analysis.
- To support Reliability Analysis by facilitating the determination of predicted MTBF in the absence of proven service data. This activity is conducted to support both the ILS effort as well as the SSA (where the failure probability approximates the inverse of the MTBF (i.e. $P \approx 1/\text{MTBF}$)).
- To assure that a product satisfies ISO 9000 process and customer requirements.⁶
- To determine design vulnerabilities in the manufacturing, maintenance and usage processes of the system or item, as well as to optimise these processes.

It is, therefore, very important to obtain from the customer (at a higher system level, refer Fig 1.3) a written specification defining the intent of the FMEA, the failure effects of interest, outputs to be considered and the required format of the final report.

5.1.4 Scope of the Failure Modes and Effects Analysis

An FMEA usefully considers single failure modes only and should be conducted by the Design Authority of the system level (i.e. System, Subsystem, Module or Part) under consideration.

The FMEA is performed by postulating the ways the chosen level's specific implementation may fail. So, with reference to the example system hierarchy in Fig. 1.1, we can consider that:

- A Level 2 (component level) FMEA is conducted to consider how a component may fail, while the same Level 2 FMECA evaluates what effects this failure may have on its use in

⁶ The manufacturers of consumer products established a different set of priorities, including quality, customer satisfaction and safety. In 1988, the International Organization for Standardization issued the ISO 9000 series of business management standards. The requirements of ISO 9000 pushed organisations to develop formalised Quality Management Systems that ideally are focused on the needs, wants, and expectations of customers. QS 9000 is the automotive analogy to ISO 9000. A Task Force representing Chrysler Corporation, Ford Motor Company and General Motors Corporation developed QS 9000 in an effort to standardise supplier quality systems. In accordance with QS 9000 standards, compliant automotive suppliers shall utilise Advanced Product Quality Planning (APQP), including design and process FMEAs, and develop a Control Plan. Advanced Product Quality Planning standards provide a structured method of defining and establishing the steps necessary to assure that a product satisfies the customer's requirements. Control Plans aid in manufacturing quality products according to customer requirements in conjunction with QS 9000. An emphasis is placed on minimising process and product variation. A Control Plan provides 'a structured approach for the design, selection, and implementation of value-added control methods for the total system'. QS 9000 compliant automotive suppliers must utilise Failure Modes and Effects Analysis (FMEA) in the Advanced Quality Planning process and in the development of their Control Plans (refer <http://www.theleanmachine.com/newsletters/December2003/FMEA.htm>, downloaded on 10/3/09).

a Level 3 and Level 4 system.⁷ This type of FMEA is often referred to as a piece-part⁸ (or structural⁹) FMEA.

- A Level 3 FMECA is conducted to consider how the system failure may affect the platform (Level 4) and its operation application (Level 5). This type of FMEA is often referred to as a functional FMEA and may be very similar to an FHA (see [Chapter 2](#)).

5.1.5 Functional Block Diagrams

Before we go further into the FMEA process, it is useful to review what a Functional Block Diagram (FBD) is, as it forms an essential part of the Functional FMEA process.

An FBD is a diagram that describes a function between input and output variables. A function is described as a set of elementary blocks with input and output variables being connected to blocks by connection lines. An output of a block may also be connected to an input of another block.

Inputs and outputs of the blocks are wired together with connection lines/links. Single lines may be used to connect two logical points of the diagram, e.g.:

- An input variable and an input of a block
- An output of a block and an input of another block
- An output of a block and an output variable

An FBD describes a function between inputs and outputs and should describe the system in one picture; see the example in [Fig. 5.1](#). It should show:

- All major system components
- Interfaces to the outside world
- Interfaces between subsystems
- Power, data and structural interfaces

The FBD therefore provides two important pieces of information:

- what functions are contained within a systems (e.g. an electronic circuit), and
- how the components within that system interface.

Once furnished with a concise system description (including its monitoring and failure detection devices); an FBD; and with a basic understanding of the technology involved (e.g. electronics) you can more easily predict the effect of failures on your system (i.e. during design), or get an idea of where a fault might be located (during maintenance).

⁷ Note: The Criticality is assigned to the severity at Level 3, and this requires detailed knowledge of the Level 3 architecture (e.g. redundancy) and Level 4 application (e.g. is the aircraft cleared for low-level IFR).

⁸ A piece-part FMEA is useful for systems that rely on redundancy (since a functional FMEA may not reveal single component failures affecting more than one redundant element) and is particularly useful for assessing electronic components, mechanical elements and assemblies (refer ARP4176 para G.3.2).

⁹ The structural approach is performed on hardware and focuses on potential hardware failure modes at any system level (i.e. system, subsystem and component or item level). For more on the difference between the structural and the functional approach, see Fig. 13.3 in Ericson (2005).

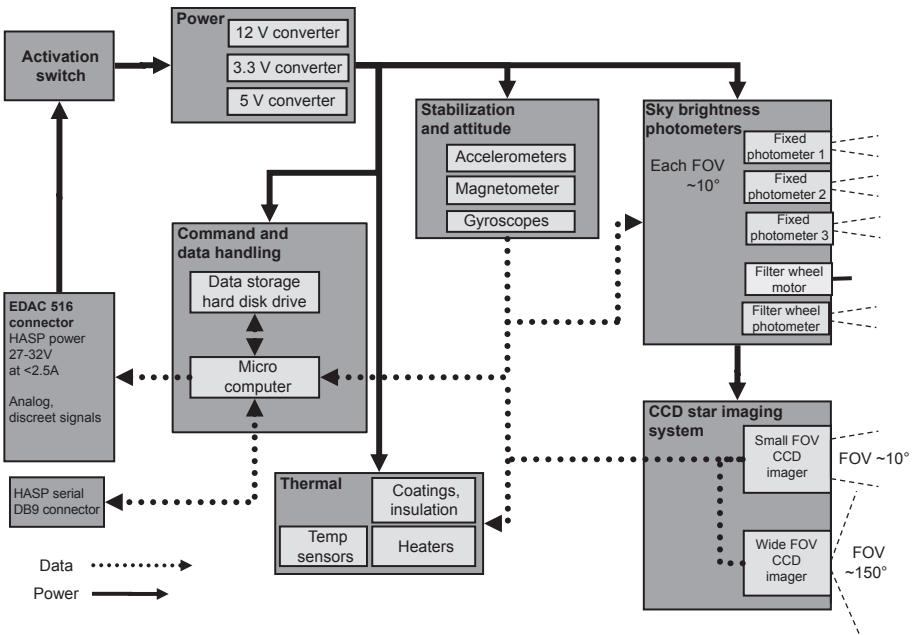


Figure 5.1 Example functional block diagram.

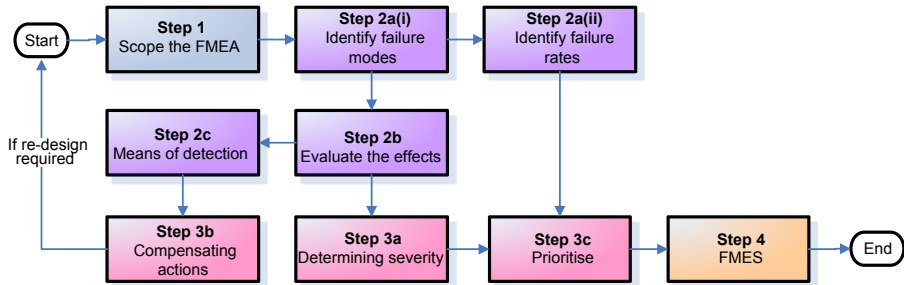


Figure 5.2 FMECA process.¹⁰

¹⁰ In this diagram, Step 2 shows the FMEA specific actions, while Step 3 shows the FMECA specific actions. Strictly speaking the FMEA process should not end here, as it should be referenced in maintenance programs as well as be a benchmark to evaluate reliability performance once the system is in service.

5.2 Conducting a Failure Modes and Effects Analysis

5.2.1 *Modelling the process*

Fig. 5.2 provides a simple illustration of the FMEA process, which essentially consists of four main steps:

- Step 1: Scoping the FMEA
- Step 2: Conducting the FMEA, which consists of four substeps
- Step 3: Completing the FMECA, which consists of three substeps
- Step 4: Summarising in an FMES

The following subsections will explore each one of these four steps in turn.

5.2.2 *Step 1: scoping the Failure Modes and Effects Analysis*

This step is essential in setting the ‘ground rules’ for the FMEA by defining the target system level, boundaries, expectations and requirements before any work on it commences.

Along with describing the system under consideration, the scope should also answer, as a minimum, the following questions:

- What level (see [Section 5.1.4](#)) of analysis will the system start and end at?
- What are the system boundaries? Specifications (and their family tree), drawings, schematics, functional block diagrams, etc., all contain useful data to help define the FMEA boundaries.
- Is this a stand-alone assessment, or does it feed from/into related studies? For instance, the scope of the FMEA might be constrained to:
 - supporting only one baseline failure condition in an FTA (see [Chapter 4](#))
 - a manufacturing process only
 - a maintenance process only
 - an operational usage process only
- Are we interested in all failure modes, or only failure modes of a specific type? For instance, when conducting an SFAR 88 assessment, an FMEA might be scoped to assess only those failure modes that could cause ignition sources (i.e. in this instance, no functional failure modes are under investigation).
- Is this a qualitative and/or quantitative FMEA? See [Section 5.2.3.2](#) for more information.
- What are the operational modes of interest? Failures often have different effects depending on the flight phase in which the failure occurs. See [Section 5.2.3.3](#) for more information.
- And finally, is this a functional or piece-part FMEA? Functional FMEAs are typically performed to support the safety analysis effort, whereas piece-part¹¹ FMEAs are performed as necessary to provide further refinement of the failure rate (ARP4761 para G.3.2).

¹¹ A piece-part FMEA is similar to a functional FMEA, except that the failure modes of each individual component contained in the item or function are analysed instead of the failures of a block of components. A piece-part FMEA is used to determine the failure effects of potential electrical, electronic or mechanical failures. Piece-part FMEAs are typically done when the more conservative failure rates from a functional FMEA will not allow the system or item to meet the FTA probability of failure budget. A piece-part FMEA may also be useful for systems that rely on redundancy, since a functional FMEA may not reveal single component failures affecting more than one redundant element. Piece-part FMEAs are also useful for safety analysis of mechanical items and assemblies.

Table 5.1 Example format for a system level 2 piece-part FMECA

ID	Item Or part (part number)	Failure mode	Mission phase/ operational mode	Failure effect			Failure detection method	Compensating provisions	Failure probability		Severity
				Local (item)	Next (subsys)	End (system)			Probability	Substan- tiation	
1.1	Low Temp Switch (#3255)	Fails Closed (remains on)	Ground and Air	Fan remains on	None	None	Fan powered from XS1 busbar, ie, not using power from the aircraft batteries	None	3.96×10^{-5}		Minor
1.2		Fails Open (remains off)	Ground and Air	Fan remains off	High temp switch will engage.	Flight deck overtempera- ture indication	High tempera- ture switch will cause indication and energise fan	None	3.96×10^{-5}		Minor
2	Fan fuse (F3)	Fuse 'blows'	Ground and Air	Fan remains off	DC PS overheats if DIRCM is operating	DIRCM system failure indication	High tempera- ture switch will cause overtemperature indication	DC PS/Mod has internal overtemperature switch which shuts down system	6×10^{-7}		Minor

Table 5.3 Example format for a system level 2 functional FMEA

Failure Mode Effect Analysis (FMEA)								
System:				Specification			Issue:	
Subsystem:				Drawing:			Prepared by:	
Assembly:				Functional diagram:			Date:	
ID	Component	Function	Failure mode	Failure effect on (1) Component (2) Subsystems (3) System	Failure probability/ failure rate	Reserve factor ^a	Who and how recognised?	Remarks (influencing conditions, ^b failure compensation, ^c etc.)

^aThe Reserve Factor(s) RF, ie, the ratio of strength/ultimate load taken from the stress report, of the component for each failure mode (if only the Margin of Safety, or MS, is known from the stress report: $RF = MS + 1$). Reserve Factor only required for sized items, ie, for items covered by stress reports; to be left blank for nonsized items).

^bAll conditions which can influence a particular type of failure, eg, Service variations (service type and/or service phase), system conditions (presence of other failures), environmental conditions (eg, temperature, vibration, humidity).

^cFailure Compensation indicates the measures through which the failure can be compensated, eg, Design by adequate reserve factor (ratio strength/ultimate load), replacement of units, redundant system.

Table 5.4 Alternative example format for a system level 2 functional FMECA (ABD0100.1.3)

[illegible]

^aA percentile breakdown of the failure rate can be performed (ABD0100.1.3) for equipment as explained below:

- Continuous Test: % of detection of failure immediately detected by equipment BITE (Built-In Test Equipment);
- Cyclic Test: % of detection of 'Continuous Test+failure not immediately detected by equipment BITE but detected during the affected flight';
- Safety Test: % of detection of 'Cyclic Test+failure not detected by the before mentioned tests, but after power on of the aircraft or by the computer itself';
- ATE: % of detection of failure detected during the 'Automatic Test Equipment' (ATE), i.e. during a lab test for PCBs, of the equipment.
- Others: failure not detected during the before mentioned tests, but detected by other means or failure not detected during aircraft life.

Table 5.5 Example format for a system level 3 functional FMECA

[illegible]

Table 5.8 Example format for an FMES

Failure Modes and Effects Summary (FMES)					Number:		Revision:		Date:	
		System:		ATA chapter:		Supplier:			Prepared by:	
		Subsystem:		Part number:		Supplier:			Approved by:	
		LRU:		Part number:		Supplier:			Authorised by:	
ID	Failure mode ^a	Failure rate ^b	Phase	Effects on system	Symptoms (flight or ground crew)	Causal failures	Causal failure reference	Check reference	Failure condition reference	Remarks

^aThe failure effects from the FMEA become the failure modes for the FMES (as identical failure effects from the FMEA are categorised as one mode in the FMES). Or, put another way, all identical failure effects listed in the FMEA(s) are summarised to one failure mode in the FMES.

^bThe failure rate for each failure mode in the FMES is the sum of the failure rates resulting from the failure modes of the individual FMEA(s). Or, put another way, it is the sum of the corresponding individual failure rates in the FMEA for all failure modes resulting in the same effect.

Table 5.9 Alternative example format for an FMES (ABD0100.1.3)

Failure modes and effects analysis (FMEA)									
Aircraft:			Document ref.:			Issue: Page of			
System:			Documents:			Supplier:			
Subsystem:			Drawings:			Prep. by:			
Assembly:			Func. diagrams:			Date:			
Ref.	Failure mode	Failure rate (1E-6 per FH)	Absolute apportionment of failure rate (1E-6 per FH)					Effects on system	Remarks
			Continuous test	Cyclic test	Safety test	ATE	Others		

The scope of an FMEA should therefore be coordinated with the user requesting it (SAE ARP4761 para 4.2) as the FMEA may have to be redone if an expectation is not met.

The assessor can start compiling the FMEA worksheets once the scope of the FMEA is agreed. [Tables 5.1–5.7](#) provide some example FMEA worksheets.

5.2.3 Step 2: Conduct the Failure Modes and Effects Analysis

5.2.3.1 Step 2a(i): identify the failure modes

Postulate every foreseeable failure mode at the level of design being analysed (ARP4761 para G.3.2). The failure mode allocation will depend on whether this is a functional or piece-part FMEA:

- A functional FMEA may be conducted at any system level (e.g. the FHAs in Section 3.3 can also be entitled Functional FMEAs) and is concerned with the function of the system under consideration and how that function might fail. The basic failure categories will be identical to those described in Section 3.2.2.

To facilitate the systematic compilation of the FMEA, it is recommended that each Part/LRU/System should be linked within a Functional Block Diagram (refer [Section 5.1.5](#) as well as ARP4761 para G.3.2.1) with each block having as few outputs as possible. For each functional block, internal and interface functions should be analysed relative to system operation.

Example (ARP4176 para G.3.2.1)

At the component level (i.e. System Level 2), we can consider an electronic component that generate 5V as its primary function. Its failure modes might be:

- loss of 5V;
- voltage <5V;
- voltage >5V;
- noise on 5V, etc.

Example (ABD0100.1.3)

For computer LRUs, in addition to the functions achieved by the hardware functional blocks (such as power supply, inputs, processor, outputs), the main operational functions of the equipment have to be considered such as:

- Engagement logic
- Monitoring of peripheral equipment
- Voltage and frequency regulation inside a general control unit
- Electrical contactor closing/opening control logic
- Computation of an optimal speed
- Equipment self monitoring
- Output failure such as poor transmission characteristics (e.g. signal amplitude, refresh rate, drift, gain error, etc.)

- The piece-part FMEA analyses the failure modes of each individual component (i.e. the smallest individual part) contained within the item under consideration. It therefore starts with a list of all the components to be covered, and a Bill of Materials (BOM) is often a key input. The basic failure categories are (1) Complete failure, (2) Partial Failure and (3) Intermittent Failure.

Note: A piece-part FMEA is often only effectively conducted by the design authority of the part being considered. For the purposes of supporting a 2X.1309 System Safety Assessment, the piece-part FMEA is thus seldom applied above System Level 3 and is only conducted (ARP4761 para G.3.2.2) when necessary (eg, when the more conservative results of a functional FMEA will not meet the FTA probability of failure budget).

Example (ARP4761 para G.3.2.2.1)

Typical failure modes to consider in a piece-part FMEA include, but are not limited to, Open, Short, Parameter shifts, Out of adjustment, Dielectric breakdown, Intermittent operation, Inoperative, Spurious operation, Wear, Mechanical failure, Sticking, Loose, Fracture, etc.

Example (Ericson (2005) para 13.5.4 and 13.5.5)

At the component level (i.e. System Level 2) we can consider the following failure modes which can be used for piece part and or process FMEAs:

- | | |
|----------------------|--------------------------|
| • Open circuit | • Misalignment |
| • Closed circuit | • Binding |
| • Out of tolerance | • Corroded |
| • Leak | • Failure to operate |
| • Hot surface | • Intermittent operation |
| • Bent | • Degraded operation |
| • Oversize/Undersize | |
| • Cracked | |
| • Brittle | • Loss of output |

At software level we can consider the following failure modes:

- | | |
|---------------------------------------|--------------------------------------|
| • Software function fails | • Software stops or crashes |
| • Function provides incorrect results | • Software hangs |
| • Unsent messages | • Software exceeds internal capacity |
| • Message sent too early or too late | • Software start-up failure |
| • Faulty message | • Software response too slow |

5.2.3.2 Step 2a(ii): identify the failure rate of each failure mode

Except for software-induced errors (see [Chapter 9](#)), each failure mode needs to be allocated a failure rate. These may be qualitative and/or quantitative:

- **Qualitative:** Make use of regulatory/customer-accepted failure rate categories (e.g. see Table 3.3), and ensure that the selection is adequately justified and/or defensible. These allocations should be conservative and should be replaced with quantitative analysis if stakeholder consensus on the substantiation cannot be reached.
- **Quantitative:** A failure rate is assigned to each failure mode. Whenever possible, failure rates should (ARP4761 para G.3.2)¹² be determined from historical (i.e. in-service) failure data of similar (but preferably identical) equipment already in field use. These must be justified by including accumulated flight hours, number of occurred failures and justification of similarity. Alternatively, industry sources of failure modes and failure rates include¹³:
 - MIL-HDBK-217, but should include all used parameter and values. Limited to electronic parts only.
 - MIL-HDBK-338.
 - MIL-HDBK-978.
 - GIDEP (Government Industry Data Exchange Program), see <http://www.gidep.org/>.
 - Rome Laboratory's 'Reliability Engineers Toolkit (1993)'.
 - NPRD95 (Nonelectronic Parts Reliability Data). Quote the NPRD component exactly to allow unambiguous identification. And ensure that it is correctly used (e.g. GB values are for 'Ground Benign' and 'M' attached to failure rate figures refer to miles instead of flight hours).

One technique is to perform a failure rate prediction for each block and apportion the failure rate across the various failure modes based on past experience of similar functions, or other sources allowing determination of probability of occurrence.

¹² ARP 4761 para G.3.2.2.1: While the failure rate and mode source documents provide a basis for failure modes of some component types, there will be many device types that are not included in these documents. This is especially true for complex digital ICs which need to be considered on a part by part basis. Determining the failure modes of digital devices generally requires engineering judgement, and it is unlikely that all of the failure modes can be determined for a complex digital IC. A method for estimating the failure modes of complex digital devices is to model the digital devices under consideration with constituent functional blocks for which a better definition of failure modes may exist. Identify the pin level effects of possible failures of the functional blocks as the device failure modes if possible. Some faults may affect multiple pins and various combinations of pins. Particular attention must be paid to potential component failure modes that may lead to the FTA basic events. Trying to determine the actual failure mechanisms and the associated effects through a physics of failure approach is not recommended for ICs as it forces the analyst to perform an 'FMEA' on each digital IC. This 'FMEA' may be more complex than the higher level FMEA being completed and may not even be possible for complex ICs. In addition, an undisclosed design enhancement by the chip manufacturer could render the entire effort obsolete. Complex IC failure modes can include intermittent faults and various fault combinations possibly affecting multiple pins. Failure modes of other component types are more readily available than for ICs. However, a look at several sources will yield different failure mode distributions for the same component type and sometimes even different failure modes. This points out that even for simple components, it is difficult to determine which potential failure modes are valid and which ones cannot happen.

¹³ For more sources of reliability prediction models, see Table 13.2 in Ericson (2005).

It may also be necessary (SAE ARP4761 para G3.2.2.1) to further break down the failure rates for components to identify percentages of failure rates applicable to specific failure modes (e.g. see [Table 5.4](#)).

The total failure rate for each failure effect category may be detailed in a summary sheet or is summarised in the Failure Modes and Effects Summary (FMES, see [Section 5.2.5](#)).

Note that all failure rates must be traceable to referenced source data. The rationale for each failure rate assignment, as well as any assumptions, must be documented for future maintenance of the FMEA and to assist in resolving any future questions.¹⁴

5.2.3.3 Step 2b: evaluate the effects

Evaluate the effect of the failure mode on the system level under consideration. This process is facilitated by the definition of the function of the component.

The failure effect is usually extended to consider the effect on the next one or two level(s) above it, but note that a component designer (at System Level 2) might not be able to postulate the effect at System Level 4 (e.g. the Level 4 designer could have incorporated a dual redundant architecture which the Level 2 provider might not be aware of).

Use of Reliability Block Diagrams may assist in determining the severity of each failure. If the analytical method of determining failure effects for a failure mode is difficult, laboratory verification should be performed where possible ([SAE ARP4761](#) para G.3.2.2). It is desirable for all significant failure effects to be verified by test, for example:

- For electrical or electronic systems, faults can be inserted by opening leads or shorting leads together or to ground.
- If device outputs can be tri-stated, logic combinations can be easily inserted.

Unfortunately, the most difficult failure modes to analyse are sometimes also difficult to confirm through testing. For example, it is impossible to insert all faults for most Integrated Circuits. Software packages may also be used to simulate failures, which allow the equivalent of the fault to be inserted into the circuit simulation and the failure effect determined.

Analysis of failures during testing and in actual use can also be used to substantiate the results of the FMEA. This failure data can also be used to create a library of failure modes for future FMEAs.

¹⁴ These rationales and assumptions are often not included in the FMEA report, but must be available for audit (by the Independent Safety Auditor) and must be kept for future reference. The customer needs to be aware of these data and may consider ensuring that its inclusion is captured in the contractual deliverable. For relative frequency of failure modes of electronic components, it is also possible to refer to Alessandro Brilini's *Reliability Engineering: Theory and Practice*, 1997. If, however, failure rates cannot be apportioned in a justified way (i.e. from in-service experience of published data), then it should be justified by qualitative argument (see [Table 3.3](#)).

It is often useful to describe (or allocate in a separate column) the flight/mission phase that the aircraft is in when the failure occurs. For example, a failure of the Inertial Navigation System (INS) during cruise at flight level (FL) 300 will have a vastly different effect compared with the same failure occurring during final approach under IFR conditions:

- FMEA: As the probability of the failure condition is independent of the flight phase, it is usefully sufficient to select only one (the worst case) phase only.
- FMECA: The severity (see [Section 5.2.4.1](#)) of the failure is very dependent on the effects in a specific phase. More than one phase might thus be allocated to each failure mode.

The worst-case effects must be assumed in cases where it may not be possible to determine conclusively the specific nature of the failure mode (refer ARP4176 para G.2). If the worst-case effect is not acceptable, then the assessor may elect to try one of the following:

- Engage with the next higher level design authority to determine/recommend improved redundancy or system monitoring.
- If it is a functional FMEA (conducted at Level 3 or 4), then drop down to a lower level (e.g. piece-part) and exclude components with no effect on the event under consideration
- If it is a piece-part FMEA, then drop down a level to consider specific failure mechanisms within that part.

For the purposes of the FMES (see [Section 5.2.5](#)), it is useful to assign failure categories to failures which have the same effect (either by allocation to a preallocated category, or by standardising terminology). Each effect category should only have one higher level effect (unless the effect categories are explained in more detail.¹⁵)

5.2.3.4 Step 2c: *identify means of detection*

Document the means by which the failure can be detected. This step demands detailed knowledge of the system, its integration into a higher level system and its operational application. For instance, consider:

- any hardware monitors (such as an overheat switch), but be aware of any potential passive failure modes in the monitor;
- any software monitors (such as PBIT and CBIT);
- failure indication to flight crew and any recommendations on how crew should compensate (see [Section 5.2.4.2](#));
- if the failure is hidden to the flight crew, then three options will must be considered:
 - redesign the system architecture to avoid this passive failure condition;
 - specify flight crew checks (e.g. preflight checklist);
 - specify Certification Maintenance Requirements, some of which (depending on failure severity) may become Critical Certification Maintenance Requirements (CSMR). See Section 11.2.3 for more information.

¹⁵ For example, if a failure mode is found to cause two higher level effects (e.g. ‘Loss of signal A’ and ‘Loss of signal B’), then it might be combined (ARP4761 para G.3.2) to form a new effect category (i.e. ‘Loss of signal A&B’).

During system integration, one must not neglect to verify that monitoring can indeed detect the failure mode. This may be accomplished by:

- Detection Coverage Analysis, which can lead to each individual failure that would have been one effect category now being a separate effect category due to the detection coverage possibilities (SAE ARP4761 para G.5). Another way to include detection coverage is for the FTA to conservatively assume that no holes in coverage due to latent failure in the detection method affect detection of all failures assigned to the failure effect category of concern. The FMEA can be revised if necessary for those cases where this conservative assumption does not allow the top event probability requirements to be met.
- Conducting ground and flight trials where the failure modes are intentionally induced. The result of such trials will also provide useful results to prove compliance to standards such as CS25.1309(c) and CS25.1302 (see [Chapter 11](#) for more information).

5.2.4 Step 3: complete the Failure Modes Effects and Criticality Analysis

5.2.4.1 Step 3a: severity analysis

Consider the severity of the worst-case end effect. This is accomplished by applying the agreed failure severity criteria (see Table 3.2) to the description of the effect of the failure.

See Section 3.2.3.1 for more information.

5.2.4.2 Step 3b: compensating actions

Recommend compensating actions (i.e. automatic or manual) which need to be undertaken upon failure detection.

Take credit (i.e. reconsider the failure mode or the probability or the severity) for diverse or redundant systems, training or competence, operational history or history of similar systems, use of established technology, reuse of software, test plans and results, coverage of tests, safety management practices, etc.

5.2.4.3 Step 3c: prioritise

This optional step may be useful to prioritise the failure modes by conducting a simple risk analysis (i.e. the product of severity and probability).

However, note that this will be the probability of the failure mode, not the probability of the end effect, as the FMEA considers single failure modes only and does not combine these in an accident sequence. Care should therefore be taken when comparing the FMEA risks (if these are used at all) with those in the Hazard Log.¹⁶

¹⁶ See Kritzinger, *Aircraft System Safety: Military and Civil Aeronautical Applications*, Chapter 3 for more information on the Hazard Log and accident sequences.

5.2.5 Step 4: Failure Modes and Effects Summary

The results of an FMEA may be used to generate the Failure Modes and Effects Summary (FMES). This is especially useful for very large FMEAs or to consolidate individual piece-part FMEAs.

The FMES need not necessarily be a separate analysis, as it simply groups single failure modes which produce the same failure effect by:

- Analysing each identified failure mode to determine its effect on the given level and usually on higher levels as well (if not already completed in the FMEA). SAE ARP4761 (para H.3.2) advises that the analyst should review the existing FMEA(s) and check all failure effects for consistency (i.e. is the same failure effect always described with the same wording and does different wording for the failure effect always mean a different failure). This check should be done with special care when an FMES on the system level is performed (i.e. summarising effects from installation failure modes and item failure modes).
- Creating failure effect categories¹⁷ for each different type of highest system level effect. (i.e. each unique failure effect has a separate grouping of single failure modes). Because the FMEA considers single failure modes only, the various failure modes are assumed to be independent.
- Summarising the results (SAE ARP4761 para G.3.2 and ABD0100.1.3):
 - The failure effects from the FMEA then become the failure modes for the FMES (as identical failure effects from the FMEA are categorised as one mode in the FMES).
 - The failure rate for each failure mode in the FMES is the sum of the failure rates resulting from the failure modes of the individual FMEA(s).
 - The references to the individual failure mode in the FMEA may be identified in the FMES 'CAUSAL FAILURE' columns.

An FMES can be compiled from a Level 2 (equipment supplier's), Level 3 (system integrator's) or Level 5 (the aircraft manufacturer's) FMEA. It can also be completed as part (i.e. a summary) of the FMEA.

The FMES reduces the data into a useful format to provide key inputs¹⁸ to higher level FMEAs and/or FTAs and should be coordinated with the user requesting it.¹⁹ See [Table 5.8](#) for an example FMES layout.

5.3 The Case Study

In Section 1.3, we defined a case study for an upgraded Attitude and Altitude Display System. One branch of the Safety Strategy of Fig. 2.4 is repeated in [Fig. 5.3](#).

¹⁷ A code may be assigned to each effect category, which simplifies the FMEA worksheet by moving the description of each effect from the worksheet to the body of the report (SAE ARP 4761 para G.3.2).

¹⁸ The FMES is an aid to simplify the FTAs (reduce the number of OR gates at the lowest level) and to combine the effects of item failures and failures of the installation that have the same effect as one single event. For calculation of failure rates, it should be remembered that an FMEA considers single failures, whereas an FTA considers both single failures and combinations of failures (SAE ARP 4761 para G.3.2).

¹⁹ Note that the FMES form may be altered to add or delete specific data entries as necessary to support the specific FMES customer requirement and the specific FMEA format being used.

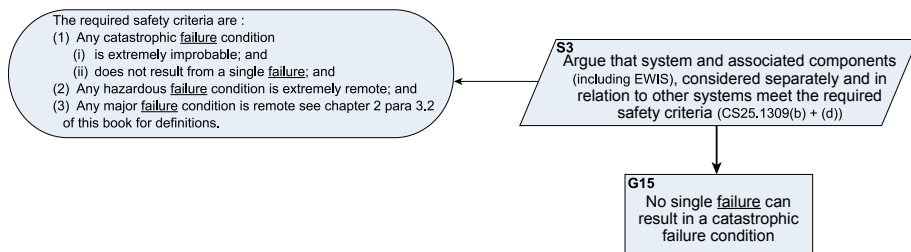


Figure 5.3 Strategy leading to FMEA requirement.

It is suggested (see [Section 5.1.1](#)) that the FMEA is the ideal tool to provide the evidence for accomplishment of Goal 15 (i.e. the FMEA report is the solution).

The following sections apply the theory of [Section 5.2](#) to this case study.

5.3.1 Step 1: scope the Failure Modes and Effects Analysis

As this case study is required to support a CS25.1309 safety assessment, it is suggested that a Functional FMEA at the system integration level (i.e. Level 3 in Fig. 1.1) would be more appropriate than a piece-part FMEA at the component level.²⁰ This Functional FMEA will have the objective of searching for single failure conditions in the Barometric Altitude Display System which might cause a catastrophic failure at the aircraft level (i.e. Level 4 in Fig. 1.1).

The Functional Block Diagram for the Level 3 system is extracted from [Chapter 1](#) and repeated in [Fig. 5.4](#) for ease of reference. The FMEA will evaluate the impact of functional failure of each of these LRUs, see [Table 5.10](#) column 2.

5.3.2 Step 2a: identify the Failure Modes and Failure Rates

In this step we use the FBD to capture each function in turn ([Table 5.10](#) columns 1 and 2) to identify the functional failure modes ([Table 5.10](#) column 3).

The functional failure modes rates (i.e. the probabilities of the failures occurring) are captured in [Table 5.10](#) column 4, while column 5 can be used to justify or reference its substantiation. Note that we are declaring the probability of the failure mode, not the probability of the end effect.

5.3.3 Step 2b: evaluate the effects

In [Table 5.10](#) we identify the effect of the failure on the component (in column 7), then the effect on the Barometric Altitude Display System (column 8), and finally the effect at the aircraft level (column 9). Note in column 9 that we have elected to use the severity descriptor from [Table 3.3](#) as it facilitates severity allocation in column 10.

²⁰ The scope of this FMEA will thus not extend to include the depth of analysis required for a Reliability Analysis.

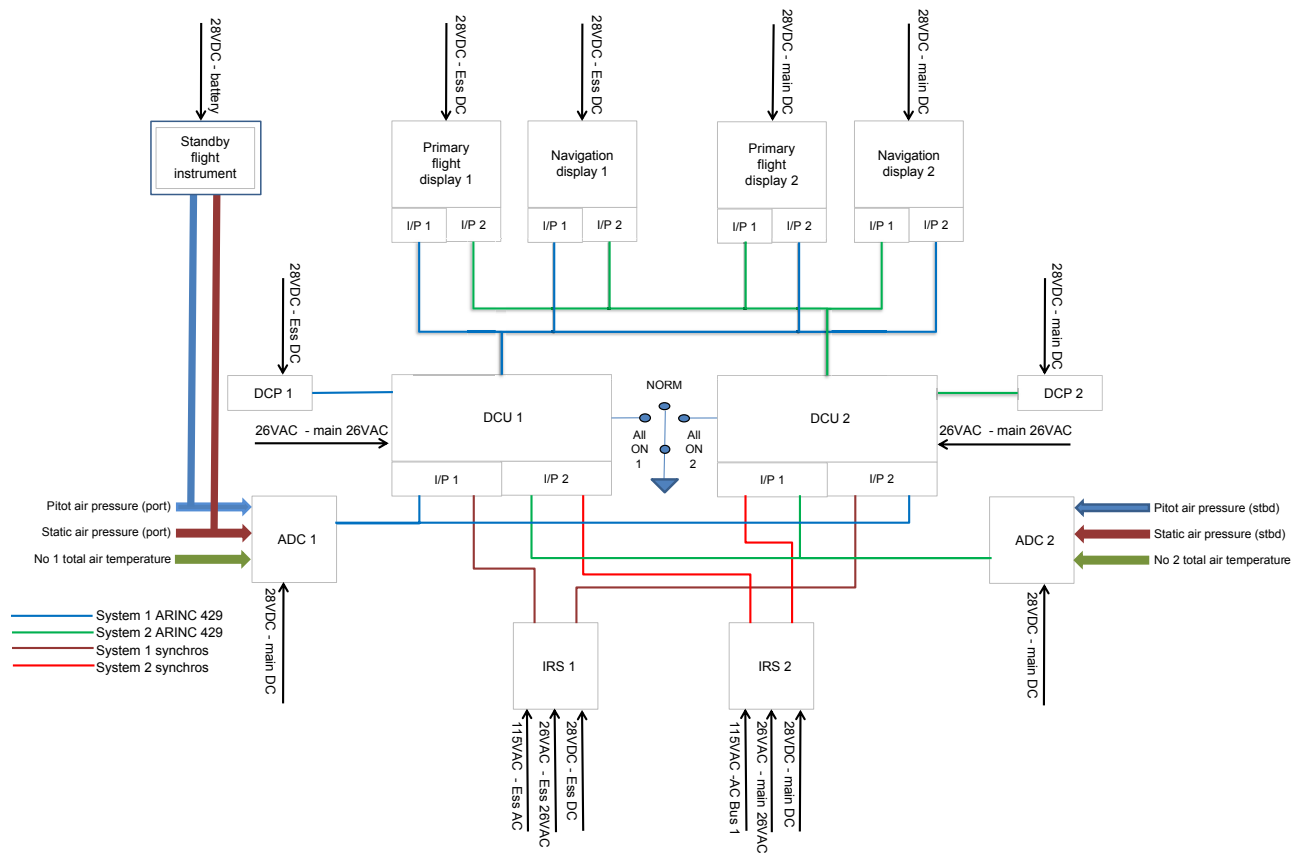


Figure 5.4 Functional block diagram: Attitude and Altitude system.

5.3.4 Step 2c: identify means of detection

In [Table 5.10](#) we identify the effect of the means by which the failure can be detected in column 11. Although we have populated the detection method for many of the failure modes in [Table 5.10](#), it could be argued (by referencing the scope/objective of the FMEA) that we only need to complete this column in the catastrophic failure condition. In practice this is seldom an accepted argument, but the assessor (and the customer/regulator) must remain mindful of the objective/scope of any given FMEA.

5.3.5 Step 3a: severity analysis

If we are conducting an FMECA, then column 10 in [Table 5.10](#) has been reserved for this purpose (where it should be evident that the severity depends on the worst-case aircraft-level effect defined in Column 7). In some cases the Level 4 system integrator might feel that the Level 3 designer does not have the required system knowledge to justify this effect, in which case their participation needs to be solicited to ensure the objective of the FMEA is satisfied.

5.3.6 Step 3b: compensating actions

In [Table 5.10](#), column 12 may be used to proactively prescribe compensating provisions (e.g. need check intervals or prescribed design features) or to reference existing provisions (e.g. existing design redundancy or existing check intervals) ([Table 5.9](#)).

5.3.7 Step 3c: prioritise

Now that the FMEA is complete, we can prioritise effort by sorting against:

- Failure Severity (column 10), in which case it is evident that row 8 is an item of concern;
- Failure Probability (column 4), as this drives the reliability the system;
- Risk of failure, in which case we need to add an extra column where we consider the product (i.e. combination) of failure probability and failure severity.

Note, the Level 3 designer we can also repeat this step for the FMES in [Section 5.3.8](#) below, although this is useably done by the Level 4 system integrator to focus their effort.

5.3.8 Step 4: Failure Modes and Effects Summary

The Level 3 system designer now needs to decide of an FMES is justified or not. For smaller FMEAs, this might not be require but, to keep the whole integration process manageable, it is essential to provide a summary of a very lengthy FMEA to the next level of system integration.

Within the scope/objective of this FMEA case study (refer [Section 5.3.1](#)), it can be argued that no FMES is needed because of the ease of extracting the few catastrophic failure modes which exist. If an FMES was required, then for

Table 5.10 Functional FMEA for the Barometric Altitude Display System^a

Objective: Find single source catastrophic failures in the barometric altitude display system											
		Functional failure mode			Failure effect						
ID (1)	Item/part (2)	Failure (3)	Probability (4)	Substantiation (5)	Phase of flight (6)	Local (item) (7)	Next (system) (8)	End (aircraft) (9)	Severity (10)	Failure detection method (11)	Compensating provisions (12)
1	28VDC – Ess DC Bus	Loss of power	1×10^{-6} /flthr	A ‘recognised assumption’, refer Lloyd and Tye (p76) (although best substantiated from the specific aircraft type’s FRACAS)	IFR conditions	Loss of ADC1, DCP1, PFD1 and ND1	DCU2 on PFD2 and ND2 becomes primary Alt reference	Slight reduction in functional capabilities or safety margins	Minor	DCU1 or DCU2 will output error flag to primary displays	Barometric altitude available off PFD2, ND2 and Stdby FI
2	28VDC – Main DC Bus	Loss of power	1×10^{-6} /flthr	A ‘recognised assumption’, refer Lloyd and Tye (p76)	IFR conditions				Minor	DCU1 or DCU2 will output error flag to primary displays	Barometric altitude available off PFD1, ND1 and Stdby FI
3	115 V AC – Ess AC Bus	Loss of power	1×10^{-6} /flthr	A ‘recognised assumption’, refer Lloyd and Tye (p76)	IFR conditions				Minor	DCU1 or DCU2 will output error flag to primary displays	Barometric altitude available on all primary displays and Stdby FI.

4	115 V AC – main AC Bus 1	Loss of power	1×10^{-6} /ft hr	A ‘recognised assumption’, refer Lloyd and Tye (p76)	IFR conditions				Minor	DCU1 or DCU2 will output error flag to primary displays	Barometric altitude available on all primary displays and Stdby FI.
5	26V AC – Ess 26V AC Bus	Loss of power	1×10^{-6} /ft hr	A ‘recognised assumption’, refer Lloyd and Tye (p76)	IFR conditions				Minor	DCU1 or DCU2 will output error flag to primary displays	Barometric altitude available on all primary displays and Stdby FI.
6	28V AC – Main 26V AC Bus	Loss of power	1×10^{-6} /ft hr	A ‘recognised assumption’, refer Lloyd and Tye (p76)	IFR conditions				Minor	DCU1 or DCU2 will output error flag to primary displays	Barometric altitude available on all primary displays and Stdby FI.
7	28V – Battery Bus	Loss of power	1×10^{-6} /ft hr	A ‘recognised assumption’, refer Lloyd and Tye (p76)	IFR conditions				Major (refer AC25-11 Table 5)	No power on Stdby FI	Revert to Primary displays

Continued

Table 5.10 Continued

[illegible]

11	No.1 TAT	Misleading TAT									
12	No.2 TAT	Misleading TAT									
13	DCU Reversion Switch	Fails									
14.1	ADC1	Fails									
14.2	ADC1	Error									
15.1	ADC2	Fails									
15.2	ADC2	Error									
16.1	DCU1	Fails									
16.2	DCUI	Error									
17.1	DCU1	Fails									
17.2	DCU2	Error									
18.1	DCP1	Fails									
18.2	DCP1	Error									
19.1	DCP2										
19.2	DCP2										
20.1	Stdbby FI										
20.2	Stdbby FI										

Continued

Table 5.10 Continued

Objective: Find single source catastrophic failures in the barometric altitude display system											
		Functional failure mode			Failure effect						
ID (1)	Item/part (2)	Failure (3)	Probability (4)	Substantiation (5)	Phase of flight (6)	Local (item) (7)	Next (system) (8)	End (aircraft) (9)	Severity (10)	Failure detection method (11)	Compensating provisions (12)
21.1	PFD1 or PFD2										
21.2	PFD1 or PFD 2										
22.1	ND1 or ND2										
22.2	ND1 or ND2										
23.1	IRU1 or IRU2										
23.2	IRU1 or IRU2										

^aThe first five failure conditions are completed below. To facilitate learning, the reader is invited to complete the rest.

each ‘effect’ (depending on contractual commitments, this could be either system level or aircraft level), suitable column headings with support from [Table 5.10](#) might include:

1. All the FMEA ‘failure modes’ contributing to that ‘effect’
2. The failure rate of each of these failure modes
3. The number of these components (causing this failure mode) in the system under consideration
4. The total failure rate of each component (i.e. the product of (b) and (c) above)
5. The total failure rate leading to the effect (i.e. the sum of all component in (d) above).

Special care should be taken of the failure effect wording in the FMEA table to allow an unambiguous processing of the FMES. This is especially important for separate FMEAs (different subsystems) or when using different specialists to assess their systems: try to use identical wording and a common lexicon whenever possible (eg, same effects should use identical narrative).

5.4 Discussion

Many crashes follow the pattern of an old proverb.

For Want of a Nail.

*For want of a nail the shoe was lost.
For want of a shoe the horse was lost.
For want of a horse the rider was lost.
For want of a rider the message was lost.
For want of a message the battle was lost.
For want of a battle the kingdom was lost.
And all for the want of a horseshoe nail.*

During the earliest conceptual stages of design, the FMEA attempts to preempt this type of problem by identifying single failure conditions and ensuring that these failures (if they do occur) do not escalate.

Within the safety context, an FME(C)A is generated to support the Safety Assessment, so it is important to understand the expectations and requirements on the FMEA before any work on it commences. It is therefore important to coordinate the required scope of FMEA with the user requesting it. For instance,

- the sole purpose for the FMEA may be to support the basic events of a specific Fault Tree;
- a Piece-part FMEA may not be necessary if the failure rates from a Functional FMEA allow the PSSA targets to be met.

As the analysis progresses, the following should be informally recorded²¹ for future maintenance of the FMEAs and to assist in resolving questions regarding the FMEA:

- Justification of each failure mode
- Rationale assigning a particular failure to a failure effect category
- Documentation of any assumptions made

Many of these are made when conducting an FMEA – especially early in the design phase. These must be carefully documented and maintained for traceability and to simplify future updates resulting from design changes and/or changes in operational application.

5.5 Conclusions

The FMEA is a useful tool with an excellent pedigree; however, it should never be blindly conducted without a thorough understanding of its advantages and limitations.

5.5.1 Advantages

The FMEA is simple and flexible tool which:

- is very systematic at lower levels (i.e. forces the assessor to consider all components in the system);
- can identify many possible causes of any failure mode and therefore is useful to support fail-safe strategies (such as BIT, failure indications and redundancy);
- ensures consideration of all possible types of failure and their effects on safety, operation, reliability and maintainability;
- can be arranged according to the importance of their effects or their probabilities (in certain cases) on different levels (e.g. system, subsystem, component);
- provides useful information for reliability programs, is good at generating maintainability data and is useful for the preparation of diagnostic routines (e.g. flowcharts or fault finding tables) by conveniently listing all the failure modes;
- can identify critical single failure events as well as any latent failures;
- provides source data for the FTA/DD/MA and complements these tools when an item has particularly significant potential consequences;
- can also be applied to evaluate operations and/or facilities.

If done early enough, the most important benefits of performing an FMEA are that it enables the system integrator to:

- choose designs of the required reliability and safety;
- obtain input for the definition of design requirements, especially with regard to fail-safe design, redundancy and failure indication;
- avoid costly modifications by the early identification of design deficiencies;
- show, at an early stage, which criteria must be given special consideration in planning and executing experiments and in developing function tests;
- establish the basis for determining the priorities of corrective actions.

²¹ This documentation is usually not included in the FMEA report but is retained for reference.

5.5.2 Limitations

Although an FMEA is very systematic and extensive, the ‘devil is in the details’.

- It lists only single failures and assumes rest of system is working perfectly (not the way accidents happen).
- It is primarily a reliability technique. Many failure modes listed will have no safety concern, and much effort may be spent in obtaining or justifying the failure rates (step 2a(ii)) of these. So, be aware of confusing reliability objectives with safety objectives.
- Often too much reliance is placed on the FMEA/FMECA, while ignoring threats that can arise from outside the system (e.g. common cause failures, human error, multiple failures, etc.).
- Cannot cope with human induced hazards/errors.
- Can be very detailed, and critical aspects may be lost in the detail. Level of analysis must be decided (piece-part/LRU/Subsystem/system) and not mixed in the same document.
- For the FMECA, severity can only be allocated if it is taken through to a higher system level (e.g. adding a safety severity to a resistor failure is meaningless), and the designer might not have access to the data to make this judgement.
- It is time-consuming and expensive to generate (often iterative). Needs continuous management to keep it current.
- System architecture needs to be defined to consider effect on the higher level assemblies.
- A piece-part FMEA is not practically feasible for modern microcircuit-based LRU and systems. Determining all the failure modes of any but the simplest components (where industry data is available) is extremely difficult and sometimes impossible (especially for integrated electronic circuits).

References

- APD0100.1.3, 2007. Airbus, Safety & Reliability Requirements, Equipment-design-general Requirements for Suppliers.
- Ericson, C.A., 2005. Hazard Analyses Techniques for System Safety. Wiley-Interscience, New Jersey.
- MIL-HDBK-217F, Reliability Prediction of Electronic Components, US Department of Defense, Rome Laboratory/ERSR, 425 Brooks Rd, Griffiss AFB, NY 13441-4505.
- MIL-HDBK-338B, 1998. Military Handbook: Electronic Reliability Design Handbook. Department of Defense, USA.
- MIL-HDBK-978B, March 1998. Military Handbook: NASA Parts Application Handbook - Connectors, Protective Devices, Switches, Relays, Wire, Cable. Department of Defense, USA.
- MIL-P-1629A, 1949. Military Standard, Procedures for Performing a Failure Mode, Effects and Criticality Analysis. US Department of Defence.
- NPRD95, Non-electronic Parts Reliability Database, Reliability Analysis Centre, 201 Mill Street, Rome, New York, 13440, USA.
- Reliability Engineer's Toolkit, Systems Reliability Division, 1993. Rome Laboratory, Air Force Materiel Command (AFMC), Griffiss AFB, NY 13441-4505.
- SAE ARP4761, 1996. Guidelines and Methods for Conducting the Safety Assessment Process on Civil Airborne Systems and Equipment.

Further reading

- Borgovini, Pemberton, R., Ross, S., 1993. Failure Mode, Effects and Criticality Analysis (FMECA). Reliability Analysis Center.
- BS5760, 1991. Reliability of Systems, Equipment and Components Part 5: Guide to Failure Modes, Effects and Criticality Analysis (FMEA and FMECA). British Standards Institute.
- Dill, R., et al., 1963. State of the Art Reliability Estimate of Saturn V Propulsion Systems. General Electric Company.
- ECSS-Q-HB-30-08A, January 2011. Space Product Assurance. Components Reliability Data Sources and Their Use. European Cooperations for Space Standardization (ECSS), ECSS Secretariat, ESA-ESTEC Requirements & Standards Division, Noordwijk, The Netherlands.
- Neal, R.A., 1962. Modes of Failure Analysis Summary for the Nerva B-2 Reactor. Westinghouse Electric Corporation Astronuclear Laboratory.
- SAE ARP926, 1967. Design Analysis Procedure for Failure Modes, Effects and Criticality Analysis (FMECA). Society for Automotive Engineers.

Common Mode Analysis

6

A common mistake that people make when trying to design something completely foolproof is to underestimate the ingenuity of complete fools.

Douglas Adams (1952–2001)

6.1 Introduction

6.1.1 Background

The acceptance of a qualitative or quantitative failure probability declaration is often based on the assumption that failures are independent (AMC25.1309). Independency is often accomplished in duplication of systems/components. Redundancy, and the independence thereof, is a key feature in fail safe designs of system requiring high level of functional integrity.

There are, however, many and various threats to the independence of the channels of redundant systems. Many of which, should they become manifest, may lead to failures at higher rates than would be forecast by derivation from the failure rates of the component channels alone (Lloyd and Tye, 1982, p. 73). Although most critical systems employ redundancy techniques, it will be found on examination that many of them have a ‘single cause’ (e.g. EMI/EMC), or ‘common point’ (e.g. common bus-bar or common controller) that could cause multiple failures. In many cases this single element is readily obvious, but this is not always the case.

Example common mode failure

- *Common part failure:* Three totally independent flying control systems may merge together in a common part – the pilots control column. A failure of this common part causes total system failure.
- *Common cause failure:* Afire in a compartment might destroy all the channels of a system running through that compartment. Likewise, contaminated hydraulic fluid could cause all the channels of the hydraulic system to fail, or mechanical failures in an electrical loom.
- *Common mode failure:* Identical software in a dual redundant system will fail when exposed to the same inputs; jamming of a mechanical system (either due to failure or due to FOD); overheating of avionic equipment; etc.
- *Cascade failures:* A single electrical bus failure may overload the remaining channels, thereby increasing the probability of their failure. In a two-channel system, each channel with a failure rate of 1 in 1000h, the probability of any one of the

Example common mode failure—cont'd

channels failing is $2 \times 1/1000$. So in a period of a million hours, there will be 2000 failures. The probability of two channels failing is $(1/1000)^2$, i.e. one double failure in a million hours. However, if the failure of the first channel will cause a 10-fold increase in the probability of the second channel also failing, then the probability of total failure is then $(1/1000) \times (1/100)$, i.e. 10 such double failures in a million hours. It is thus evident that the combined failure rates increase proportionally with increase of risk under the added load, and hence, it is important to take this into account and preferably design channels to cope with the added load without materially worsening the failure rate (Lloyd and Tye, 1982, p. 75). See MIL-HDBK-217B, 'Reliability Prediction of Electronic Equipment' where attempts have been made to quantify the relationship between load and failure rate of some electrical components.

It is necessary to recognise that such independence may not actually exist in the practical sense, and specific studies are often necessary to establish whether such independence (AMC25.1309 App 1 para (f) in CS25) can be either confirmed or, justified as sufficient (i.e. by incorporation into the System Level 3 or 4 (see Fig. 1.1) probability declaration).

The advisory material to CS25.1309, FAR25.1309 and SAE ARP4761 group these 'specific studies' under the term common cause analysis, which is further subdivided into three approaches:

- Common Mode Analysis (CMA), which is the subject of this chapter;
- Particular Risk Analysis (PRA), which is discussed in Chapter 7;
- Zonal Safety Analysis (ZSA), which is discussed in Chapter 8.

The CMA purposefully looks for common cause failure in system architecture (e.g. threats to full redundancy) *as well as* considering how that architecture is to be installed, operated and maintained. Historically, a system design was often the product of engineers who cared passionately that their creations function well, but did not always give sufficient thought to those who would use it in operation or those who would maintain it. People (and the jobs they do) play an important safety role, nowhere perhaps, is this more sharply brought into relief than in the study of aviation disasters, where accident investigators often conclude that human error during maintenance and/or operation acted as a major contributory factor (Edwards, 1988).

Unfortunately many products and systems are still designed without adequate consideration of the human factor (HF). Designers still tend to focus primarily on the technology and its features without fully considering the use of the product from the human perspective. Clearly it is neither practical nor cost-effective to develop

and/or test all possible combinations of conditions that may affect human performance¹; however, a systematic and informed consideration of the human as part the safety assessment process can provide significant additional opportunities for risk reduction.

6.1.2 Aim of the Common Mode Analysis

The aim of the CMA is to identify failures which bypass or invalidate redundancy/independency assertions (i.e. assumed independence) in a Safety Assessment so that appropriate steps can be taken to preclude them.

6.1.3 Objectives of the Common Mode Analysis

The objective of the CMA is to identify all possible common mode failures which have the potential to fail, or degrade, system redundancy or to cause another initiating event leading to system failure.

6.1.4 Scope of Common Mode Analysis

The CMA is scoped to focus on:

- The independence of functions and their respective monitors. In particular, items with identical hardware and/or software that could be susceptible to generic faults which could cause simultaneous malfunctions in multiple components (e.g. similar development errors, duplex systems using a common power supply etc.).
- The system's vulnerability to errors in manufacture, maintenance and operation. These cause a substantial proportion of accidents. In order to reduce the likelihood of such accidents, it is necessary to make an analysis of the design to determine those features of it which are vulnerable to error and to try to eliminate these, or to ensure that instructions, checks, training, etc., can be relied upon to safeguard against the predictable errors (Lloyd and Tye, 1982, p. 111).

The CMA may be scoped to be conducted at System Levels 3, 4 or 5 (refer Fig. 1.1), and effort should be focussed (refer Fig. 3.3) on those failures which have either Catastrophic or Hazardous effects (refer Tables 1.1 and 3.2).

¹ The term 'Human Factors' (HF) has grown increasingly popular as the aviation industry has realised that human error, rather than mechanical or electrical failure, underlies a significant number of aviation accidents and incidents. If interpreted narrowly, HF is often considered synonymous with Crew Resource Management (CRM) or Maintenance Resource Management (MRM). However, it is much broader in both its knowledge base and scope. HF involves gathering information about human abilities, limitations and other characteristics and applying it to tools, machines, systems, tasks, jobs and environments to produce safe, comfortable and effective human use. In aviation, HF knowledge is dedicated to better understanding how humans can most safely and efficiently be integrated with the technology. That understanding is then translated into design, training, policies or procedures to help humans perform better (Graeber, 2010).

6.2 Conducting a Common Mode Analysis

A typical CMA process is illustrated in Fig. 6.1, and each step is explained in Sections 6.2.1–6.2.5.

6.2.1 Step 1: identify common mode vulnerabilities

There are two distinct set of inputs used to identify the common mode vulnerabilities:

- Step 1a: This CMA process is performed to verify that ‘AND’ events (e.g. in the Fault Tree Analysis (FTA) or Dependence Diagrams (DD), or any qualitative probability declaration) are truly independent.

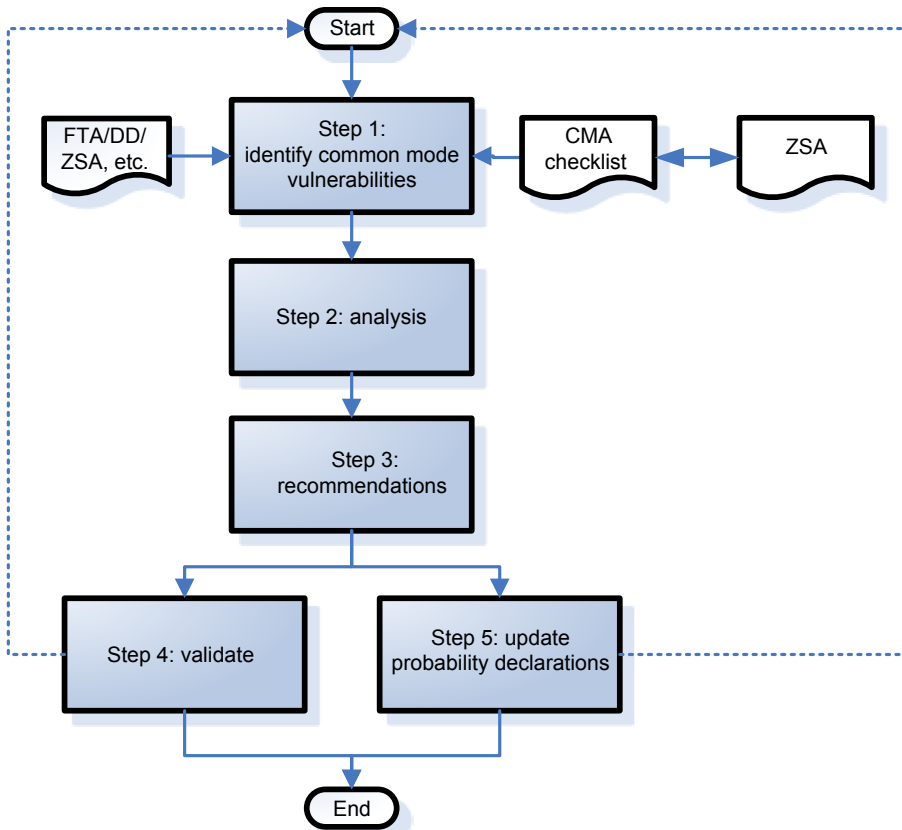


Figure 6.1 Typical CMA process.²

² SAE ARP4761 Appendix K advises that the main input into the CMA is the FHA (which highlights the Catastrophic and Hazardous failure conditions which need to be subject to a CMA). The assumption (refer Fig. 3.2) made in this chapter is that the FHA already scoped all the FTA/MA/DD required. So, the FHA drives the requirement for FTA/DD/MA which in turn needs to be subject to CMA.

In terms of assessment sensitivity, it is the ‘AND’ gates which are the vulnerable parts of the design. The reason for this is that, by definition of an ‘AND’ gate, the event is only going to happen if more than one of the necessary conditions are met. These conditions are normally multifailure events. Since the probability of the resulting top-level failure event is the multiplication of failure probabilities of the individual events, the result is that the top-level failure event is far less probable than the individual causes. However, any common cause failure could dramatically alter the probability of the top event.

Note: The review of all the ‘AND’ gates is best performed by someone other than the originator of the qualitative or quantitative analysis (e.g. an independent Compliance Verification Engineering or an Independent Safety Assessor). It is thus unlikely that this step can be completed before the Preliminary Design Review (PDR) and is likely to be iterative as the system architecture becomes more mature.

- **Step 1b:** There are CMA requirements which may not be readily derived from Step 1a, but are attributable to vulnerabilities and/or systemic errors in the design, build or operation of the system (i.e. systemic errors). The reason such additional assessment is required is due to the ease of designing a system which is complex to use, difficult to maintain and hard to recover in the event of failures/malfunctions. If something can be done incorrectly during the system life cycle, then at some point in time, chances are that it will be done incorrectly. Failure to consider the impact of design on expected human performance may lead to gross misjudgement of both total system safety and operational effectiveness.

Step 1b is therefore based on the application of a ‘lessons learned’ checklist, an example of which is contained in [Table 6.1](#). Its purpose is to identify potential sources of systemic errors³ (which could lead to systematic failures). It considers the possibility of requirement, design and implementation errors (as well as reasonably anticipated crew errors after the failure occurrence of a failure condition). The checklist may be applied to consider system architecture vulnerabilities as well as physical installation vulnerabilities in the ZSA (see [Chapter 8](#)).

It should be readily obvious from the column entitled ‘Source of Common Mode Errors’ that this checklist is best reviewed by a multidisciplinary working group tasked to brainstorm any possible condition which could contribute to a CMA event. In reviewing the ‘Design Architecture’, some effort in the working group may duplicate Step 1a, but this is inevitable and provides additional confidence in the thoroughness of the process.

Note: It is recommended that this step be conducted proactively early in the design process (so that safety features can be designed into the system from the start) and then repeated after major design reviews (such as at PDR and CDR).

³ Errors in this case are taken to include both mistakes and omissions. Vulnerability to systemic errors can be introduced during any part of the life cycle, and its realisation comes into effect during manufacture, installation, maintenance or operation (where the blame is then allocated under the umbrella term of Human Factors). These systematic failures may not all be derivable from assessments such as the FHA or FTA, and many are best identified from service experience.

Table 6.1 Example CMA checklist^{a,b}

Life cycle phase	Common mode type	Potential common mode sources/failures/errors	Comments
Design	External interfaces	Electrical power supply Ventilation supply Discharge/Exhaust	
	Systemic	Software errors	<p>Multiversion programming is employed in fault-tolerant computer systems in order to provide protection against common mode failures in software design.</p> <p>Multiversion programming uses diverse software implementations of critical functions such that an error in one version will not fail in an identical manner as in another diverse version.</p> <p>Two critical issues for determining if multiversion programming can increase reliability:</p> <ul style="list-style-type: none"> • the likelihood of common mode failures in diverse versions and • the reliability of the voter in a multiversion programming system
	Systemic	Hardware environmental deficiencies	Usage outside of operating ranges (e.g. vibration, temperature, altitude, etc).
Manufacture	Quality control	Incorrect process Inadequate manufacturing control Inadequate inspection Inadequate testing	<p>Batch of faulty components can grossly affect failure probability. This type of failure is usually picked up by the operator, but until then the system may run at high risk. Two safety recommendations may include:</p> <ul style="list-style-type: none"> • The manufacturer needs to understand the ramifications of quality deficiencies and put appropriate controls in place. • The operator needs to set 'alert levels' at which corrective action needs to be taken to maintain the required level of safety.

Integration	Installation error	Mal-assembly Mal-rigging Common fitter Installation/maintenance staff competence	Multichannel systems (e.g. wiring into back of Engine instrument Display System) are particularly vulnerable to installation error. CMA teams must consider what could go wrong if incorrectly assembled/connected). Recommendations could include: <ul style="list-style-type: none"> • structural provisioning (e.g. so loom position is obvious); • using different keys in similar electrical connections; • postinstallation check procedures. The result of this assessment must be fed into the Operational Suitability Data, specifically the means of compliance to CS-MCSD (for Maintenance Certifying Staff).
Test	First flight tests for certification	Incorrect assembly Unknown systems characteristics	
	Postmaintenance flight tests	Incorrect maintenance Maintenance staff competence (refer CS – MCDS)	As for Integration, consider which tests are required after maintenance, and how they may go wrong. The result of this assessment must be fed into the Operational Suitability Data, specifically the means of compliance to CS-FCD (for Flight Crew Data, and CS-SIMD (for simulator devices).
Operate	Common crew	Inadequately trained personnel Overstressed operator (omission of action, incorrect or inadequate commission or action) Faulty operating instructions Misdiagnoses (following errors/failures) Incorrect anthropometric considerations Competence	Working group should consider all potential failures (i.e. if it can go wrong, then it will go wrong one day) and the consequences. Update flight reference cards and ensure that crew are trained to diagnose and rectify foreseeable errors. The result of this assessment must be fed into the Operational Suitability Data, specifically the means of compliance to CS-FCD, CS-CCD (for cabin crew) and CS-SIMD. Example: Frozen pitot-static ports robbing the aircraft of airspeed and altitude information was determined by Turkish investigators as a primary cause of THY Turkish Airlines crash (7/4/99). The pilots failed to switch on their pitot/static heating systems as the 747-400 entered icing conditions not long after leaving Jeddah. Highlighted by UKCAA (CAA PAPER 2011/03) as a ‘significant 7’ risk of a loss of control through operation of increasingly complex and highly automated aircraft types.

Table 6.1 Continued

Life cycle phase	Common mode type	Potential common mode sources/failures/errors	Comments
	Aircraft settings	Aircraft weight and balance Airport altitude Trim	The result of this assessment must be entered into the Operational Suitability Data, specifically the means of compliance to CS-FCD and CS-SIMD. Example: The inquiry into the November 2006 event has discovered that the Iranian-operated aircraft had been cleared to 2500 ft (760 m) as it arrived from Tehran. Instead of keeping to this height, the A310 continued to lose altitude even after Birmingham air traffic control contacted the aircraft three times to alert the pilots. The presence of a tall mast in the vicinity then led the controller to order the A310 to perform an immediate climb. The crew had not reset the altimeter from the standard pressure setting of 1013 hPa to the local figure of 982 hPa – a difference of 31 hPa that would have led to a 930 ft error in the altimeter reading. Investigators have found that a Mahan Air crew's incorrect interpretation of navigation data, reinforced by poor cockpit communication, led an Airbus A310-300 to descend to an extremely low altitude while attempting to land at Birmingham, United Kingdom.
	Software updates	Flight line software updates	Flight Plan (OFP) updates. Counter Measure Dispensing Software (CMDS) updates. Flight control software updates. Engine control software updates.
	Mechanical failures	Temperature exceedance Dust and Dirt Vibration Pressure Humidity Stress Fatigue	Mitigations will include: <ul style="list-style-type: none"> • Environmental qualification status of all LRUs (see RTCA/DO-160 and Mil-Std-810); • Limiting life of components, etc.

	<p>Electrical</p> <p>Radiation</p> <p>Contamination</p>	<p>Power surge, voltage Power surge current Short circuit EMC</p> <p>EMI HIRF HERTA</p> <p>Increased friction and binding between sliding surfaces Clogging and blocking of lines, valves, regulators, filters, nozzles, orifices Scoring and abrading of closely fitted moving surfaces Deterioration of fluids Clogging of cooling fans Introduction of FOD</p>	<p>Refer the means of compliance to CS25 Appendix H, as well as AMC20 (Leaflets 21, 22 & 23).</p> <p>Consider the vulnerability of aircraft being exposed to military system at military bases, including the possibility of being painted by targeting radars on ships.</p> <p>Can be caused by condensation, airborne dirt or other environmental particles, leakage or spillage of petroleum products, filtration system overload or failure, corrosion, fire extinguishing fluids/powder, etc.</p> <p>Contamination (e.g. water) can cause short circuits, for this reason pins of electrical components are normally arranged to minimise the effect.</p> <p>Contamination (e.g. water) of fuel has caused engine flameouts (AAR 1/2010).</p> <p>FOD (dust and lint) provide fuel for an EWIS-related fire (CS25 Subpart H).</p>
Maintain ^a	Common staff	<p>Inadequately trained personnel Incorrect anthropometric considerations Overstressed technician (omission of action, incorrect or inadequate commission or action) Incomplete, ambiguous or incorrect maintenance instructions Misdiagnoses of fault</p>	<p>Working group should consider all potential failures (i.e. if it can go wrong, then it will go wrong one day) and the consequences. Mitigations could include</p> <ul style="list-style-type: none"> ensuring that technicians understand the safety significance of the systems they are working on, reducing probability via peer review/inspection/sign-off. <p>Using techniques (e.g. routing clipping, keyed connections) to prevent cross connection of duplicated systems.</p> <p>Identification of safety critical maintenance tasks are instrumental in highlighting potential ‘maintenance system’ failures (Leaflet B150 in CAP 562).</p> <p>The result of this assessment must be fed into the Operational Suitability Data, specifically the means of compliance to CD-MCDS.</p>

Continued

Table 6.1 Continued

Life cycle phase	Common Mode Type	Potential common mode sources/failures/errors	Comments
	Interchange of equipment and availability of spares	Not making necessary adjustments such as those to sensitivity and gain Failure to embody applicable modifications 'Robbery'/interchange of components, parts or articles	Example: ILS and Glide Slope settings on one aircraft can cause the course indicator of another aircraft to be either oversensitive or inhibit the warning system. Example: Fitting 'pool stock' items (such as engines) held in storage at a 'mod standard' incompatible with the donor aircraft (Air Transat Accident report 22/ACCID/GPIAA/2001).
	Continuing Airworthiness data	Contractual ambiguities Availability of data Critical updates of system failures in service (e.g. Airworthiness Directives, Service Bulletins)	Owner/operators contacts with maintenance service providers (e.g. outlining required maintenance tasks i.a.w. AMP data). The emphasis is on the owner/operators/maintenance providers to source available regulatory requirements (ie, failure to action ADs in a timely fashion) (refer European Aviation Safety Plan 2014–2017 , p. 34). Continuing Airworthiness data supplied with anomalies' or 'latent defects' leading to incorrect maintenance being performed.
	Calibration	Inadequately trained personnel Tools not calibrated (or inadequate)	

^aThis table assesses the vulnerability of the system to systematic failures. Systematic failures are due to systemic errors in the design, build, maintenance or operation of the system. The likelihood of many of these errors can be reduced by careful design.

Note: This checklist is derived from [SAE ARP 4761](#) (p. 162–164) and includes the author's experience. This list is intended to be thought provoking but has all the limitations of generic data. In no circumstances should it be considered complete or necessarily applicable to all systems. Note also, with reference to CS25.1309(c), it is interesting to note that there are currently no CS requirements to actually address errors by maintenance personnel in the System Safety Assessment.

It is recommended that the reader compiles their own CMA checklist tailored to the circumstances and that such a checklist be maintained as a live document within the procedures of the Safety Management System.

^bStudies by [Rasmussen and Mosleh \(2007\)](#) showed a major contributor to common-cause failure events in the nuclear power industry is programmatic maintenance practices (frequency, quality in both procedures and performance). Human errors related to procedures caused a small percentage of the total events. A vast majority of the events were not due to multiple failures in response to an operational demand, but result from a 'condition of equipment' (eg, inspection of one component revealing a deficiency leads to inspection of the redundant component, resulting in the discovery of the discovery of the same deficiency).

6.2.2 Step 2: analyse each Common Mode Analysis vulnerability to identify/verify independence criteria

6.2.2.1 Independence of 'AND' events

It is recommended that an independent specialist or Compliance Verification Engineer [CVE, refer EASA Part 21.A.239(b)] challenges each AND gate to ensure the robustness of the architecture.

6.2.2.2 The Common Mode Analysis checklist

For each CMA item identified, the System Safety Working Group (SSWG) must identify all sources of the failure/error/vulnerability. The SSWG might elect to delegate tasks to an analyst (or a subworking group) to investigate the issue further. In order to do this, the analysts need (refer ARP4761 para k.3.2) to have, or solicit, a thorough understanding of the subject system characteristics with regard to system operation and installation, for example:

- Design architecture and installation plans
- Equipment and components characteristics
- Maintenance and test tasks
- Crew procedures
- Systems, equipment and software specifications

The intent is that the SSWG and supporting analysts should proactively influence the design with the operator and maintainer in mind to minimise the possibility of error during the system's operational life. Common mode vulnerabilities due to HF may require process mapping to identify the critical steps in each task so that appropriate barriers or protection mechanisms can be put in place to during Step 3 to prevent error propagation.

6.2.3 Step 3: recommend corrective or preventative action

Following their identification, consideration should be given to design changes, manufacturing techniques, maintenance actions and system operating procedures to eliminate and/or mitigate Common Cause Failures. For example, for possible errors during maintenance, the HF specialist could ([SCF/SYS/A/108/4523](#) para 2.4.1) pose the following type of questions:

- Could a low-cost redesign eliminate the safety critical maintenance task?
- Could the frequency for a safety critical task be reduced?
- Could the functionally isolated step in a procedure be integrated into the procedure?
- Could a procedure which is subtly different from a more common procedure be made identical or more transparently different?
- Could a task which is inherently complex be simplified?
- Could documentation for a task be better related to the actual known reality of task performance?
- Could better cues be established to guide the correct progress of reassembly tasks?
- Could steps which are sometimes omitted in procedures be always included?
- Could components which are regularly inspected be repositioned to limit the impact on nearby systems?

- Could design differences for similar systems and subsystems be eliminated?
- Could permissive action links⁴ (PALs) be established to prevent systems on which an erroneous task has been undertaken being activated?

For any mitigation, the analysts needs (refer ARP4761 para k.3.2) to be familiar with all required safeguards required to eliminate or minimise common mode effects. These include (SCF/SYS/A/108/4523 para 3.3 and ARP4761 para k.3.2):

- Eliminate Error Potential by design:
 - Dissimilar parts
 - Redundancy
 - Design control and design quality level
 - Parts impossible to assemble incorrectly
 - Validation of maintenance task and data
- Reduce the severity of the effect by design:
 - System logic to inhibit or challenge incorrect actions
 - Failure paths
- Reduce the probability of the occurrence of the error by design:
 - Switches that are locked or guarded
 - Safety notices
 - Barriers
 - Improve reliability (to extend the maintenance interval)
- Improve the detectability (and correctability) of the error by design:
 - Functional testing
 - Built-In Test Equipment (BITE) checks
 - Reduce the inspection/scheduled maintenance interval
- Mitigation by Procedural Change:
 - Secondary sources of information
 - Safety warnings
- Mitigation by Training Change:
 - Educate on error potential
 - Use of procedures or specifications
 - Training of personnel
- Mitigation by Operational Change:
 - Duplicate inspection
 - Preflight checks
 - Safety tags
 - Checklists
 - Briefings

6.2.4 Step 4: provide validation of independence

Interdependency assertions will require some form of validation. Validation activities may include:

- Confirmation of revised design documentation, which shows that the common mode events have been designed out of the system.

⁴ PALs are failsafe devices used in the military to prevent the unintentional or illicit use of nuclear weapons. PAL refers to the fact that a predetermined sequence of actions must occur for the system to work.

- Confirmation of revised aircraft manuals, which specifies maintenance requirements, as well as makes recommendations for reactive fault diagnostics and proactive crew checks.
- Tests which induce specific failure modes. For instance, use hardware/software fault-injection techniques to subject redundant diverse versions to anomalous behaviour. The effect of the injected faults is observed to determine if common mode failures have occurred.

6.2.5 Step 5: update qualitative/quantitative assessments

Where there is lack of independence, the assessors need to update all applicable qualitative/quantitative assessments to reflect the remaining presence of any common mode events or systematic failures.

Systematic failures occur whenever a set of particular conditions is met and are therefore repeatable (i.e. items subjected to the same set of conditions will fail consistently) and thus apply to both hardware and software. It is difficult to quantify the rate at which systematic failures will occur and a qualitative figure based on the robustness of the development/build process is normally used.

6.3 The Case Study

In [Chapter 2](#) we defined a case study for the upgraded Attitude and Altitude Display System in which one safety argument (i.e. Strategy S7) in [Fig. 2.4](#) looked as shown in [Fig. 6.2](#).

The following sections will use the process in [Fig. 6.1](#) to result in a CMA report which could be issued as the solution to goal G14.

6.3.1 Step 1: identify common mode vulnerabilities

6.3.1.1 Step 1a: verify that ‘AND’ events are truly independent

In this case study we have used AND gates in the FTA (see [Section 4.3](#)), and each of these need to be challenged to ensure that there is no common mode failure which would oppose or negate the redundancy claims. [Table 6.2](#) is a proposed format to

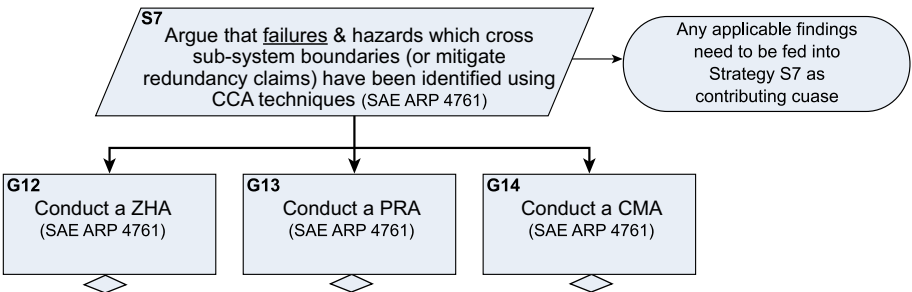


Figure 6.2 Strategy leading to CMA requirement.

Table 6.2 Common more failure assessment of FTA#4.1.1.a.1

CMA of FTA#4.1.1.a.1 (Revision 1)				
FTA proposed by			CMA assessed by	
Event	AND-gate concern	Assessments (i.e. Step 2)	Action required (i.e. Step 3)	Status
1	PFD1 and PFD2 might have a common mode failure	Data Bus connections and EWIS could introduce common mode failures	The FTA needs to be updated to incorporate the information in Fig. 1.11 and also to be in compliance with CS25.1309(d)	Open
2	DCU1 and DCU2 might have a common mode failure	As above	As above	Open
3	ADC1 and ADC2 might have a common mode failure	As above	As above	Open

capture events of concern. This proposed format can be used in the CMA report and/or can be inserted in the FTA report to support FTA validation (see Section 4.2.3).

6.3.1.2 Step 1b: identify common mode vulnerabilities via the Common Mode Analysis checklist

Use checklist in Table 6.1 to stimulate debate about potential common mode vulnerabilities in the design. The result of this action can then be captured in columns 1–5 of example Table 6.3.

6.3.2 Step 2: analyse each Common Mode Analysis vulnerability to identify/verify independence criteria

6.3.2.1 Independence of ‘AND’ events

With reference to Section 6.2.2.1, the independent assessor might wish to capture the result of this analysis in column 3 of Table 6.2.

6.3.2.2 The Common Mode Analysis checklist

With reference to Section 6.2.2.2, the SSWG can capture this step in column 6 of example Table 6.3. Note that this column may contain either:

- a short analysis/discussion/mitigation of the concern, and/or
- a reference out to a report where the issue is analysed in greater detail.

Column 9 in Table 6.3 is reserved for the evidence which will close off any outstanding actions.

Table 6.3 Example format for capturing common mode vulnerabilities^a

COMMON MODE ANALYSIS							REV	
Project title and number:								
Meeting date:			Attendees:					
Meeting date:			Attendees:					
Meeting date:			Attendees:					
(1) ID	(2) Life cycle phase	(3) Common mode type	(4) Potential common mode sources/ failures/errors	(5) Applica- ble? (Yes or No)	(6) Discus- sion/analysis/ mitigation	(7) Action/ recommendation	(8) Actionee/ status	(9) Evidence
1.1	Design	External Interfaces	Electrical power supply	Yes	Primary system is fed from DC buses Back-up system is fed from Battery bus	Ensure dual redundant primary system is fed from 2 separate DC busses (Main DC and Essential DC)	A. Sparky	
1.2			Ventilation supply	Yes	Possibility that forced air cooling is required behind instrument panel	Investigate and report back before PDR	B. Fitter	
1.3			Discharge/ Exhaust	No	No discharge/ exhausts within the scope of this modification		Closed	

Continued

Table 6.3 Continued

COMMON MODE ANALYSIS							REV	
Project title and number:								
Meeting date:			Attendees:					
Meeting date:			Attendees:					
Meeting date:			Attendees:					
(1) ID	(2) Life cycle phase	(3) Common mode type	(4) Potential common mode sources/ failures/errors	(5) Applicable? (Yes or No)	(6) Discussion/analysis/ mitigation	(7) Action/ recommendation	(8) Actionee/ status	(9) Evidence
1.4		Systemic	Software				C. Soffe	
1.5			Hardware environmental deficiencies				L. Hardy	
2.1	Manufacture	Quality Control	Incorrect process	No	No System Level 4 vulnerabilities identified to date		Closed	
2.2			Inadequate inspection	No			Closed	
2.3			Inadequate testing	No			Closed	
3.1	Integration	Installation error	Mal-assembly	Yes	ATT and ALT displays may be vulnerable to cross connection	Ensure connector keys will prevent this	A. Sparky	

3.2			Mal-rigging	No	No mechanical vulnerabilities identified			
3.3			Common fitter	Yes		Ensure job-card has inspector sign-off before zone is closed	A. Sparky	
4.1	Test	First test	Post maintenance tests					
5.1	Operate	Common pilot	Inadequately trained personnel					
5.2			Overstressed operator (omission of action, incorrect or inadequate commission or action)					
5.3			Faulty operating instructions					
5.4			Misdiagnoses (following errors/failures)					
5.5		Aircraft settings	Aircraft weight and balance					
5.6			Airport altitude					

Continued

Table 6.3 Continued

COMMON MODE ANALYSIS							REV	
Project title and number:								
Meeting date:			Attendees:					
Meeting date:			Attendees:					
Meeting date:			Attendees:					
(1) ID	(2) Life cycle phase	(3) Common mode type	(4) Potential common mode sources/ failures/errors	(5) Applica- ble? (Yes or No)	(6) Discus- sion/analysis/ mitigation	(7) Action/ recommendation	(8) Actionee/ status	(9) Evidence
5.7		Software updates	Flight line updates					
5.8		Mechanical failures	Temperature exceedance					
5.9			Dust & Dirt					
5.10			Vibration					
5.11			Pressure					
5.12			Humidity					
5.13			Stress					
5.14			Fatigue					

5.15		Electrical	Power surge, voltage					
5.16			Power surge, current					
5.17			Short circuit					
5.18			EMC					
5.19		Radiation	EMI					
5.20			HIRF					
5.21			HERTA					
6.1	Maintain	Common Staff	Maintenance error could block pitot-static ports	Yes	(see Chapter 9 App A4)	Ensure warning labels are provided (“ <i>Do not block or deform ports. Indicated area must be smooth and clear</i> ”) Ensure recurring pilots training ensures they can correctly identify and diagnose pitot-static failures	TBD	TBD (See example in Section 11.2.3)
6.2								
6.3								

^aA few failure conditions are explored below to illustrate the CMA concept. To facilitate learning, the reader is invited to further complete the exercise

6.3.3 Step 3: recommend corrective or preventative action

This may be captured in [Table 6.2](#) (column 4) and [Table 6.3](#) (column 7). During the early stages of the design, these recommendations will be aimed at the design process. As the system approaches flight trials and certification, these recommendations will be for specific entry into the aircraft Technical Publications (e.g. Flight Reference Cards, Maintenance Manuals, etc.).

6.3.4 Step 4: provide validation of independence

[Table 6.2](#) (column 4) and [Table 6.3](#) (column 9) allow for validation that independence is properly evidenced or that any corrective/preventative action is satisfactorily accomplished.

6.3.5 Step 5: update qualitative/quantitative assessments

The final step in the CMA process (assuming it has completed as many iterations as required to reflect the evolving design and the resulting probabilistic assessments) is to ensure that all qualitative/quantitative probability declarations are appropriately amended. Again, column 9 in [Table 6.3](#) can be used to reference the evidence.

6.4 Discussion

Fail Safe design principles (see [Kritzinger \(2006\)](#) Chapter 7) will go a long way in preventing CMA vulnerabilities. More specifically, the following strategies should keep common failure modes to a minimum:

- Use detailed design precautions which minimise the risk of mal-assembly, poor rigging and cross connections. If not possible resort to clearly defined procedures which not only specify the action required, but also provide the independent verification required to check that the job has been completed correctly.
- Ensure that critical areas are readily inspectable with an indication of the failure/fault/hazard that the procedure is intended to guard against.
- Devise adequate check-out procedures to cater for maintenance errors which could result in a failure/fault/hazard.
- Mechanical and electrical segregation of duplicated systems is good design practice, whether required by the Safety Assessment or not. However, this is very difficult and expensive to rectify if this deficiency is discovered too far down the development life cycle.
- Add special protective features (e.g. waterproofing) to protect safety critical equipment, or equipment that can fail hazardously.
- Make appropriate use of alternative technologies (e.g. using fibre-optics to transmit data as protection from EMI/EMC).
- Make wise use of dissimilar redundancy (e.g. electrical systems backed-up with a mechanical alternative). The virtue of dissimilar redundancy is that, because the channels are fundamentally different in their design, it is much less likely for an external event to affect them all in the same way ([Lloyd and Tye, 1982](#), p. 94).

- Provide warning systems which allow the crew to immediately recognise the problem and take appropriate corrective action. Consider the implication of multiple simultaneous warning (e.g. from cascade failures) and their impact on crew workload and performance (see [Chapter 10](#)).
- Ensure that the pilot is kept ‘in the loop’. For instance, an autopilot may automatically cope with a series of developing problems without informing the crew. When it reaches a point where the autopilot cannot control the aircraft, it may then suddenly drop a complex situation in the lap of the pilot.
- Provide automatic inhibition of controls in the phases of flight where improper operation would produce a catastrophe. If not feasible, then ensure that there is enough time for improper operation to be recoverable.
- Install flight instruments that will minimise the risk of ‘misreading’ (due to ambiguity or lack of clarity). Consider the instrument scanning patterns of the crew when deciding on instrument location.

6.5 Conclusion

Since the inception of the concept of system safety engineering, there has always been a concern with regard to common failure modes and how to identify them. Many analysts attempted to identify common failure modes with hit-or-miss analyses without utilising any sort of systematic process for the entire product life cycle, and one which proactively mitigates any HF vulnerabilities which can preemptively be designed out of the system. Extensive HF interventions in the Continuing airworthiness phase is systematic of HF neglect during the Initial Airworthiness phase.

The CMA process defined in this chapter is derived from [SAE ARP 4761](#) (which states on p.159, that it is a qualitative analytical tool used to ensure the ‘goodness’ of a design).

Application of this CMA process is not a guarantee that all common failure modes have been identified and mitigated, but it does provide a structured (i.e. systematic) approach, with the following advantages and limitations.

6.5.1 Advantages

- Most other techniques concentrate on the as-designed system functionality. The CMA ensures that the *installed* design is free from common causes which can undermine design, qualitative and quantitative predictions.
- Verifies that the ‘AND’ events in the FTA/DD/MA are independent in actual implementation.
- A good second line check on design to identify common development errors (e.g. software design errors, avoidable installation error, etc.).
- Supports the selection of system architecture through verification that appropriate independence has been achieved.
- Analyses system architectures that rely on redundancies. Establishes and validates physical and functional separation and isolation requirements between systems.
- May identify common environmental hazards (e.g. HIRF, moisture, temperature, etc.).
- Although not required for 25.1309 compliance, the checklist can easily be extended to identify health- and safety-related hazards in each phase of the project life cycle.

6.5.2 Limitations

- Used throughout design process, but more cost-effective if done earlier because of the influence on system architecture. However, confirmation is often only feasible when the implementation is complete.
- Difficult to be rigorous and systematic.
- Requires detailed knowledge of the system.
- Analysis of FTA/MA/DD is best done by an independent third party (independent of the probabilistic assessment, but familiar enough with the actual system architecture to spot vulnerabilities in the argument).
- Application of the checklist is best suited to brainstorming sessions with multiple input, and these meetings are difficult to chair.
- Relies on acceptance that seemingly unlikely events will occur.
- The checklist has all the limitations of generic data. In no circumstances should it be considered complete or necessarily applicable to all systems.

References

- CAA Paper 2011/03. CAA “Significant Seven” Task Force Reports, Safety Regulation Group, Gatwick, UK.
- CAP562, November 23, 2013. Civil Aircraft Airworthiness Information and Procedures. Safety Regulation Group, Gatwick, UK.
- CS25, July 17, 2015. Certification Specifications and Acceptable Means of Compliance for Large Aeroplanes, Amendment. European Aviation Safety Agency, Cologne.
- Edwards, E., 1988. Part 1 Introductory overview. In: Wieger, Nagel (Eds.), Human Factors in Aviation. Academic Press.
- European Aviation Safety Plan 2014–2017. EASA, Cologne.
- FAR25, November 2007. Airworthiness Standards: Transport Category Airplanes, Amendment 25-123m. Federal Aviation Authority, Washington.
- Graeber, C., August 2010. Human Factors, Boeing Human Factors Engineering, http://www.boeing.com/commercial/aeromagazine/aero_08/human_textonly.html.
- Kritzinger, D.E., 2006. Aircraft System Safety: Military and Civil Aeronautical Applications. Woodhead Publishing Ltd, Cambridge, CB1 6AH.
- Lloyd, E., Tye, W., 1982. Systematic Safety. CAA, London.
- Rasmuson, D.M., Mosleh, A., June 20, 2007. A Brief History of Common-Cause Failure Analysis, Presentation at the IAEA Technical Meeting on “CCF in Digital Instrumentation and Control Systems for Nuclear Power Plants”, Bethesda, Maryland, USA.
- SAE ARP 4761, December 1996. Guidelines and Methods for Conducting the Safety Assessment Process on Civil Airborne System and Equipment.
- SCF/SYS/A/108/4523, 2001. Human Hazard Analysis: A Demonstrator for a Means of Compliance, a Technical Report by the Systems Centre of Competence. Airbus UK Ltd, Filton..

Further reading

- Mosleh, A., 1993. Procedure for Analysis of Common-cause Failures in Probabilistic Safety Analysis, NUREG/CR-5801. US Nuclear Regulatory Commission.

Particular risk analysis

7

There is no reason to fly through a thunderstorm in peacetime.

Sign over squadron ops desk, Davis-Monthan AFB, AZ, 1970

7.1 Introduction

7.1.1 Background

As discussed in [Chapter 6](#), the acceptance of a System Level 3 or 4 (see Fig. 1.1) failure probability is often based on the assumption that failures are independent (AMC25.1309). However, this approach does not sufficiently recognise (refer, inter alia, [SAE ARP4761](#) App I para 1) the threats which external events (outside of the immediate system boundary) have on assumptions made about the robustness of our fail safe designs.

It is necessary, therefore, to conduct specific studies to ensure that systems are adequately protected against all foreseeable external events, and the FAA (FAR25.1309), EASA (CS25.1309) and [SAE \(ARP4761\)](#) refer to this study as a Particular Risk Analysis (PRA).

Note: The term ‘Particular Risk Analysis’ may create a bit of confusion, as ‘risk’ is usually defined as the combination of the probability of the occurrence of an event (in safety terms this event is the accident) and the severity of its outcome. The PRA (as promulgated in guidance material such as AMC25.1309 and [ARP4761](#)) uses goal/failure based safety criteria rather than risk/accident based criteria (see [Kritzinger, 2006](#), Chapters 4 and 5) for more on this distinction). With this in mind, it may be more appropriate to rather refer to this tool as a Particular Event Analysis, which avoids potential confusion and distinguishes this hazard identification technique from a Hazard Log (which is used to manage the risk of hazards in the operational Safety Case, see Chapter 110029).

7.1.2 Aim of the particular risk analysis

The aim of the PRA is to identify those events or influences, which exist outside of the system (or item) concerned, but may generate hazards or violate system independence claims.

7.1.3 Objectives of the particular risk analysis

The PRA considers any event outside the immediate boundaries of a system which could cause system failure and impact airworthiness. Once identified, each particular

event is subject to a specific study to examine and document its effect on the system and the aircraft.

7.1.4 Scope of the particular risk analysis

By definition, it is not necessarily straightforward to limit the scope of the PRA. It is meant to be a catch-all tool to account for any external events (e.g. in the operation environment) which may cause individual or simultaneous failures, which may subsequently lead to a safety incident or accident. Depending on the scope of the checklist utilised (see [Table 7.1](#)), the PRA might identify similar vulnerabilities¹ as other tools/techniques (e.g. ZSA, CMA or even the FHA).

7.2 Conducting a particular risk analysis

7.2.1 Modelling the process

Considering the objective of the PRA in [Section 7.1.3](#) above, a typical process for every particular hazard/event identified is illustrated in [Fig. 7.1](#).

7.2.2 Step 1: identify any particular hazard/event

Identify any possible events outside the immediate boundaries of the system which could have an implication on the safety of the aircraft. Both the cause of the event as well as its possible result (e.g. system failure) must be identified, as either may be mitigated in Step 5.

The identification process is most usefully done via a brainstorming session in a System Safety Working Group (SSWG). Regulations (e.g. for bird strike and lightning strike requirements) provide a useful source of data; however, as with the Common Mode Analysis, the SSWG may elect to use a PRA Checklist (see example in [Table 7.1](#)) which has the purpose of making this part of the assessment as systematic as possible. In any approved Design Organisations with robust processes, one would expect such a table to grow through a number of iterations to include the lessons learned from past SSWG meetings and regulatory updates.

7.2.3 Step 2: derive a safety target

Assess the failure modes that could arise from these events of concern. Use the agreed safety criteria to allocate failure mode severity as well as a target probability for the failure mode occurrence.

This step is identical to Step 3 of the FHA process (see [Section 3.2.3](#)).

¹ This might seem like a duplication of effort, but does not have to be. Each tool/technique approaches system vulnerabilities from its own perspective and may all reference the same qualitative/quantitative analysis which evidences its mitigation and or probability deceleration.

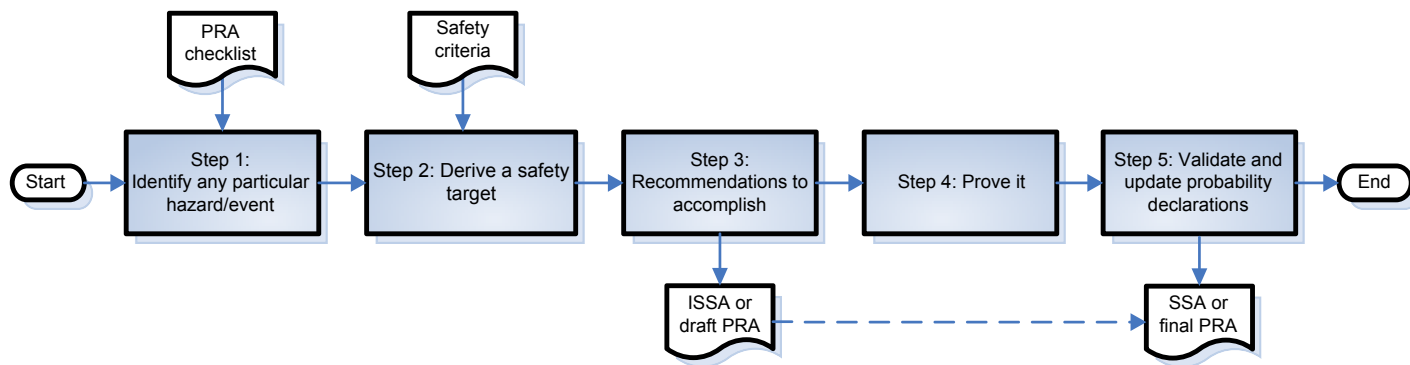


Figure 7.1 Typical PRA process.

Table 7.1 Example PRA checklist

PRA topic	Potential PRA sources/failures/events	Comments
Fire and smoke	<p>Chafing of wiring looms near combustible material. Normally low-combustible materials in the presence of strong oxidisers or high temperatures.</p> <p>Hydrogen from charging batteries.</p> <p>Ignition sources could include:</p> <ul style="list-style-type: none"> • Open flames • Arcs and sparks • Hot surfaces • Lightning strikes • Adiabatic compression • Hypergolic mixtures • Prophoric mixtures • Water-sensitive reactive materials • Lithium batteries 	<p>A fire could destroy dual redundancy of a safety critical system, so ensure electrical and physical segregation in the design.</p> <p>On-board fire, smoke and fumes represents such a significant hazard in aviation (the seventh most frequent cause of accidents), that EASA has added it as a significant operational risk area (European Aviation Safety Plan, 2014–17). It can lead to loss of control, either as a result of structural or control system failure or crew incapacitation.</p>
Kinetic energy	<p>Noncontainment of moving components (e.g. RAT, Fans, APU Engine, burst pressure vessel).</p> <p>Wheels (e.g. tyre burst, flailing tread, rim release, etc).</p> <p>Flailing shafts.</p>	<p>Need to consider issues such debris trajectory (e.g. see ARP4761, p. 304) and resulting vibration, loss of function, etc.</p>
Potential energy	<p>Rupture of pressure vessels, pipes or ducts.</p> <p>Bulkhead rupture.</p>	<p>Consider issues such as pressure vessel containment, damage to areas subjected to very high/low temperature, explosive decompression, etc.</p>
Hazardous materials	<p>Mercury (damages aluminium and is toxic in liquid or vapour form).</p> <p>Ignition of hydrogen produced by battery charging.</p> <p>Fuel, lubricant or solvent in contact with strong oxidiser.</p>	<p>Many of these will be subject to the restriction of Health & Safety Regulations.</p> <p><i>Note, however, that CS25.1309 is concerned over materials which could impact airworthiness. Health and environmental concerns may form part of a different assessment, targeted at a different authority with its own safety criteria (i.e. CS25.1309 criteria is not applicable to hazards faced by aircraft technicians).</i></p>

Toxic gas	<p>Crew and passenger exposure to:</p> <ul style="list-style-type: none"> • cryogenic, carcinogenic and/or highly toxic chemicals. • any gas which could displace oxygen. 	Need to consider any component than can release toxic gas/fumes when overheated or exposed to fire.
Electrical	<p>Arcing and sparking causes:</p> <ul style="list-style-type: none"> • Ignition of combustibles • Build-up and welding of contacts • Surface damage to metals • Interference with electrical equipment operation • Electrical noise and cross talk • Electrostatic discharge <p>Events could include:</p> <ul style="list-style-type: none"> • Inadequate or deteriorated insulation • Erroneous connection • Bare conductors touching • Dirt, contamination or moisture • Corrosion • Bent connector pins • Component failure • Insufficient of deteriorated grounding • Stray current from cross-connection, sneaks-circuits, static electricity discharge, coupling, etc. • System overloading 	<p>Secondary effects could include:</p> <ul style="list-style-type: none"> • System inoperative in hazardous situation • Release of holding devices • Detection and warning devices inoperative
EMI/EMC	<p>Electromagnetic interference and compatibility. All electrical and electronic technologies emit EM disturbances that can interfere with the correct operation of radio communications or other electronics.</p> <p>Other threats include lightning strike and radiation.</p>	Safety/Mission critical systems are required to demonstrate that operational capability is maintained for all the threat scenarios. Safety is to be demonstrated for ordnance, fuels and personnel.

Continued

Table 7.1 Continued

PRA topic	Potential PRA sources/failures/events	Comments
Lightning	<p>Indirect effects of lightning can cause power transients and anomalies that can damage sensitive/critical LRUs or damage wire looms. EMI can be induced into wiring by the lightning currents flowing on the outside of the fuselage.</p> <p>Direct effects of lightning include fire, damage to external components/structure, loss of function of equipment hit, acoustic shock, delamination of composite material etc.</p> <p>Main strikes occur between the extremities of the aircraft (wing tips, nose, tail), but they can sweep along the fuselage or across the wing behind projections such as the engines.</p>	<p>Also required by FAR25.981, 25.954, 25.1316.</p> <p>Principle concerns:</p> <ul style="list-style-type: none"> • ignition of fuel vapour at vents disruption of nonmetallic unbonded parts • voltage injection into system (particularly of earthed) from a charged aircraft skin • Localised heating of nonmetallic panels • lightning dwell (e.g. where paint is too thick) causing localised heating and even penetration <p>Direct effects can [Moore, P] be minimised by:</p> <ul style="list-style-type: none"> • Maintaining a highly conductive metal aircraft fuselage surface. For composites this can be achieved via a mesh of aluminium/copper/bronze or a zinc/aluminium spray. • Carefully design bolt receptacles to prevent interior sparking, especially in wet wings. • Ensure all metal components are electrically bonded with very low resistance ($<5\text{ m}\Omega$). <p>Indirect effects can be minimised by:</p> <ul style="list-style-type: none"> • Keeping radiation away from wiring (through screened/braided conduits, shields, etc). • Using high standard wiring and not locating close to the fuselage skin. • Harden LRUs (enclose the design, use filters, conform to data standards and signal levels).
Radiation	<p>High intensity radiated fields caused by:</p> <ul style="list-style-type: none"> • Microwave/radio frequency from radar and communications equipment operation/malfunction or high power microwave equipment operation. • Electromagnetic pulse, which can cause damage to electrical/electronic components, equipment and systems, loss of magnetically stored data, lack of communications. 	<p>Also required by FAR25.1317</p>

Bird strike	Bird strike on cockpit windows, all leading edges, engine intakes, other cooling intakes, antennas, etc.	Also required by FAR25.631
Fluid leakage	Fuel, hydraulics, water, battery acid, etc. Condensation against inner fuselage. Drink spillage in cockpit.	These are usually examined as part of the ZSA, but may sometimes require specific additional assessment.
Cabin air and Pressurisation	Inadequate oxygen for respiration due to: <ul style="list-style-type: none"> • High altitudes • Dilution by inert gases • Combustion that consumes all available oxygen. • Insufficient ventilation of occupied, enclosed space • Atmospheric pollution by industrial, automobile or other exhausts 	
High temperature	Generation or absorption of heat from: <ul style="list-style-type: none"> • Engine operation, fire or explosion • Electrical, laser or solar heating • Aerodynamic friction • Friction between moving parts • Gas compression • Weather • Lack of insulation from thermal sources • Inadequate heat dissipation capacity or cooling system failure 	Secondary effects of high temperature exposure could include: <ul style="list-style-type: none"> • Reduced strength of metals and other materials. • Distortion and warping of parts • Decreased viscosity of lubricants • Increased fuel tank flammability • Ignition due to hot spots
Low temperature	<ul style="list-style-type: none"> • Icing of operating equipment (e.g. pitot tubes) • Freezing of liquids (e.g. water in fuel tanks) • Condensation of moisture and other vapours 	Secondary effects of low temperature exposure could include: <ul style="list-style-type: none"> • Reduced viscosity of liquids and lubricants
Cyber security	<ul style="list-style-type: none"> • Computer viruses • Hackers 	The use of computer-based systems in sophisticated air navigation systems and on-board aircraft control and communications systems

Continued

Table 7.1 Continued

PRA topic	Potential PRA sources/failures/events	Comments
Other	<i>TBD – each SSWG to consider how this checklist has to evolve.</i>	<i>Examples could include:</i> <ul style="list-style-type: none">• <i>Wheel-up landing</i>• <i>Rapid decompression, including air pressure bulkhead rupture</i>• <i>High vibration analysis (e.g. engine fan blade out and nose wheel imbalance)</i>• <i>System vulnerability to terrorist action (e.g. Bomb on board and Cockpit intrusion)</i>• <i>Cargo shifting</i>• <i>Tail strike</i>• <i>Ditching</i>• <i>Fuel tank ignition, etc.</i>

Note: This checklist is partially derived from [SAE ARP4761](#) (para J.1) and includes the author’s experience. This list is intended to be thought provoking but has all the limitations of generic data. In no circumstances should it be considered complete or necessarily applicable to all systems. It is recommended that the reader compiles their own PRA checklist tailored to the circumstances and that such a checklist be maintained as a live document within the procedures of the Safety Management System.

7.2.4 Step 3: recommendations to accomplish

Stipulate what needs to be done to prove accomplishment of the safety objectives allocated in Step 2. To eliminate or mitigate any particular hazard or event, recommendations may take the form of changes to the design, testing required to evaluate effects following an initiating event, further analysis to be conducted, manuals to be amended, etc.

At this point the author of the PRA must ensure visibility of these recommendations to all relevant stakeholders. Consideration needs to be given to issuing the first draft of the PRA as early as possible, especially if the output influences the requirements management process (see Fig. 1.3). This may take the form of a stand-alone report or may be contained within any interim updates of the System Safety Assessment (i.e. PSSA is the first issue, SSA is the final issue, with as many ISSAs as required to keep track with the evolving design and maturing System Safety Assessment).

7.2.5 Step 4: prove it

Having identified the event of concern, each should then be subject to a separate study which is subsequently used to validate the probability declaration in the PRA.

These studies are primarily qualitative analyses ([SAE ARP4761](#) para J.3) performed by:

- defining the details of the particular event to be analysed;
- defining the affected zones/areas/systems;
- listing the certification requirements to be fulfilled;
- reviewing the existing design and installation precautions;
- conducting appropriate work to mitigate the probability or the severity of the event (e.g. analysis, simulation, tests, maintenance procedures, etc.);
- reviewing the safety effect of the event being realised.

The objective of each analysis is to ensure that any safety-related effects are either designed out or shown to be acceptable by virtue of their probability of occurrence.

In some circumstances the analyst may need to take account of the likelihood of the initiating event. [Table 7.2](#) is tailored from [Kritzinger \(2006, Chapter 10\)](#) and may contain some useful information for the reader. If used, such probabilities will need to be declared to (and agreed with) the certification authority.

7.2.6 Step 5: validate and update the probability declarations

Using the event probability, in combination with all mitigations accomplished, declares the probability of the failure mode.

Table 7.2 Useful aircraft-related event probabilities^a

Event	Probability of occurrence	Source	Comments
Cabin high altitude requiring passenger oxygen	No accepted standard data	AMC25.1309 (Amd1, App 4)	
Fire in lavatory, cargo compartment, APU compartment, engine	No accepted standard data	AMC25.1309 (Amd1, App 4)	
Flight conditions $\leq 0g$	No accepted standard data	AMC25.1309 (Amd1, App 4)	
Flight conditions $\geq 1.5g$	No accepted standard data	AMC25.1309 (Amd1, App 4)	
Flight conditions requiring stall warning	10^{-2} per flight	An assumption in AMC25.1309 (Amd1, App 4)	
Flight conditions resulting in a stall	10^{-5} per flight	An assumption in AMC25.1309 (Amd1, App 4)	
Excessiveness of V_{MO}/M_{MO}	10^{-2} per flight	An assumption in AMC25.1309 (Amd1, App 4)	
Go-around	No accepted standard data	An assumption in AMC25.1309 (Amd1, App 4)	
Any rejected take-off	No accepted standard data	An assumption in AMC25.1309 (Amd1, App 4)	
High energy rejected take-off	No accepted standard data	An assumption in AMC25.1309 (Amd1, App 4)	
Need to jettison fuel	No accepted standard data	An assumption in AMC25.1309 (Amd1, App 4)	
HIRF conditions	No accepted standard data	AMC25.1309 (Amd1, App 4)	
Normal icing (trace, light, moderate icing)	1 (ie, assumes to be always present in a flight)	AMC25.1309 (Amd1, App 4)	
Severe icing	10^{-2} per flight	ACJ25.1309 (Amd16, App 4)	Note that the updated AMC25.1309 (Amd1) does not contain a recommended probability for this event
Air temp $< 70^{\circ}C$	No accepted standard data	AMC25.1309 (Amd1, App 4)	

Lightning strike	No accepted standard data	ACJ25.1309 (Amd16, App 4)	Lloyd and Tye (p. 84) suggested for large transport aircraft: • 1 strike every 6000h worldwide • 1 strike every 2400h in EU
Wind: Head wind >25 kts during take-off and landing	10^{-2} per flight	AMC25.1309 (Amd1, App 4)	See also AC120-28 and CS-AWO
Wind: Cross wind >20 kts during take-off and landing	10^{-2} per flight	AMC25.1309 (Amd1, App 4)	See also AC120-28 and CS-AWO
Wind: Tail Wind >10 kts during take-off and landing	10^{-2} per flight	AMC25.1309 (Amd1, App 4)	See also AC120-28 and CS-AWO
Wind: Limit design gust and turbulence	10^{-5} per flight hour	AMC25.1309 (Amd1, App 4)	See also CS25.342 (under review by Structures Harmonisation Working Group)
Event rate for fires occurring in Class D and Class C compartments	1×10^{-7} per departure	EASA NPA 2013-23 para 3.1	

Note: AC25.1309 (page 2-F-30) advises that if '*no accepted standard data*' appear in this table, then the designers must provide a justified value if the probability used is less than 1.

Note: Sometimes data are valid only in special circumstances. For instance, a statistical source may indicate that a specific number of aircraft accidents due to bird strikes take place every 100,000 hours. One may conclude from this data that the probability of a bird strike is comparatively low. Hidden by the data analysis approach is the fact that at certain airfields, such as Boston, the Midway Islands and other coastal and insular areas where birds abound, the probability of a bird strike accident is much higher than the average. This example demonstrates that generalized probabilities will not serve well for specific, localized areas. This applies to other environmental hazards such as lightning, fog, rain, snow and hurricanes.

^aFor suggested additions, or an up-to-date version of this table, please contact the author at www.aircraftsystemsafety.com.

Mitigation assertions will require some form of validation. Validation activities may include:

- confirmation of revised design documentation, which shows that the PRA events have been designed out of the system;
- confirmation of revised aircraft manuals, which specify maintenance requirements, as well as makes recommendations for reactive fault diagnostics and proactive crew checks;
- tests which induce specific particular threats (e.g. EMI test, bird strike tests, etc.), etc.

7.3 The Case Study

In Section 1.3 we defined a case study for an upgraded Attitude and Altitude Display System. One branch of the Safety Strategy of Fig. 2.4 is repeated in Fig. 7.2:

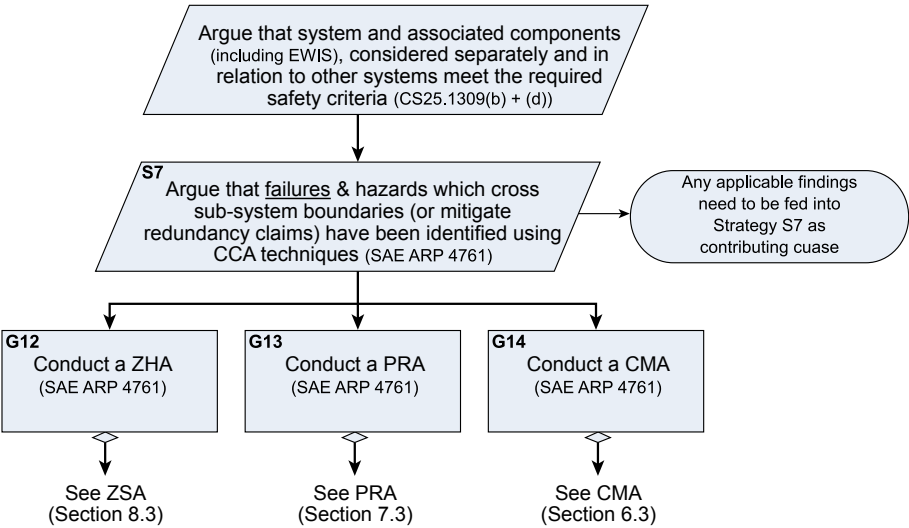


Figure 7.2 Strategy leading to PRA requirement.

In the following subparagraphs we apply the PRA approach of Section 7.2 to accomplish goal G13.

7.3.1 Step 1: identify any particular hazard/event

Using the checklist in Table 7.1, the SSWG may elect to capture this activity as per the format shown in the first four columns of Table 7.3.

Remember, the checklist is simply meant to stimulate thought and debate. It is therefore recommended that:

- The SSWG should not feel restricted by the checklist (i.e. it is often necessary to update the checklist following a good SSWG meeting). However, always remember the aim of the PRA (see [section 7.1.2](#)) and try not to stray into the domain of other tools (such as the ZSA). Some duplication will inevitably do arise, in which case continuity may be accomplished by cross-referencing to the same qualitative/quantitative analysis where applicable.
- The SSWG be encouraged to not delete a PRA topic (column 2) if no concerns are identified. Capturing that decision shows that it has been systematically assessed and may stimulate participants to reconsider any missed concerns whenever the PRA is reviewed.

Note: In the fourth column of [Table 7.3](#), the author has elected to capture the causes and/or consequences of the hazard/event. The reader may elect to capture this in separate columns (space allowable). Both causes and consequence are needed to accomplish Step 2.

7.3.2 Step 2: categorise severity and allocate probability

Using the failure mode severity categories in [Table 3.2](#), allocate allowable system failure probability targets using [Table 3.3](#). The SSWG may elect to capture this activity as shown in the columns 5 and 6 of [Table 7.3](#).

Note: For those assessors used to working with Hazard Logs (and the risk/accident-based criteria of standards such as DEF STAN 00-56 or MIL-STD-882), it is worth pointing out that the SSWG must be consistent in their use of safety criteria. Mixing risk/accident-based criteria with goal/failure-based criteria is fraught with difficulties ([Kritzinger, 2006](#), Chapters 4 and 5). When using the goal-based requirements of specifications (such as CS25.1309), the SSWG must be diligent in allocating the severity and probability target to the system, not to the hazard/accident. For example in ID1.1 of [Table 7.3](#), we are concerned that there may be a fire in the avionics bay. The result (or end effect) is considered to be CATASTROPHIC, and the subsequent steps will aim to prove that the likelihood of the fire is EXTREMELY IMPROBABLE. Using the goal/failure-based requirements of CS25.1309, we do not build an accident sequence (required by risk/accident-based criteria) to consider the probability of various types of resulting accidents (e.g. ranging from aircraft crash, to hull loss with no loss of life, to passengers inhaling smoke or receiving burn wounds). This does not mean that this event should not appear in a Hazard Log (see [Chapter 12](#)), but that the PRA is a different tool compared to the Hazard log-with a different scope and objective.

7.3.3 Step 3: recommendations to accomplish

The SSWG may elect to capture the recommendations as shown column 7 of [Table 7.3](#). These recommendations may include instructions for further assessment to be conducted, additional tests or changes to design requirements, etc. In order to ensure accomplishment, it is vital that such recommendations are formally allocated as actions resulting from the PRA.

7.3.4 Step 4: prove it

The last column in [Table 7.3](#) has been reserved to evidence the accomplishment of any recommendations. This column might include a qualitative substantiation that the

Table 7.3 Particular risk analyses for the Altitude & Attitude Display System^a

Particular risk analysis								REV	
Project title and number:									
Meeting date:			Attendees:						
Meeting date:			Attendees:						
Meeting date:			Attendees:						
ID	PRA topic	Event	Causes/results	Failure mode severity	Qualitative objective	Mitigation required	Declared probability	Substantiation	
1.1	Fire	Fire in avionics bay	Various causes (e.g. overheating LRU, short circuit, etc.) causing smoke in cabin and may destroy surrounding equipment	Catastrophic	Extremely Improbable	1. Ensure physical segregation of LRUs. 2. Ensure electrical segregation of duplicated EWIS. 3. Ensure all LRU are connected to Circuit Breakers. 4. Ensure that all LRU's requiring cooling have built in temp switches, or add temp sensors. 5. Provide smoke detection devices and fire suppression equipment. 6. Provide crew training instructions to combat this event.	TBD	1. TBD 2. TBD 3. TBD	
1.2	Fire	Fire in hidden areas	Chaffing of EWIS behind instrument panel or behind cabin insulation Causes smoke in cabin and disruption of other aircraft systems	Catastrophic	Extremely Improbable	1.Ensure cable and clipping in accordance with procedure XXX. 2.Provide zonal inspection procedure at every C-check. 3.Provide crew training instructions to combat this event.			

2	Kinetic Energy	TBD – check if any LRUs contain cooling fans which may shatter	TBD	TBD	TBD	Revisit this topic at CDR	TBD	TBD
3	Potential Energy	No particular events identified to date.	N/A	N/A	N/A	Revisit this topic at CDR	N/A	N/A
3	Hazardous Materials	TBD – review each supplier's Haz Mat deliverable.	TBD for causes Negative airworthiness consequence only if material leaks in failure conditions (incl. overheat)			Revisit this topic at CDR. Ensure that the H&S departments see the HazMat List so that D-level and I-level maintenance procedures can be amended accordingly		
4	Toxic Gas	TBD – review each supplier's Hat Mat deliverable.	TBD for causes Negative airworthiness consequence only if material releases gas in failure conditions (incl. overheat)			Revisit this topic at CDR. If gas can be released in failure conditions (including fire), then ensure that aircraft manuals flag this warning		
5	Electrical							

Continued

Table 7.3 Continued

Particular risk analysis								REV	
Project title and number:									
Meeting date:			Attendees:						
Meeting date:			Attendees:						
Meeting date:			Attendees:						
ID	PRA topic	Event	Causes/results	Failure mode severity	Qualitative objective	Mitigation required	Declared probability	Substantiation	
6	EMI/EMC	Could cause display of hazardously misleading information	Any aircraft electrical systems Any Passenger Electronic Devices	Catastrophic	Extremely Improbable	Conduct EMI/EMC tests for to evaluate aircraft system compatibility. Restrict use of PED until further notice.	TBD	TBD – EMI/EMC report required	
7.1	Lightning	Direct strike	Damage to aircraft nose or pitot-static ports can impact sensor accuracy Unlikely to impact both side in identical manner, but note that Captain's primary display and the standby display are fed from port side pitot statics						
7.2	Lightning	Indirect effects							

8.1	Radiation	HIRF						
8.2	Radiation	Electro-magnetic Pulse (EMP)	Caused by EMP weapon. Will destroy all electronic data	Cata-strophic	Extremely Improbable	System not contracted to be EMP hardened. Ensure that flight reference cards provide instructions for loss of all electronic data (whatever the cause)	Extremely Improbable	Likelihood of event is historically Extremely Improbable. No further action proposed.
9	Bird strike	Strike on/near pitot-static ports	Damage to aircraft nose or pitot-static ports can impact sensor accuracy Unlikely to impact both side in identical manner, but note that Captain's primary display and the standby display are fed from port side pitot statics					
10								
11								

^aNote the following:

- A few failure conditions are explored to illustrate the PRA concept. To facilitate learning, the reader is invited to further complete the exercise.
- This is a suggested layout only. There are many ways to conduct and capture the results of a PRA.
- It is assumed that this PRA is conducted at PDR, and the 'TBD' indicates items which will need to be confirmed/completed before certification.
- The user may wish to add extra columns, such as:
 - Actionee (who is responsible for providing event probability mitigation);
 - CVE (who is the independent Compliance Verification Engineer);
 - Status (e.g. Open/Closed/Pending).

HazMat = Hazardous Material.

probability target has been met, but more often than not, it will reference another report where the practical concern has been fully analysed and the probability of the resulting system failure has been properly proven.

7.3.5 Step 5: validate and update the probability declarations

Step 5 allows the SSWG to declare that the safety objectives (which were allocated in Step 2) are indeed accomplished in the final design.

The second last column in [Table 7.3](#) has been reserved for this purpose.

At this point in time, it may be required that the events of concern are added to any qualitative/quantitative analyses (such as FTA) that have been conducted to prove the functional failure probability targets of [Chapter 3](#).

If the target has not been accomplished, then there are three options available:

- Change the design in order to meet the target. This could add cost and delays to the programme, so it is vital that such requirements are identified early (preferably before PDR). This is why it is important to start the PRA early and to ensure that company procedures (incl. the PRA Checklist) are constantly evolving to be useful to the next project so that unexpected design changes can be avoided.
- Mitigate using procedures: These Instructions for Continued Airworthiness (ICA) provide the operator with limitations of use as well as information on how to cope with the failure event when it does occur (see [Chapter 11](#) for more information).
- Declare and obtain authority dispensation: This is a last resort and should not be relied upon. Acceptance is not guaranteed, and its implication on the programme could be dramatic if not accepted.

7.4 Discussion

The PRA can commence early in the design life cycle (via proactive application of a ‘lessons learned’ checklist to influence the requirements management process) and then be repeated to challenge the maturing design (e.g. when the FTA/DD/MA have been completed, or by reexamining the checklist at Design Reviews).

Many PRA events are adequately mitigated by compliance to other regulations (e.g. bird strike requirements of CS25.631) and good design principles (e.g. electrical segregation; dual electrical grounding/earth paths; using LRU adequately qualified to well defined [RTCA/DO-160](#) or [MIL-STD-810](#) requirements; etc.).

7.5 Conclusions

The PRA is an essential part of Common Cause Analyses and complements the ZSA and CMA by considering the vulnerability of the system to any events outside its immediate boundaries.

7.5.1 Advantages

As a hazard identification tool, the PRA has some distinct advantages, which may include:

- Crosses system boundaries, and should identify the fault containment strategies needed.
- Allows effects of nonrelated systems on each other to be evaluated.
- Consider vulnerabilities to outside interferences.
- May address several zones at the same time.

7.5.2 Limitations

The PRA, however, is not without its shortcomings and challenges, which may include:

- Best done at a late design stage to ensure complete picture, and changes required then may be costly to implement. Can be countered by initially analysing drawings and models, but as the project progresses it may require complex calculations, simulation (e.g. trajectories of debris after fan/tyre burst) or tests (e.g. bird strike test, EMI tests, etc.).
- Using a checklist has all the limitations of generic data. It is meant to be thought provoking and stimulate debate.
- The success of the PRA is heavily dependent on good SSWG chairmanship. It requires skilled management to efficiently solicit the participation of a team of knowledgeable people to work in an imaginative and noncritical atmosphere.

References

- EASA, European Aviation Safety Plan, 2014–17, TE.GEN.00400-002, EASA Cologne.
- Kritzinger, D., 2006. Aircraft System Safety: Civil and Military Aeronautical Applications. Woodhead Publishing.
- MIL-STD-810G, 2008. Environmental Engineering Considerations and Laboratory Tests. US Department of Defence Test Method Standard.
- RTCA/DO-160G, 2010. Environmental Conditions and Test Procedures for Airborne Equipment. RTCA Inc., Washington, DC.
- SAE ARP4761, December 1996. Guidelines and Methods for Conducting the Safety Assessment Process on Civil Airborne System and Equipment.

Further reading

- FAA, 2013a. Docket No. FAA-2013-0958. Special Conditions: Boeing Model 777-200, -300, and -300ER Series Airplanes; Aircraft Electronic System Security Protection From Unauthorized Internal Access. Federal Aviation Administration, US Department of Transportation, Washington, DC.
- FAA, 2013b. Docket No. FAA-2013-0910. Special Conditions: Airbus Model A350-900 Airplanes; Isolation or Protection of the Aircraft Electronic System Security From Unauthorized Internal Access. Federal Aviation Administration, US Department of Transportation, Washington, DC.

Zonal Safety Analysis

8

Arguing serves the function of being a zone of familiarity into which you can retreat when you are afraid of making a creative breakthrough.

Gay Hendricks

8.1 Introduction

8.1.1 Background

The acceptance of a System Level 3 or 4 (see Fig. 1.1) failure probability is often based on the assumption that failures are independent [AMC25.1309 App 1 para f]. However, this assumption does not sufficiently recognise the implications of the physical installation (i.t.o. both the act of its installation as well as the surrounding location) which could significantly impair the assumed independence between items (refer, inter alia, [SAE ARP4761](#) App I para 1).

In spite of our best efforts, component proximity vulnerabilities are a fact of life as demonstrated by the following examples.

Example: B707-300 taking off from Aden (15 Aug 1985)

During a climb at FL230, water inadvertently spilled on the autopilot panel. The stabiliser trim wheel started to rotate, and the pilot disengaged the autopilot. The aircraft then began to pitch up and down, and the pilot declared an emergency. Altitude could not be maintained, but the aircraft finally recovered at 1000 ft. The pilot returned and was cleared to land on Runway 26. He overshot the extended centreline and made a normal landing on Runway 08.

There were 8 crew and 65 passengers aboard. 1 crew member and 2 passengers suffered fatal injuries. 1 crew member and 7 passengers suffered serious injuries. 6 crew and 56 passengers escaped with minor or no injuries.

Example: JAL792 taking off from Shanghai (17 Sept 1982)

Nine minutes after take-off, the crew heard a strange noise coming from the lower-middle part of the aircraft. This was immediately followed by a hydraulic low level warning, a hydraulic reservoir air low pressure warning, a complete loss of hydraulic system pressure, abnormal flap position indications, and a complete loss of air brake pressure.

Example: JAL792 taking off from Shanghai (17 Sept 1982)—cont'd

The crew elected to return to Shanghai for an emergency landing. The DC-8 touched down fast on runway 36 and overran and came to rest in a drainage ditch.

The General Administration of Civil Aviation of China identified the probable cause as: ‘The explosion of the air brake bottle damaging 13 hydraulic system tubes and two emergency air brake system tubes, some of which resulted in the failure of extension of flaps and a loss of normal as well as emergency wheel braking, thus increasing the roll after touchdown distance to a value greater than available runway and stop way length. These factors prevented the captain from stopping the aircraft within the runway and stop way confines’.

It is therefore necessary to recognise that such independence may not exist in the practical sense, and specific studies are necessary to ensure that independence in physical location can either be assured or, if not possible, deemed acceptable by incorporation into the probability declaration. The FAA (FAR25.1309), EASA (CS25.1309), and SAE ARP4761 refer to this study as a Zonal Safety Analysis (ZSA).

8.1.2 Aim

The aim of this chapter is to explore the use of the Zonal Safety Analysis (ZSA) as one of the tools in the System Safety process. The ZSA considers the proximity aspects of individual system/item installations and the potential for mutual influence between several systems/items installed in close proximity.

8.1.3 Objectives of a Zonal Safety Analysis

AMC25.1309 Amendment 17 (App 1 para f(1)) advises that the ZSA has the objective of:

“ensuring that the equipment installations within each zone of the aeroplane are at an adequate safety standard with respect to design and installation standards, interference between systems, and maintenance errors. In those areas of the aeroplane where multiple systems and components are installed in close proximity, it should be ensured that the zonal analysis would identify any failure or malfunction which by itself is considered sustainable but which could have more serious effects when adversely affecting other adjacent systems or components.”

8.1.4 Scope of a Zonal Safety Analysis

SAE ARP4761 (para 4.4.1 & I.3) states that the ZSA should be performed on each zone of the aircraft to ensure that the equipment installation meets the relevant safety requirements with respect to:

- **Basic Installation:** The installation should be checked against the appropriate design and installation requirements.
- **Interference between Systems:** The effects of failure of equipment should be considered with respect to their impact on other systems and structures within their physical sphere of influence.

- **Maintenance Errors:** The installation should be checked for vulnerabilities against maintenance errors and their effect on the system or aircraft.
- **Event Independence:** Upon completion of the ZSA, the assessor needs to verify that the design meets any event independence claims made in other parts of the safety assessment (such as in the FTA).

8.2 Conducting the Zonal Safety Analysis

The ZSA is primarily a qualitative analysis comprising four main tasks derived from the objectives stated above:

- Task 1: Preparation of Design and Installation Guidelines
- Task 2: Identification of any potential proximity issues hazards
- Task 3: Inspection of each zone
- Task 4: ZSA reporting

Fig. 8.1 is tailored from SAE ARP4761 Fig. I2 and illustrates a suggested¹ process of events or steps in the ZSA process.

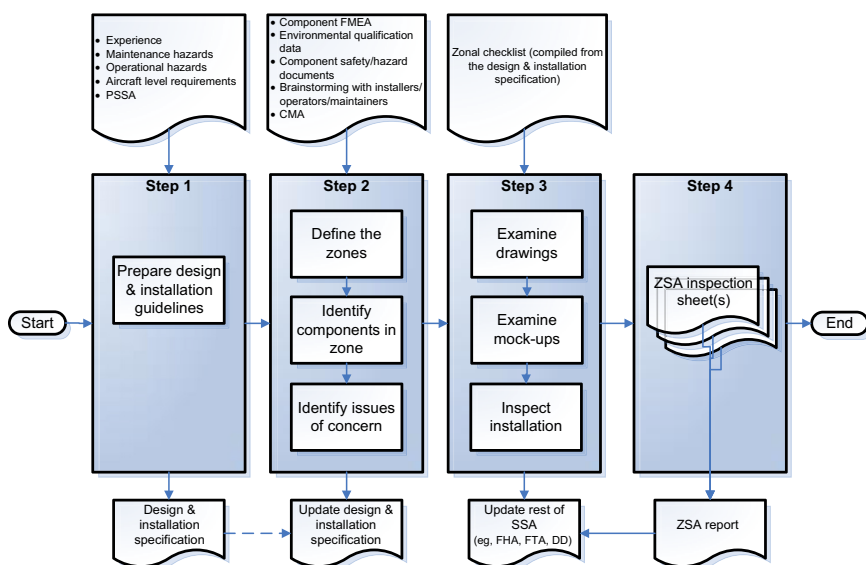


Figure 8.1 ZSA process.

8.2.1 Step 1: preparation of design and installation guidelines

SAE ARP4761 (para I.3.1) advises that this task is independent of the zone itself. The objective is to proactively specify the project's design and installation guidelines which, if followed, should preclude later concerns about common-cause failures.

¹ This simplified process is suggested to support the AMCs to the regulatory requirements. It should be read in conjunction with the referenced source data. Each company needs to decide how to apply the process within their own organisation and its scope of work.

As illustrated in Fig. 8.1, the following list provides useful input sources to this task:

- Existing hazards: Experience from operators and maintainers (and, if available, any Hazard Log) can provide valuable guidelines relating to concerns and desired characteristics of each zone.
- Preliminary System Safety Assessment: The PSSA should have already been conducted and should have highlighted some of the architectural/installation requirements (such as physical and electrical segregation) needed to ensure appropriate levels of redundancy in system functionality during potential failure events.
- Aircraft Level Requirements: For aircraft modifications, guidelines of the baseline aircraft should be used as much as possible to ensure that design principles are not contradicted or that the certification basis is not impacted.

In terms of process efficiency, it may prove beneficial to capture such requirements in company Procedures/Checklists/Design Requirements Manuals, where they may be kept current and appropriately referenced during the design life cycle (as well as for future proactive use).

Specific design and installation guidelines for each system (or ATA chapter) should be derived from experience, in-service data, the relevant PSSA and aircraft level requirements and objectives. As far as possible, their origin should be traceable (in the event of any queries on possible noncompliance) and they should be agreed by all parties concerned.

Example of system-specific design and installation guidelines (ARP4761, p286)

ATA 29 – Hydraulics:

- a) The air conditioning piping should normally be routed above the hydraulics.
- b) When proximity of hydraulic and air conditioning systems is unavoidable, a protective shielding is necessary. The ducting used for cabin air should be inert to hydraulic and other toxic or reactive contaminants likely to come in contact with it.
- c) It should be possible to manually operate valves without use of special tools or dismantling, etc.

8.2.2 Step 2: identification of any potential proximity issues²

As illustrated in Fig. 8.1, this step is divided into three discreet activities³:

- Define the zones
- Identify components in the zone
- Identify potential issues of concern

² Note, We have deliberately not used the term ‘hazard’, due to the difficulties that when assessors do not clearly distinguish between hazards, their causes and the resulting accident emerge (especially when a Hazard Log is also a deliverable). See Kritzinger (Chapter 6) for more on this.

³ See also the EWIS EZAP procedure in the FAA’s AC25-27A and EASA’s AMC20-22.

8.2.2.1 Define the zones

We first need to identify/describe the aircraft zones under consideration. ARP4761 (Fig. I1) shows an example of aircraft zones as defined by the manufacturer. For legacy platforms (where OEM data are not always available), it is often useful to use the zones already defined in the relevant maintenance documentation (see example in Fig. 8.2).

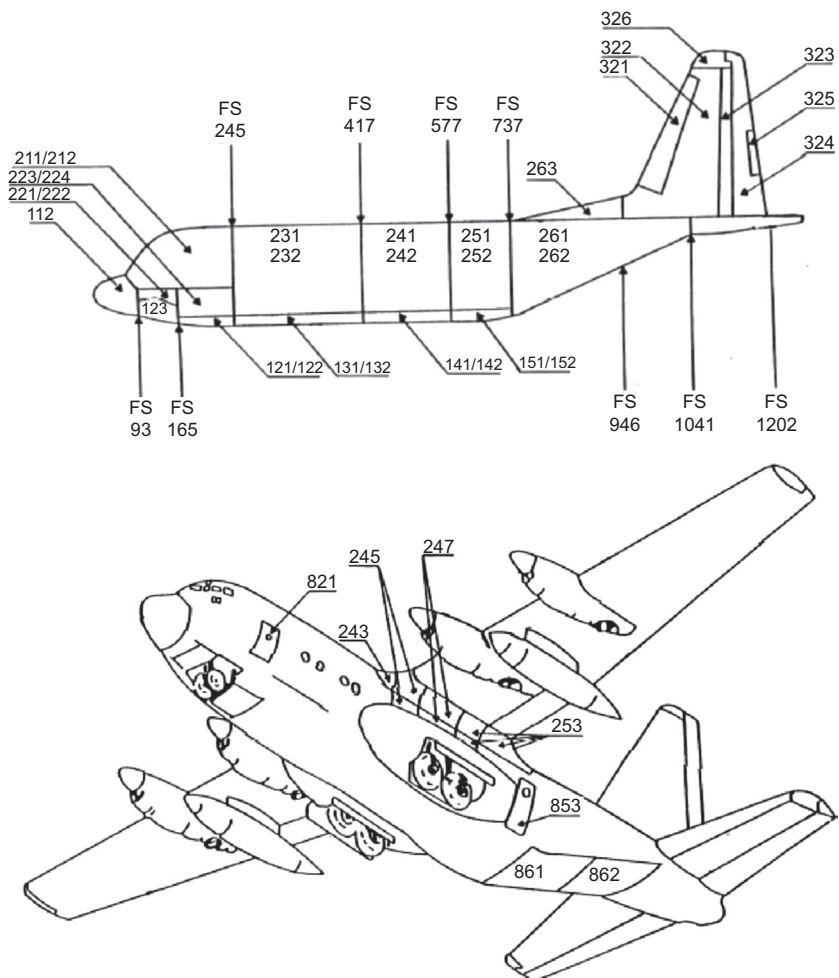


Figure 8.2 Example of aircraft zones.

Example of a zonal description (ARP4761, p288)

The following systems/components are installed in the main landing gear bay.

- a. Blue hydraulic system pipes on the left hand side
- b. Yellow hydraulic system pipes on the right hand side
- c. Green hydraulic system pipes in the lower part
- d. Reservoir of green system
- e. Manifold green system equipped
- f. Power Transfer Unit (PTU)
- g. Maintenance lights
- h. Brake system components
- i. Main Landing Gear
- j. Main Landing Gear free fall system
- k. Slat drive power control unit (PCU)
- l. Flap drive PCU
- m. Flap drive transmission shafts
- n. Gearbox for slat and flap drive shaft
- o. Constant speed motor generator (CSMG) driven by hydraulic power
- p. Auxiliary Power Unit (APU) bleed duct

8.2.2.2 Identify components in the zone

To meet the objectives of the ZSA (see [Section 8.1.3](#)), it is important to identify the equipment and systems within each zone of concern as shown in the example above.

It is useful (and often essential) to describe each of the systems/components in greater detail. Illustrations (where available) should be used liberally to aid assessment and facilitate understanding by the reader of the ZSA report.

8.2.2.3 Identify potential issues of concern

Now that we have defined zones and identified the systems within those zones, we are in a position to assess zonal vulnerabilities resulting from:

- Component external failure modes: This is where a component may physically fail in such a way to have negative impact on its surrounding. Examples include leakage, overheat, burst, etc. These data may be obtained from the FMEA (see [Chapter 5](#)).
- Component intrinsic hazards: These are any hazardous features/attributes of the components which could have negative consequences without the equipment itself failing, but with due consideration of human error and operational conditions. Examples include high operating temperature in closed spaces or exposed terminals in a fuel rich environment. If the scope of the ZSA extends to the consideration of Occupational Health & Safety Hazards, then examples of intrinsic hazards might include high voltages, sharp corners and heavy weights. [Table 8.1](#) provides a useful checklist for the consideration of intrinsic hazards.

- Systemic deficiencies in the design. This is where the design lends itself to a tendency for maintenance errors to be realised. An example might be the potential for incorrect connection of duplicated components due to inadequate cable clipping or insufficient provision of different electrical connector keyways.⁴ These vulnerabilities could be obtained from conducting the CMA (see [Chapter 6](#)) on each zone.

Table 8.1 Intrinsic hazard analysis checklist

Type	Subtype	Hazard
Thermodynamic and fluid hazards	Pressure	High pressure
		Low pressure
		Vacuum
	Temperature	High temperature
		Low temperature
	Heat transfer	Heat radiation
		Heat convection
		Heat conduction
	Fluid	Fluid jet
		Hydraulic shock
Electrical hazards	Voltage	High voltage
		Static electricity
	Current	High current
	Fire	Combustible materials
	Short circuits	FOD, proximity issues
Electromagnetic hazard	Radiation	X-ray, RF, laser, etc
		Nuclear radiation
	Magnetic field	Induced magnetic field
		External magnetic field
		Ionisation

Continued

⁴ Many connectors are keyed, meaning that they have some component which prevents mating except with specific connectors or in a specific orientation. This can be used to prevent incorrect or damaging interconnections, either preventing pins from being damaged by being jammed in at the wrong angle or fitting into imperfectly fitting plugs, or to prevent damaging connections, such as plugging an audio cable into a power outlet. For instance, XLR connectors have a notch to ensure proper orientation, while Mini-DIN plugs have a plastic projection, which fits into a corresponding hole in the socket and prevent different connectors from being pushed together (they also have a notched metal skirt to provide secondary keying).

Table 8.1 Continued

Type	Subtype	Hazard
Chemical hazards	Reactivity	Acidity
		Alkalinity
		Corrosive potential
		Flammability
		Explosive potential
		Hypergolicity
		Pyrophoricity
	Toxicity	High toxicity
		Low toxicity
		Carcinogenic potential
		Asphyxiant
Mechanical hazard	Energy	Potential energy
		Kinetic energy
		Rotation energy
	Mechanical property	Sharpness
		Cutting edge
		Roughness
		Slipperiness
	Stresses	Tension
		Compression
		Friction
	Forces	Brittleness
		Tearing susceptibility
Biological hazards	Microorganisms	Fungus
		Bacteria
		Yeast
Contamination hazards		Dust
		Smoke
		Lubricants

The identification of these zonal issues benefits from multiple sources of input, such as:

- component FMEA (or equivalent), which should identify the effect component failure might have on the surrounding zone;
- environmental qualification data, which will show if the component is susceptible to environmental issues (such as providing a source of ignition in a fuel rich environment);
- brainstorming workshops with installers, operators and maintainers to collect their experience and identify any avoidable hazards resulting from component interference and/or incorrect installation (refer [Section 8.2](#));
- checklist resulting from previous ZSAs (especially ZSAs conducted on that zone);
- event independence claims made in other parts of the System Safety Assessment (such as FTA, ETA, DD, CMA, etc.).

This part of the assessment is documented in the ZSA report, where it:

- proactively mitigates any vulnerabilities by providing specific zonal design and/or installation guidelines (i.e. update Step 1) and
- directs reactive inspection (i.e. input to Step 3) to ensure compliance or address concerns.

8.2.3 Step 3: inspection of each zone

This step in the ZSA process is the validation (typically pre-CDR) and verification (typically during physical integration) exercise of inspecting the zones against:

- the design and installation requirements (i.e. Step 1), and
- the resulting effects on aircraft of any system interference issues identified (i.e. Step 2).

These inspections should start as early as possible (e.g. inspection of drawings to find and address any issues of concern when it is still relatively easy to change the design) and be repeated to be in line with the evolving design or configuration baseline.

Inspection activities should account for the installation process to ensure that assumptions made during modelling do not differ from reality (this is especially true when the system is difficult to install and/or maintain, and technicians require changes which might not be fed back to the safety assessor).

It is often useful to compile a checklist (from Task 1) to guide the inspection process.

Example inspection checklist

For the installation of any equipment in the nose landing gear bay:

- Is it possible for the new equipment to fail in such a way that it prevents the normal operation of the landing gear?
- What effects does the normal operation/failure of the landing gear have on the new equipment?
- Will the new equipment be affected by stones/water thrown up by the gear?
- What effect will a tyre burst have?

The result of the inspections of the zone should be documented for inclusion in the ZSA report.

8.2.4 Step 4: Zonal Safety Analysis report

The results of the ZSA process are captured in a ZSA report, which should be a living⁵ document and which should be updated and issued periodically as the design progresses.

It is recommended that each ZSA report contains the following data:

- How, when and by whom the assessment was made (i.e., mock-ups, aircraft, etc.).
- The correct identification of any equipment that is highlighted as a potential problem.
- Any deviations from the guidelines, or any significant failures, resulting from later interaction of systems or maintenance errors.
- The system/item external failures and their implications should be analysed and listed. References should be given for the source of the failure mode, the rationale for the failure effect on the adjacent system and the reference to the relevant SSA which describes the effect on the aircraft.
- The manner in which problems highlighted by the analysis have been resolved (giving reference to associated documents). Any problem or deviation should be brought to the attention of the responsible design organization and should be considered for design change.
- Recommendations as to how the System Safety Assessment needs to take account of any event independence vulnerabilities (refer [Section 8.2](#)).

8.3 The Case Study

In [Chapter 2](#) we defined a case study for the upgraded Attitude and Altitude Display System in which one safety argument (i.e. Strategy S7) in [Fig. 2.5](#) looked as shown in [Fig. 8.3](#).

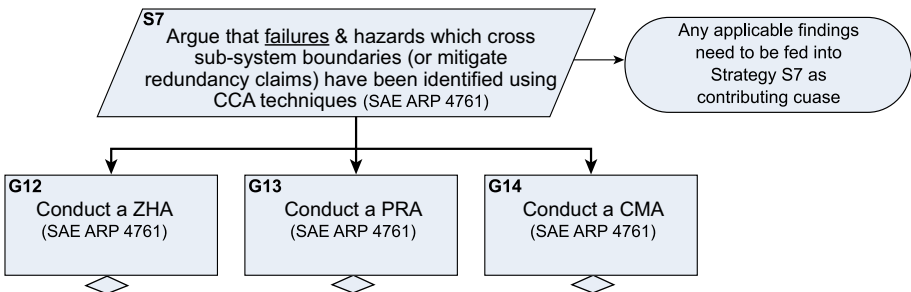


Figure 8.3 Strategy leading to ZSA requirement.

The following sections will use the process in [Fig. 8.1](#) to result in a ZSA report which can be issued as the solution to goal G12.

⁵ SAE ARP4761 (para I.4) advises that records of the analysis should be made on a day-to-day basis.

8.3.1 Step 1: preparation of design and Installation guidelines

Using past experience (e.g. existing hazards) and inputs from aircraft-level requirements and from the PSSA, we can derive general design requirements for the modification as shown in the following example:

Examples of general design and installation guidelines include (refer ARP4761, p285)

- Component Removal and Replacement
 - a) A change of similar but not identical components should not have an unacceptable effect on system performance.
 - b) Any component which could be installed in an incorrect orientation should not produce a problem (e.g., cause a significant reduction in clearance or cause unacceptable stress on any connecting wire, cable, hose, etc.).
 - c) Cross connection of connectors, pipes, etc., shall be prevented.
- Drainage
 - a) Consideration should be given to implications on drainage of incorrect component/equipment installation.
 - b) There should be drainage in those areas or components where accumulation of liquid would be dangerous.
- Maintenance and Servicing
 - a) All ground service connection points should be identified and/or arranged such that it is obvious which fluids should be used or which equipment connected.
 - b) Where possible, the design should allow replacement of items without removal of other equipment, in particular, equipment in other systems. If not, a check of all the involved systems should be made if a risk exists.
 - c) Consideration should be given to any possible hazards that might result from tools, bolts, etc., being inadvertently left on the aircraft.

We can also provide more specific design and installation requirements for the system in each zone, for example:

Example of system specific design and installation guidelines for the avionic upgrade

Avionics Bay (Zone 223/224 in Fig. 8.2)

- a) Ensure physical/spatial segregation of all dual redundant system components to ensure the functional and physical integrity of critical systems due to single point failures.
- b) Electrical equipment controls and wiring must be installed so that operation of any one unit or system of units will not adversely affect the simultaneous operation of any other electrical unit or system essential to the safe operation (FAR25.1353).
- c) Cables must be grouped, routed and spaced so that damage to essential circuits will be minimized if there are faults in heavy current-carrying cables (FAR25.1353).

Example of system specific design and installation guidelines for the avionic upgrade—cont'd

Circuits shall be separated and segregated to minimize the risk of interference during operation and maintain redundant operation.

- d) Avionics equipment bays tend to attract dust, dirt and other contamination. Because of the heat generated by these components and their relatively tightly packed installations, your consideration shall be given to the potential for accumulating combustible material. Forced air ventilation may be required causing lint and dust to be blown about the area and prevent a build-up of dust and lint on the surfaces of components and wiring.

Wiring for Engine Indicating System

- a) If a wire bundle was to be routed along the side of the bleed air duct, a minimum clearance of 1 inch has to be maintained.
- b) If a wire bundle was to be routed along the side of a fuel line, a minimum clearance of 2 inches has to be maintained.

8.3.2 Step 2: identification of any potential proximity issues

As illustrated in [Fig. 8.1](#), this step is divided into three discreet activities:

- Define the zones
- Identify components in the zone
- Identify potential issues of concern

With reference to [Fig. 8.2](#), [Table 8.2](#) captures the proximity issues for Zone 211/212.

8.3.3 Step 3: inspection of each zone

This step in the ZSA process is the validation exercise of inspecting the zones against:

- the design and installation requirements (i.e. Step 1), and
- the resulting effects on aircraft of any system interference issues identified (i.e. Step 2).

These inspections should start as early as possible, and each inspection is recorded in a ZSA Inspection Sheet, an example of which is shown in [Table 8.3](#).

8.3.4 Step 4: Zonal Safety Analysis report

The results of the ZSA process are captured in a ZSA report, which is up-issued as the project progresses and as additional data become available. It is recommended that the ZSA report contains:

- Every single ZSA Inspection Sheet (at its latest revision status) in the Annex.
- A summary of the findings to date, including any open actions/activities. These findings should include all prescribed maintenance activities needed to ensure continued airworthiness.

Table 8.2 Proximity issues in zone 211/212: Cockpit Instrument panel

ID	Component in zone	External failure mode(s)	Intrinsic hazards	Systemic vulnerabilities	Effect on the aircraft	Corrective/preventative action and/or mitigations
1.1	Flight Deck Armour	Velcro attachments could shift during flight	N/A	N/A	Interference with rudder pedals	Ensure adequate Velcro attachments are provided and validate by actually trying to move it once installed Ensure close fitting to aircraft floor contours to prevent any movement
1.2	Flight Deck Armour	N/A	N/A	Armour panels cover lower Pilots and Co-pilots CB panels	Circuit Breakers are not easily assessable during an emergency	Panels can be removed using a Velcro separating tool (retained with the aircraft) to allow access in emergency. Ensure this hazard ins transferred to the operator's Hazard Log
2	RWR display	N/A	Operates at high temperature	N/A	Could cause adjacent instruments to fail prematurely	Test conducted [refer Report Nr 1234] concludes 'Temperatures of the switches reached @ 80°C, but the aft top face of the unit stabilised @ 49°C'. This suggests that the temperature of the unit did not reach a level that would cause adjacent equipment to attain dangerous temperatures as all LRU should be cleared to operate in a temperature range of -40°C to 71°C

Continued

Table 8.2 Continued

ID	Component in zone	External failure mode(s)	Intrinsic hazards	Systemic vulnerabilities	Effect on the aircraft	Corrective/preventative action and/or mitigations
3	Engine indicating system	N/A	N/A	Electrical connectors are identical	Cross connection could cause the wrong engine to be shut down in the case of an emergency	Ensure that all connectors are keyed and properly labelled to prevent cross connection
4	Etc.					

Table 8.3 Zonal safety analysis inspection sheet

Aircraft:	C-130	Zone:	211/212	System	ATA31 ATA34 ATA42	Inspection sheet issue:	1
Date:	1/10/14	Inspected by:	J. Nervous B. Ready N. Happy	Inspection sheet prepared by:	J. Nervous	Inspection sheet authorised by:	D. Boss
Inspected against:	Design Spec 758 Iss1		ZSA Report 758 draft A Table 8.1				
ID	Concern	New issue or non-compliance?	Action/mitigation required	Actionee	Target date	Status	Comments and references
1	Area behind instrument panel is vulnerable to condensation (moisture drip), which could cause corrosion or short circuits in connectors	New issue	Ensure adequate ventilation (forced air cooling) behind instrument panel Ensure that all wiring incorporate drip lips to prevent water from running into connectors	N. Happy		Open	
2	Dual redundant Avionic CPU have electrical power supply routed in same loom	Non-compliance to System Spec 123 (no electrical segregation)	TBD	N. Happy		Open	
3	Etc.						

- Recommendations as to how the System Safety Assessment needs to take account of any event independence vulnerabilities identified.
- Recommendations of talk and intervals to go into the Zonal Inspection Programme (ZIP) and other Instructions for Continued Airworthiness (ICA).

8.4 Discussion

The ZSA Process of [Fig. 8.1](#) can be mapped against the project life cycle, and an example is shown in [Fig. 8.4](#).

8.5 Conclusion

The ZSA is an invaluable tool to use during system integration. However, an understanding of its benefits and limitations is required for its efficient application.

8.5.1 Advantages

As a safety assessment technique, the ZSA has a number of advantages which include:

- Looks at the complex interactions that can occur between high-energy systems and is specifically concerned with their physical position in relation to each other.
- Highlights potential hazards from adjacent nonrelated systems (e.g. heating pipes near sensitive electronic equipment, hot air leaks, drips from pipes, multichannels through same connectors, EMI effects on multichannel configurations, etc.).
- Considers any potential interactions between high-energy sources and sensitive items.

8.5.2 Limitations

The ZSA also has its limitations:

- Best done at a later stage in the design when all equipment can be considered. This means that changes are likely to be expensive.
- Tends to be very subjective, difficult to systematise and is restricted to each specific zone considered.
- Requires system experience. Checklists can be utilised in the process to identify hazards; they can also be used to check that designs comply with certain standards and codes of practice or that protective measures are correctly employed. They are, however, reliant on the knowledge and experience of those persons compiling the lists.

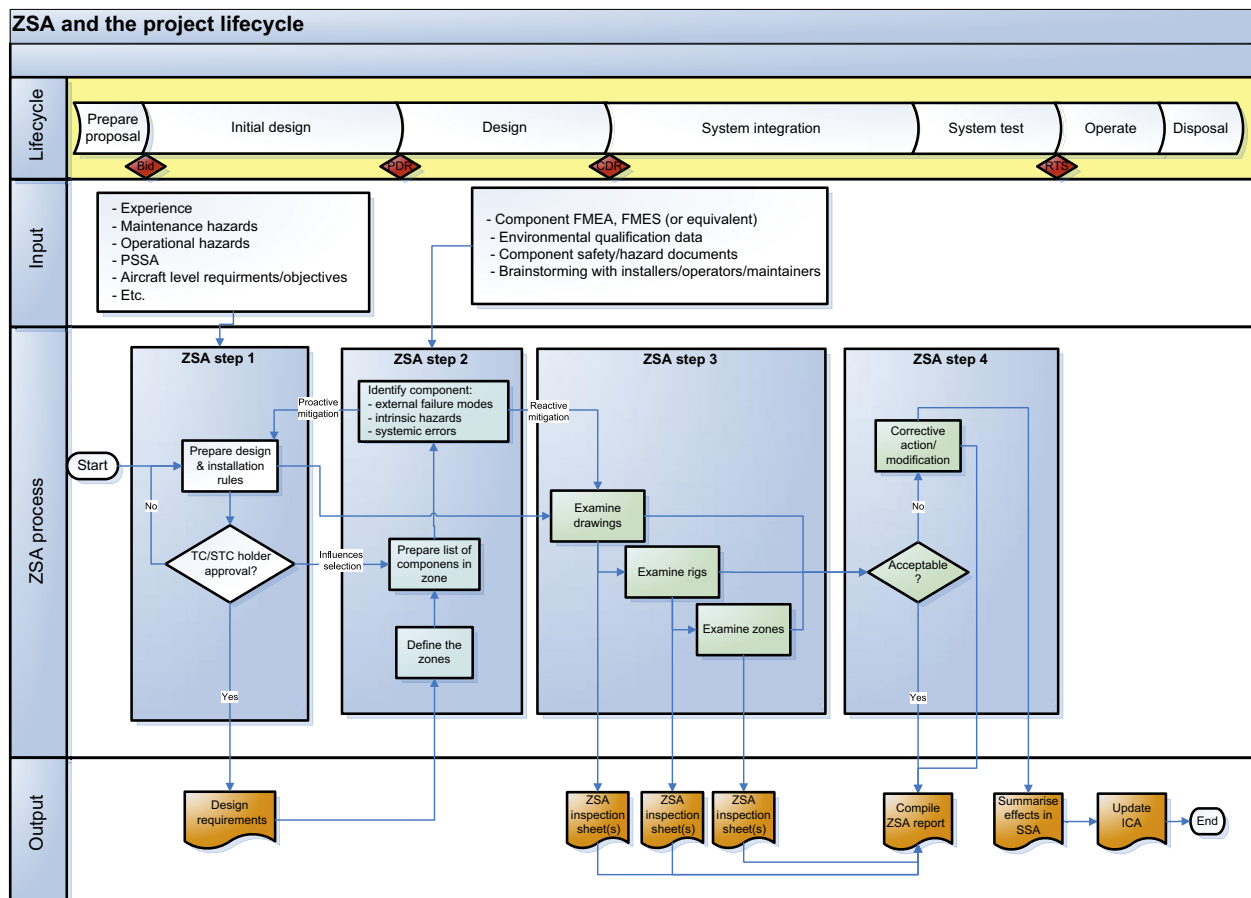


Figure 8.4 ZSA and the project lifecycle.

References

- CS25, July 17, 2015. Certification Specifications and Acceptable Means of Compliance for Large Aeroplanes, Amendment. European Aviation Safety Agency, Cologne.
- FAR25, November 2007. Airworthiness Standards: Transport Category Airplanes, Amendment 25–123. Federal Aviation Authority, Washington.
- SAE ARP4761 Aerospace Recommended Practise, 1996. Guidelines and Methods for Conducting the Safety Assessment on Civil Airborne Systems and Equipment. The Engineering Society for Advanced Mobility Land Sea Air and Space, Warrendale, USA.

Further reading

- DOT/FAA/AR-TN06/17, 2006. Development of an Electrical Wire Interconnect System Risk Assessment Tool. National Technical Information Services (NTIS), Springfield, Virginia. <http://www.tc.faa.gov/its/worldpac/techrpt/artn06-17.pdf>.
- Caldwell, R.E., Merdgen, D.B., 1991. Zonal analysis: the final step in system safety assessment (of aircraft). In: Reliability and Maintainability Symposium. http://ieeexplore.ieee.org/xpl/articleDetails.jsp?tp=&arnumber=154447&url=http%3A%2F%2Fieeexplore.ieee.org%2Fxppls%2Fabs_all.jsp%3Farnumber%3D154447.
- Chiesa, S., Corpino, S., Fioriti, M., Rougier, M., Viola, N., 2012. Zonal Safety Analysis in Aircraft Conceptual Design: Application to Save Aircraft. Department of Mechanical and Aerospace Engineering, Politecnico di Torino, Italy. <http://pig.sagepub.com/content/227/4/714>.
- Xiaolei, L., Jin, T., Tingdi, Z., 2008. An improved zonal safety analysis method and its application on aircraft CRJ200. In: Third International Conference on Availability, Reliability and Security. <http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=4529374&url=http%3A%2F%2Fieeexplore.ieee.org%2Fiel5%2F4529302%2F4529303%2F04529374.pdf%3Farnumber%3D4529374>.

Development Assurance

9

A program which does not work is undoubtedly wrong; but a program which does work is not necessarily right.

Michael A. Jackson [*Principles of Program Design, Academic Press, 1975*]

9.1 Introduction

9.1.1 Background

Before the proliferation of digital technologies [i.e., software (S/W) and complex electronic hardware (CEHW)], the behaviours and properties of a system (or item) were easy to establish (or characterise) by direct inspection, analysis or test. Random failure¹ rates were relatively straightforward to deduce, and these failure rates could be used to feed (e.g., see Table 3.3) traditional system safety methodologies.

However, estimating failure rates for more complex systems is much more difficult² because the failure rates are dominated by systematic failures, which do not lend themselves to prediction like random failure rates do.³

Design and implementation errors made by developers (i.e., humans or tools during system specification, design, development or manufacture), or by human error during operation or maintenance are referred to as systematic faults and failures (Weaver, 2003). Such faults are labelled as systematic because they originate from specific instances of a breakdown in the degree to which these activities are methodical. The result of failing to be systematically methodical is usually that the behaviour of the system under specific contextual circumstances will vary from the behaviour intended

¹ Random failures occur as a result of physical causes typically traceable to the usage of the system. The metrics used to estimate the random failure occurrence and to decide their acceptability is usually based on service experience (e.g., failure rates), predictive probabilities calculations (e.g., piece part FMEA, MTBF evaluation, etc.) and accelerated life testing run by manufacturers ahead of service experience.

² For more complex or integrated systems, adequate testing may either be impossible (because all the system states cannot be determined) or it may be impractical (due to the large number of tests which must be accomplished).

³ The probabilistic approach is not appropriate to evaluate development error occurrence. These 'systematic' [SAE ARP4754A] errors will always occur when the system is exposed to the same circumstances. Due to the nature of systematic errors (i.e., it is difficult to predict the needed circumstances), exhaustive testing and traditional safety analysis techniques are not considered sufficient protection.

or necessary for safety.⁴ The failure that results from the activation of a systematic fault is a systematic failure.⁵

The behavioural nature and probability of such systematic errors of events are not easily predictable or quantifiable in numerical terms because they do not relate to the normal properties of reliability or wear-out typically modelling by failure probability distributions. For this reason modelling their probability density function is very difficult. They relate to a lack of knowledge (and thus uncertainty) in the existence of the fault and the resulting behaviour. Hence, systematic failures are not random. Systematic failures are repeatable though, although knowledge of the internal and external conditions required to repeat them may be difficult to determine for some unintuitive faults.

Highly integrated and complex systems present greater opportunities [refer, inter alia, to [SAE ARP4754A](#) (paragraph 4.1) and RTCA/DO-254 (paragraph 1.0)] for development error (e.g., requirements determination and design errors) and undesirable or unintended effects:

- Although system architectural features (e.g., redundancy, monitoring and partitioning) are used to help prove the safety objectives set in the Functional Hazard Analysis (see Table 3.3), it is practically impossible to guarantee the correctness and completeness of requirements definition or design implementation.
- Any attempt to justify that a complex or highly integrated system is sufficiently error free, solely by means of testing, quickly becomes impractical as the system complexity increases. In such cases, the use of structured processes, including formal controls, in the system development may be used to provide additional evidence to support a practicable level of testing ([SAE ARP4754A](#) paragraph D2).
- There is evidence accumulated from many projects that more operational failures are caused by the misunderstanding of requirements than by the incorrect implementation of requirements. For S/W and complex hardware, the problem may be traced to the development organisation that is responsible for the decomposition of system requirements and their translation into S/W and hardware requirements ([ASSC, 2009](#), p. 24).
- CEHW [such as Field-Programmable Gate Arrays (FPGAs), Application Specific Integrated Circuits (ASICs), and other custom micro-coded devices] are often just as complex as S/W-controlled microprocessor-based systems ([CAST-27](#) paragraph 2a).

For these reasons, a different methodology is needed to assure the behaviours and properties of systems and items. Development Assurance is such a methodology.

Development Assurance provides confidence in the behaviours and properties of a system (or item). It is based on setting objectives and configuration control requirements

⁴ For instance, it is the necessity for the system to behave in a particular way under specific conditions. Under other conditions, the system may behave acceptably.

⁵ A systematic failure should not be confused with systemic failure. A systemic failure relates to something that affects the entire system or organisation [Oxford English Dictionary], and is used to characterise widespread shortfalls with respect to a particular characteristic of a system or organisation. For example, if an organisation chooses to downsize QA such that the function becomes ineffective, then this would be labelled a systemic organisational failure. In recent times, the failing of the finance system across the USA and Europe are from systemic causes. While such a failure could credibly lead to the introduction of systematic faults in any products designed by this organisation, systematic faults can also originate from localised activity shortfalls that are not necessarily systemic.

for the lifecycle evidence (i.e., in both artefact/product⁶ and process). It also provides a process for establishing that the evidence satisfies the desired objectives. The decision that development errors have been sufficiently removed from a product is based on an evaluation of the integrity of the product development process. The rigour of this evaluation is determined by the allocated '*Development Assurance Level*' (DAL), where top-level failure conditions classified as '*Catastrophic*' are assigned the highest level (i.e., DAL A) and those with '*No safety effect*' are assigned the lowest level (i.e., DAL E) as shown in Table 4.2.

9.1.2 Aim of this chapter

The aim of this chapter is to describe the purpose and role of Development Assurance and to outline the general approach to satisfy the allocated Functional Development Assurance Level (FDAL) or Item Development Assurance Level (IDAL).

9.1.3 Objectives of this chapter

The objective of this chapter is to provide an explanation of the fulfilment of DALs at a summary level to allow programme managers (and the average system safety assessor) to have a basic understanding of how Development Assurance is accomplished. In doing this, this chapter provides explanation of:

- The general principles of Development Assurance that apply to Development Assurance under [SAE ARP4754A](#), RTCA/DO-178C and RTCA/DO-254.
- The relationship between the FDAL and IDAL.
- The DAL deliverables (called '*Data Items*') which need to be managed, and what is required of them (i.e., the '*objectives*' they should meet to satisfy the allocated DAL).

9.1.4 Scope of this chapter

The scope of this chapter does not extend to the detail application of the IDAL approach to:

- S/W, as the intent of this chapter is not to duplicate the specialist detail in RTCA/DO-178C.
- CEHW, as the intent of this chapter is not to duplicate the specialist detail in RTCA/DO-254.

9.1.5 Guidance material pertaining to Development Assurance

Fig. 1.4 illustrates the key regulatory and advisory documentation which support the development of Safety Assessments. As indicated in the bottom part of that illustration, there are three guidelines (these have become widely acceptable standards)

⁶ 'Product' and 'Artefact' are used interchangeably here and refer to the physical design output (i.e., the hardware or S/W code). Some standards (such as RTCA/DO-178C) use the word 'product' to describe these outputs, while some regulations (such as EASA Part 21) use the word 'product' to describe three things only: an Aircraft, Engine or Propellers (these are the only things for which you can get a Type Certificate, everything else fitted to a product is a 'part' or 'appliance'. The problem is that Part 21 and the S/W standards are written by different people using the same terminology but with different meanings.

which specifically address Development Assurance, which are explored in the sub-sections below:

9.1.5.1 SAE ARP4754A

Titled '*Guidelines for Development of Civil Aircraft and Systems*', this document addresses the development process of aircraft systems and the Development Assurance processes of (a) validation of requirements and (b) verification of the design implementation.

This document does not include specific coverage of S/W or electronic hardware, as these are covered by RTCA/DO-178C and RTCA/DO-254, respectively.

9.1.5.2 RTCA/DO-178C

Titled '*S/W Considerations in Airborne Systems and Equipment Certification*', this document addresses the production of S/W for airborne systems and equipment. It does not define firmware explicitly, although the approach may be applied⁷ to firmware.

The RTCA/DO-178C document provides the following information:

- Definition of S/W Development Assurance objectives as they relate to the S/W lifecycle processes, including the relationship between DAL, applicable objectives, independence, and S/W lifecycle data.
- Guidance on the Development Assurance activities that provide a means of satisfying those objectives.
- Guidance on additional considerations that apply in certain situations.

Advisory Circular AC 20–115C specifies DO-178C as an acceptable means, but not the only means, for receiving regulatory approval for S/W in systems or equipment being certified under a Technical Standard Order (TSO) Authorization, Type Certificate (TC) or Supplemental Type Certificate (STC). Even though DO-178 was written as a guideline, it has become the standard practice within the industry, with DO-178 (currently at revision C) officially recognised as a de facto international standard by the International Organisation for Standardisation (ISO). [Kerrel and Ferrell \(2001\)](#) advise that most applicants use DO-178 to avoid the work involved in showing that other means are equivalent to DO-178.

See Annex B to this chapter for more information on the approaches to S/W Assurance.

9.1.5.3 RTCA/DO-254

Titled '*Design Assurance Guidance for Airborne Electronic Hardware*', this document addresses hardware such as Circuit Board Assemblies; custom micro-coded components [e.g., ASICs and Programmable Logic Devices (PLDs)]; integrated technology

⁷ Firmware should be explicitly classified as either S/W or hardware and addressed by the applicable processes. This means it may be classified as S/W and RTCA/DO-178C applied or hardware and RTCA/DO-254 applied.

components (e.g., hybrids and multichip modules) as well as Line Replaceable Units (i.e., LRUs containing any of these technologies). It does not define firmware explicitly, although the approach may be applied to firmware.

The purpose of DO-254 is to establish a framework for the development of such hardware to ensure, by way of a standard framework, that the resulting hardware devices perform their intended function under all foreseeable conditions.

The focus of DO-254 is design assurance, involving a number of key principles including extensive and detailed planning, establishing a requirements-driven flow, ensuring a repeatable, structured, controlled and well-documented process, holding numerous internal reviews and external audits, and having thorough verification throughout the process.

The RTCA/DO-254 document provides the following information:

- Definition of hardware Design Assurance objectives.
- Description of the basis of the objectives to help ensure correct interpretation of the guidance.
- Provide descriptions of the objectives to allow the development of means of compliance with this and other guidance.
- Provide guidance for Design Assurance activities to meet the Development Assurance objectives. It does allow flexibility in choice of processes necessary to meet the objectives of this document as new process technologies become available.

See Annex A to this chapter for more information on the approaches to Hardware Assurance.

All three of these documents (i.e., [SAE ARP4754A](#), RTCA/DO-178C and RTCA/DO-254) utilise the concept of DAL Objectives, which are explored in more detail below:

- The terms ‘*Design Assurance*’ or ‘*Development Assurance*’ and ‘*S/W Level*’ are used interchangeably, for instance:
 - ARP4754A (paragraph 3.1) refers to ‘*Development Assurance*’, which ‘*establishes confidence that system development has been accomplished in a sufficiently disciplined manner to limit the likelihood of development errors that could impact aircraft safety*’. Development Assurance is defined as ‘*a process involving specific planned and systematic actions that together provide confidence that errors or omissions in requirements or design have been identified and corrected to the degree that the system, as implemented, satisfies applicable certification requirements*’.
 - RTCA/DO-254 (App C) refers to ‘*Design Assurance*’, which are ‘*all of those planned and systematic actions used to substantiate, at an adequate level of confidence, that design errors have been identified and corrected such that the hardware satisfies the application certification basis*’.
 - RTCA/DO-178C (paragraph 2.3) uses the term ‘*S/W Level*’, which in practice inherits the ARP 4754A definition in its application of Development Assurance.
- The term ‘*Assurance*’ involves the application of techniques and methods, commensurate with the worst credible failure, to generate specific evidence with respect to attributes of the development lifecycle and product of the following ([McDermid, 2012](#)):
 - Requirements Validity – does the component have the right behaviours?
 - Requirements Satisfaction – are the required behaviours implemented in the configuration item?

- Requirements Traceability – are the behaviours of the S/W fully accounted for and are there any additional behaviours?
- Non-Interference – are any of the behaviours necessary for safety interfered with by other behaviours of the product?
- Configuration Consistency – does the evidence produced throughout the lifecycle have traceability to the delivered product?
- Design Integrity – have reasonable steps been employed to adopt techniques and methods that contribute to design integrity rather than providing opportunities for vulnerabilities to be introduced that might be a source of counter evidence to requirement validity or satisfaction?
- The term ‘*Objectives*’ refers to the measurable outputs which should be met in the evidence (or deliverables) needed to demonstrate compliance to the requirements of these standards.
 - In ARP4754A and RTCA/DO-178C, this information is provided in specific tables contained in Annex A of both documents.
 - In RTCA/DO-254, the information is embedded within the main body of the standard, although the information can be organised into a similar structured table.

In [Section 9.2](#), the ‘*objectives*’ are copied from these documents to be reallocated against each of the steps in [Fig. 9.1](#).

In essence, the DAL Objectives provide a way for the certification authority to evaluate the applicant’s product lifecycle data⁸ and processes⁹ to establish if the data supports the assertion that the system is assured. It provides a measureable way to deal with evidence that may have qualitative properties (e.g., human review evidence) and/or quantitative properties (e.g., test results of an algorithm). By using an objective-based approach, these guidelines/standards provide the developer with some flexibility in choosing methods and techniques that best suit their needs, provided that the evidence (i.e., output or deliverables or data items) satisfies these objectives.¹⁰

⁸ *Lifecycle Data*: Lifecycle processes include the planning process, the S/W development processes (requirements, design, code and integration) and the integral processes (verification, configuration management, S/W QA and certification liaison). For instance, RTCA/DO-178C Section 11 addresses S/W lifecycle data, which includes the Plan for S/W Aspects of Certification (PSAC), the S/W Development Plan (SDP), the S/W Verification Plan (SVP), the S/W Configuration Management (SCM) Plan, the S/W Quality Assurance Plan (SQA) and so on. Kerrel and Ferrell (2001) advise that DO-178B discusses the S/W lifecycle processes and transition criteria between lifecycle processes in a generic sense without specifying any particular lifecycle model.

⁹ *Processes*: Kerrel and Ferrell (2001) advise that DO-178B specifies the information flow between system processes and S/W processes. The focus of the information flow from the system process to the S/W process is to keep track of requirements allocated to S/W, particularly those requirements that contribute to the system safety. The focus of information flow from the S/W process to the system process is to ensure that changes in the S/W requirements, including the introduction of derived requirements (those not directly traceable to a parent requirement), do not adversely affect system safety.

¹⁰ *Objectives*: It is important not to confuse S/W assurance standards (such as RTCA/DO-178/ED-12B) with S/W development and lifecycle standards (such as MIL-STD-498, DoD-STD-2167A and IEEE12207). S/W assurance standards define objectives that should be satisfied to provide confidence in the behaviour of the S/W with respect to safety objectives. S/W development standards are more synonymous with a shopping list of documentation, activities and processes that S/W developers might use to undertake S/W. However, unlike assurance standards, development standards are not outcome or objective based. For example, a S/W development (or sometimes called a S/W lifecycle standard) might identify unit testing as a relevant technique, whereas a S/W assurance standard would instead provide an objective that provides a measure of when sufficient unit testing has been completed.

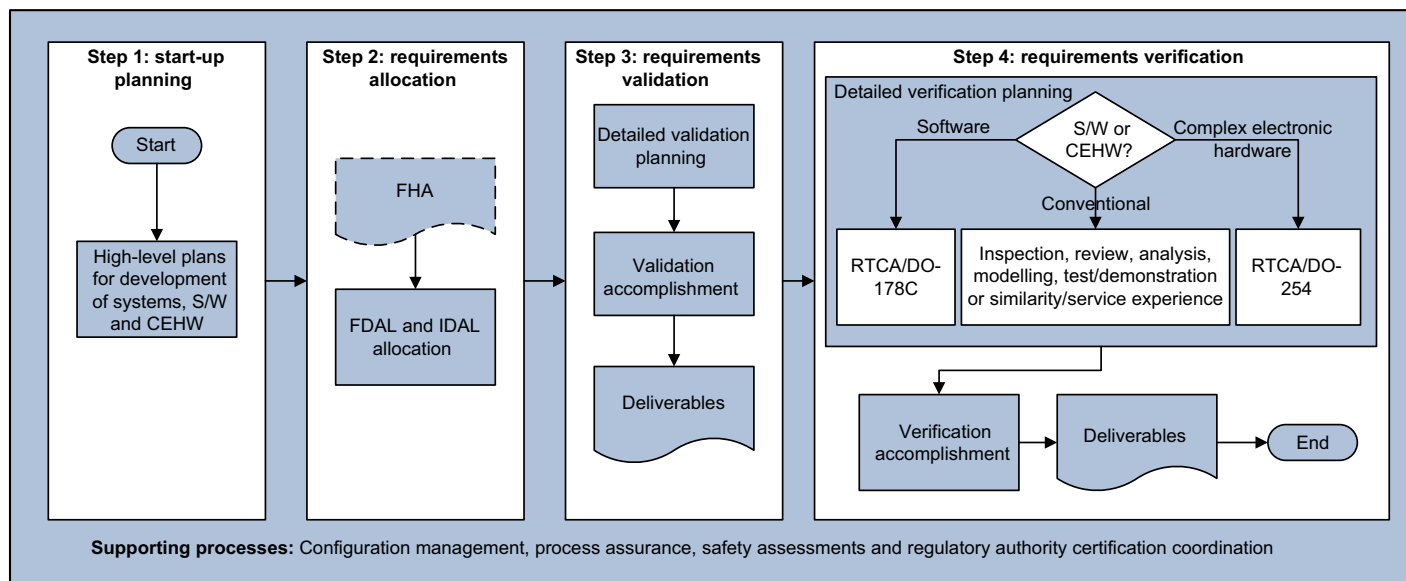


Figure 9.1 The Development Assurance process.

9.2 The Development Assurance process

Fig. 9.1 provides a simple illustration of the Development Assurance process, and each step is discussed in greater detail below.

The application of Fig. 9.1 to the V-diagram of Fig. 1.3 can be summarised as follows:

- Start-up Planning (**Step 1**), which will encompass all plans to drive the development through Steps 2 to 4 to accomplish certification.
- Requirements Allocation (**Step 2**), which includes:
 - The refinement and allocation (or ‘*cascading*’ to use the nomenclature of Fig. 1.3) of safety requirements from aircraft down to item level.
 - FDAL allocation, from aircraft functions down to item function level.
 - IDAL allocation at the bottom left side of the V-diagram.
- Validation (**Step 3**), where we confirm that all requirements are [refer, inter alia, RTCA/DO-178C (paragraph 11)] unambiguous, complete, verifiable, consistent, modifiable and traceable (i.e., see arrows going back up to aircraft level on the left side of the V-diagram in Fig. 1.3).
- FDAL and IDAL Verification (**Step 4**) are all the requirements satisfaction activities performed on the right side of the V-diagram. These are accomplished in different ways:
 - For CEHW, we do verification via ARP 4754A (to satisfy the FDAL at each level of integration) and DO-254 (to satisfy the IDAL at the bottom right of the V-diagram).
 - For S/W, we do verification via ARP 4754A (to satisfy the FDAL at each level of integration) and DO-178 (to satisfy the IDAL at the bottom right of the V-diagram).
 - For simple hardware (see Annex A1), we do verification via inspection, tests and analysis to satisfy the applicable FDAL verification objectives of ARP 4754A.
- All four steps are underpinned by supporting procedures (which should be mature in any approved Design Organisation) for:
 - Configuration of H/W, S/W and all supporting data.
 - Process Assurance, that is all associated quality assurance (QA) and quality control processes.
 - Safety Assessment process, which drives the DAL allocation and the required derived system behaviours.
 - Regulatory Certification Liaison process, when determining the required level of involvement (LOI) of the accepting authority (be it the regulator or the system integrator).

In the context of this summary, the following subsections explore each element of Fig. 9.1 and endeavour to match the ARP4754A, RTCA/DO-178C and RTCA/DO-254 outputs to each of these four steps. Although the following tables might lead to duplication in some data items, this purpose here is to:

- better support the approaches communicated in Figs 12 and 13 of ARP4754A;
- present the data against the four-step process to facilitate better understanding by Programme Managers.¹¹

¹¹ See www.aircraftsystemsafety.com for an up-to-date version of these tables in MS Excel format. These can be downloaded and sorted and/or re-allocated as required.

9.2.1 Step 1: start-up planning

Benjamin Franklin is known to have said “*If you fail to plan, you plan to fail*” and the sentiment applies especially to Development Assurance. Therefore, in Step 1 the Programme Managers need to ensure that time is spent to provide the high-level planning required to drive all the activities needed in all the following steps to ultimately achieve certification.

From the guiding standards we can extract start-up planning related outputs as follows:

- The system wide requirements from [SAE ARP4754A](#) (Appendix A) can be extracted into [Table 9.1](#).
- The S/W requirements from RTCA/DO-178C (Appendix A) can be extracted into [Table 9.2](#).
- The CEHW requirements from RTCA/DO-254C can be extracted into [Table 9.2](#).

9.2.2 Step 2: requirements allocation

Requirements are managed via two distinct processes:

- Assignment of the DAL, which will specify the lifecycle data required to support certification.
- The management of allocated and derived safety requirements in support of the V-model in Fig. 1.3.

These two interrelated processes are explored in more detail below:

9.2.2.1 Assign the Development Assurance level

The process starts with the determination of functional failure severity, which is captured in the FHA process (see [Chapter 3](#)).

The severity is then allocated a FDAL target using Table 3.3, which is then cascaded down the system architecture to item level using the approach described in Section 4.2.3.3 (Table 4.1). Each item is then allocated¹² an FDAL and an IDAL:

- FDAL controls the functional requirements (both allocated and derived) and is subject to the verification and validation (V&V) processes of ARP 4754A.
- IDAL controls the specific item’s development processes (i.e., DO-178 processes for S/W and DO-254 processes for CEHW).

For S/W and electronic hardware, partitioning analysis (temporal¹³ and spatial¹⁴ analysis) may be used to justify a lower DAL for a portion of the item. Partitioning is a means for providing isolation between components to contain and/or isolate faults. Partitioning may be achieved:

- within the system architecturally by allocating unique target hardware and hardware resources to each component.
- within the component architecture to allow multiple S/W or hardware items to run within the same hardware platform.

¹² See ARP4754A (paragraph 5.2.2) for more information.

¹³ Temporal partition is assurance that two components do not interfere with each other’s allocated timeslots.

¹⁴ Spatial partition is assurance that two components do not interfere with each other’s resources (i.e., memory, processing, etc.).

Table 9.1 **SAE ARP4754A** start-up planning

Objective		Applicability by FDAL					Output/associated evidence		Configuration control category by FDAL				
No.	Description	A	B	C	D	E	Data item	Refs	A	B	C	D	E
1.1	System development and integral processes are defined	R	R	R	R		Certification Plan	5.8.1 5.8.4.1	①	①	①	①	①
1.1	System development and integral processes are defined	R	R	R	R	N	Safety Program Plan	3.1 5.1.5 App B	②	②	②	②	
1.1	System development and integral processes are defined	R	R	R	R	A	Configuration Management Plan	5.6.2.1	②	②	②	②	
1.1	System development and integral processes are defined	R	R	R	R	N	Process Assurance Plan	5.7.2	②	②	②	②	
1.2	Transition criteria and interrelationship among processes are defined	R	R	R	A	N	Plans in Objective 1.1	3.2	②	②	②	②	

A=as negotiated for certification.

N= not required for certification.

R= recommended for certification.

R*= recommended for certification with process independence.

Each of the outputs (or deliverables) also has configuration management objectives:

- ① Requires Configuration Identification, Change Control, Configuration Index, Archiving/Retrieval, Configuration Baseline Establishment and Problem Reporting.
- ② Does not require Configuration Baseline Establishment and Problem Reporting.

Table 9.2 RTCA/DO-178C start-up planning

Objective		Applicability by IDAL				Output/associated evidence		Configuration control category by IDAL			
No.	Description	A	B	C	D	Data item	Refs	A	B	C	D
A-1	The activities of the S/W life-cycle processes are defined.	○	○	○	○	Plans for S/W Aspects of Certification	11.1	①	①	①	①
A-1	The activities of the S/W life-cycle processes are defined.	○	○	○	○	S/W Development Plan	11.2	①	①	②	②
A-1	The S/W lifecycle(s), including the interrelationships between the processes, their sequencing, feedback mechanisms and transition criteria, is defined.	○	○	○		Plans for S/W Aspects of Certification	11.1	①	①	①	
A-1	The S/W lifecycle(s), including the interrelationships between the processes, their sequencing, feedback mechanisms and transition criteria, is defined.	○	○	○		S/W Development Plan	11.2	①	①	②	
A-1	S/W lifecycle environment is selected and defined.	○	○	○		Plans for S/W Aspects of Certification	11.1	①	①	①	
A-1	S/W lifecycle environment is selected and defined.	○	○	○		S/W Development Plan	11.2	①	①	②	

Continued

Table 9.2 Continued

Objective		Applicability by IDAL				Output/associated evidence		Configuration control category by IDAL			
No.	Description	A	B	C	D	Data item	Refs	A	B	C	D
A-1	Additional considerations are addressed.	○	○	○	○	Plans for S/W Aspects of Certification	11.1	①	①	①	①
A-1	Additional considerations are addressed.	○	○	○	○	S/W Development Plan	11.2	①	①	②	②
A-1	S/W development standards are defined.	○	○	○		S/W Requirements Standards	11.6	①	①	②	
A-1	S/W development standards are defined.	○	○	○		S/W Design Standards	11.7	①	①	②	
A-1	S/W development standards are defined.	○	○	○		S/W Coding Standards	11.8	①	①	②	

Blank – Satisfaction of objective is at applicant's discretion.

- The objective should be satisfied with independence (intellectual, not organisational).
- The objective should be satisfied.

Each of the outputs (or deliverables) also has configuration management objectives:

- ① Requires Configuration Identification, Traceability, Change Control, Retrieval, Protection against Unauthorised Changes, Data Retention, Baselines, Problem Reporting, Change Review, Configuration Status Accounting, Media Selection/Refreshing/Duplication, Release.
- ② Does not require Baselines, Problem Reporting, Change Review, Configuration Status accounting, Media Selection/ Refreshing/ Duplication and Release.

Table 9.3 RTCA/DO-254 start-up planning

Objective		Applicability by IDAL				Output/associated evidence		Configuration control category by IDAL			
No.	Description	A	B	C	D	Data item	Refs	A	B	C	D
4.1	The hardware design lifecycle processes are defined.	●	●	○	○	Plan for Hardware Aspects of Certification	10.1.1	HC1	HC1	HC1	HC1
4.1	The hardware design lifecycle processes are defined.	●	●	○	○	Hardware Design Plan	10.1.2	HC2	HC2	HC2	NA
4.1	Standards are selected and defined.	●	●	○	○	Requirements Standards	10.2.1	HC2	HC2	NA	NA
4.1	Standards are selected and defined.	●	●	○	○	Hardware Design Standards	10.2.2	HC2	HC2	NA	NA
4.1	Standards are selected and defined.	●	●	○	○	Verification and Validation Standards	10.2.3	HC2	HC2	NA	NA
4.1	Standards are selected and defined.	●	●	○	○	Requirements Standards	10.2.1	HC2	HC2	NA	NA
4.1	Standards are selected and defined.	●	●	○	○	Hardware Design Standards	10.2.2	HC2	HC2	NA	NA
4.1	Standards are selected and defined.	●	●	○	○	Validation and Verification Standards	10.2.3	HC2	HC2	NA	NA
4.1	The hardware development and verification environments are selected or defined.	●	●	○	○	Plan for Hardware Aspects of Certification	10.1.1	HC1	HC1	HC1	HC1

Continued

Table 9.3 Continued

Objective		Applicability by IDAL				Output/associated evidence		Configuration control category by IDAL			
No.	Description	A	B	C	D	Data item	Refs	A	B	C	D
4.1	The hardware development and verification environments are selected or defined.	●	●	○	○	Hardware Design Plan	10.1.2	HC2	HC2	HC2	NA
4.1	The means of compliance of the hardware design assurance objectives, including strategies identified using guidance in Section 2.3.4, are proposed to the certification authority.	●	●	○	○	Plan for Hardware Aspects of Certification	10.1.1	HC1	HC1	HC1	HC1
4.1	The means of compliance of the hardware design assurance objectives, including strategies identified using guidance in Section 2.3.4, are proposed to the certification authority.	●	●	○	○	Hardware Design Plan	10.1.2	HC2	HC2	HC2	NA

Blank – Satisfaction of objective is at applicant’s discretion.

- The objective should be satisfied with independence (intellectual, not organisational).
- The objective should be satisfied.

Each of the outputs (or deliverables) also has configuration management objectives:

- HC1: Requires Configuration Identification, Baseline Traceability, Change Control, Retrieval, protection against Unauthorised Changes, Data Retention, Baselines, Problem Reporting, Change Review, Media Selection/Refreshing/Duplication and Release.
- HC2: Does not require Baselines, Problem Reporting, Change Review, Media Selection/Refreshing/Duplication, and Release.

Partitioning should ensure that:

- A partitioned component cannot contaminate (spatially) another partitioned component's implementation code, input/output (I/O) or data storage areas. Note, however, that a partitioned component may be allowed to consume shared resources (temporally) only during its scheduled period of execution.
- Failures of S/W or hardware unique to a partitioned component cannot cause adverse effects on other partitioned components.
- Any S/W or hardware providing the underlying mechanisms of partitioning should have the same or higher DAL as the highest level assigned to any of the partitioned components.
- Any S/W or hardware providing partitioning should be assessed by the system safety assessment (SSA) process to ensure that it does not adversely affect safety.
- S/W architecture may be used to limit the applicability of higher DALs by limiting the extent of the system that deals with the more severe failure modes, and providing partitioning between these more critical areas and the less severe failure mode contributors.

9.2.2.2 Requirements allocation

Requirements describe the necessary functions and features of the system we are to conceive, design, implement and operate. They are often organised hierarchically:

- At a high level, requirements focus on what should be achieved, not how to achieve it.
- Requirements are specified at every level, from the overall system to each hardware and S/W component.

Requirements Engineering¹⁵ (also known as System Engineering) is the process of eliciting individual stakeholder requirements and needs and developing them into detailed, agreed requirements documented and specified in such a way that they can serve as the basis for all system development and certification activities. There are three core activities associated with robust Requirements Engineering, which are requirements allocation (see the following sections), requirements validation (see Step 2) and requirements verification (see Step 3). These are illustrated in Fig. 1.3.

Requirements Allocation is the process of ensuring that all requirements are assigned to each level of system integration. All requirements (including the allocated FDAL) are cascaded down the system hierarchy and, along with the derived equipment, provide the contents of the applicable system-level specifications (see Fig. 1.3).

¹⁵ For more requirements engineering, see

- NASA Systems Engineering Handbook (NASA/SP-2007-6105), specifically
 - Section 4.2 (pp. 40–48) – Technical Requirements Definition
 - Section 6.2 (pp. 131–135) – Requirements Management
 - Appendix C (pp. 279–281) – How to write a good Requirement
 - Appendix D (pp. 282–283) – Requirements Verification Matrix
- Systems Engineering Handbook, International Council of Systems Engineering (INCOSE) <http://www.incose.org/practice/techactivities/wg/rqmts/ISO/IEC 15288> (IEEE STD 15288-2008).

With respect to requirements allocation to S/W/hardware, the identification and allocation of behavioural safety requirements¹⁶ in the context of the system are fundamental to the realisation of acceptably safe systems in aircraft. There are numerous approaches to identify these specific behaviour safety requirements, for instance:

- Tailoring of safety requirements sources from generic lists, such as those in [Leveson \(1995\)](#) and the DoD's [Software System Safety Handbook \(2010\)](#). The purpose here is to use the documented experience of those before us to determine if we do not know that we do not know, and to prompt us on what requirements might be relevant.
- Specific Software/Hardware Safety Analysis targeting the specific application in the context of its intended system and operating environment, such as techniques identified in [Leveson \(1995\)](#), the DoD's [Software System Safety Handbook \(2010\)](#), [Pumphrey \(1999\)](#) and Defence Standard 00-58.

Whichever approach is used, it is recommended that the systems engineer go about the analysis systematically (which means using structured techniques to avoid omissions or errors) to help understand the required properties of the system in its explicit context, as this may be different to what others have done before. The developer needs to achieve unambiguous expression, communication and traceability of safety requirements, using methods such as those described by [Knight \(2007\)](#). The intent is to make sure those responsible for reading, writing, translating, manipulating and implementing the S/W safety requirements unambiguously understand them, and systematically account for any refinement to them.

These requirements can then all be captured (or modelled¹⁷) using tools such as DOORs, 3SL's Cradle, Vector Cast's predicate table features and so on. These tools not only allow us to capture the requirements effectively, but have significant Validation (see Step 2) and Verification (see Step 3) benefits.

9.2.3 Step 3: requirements validation

Validation is accomplished during the Functional Development Phase, and it is the determination that the requirements for a product are correct and complete. It can be summarised in the question: *'Are we building the right thing?'* With reference to the left hand side of the V-diagram in Fig. 1.3, Step 2 is all about validating that the proposed solution (system or item) will indeed meet all the requirements (e.g. safety, contractual, regulatory, derived imposed on it).

¹⁶ 'Behavioural safety requirements' means those safety requirements that specify the need for the behaviour of the system to be constrained to a valid set of behaviours, usually those associated with achieving the absence or handling of failure conditions.

¹⁷ This type of modelling may be done at the system, hardware or S/W levels – most tools support hierarchies of items and requirements.

Validation consists of 3 phases: The planning phase, the doing (or accomplishment) phase and the delivery (of evidence) phase as explored in more detail below:

9.2.3.1 *Validation planning*

Requirements for functions which have been allocated to each level in the system hierarchy (down to items) need to be validated at each level of integration, and the rigour of this validation process is defined via the FDAL allocated to it.

From the guiding standards we can extract validation planning related outputs as follows:

- The system wide requirements from [SAE ARP4754A](#) (Appendix A) can be extracted into [Table 9.4](#).
- The S/W requirements in RTCA/DO-178C (Appendix A) does not specifically address ‘Validation Planning’, but the outputs in [Table 9.5](#) could be made applicable to this phase.
- RTCA/DO-254 could have the CEHW ‘Validation Planning’ related outputs extracted into [Table 9.6](#).

9.2.3.2 *Validation accomplishment*

Validation is accomplished when we have ensured that:

- Requirements are complete (i.e., sufficient and no omissions).
- Requirements are necessary (i.e., limit the potential for unintended functions in the system or for unintended functions to be induced in interfacing systems).
- Requirements are accomplishable (i.e., not ambiguous or incorrect).
- Assumptions are justified and validated.
- Derived requirements are justified and validated.
- All requirements are traceable.¹⁸

Requirements traceability is the explicit association between high-level requirements, and low-level requirements such that:

- the behaviours of the S/W and/or hardware can be systematically accounted for in the implementation;
- the basis for any additional code or design that might provide additional behaviours can be identified (and subject to appropriate analysis and verification).

¹⁸ ‘Traceability’ refers to the establishment of connections from a ‘requirement’ at a certain development level (e.g., from a system, hardware or S/W ‘requirement’) possibly through intermediate levels to the test(s) that verify its implementation, and the design decompositions and representations down to the hardware and/or S/W by which the requirement is implemented. DO-254 paragraph 6.1 advises that derived requirements that are not traceable to a higher-level requirement should be validated against the design decision from which they are derived. It is a common problem that traceability cannot be demonstrated because there are problems with the configuration management of the various documents involved such as requirements, design, tests, test environment and implementation. A traceability tool can perform checks to support the claim that the traceability is provided for each requirement. In general, the lack of an automated traceability tool can affect the programme. It is generally more difficult to maintain and demonstrate full traceability when the changes are performed ‘by hand’. In these cases, the effort to demonstrate complete traceability at key milestones can be considerable.

Table 9.4 SAE ARP4754A validation planning

Objective		Applicability by FDAL					Output/associated evidence		Configuration control category by FDAL				
No	Description	A	B	C	D	E	Data item	Refer	A	B	C	D	E
1.1	System development and integral processes are defined	R	R	R	R	N	Validation Plan	5.4.2.a 5.4.7.1	②	②	②	②	
1.2	Transition criteria and interrelationship among processes are defined	R	R	R	R	N	Plans in Objective 1.1	3.2	②	②	②	②	
2.1	Aircraft-level functions, functional requirement, functional interfaces and assumptions are defined	R	R	R	R	N	List of Aircraft-level functions Aircraft-level requirements	4.1.4 4.2 5.3	①	①	①	②	
3.1	The aircraft/system functional hazard assessment is performed	R*	R*	R	R	R	Aircraft FHA System FHA	5.1.1 5.2.3 5.2.4	①	①	①	①	①
3.2	The preliminary aircraft safety assessment is performed	R*	R*	R	A	N	Preliminary Aircraft Safety Assessment (PASA)	5.1.2 5.2.3 5.2.4	①	①	①	①	
3.3	The preliminary system safety assessment is performed	R*	R*	R	A	N	Preliminary System Safety Assessment (PSSA)	5.1.2 5.1.6 5.2.3 5.2.4	①	①	①	②	

A=as negotiated for certification.

N=not required for certification.

R=recommended for certification.

R*=recommended for certification with process independence.

Each of the outputs (or deliverables) also has configuration management objectives:

① Requires Configuration Identification, Change Control, Configuration Index, Archiving/Retrieval, Configuration Baseline establishment and Problem Reporting.

② Does not require Configuration Baseline Establishment and Problem Reporting.

Table 9.5 RTCA/DO-178C validation planning

Objective		Applicability by IDAL				Output/associated evidence	Configuration control category by IDAL				
No.	Description	A	B	C	D	Data item	Refs	A	B	C	D
A-9	Assurance is obtained that S/W plans and standards are developed and reviewed for compliance with this document for consistency.	●	●	●		Plan for S/W Aspects of Certification*	11.1	①	①	①	
A-9	Assurance is obtained that S/W plans and standards are developed and reviewed for compliance with this document for consistency.	●	●	●		S/W Development Plan*	11.2	①	①	②	

Blank – Satisfaction of objective is at applicant's discretion.

- The objective should be satisfied with independence (intellectual, not organisational).
- The objective should be satisfied.

* Data Items not explicitly listed in the tables in DO-178C, but are implied by the main body text.

Each of the outputs (or deliverables) also has configuration management objectives.

- ① Requires Configuration Identification, Traceability, Change Control, Retrieval, Protection against Unauthorised Changes, Data Retention, Baselines, Problem Reporting, Change Review, Configuration Status Accounting, Media Selection/Refreshing/Duplication and Release.
- ② Does not require Baselines, Problem Reporting, Change Review, Configuration Status Accounting, Media Selection/Refreshing/Duplication and Release.

Table 9.6 RTCA/DO-254 validation planning

Objective		Applicability by IDAL				Output/associated evidence		Configuration control category by IDAL			
No.	Description	A	B	C	D	Data item	Refs	A	B	C	D
4.1	The hardware development and verification environments are selected or defined.	●	●	○	○	Hardware Validation Plan	10.1.3	HC2	HC2	HC2	NA
4.1	The means of compliance of the hardware design assurance objectives, including strategies identified using guidance in Section 2.3.4, are proposed to the certification authority.	●	●	○	○	Hardware Validation Plan	10.1.3	HC2	HC2	HC2	NA
6.1.1	Derived hardware requirements against which the hardware item is to be verified are correct and complete.	●	●	○	○	Hardware Validation Plan	10.1.3	HC2	HC2	HC2	NA
6.1.1	Derived hardware requirements against which the hardware item is to be verified are correct and complete.	●	●	○	○	Hardware Test Procedures	10.4.4	HC1	HC1	HC2	HC2
6.1.1	Derived requirements are evaluated for impact on safety.	●	●	○	○	Hardware Review and Analysis Procedures	10.4.2	HC1	HC1	NA	NA
6.1.1	Derived requirements are evaluated for impact on safety.	●	●	○	○	Hardware Test Procedures	10.4.4	HC1	HC1	HC2	HC2

Blank – Satisfaction of objective is at applicant's discretion.

● The objective should be satisfied with independence (intellectual, not organisational).

○ The objective should be satisfied.

Each of the outputs (or deliverables) also has configuration management objectives:

HC1: Requires Configuration Identification, Baseline Traceability, Change Control, Retrieval, Protection against Unauthorised Changes, Data Retention, Baselines, Problem Reporting, Change Review, Media Selection/Refreshing/Duplication and Release.

HC2: Does not require Baselines, Problem Reporting, Change Review, Media Selection/Refreshing/ Duplication and Release.

The intent of traceability is that there should be no childless parents (at any level of requirement) and there should be no orphans (no code or requirement that do not trace to, and are in technical agreement with a higher-level requirement). If these two outcomes are achieved, then the traceability is systematic enough that the review/analysis and verification activities are more likely to provide appropriate coverage of the S/W. Shortfalls in traceability on the other hand may be representative of either a failure to properly record traceability, which is often the case if development leads the requirements/design process, or there has been a breakdown in the design process.

Requirements traceability has been traditionally recorded using traceability matrices presented in the requirements specifications, design descriptions and implementation annotations/comments. Modern developments may use various tools to manage design refinement and record traceability between high-level requirements, low-level/detailed design requirements and implementation.

9.2.3.3 *Validation deliverables*

From the guiding standards we can extract validation deliverables as follows:

- The system wide validation deliverables from [SAE ARP4754A](#) (Appendix A) can be extracted into [Table 9.7](#).
- The S/W validation deliverables from RTCA/DO-178C (Appendix A) can be extracted [Table 9.8](#).
- The CEHW validation deliverables from RTCA/DO-254C can be extracted into [Table 9.9](#).

Table 9.7 SAE ARP4754A validation deliverables

Objective		Applicability by FDAL					Output/associated evidence		Configuration control category by FDAL				
No	Description	A	B	C	D	E	Data item	Refs	A	B	C	D	E
2.2	Aircraft functions are allocated to systems	R	R	R	R	N	System Requirements	4.1.5 4.3	①	①	①	②	
2.3	System requirements including assumptions and system interfaces are defined	R	R	R	R	N	System Requirements	5.3	①	①	①	②	
2.4	System derived requirements (including derived safety-related requirements) are defined and rationale explained	R	R	R	A	N	System Requirements	4.4 5.3.1.4 5.3.2	①	①	①	②	
2.5	System architecture is defined	R	R	R	A	N	System Design Description	4.1.6 4.4 5.8.4.4	①	①	①	②	
2.6	System requirements are allocated to items	R	R	R	R	N	Item Requirements	4.1.7 4.5 4.6 5.3	①	①	①	②	
3.7	Independence requirements in functions, systems and items are captured	R*	R*	R	R	N	System, Hardware, S/W Requirements	5.3.2 5.2.3 5.1.2	①	①	①	②	
4.1	Aircraft, system, item requirements are complete and correct	R*	R*	R	A	N	Validation Results	5.4 5.4.2.c 5.4.3 5.4.4	①	①	①	②	
4.2	Assumptions are justified and validated	R*	R	R	A	N	Validation Results	5.4.2.d	②	②	②	②	

4.3	Derived requirements are justified and validated	R*	R*	R	A	N	Validation Results	5.3.1.4 5.3.2 5.4.2	②	②	②	②	
4.4	Requirements are traceable	R	R	R	A	N	Validation Results	5.4.3 5.4.4	②	②	②	②	
4.6	Validation compliance substantiation is provided	R	R	R	A	N	Validation Summary (including Validation Matrix)	5.4.2.e 5.4.2.f 5.4.8 5.4.7.4	②	②	②	②	

A=as negotiated for certification.
N=not required for certification.
R=recommended for certification.
R*=recommended for certification with process independence.

Each of the outputs (or deliverables) also has configuration management objectives:

- ① Requires Configuration Identification, Change Control, Configuration Index, Archiving/Retrieval, Configuration Baseline Establishment and Problem Reporting.
- ② Does not require Configuration Baseline Establishment and Problem Reporting.

Table 9.8 RTCA/DO-178C validation deliverables

Objective		Applicability by IDAL				Output/associated evidence	Configuration control category by IDAL				
No.	Description	A	B	C	D	Data item	Refs	A	B	C	D
A-2	High-level requirements are developed.	○	○	○	○	S/W Requirements Data	11.9	①	①	①	①
A-2	Derived high-level requirements are defined and provided to the system processes, including the system safety assessment process.	○	○	○	○	S/W Requirements Data	11.9	①	①	①	①
A-2	S/W architecture is developed.	○	○	○	○	Design Description	11.1	①	①	①	①
A-2	Low-level requirements are developed.	○	○	○		Design Description	11.1	①	①	①	
A-2	Derived low-level requirements are defined and provided to the system processes, including the system safety assessment process.	○	○	○		Design Description	11.1	①	①	①	
A-3	High-level requirements conform to standards	○	○	○		S/W Requirements Standards*	11.6	①	①	②	
A-4	Low-level requirements conform to standards.	○	○	○		S/W Design Standards*	11.7	①	①	②	

Blank – Satisfaction of objective is at applicant's discretion.

● The objective should be satisfied with independence (intellectual, not organisational).

○ The objective should be satisfied.

* Data Items not explicitly listed in the tables in DO-178C, but are implied by the main body text.

Each of the outputs (or deliverables) also has configuration management objectives:

① Requires Configuration Identification, Traceability, Change Control, Retrieval, Protection against Unauthorised Changes, Data Retention, Baselines, Problem Reporting, Change Review, Configuration Status Accounting, Media Selection/Refreshing/Duplication and Release.

② Does not require Baselines, Problem Reporting, Change Review, Configuration Status Accounting, Media Selection/Refreshing/Duplication and Release.

Table 9.9 RTCA/DO-254 validation deliverables

Objective		Applicability by IDAL				Output/associated evidence		Configuration control category by IDAL			
No.	Description	A	B	C	D	Data item	Refs	A	B	C	D
5.1.1	Requirements are identified, defined and documented. This includes allocated requirements from the PSSA and derived requirements from the hardware safety assessment.	●	●	○	○	Hardware Requirements	10.3.1	HC1	HC1	HC1	HC1
5.1.1	Derived requirements produced are fed back to the appropriate process.	●	●	○	○	Hardware Requirements	10.3.1	HC1	HC1	HC1	HC1
5.1.1	Requirement omissions and errors are provided to the appropriate process for resolution.	●	●	○	○	Problem Reports	10.6	HC2	HC2	HC2	HC2
5.2.1	The hardware item conceptual design is developed consistent with its requirements.	●	●	○	○	Conceptual Design Data	10.3.2.1	HC2	HC2	NA	NA
5.2.1	Derived requirements produced are fed back to the requirements capture or other appropriate processes.	●	●	○	○	Hardware Requirements	10.3.1	HC1	HC1	HC1	HC1
5.2.1	Requirement omissions and errors are provided to the appropriate processes for resolution.	●	●	○	○	Problem Reports	10.6	HC2	HC2	HC2	HC2
5.3.1	Derived requirements are fed back to the conceptual design process or other appropriate processes.	●	●	○	○	Hardware Requirements	10.3.1	HC1	HC1	HC1	HC1
5.4.1	Derived requirements are fed back to the detailed design process or other appropriate processes.	●	●	○	○	Hardware Requirements	10.3.1	HC1	HC1	HC1	HC1
5.4.1	Requirement omissions and errors are provided to the appropriate processes for resolution.	●	●	○	○	Problem Reports	10.6	HC2	HC2	HC2	HC2
5.5.1	Derived requirements are fed back to the implementation process or other appropriate processes.	●	●	○	○	Hardware Requirements	10.3.1	HC1	HC1	HC1	HC1

Continued

Table 9.9 Continued

Objective		Applicability by IDAL				Output/associated evidence		Configuration control category by IDAL			
No.	Description	A	B	C	D	Data item	Refs	A	B	C	D
5.5.1	Derived requirements are fed back to the implementation process or other appropriate processes.	●	●	○	○	Hardware Acceptance Test Criteria	10.5	HC2	HC2	HC2	HC2
6.1.1	Derived hardware requirements against which the hardware item is to be verified are correct and complete.	●	●	○	○	Hardware Requirements	10.3.1	HC1	HC1	HC1	HC1
6.1.1	Derived hardware requirements against which the hardware item is to be verified are correct and complete.	●	●	○	○	Hardware Traceability Data	10.4.1	HC2	HC2	HC2	HC2
6.1.1	Derived hardware requirements against which the hardware item is to be verified are correct and complete.	●	●	○	○	Hardware Review and Analysis Results	10.4.3	HC2	HC2	HC2	HC2
6.1.1	Derived hardware requirements against which the hardware item is to be verified are correct and complete.	●	●	○	○	Hardware Test Results	10.4.5	HC2	HC2	HC2	HC2
6.1.1	Derived requirements are evaluated for impact on safety.	●	●	○	○	Hardware Requirements	10.3.1	HC1	HC1	HC1	HC1
6.1.1	Derived requirements are evaluated for impact on safety.	●	●	○	○	Hardware Review and Analysis Results	10.4.3	HC2	HC2	HC2	HC2
6.1.1	Derived requirements are evaluated for impact on safety.	●	●	○	○	Hardware Test Results	10.4.5	HC2	HC2	HC2	HC2
6.1.1	Omissions and errors are fed back to the appropriate processes for resolution.	●	●	○	○	Problem Reports	10.6	HC2	HC2	HC2	HC2
6.1.4	Evidence is provided that the hardware implementation meets the requirements.	●	●	○	○	Hardware Review and Analysis Results	10.4.3	HC2	HC2	HC2	HC2

6.1.5	Evidence is provided that the hardware implementation meets the requirements.	●	●	○	○	Hardware Test Results	10.4.5	HC2	HC2	HC2	HC2
6.1.6	Acceptance test criteria are identified, can be implemented and are consistent with the hardware design assurance levels of the hardware functions.	●	●	○	○	Hardware Acceptance Test Criteria	10.5	HC2	HC2	HC2	HC2
6.1.7	Omissions and errors are fed back to the appropriate processes for resolution.	●	●	○	○	Problem Reports	10.6	HC2	HC2	HC2	HC2

Blank – Satisfaction of objective is at applicant's discretion.

● The objective should be satisfied with independence (intellectual, not organisational).

○ The objective should be satisfied.

Each of the outputs (or deliverables) also has configuration management objectives:

HC1: Requires Configuration Identification, Baseline Traceability, Change Control, Retrieval, Protection against Unauthorised Changes, Data Retention, Baselines, Problem Reporting, Change Review, Media Selection/Refreshing/Duplication and Release.

HC2: Does not require Baselines, Problem Reporting, Change Review, Media Selection/Refreshing/Duplication and Release.

9.2.4 Step 4: requirements verification

Verification is the evaluation of an implementation of requirements to determine that they have been met. It can be summarised in the question: ‘*Did we build the thing right?*’ With reference to the right hand side of Fig. 1.3, Step 2 is all about verifying that the implemented solution (system or item) indeed conforms with all the requirements (safety, contractual, regulatory, derived and so on) imposed on it.

Verification is accomplished during the Item Development Phase and is accomplished via reviews, checking, inspection, analysis and test activities; see the right hand side of Fig. 1.3.

Verification consists of 3 phases: The planning phase, the doing (or accomplishment) phase and the delivery (of evidence) phase as explored in more detail below:

9.2.4.1 Verification planning

Verification is accomplished as follows:

- For all systems (except S/W and CEHW), verification is accomplished at each level of abstraction via traditional techniques such as assessments and demonstrations (e.g., simulations, inspection and testing) and [SAE ARP4754A](#) provides us with the level of rigour against the allocated FDAL.
- For S/W, the rigour of the development process is given by the IDAL approach in RTCA/DO-178C.
- For CEHW, the rigour of the development process is given by the IDAL approach in RTCA/DO-254.

From the guiding standards we can extract verification planning related outputs as follows:

- The system wide requirements from ARP4754A (Appendix A) can be extracted into [Table 9.10](#). Unlike the Software and CEHW IDAL verification activities (which address the bottom part of the V-model in Fig. 1.3 only), verification to ARP4754A needs to be accomplished at each level of system integration (i.e., the whole right side of the V-model in Fig. 1.3).
- RTCA/DO-178C (Appendix A) could have the S/W Verification Planning–related outputs extracted into [Table 9.11](#).
- RTCA/DO-254 (Appendix A) could have the CEHW Verification Planning–related outputs extracted into [Table 9.12](#).

9.2.4.2 Verification accomplishment

Verification is accomplished when we have:

- established that the implementation of the system architecture complies with item, system and aircraft requirements (including verification of safety requirements);
- ensured the validity of the Safety Assessment, that is:
 - ensured that verification demonstrates the intended functions and that no unintended function impacts safety;
 - assessed deficiencies and their impact on safety.
- ensured that the intended functions have been correctly implemented;
- ensured that the requirements have been satisfied in the implemented S/W and hardware (i.e., traceability from requirements to evidence);
- provided compliance substantiation of verification.

Table 9.10 **SAE ARP4754A** verification planning

Objective		Applicability by FDAL					Output/associated evidence		Configuration control category by FDAL				
No.	Description	A	B	C	D	E	Data item	Refs	A	B	C	D	E
1.2	Transition criteria and inter-relationship among processes are defined	R	R	R	A	N	Plans in Objective 1.1	3.2	②	②	②	②	
1.1	System development and integral processes are defined	R	R	R	R	N	Verification Plan	5.5.3 5.5.5.1	②	②	②	②	
5.1	Test or demonstration procedures are correct	R*	R	R	A	N	Verification Procedures	5.5.4.3	①	①	②	②	
5.2	Verification demonstrates intended function and confidence of no unintended function impacts to safety	R*	R	R	A	N	Verification Procedures	5.5.1 5.5.5.3	①	①	②	②	
5.3	Product implementation complies with aircraft and system requirements	R*	R	R	A	N	Verification Procedures	5.5.1	①	①	②	②	

A=as negotiated for certification.

N=not required for certification.

R=recommended for certification.

R*=recommended for certification with process independence.

Each of the outputs (or deliverables) also has configuration management objectives:

- ① Requires Configuration Identification, Change Control, Configuration Index, Archiving/Retrieval, Configuration Baseline Establishment and Problem Reporting.
- ② Does not require Configuration Baseline Establishment and Problem Reporting.

Table 9.11 RTCA/DO-178C verification planning

Objective		Applicability by IDAL				Output/associated evidence		Configuration control by IDAL			
No.	Description	A	B	C	D	Data item	Refs	A	B	C	D
A-1	The activities of the S/W lifecycle processes are defined.	○	○	○	○	S/W Verification Plan	11.3	①	①	②	②
A-1	The S/W lifecycle(s), including the interrelationships between the processes, their sequencing, feedback mechanisms, and transition criteria, is defined.	○	○	○		S/W Verification Plan	11.3	①	①	②	
A-1	S/W lifecycle environment is selected and defined.	○	○	○		S/W Verification Plan	11.3	①	①	②	
A-1	Additional considerations are addressed.	○	○	○	○	S/W Verification Plan	11.3	①	①	②	②

Blank – Satisfaction of objective is at applicant's discretion.

● The objective should be satisfied with independence (intellectual, not organisational).

○ The objective should be satisfied.

Each of the outputs (or deliverables) also has configuration management objectives:

- ① Requires Configuration Identification, Traceability, Change Control, Retrieval, Protection against Unauthorised Changes, Data Retention, Baselines, Problem Reporting, Change Review, Configuration Status Accounting, Media Selection/Refreshing/Duplication and Release.
- ② Does not require Baselines, Problem Reporting, Change Review, Configuration Status Accounting, Media Selection/Refreshing/Duplication and Release.

Table 9.12 RTCA/DO-254 verification planning

Objective		Applicability by IDAL				Output/associated evidence		Configuration control by IDAL			
No.	Description	A	B	C	D	Data item	Refs	A	B	C	D
4.1	The hardware design lifecycle processes are defined.	●	●	○	○	Hardware Verification Plan	10.1.4	HC2	HC2	HC2	HC2
4.1	The hardware development and verification environments are selected or defined.	●	●	○	○	Hardware Verification Plan	10.1.4	HC2	HC2	HC2	HC2
4.1	The means of compliance of the hardware design assurance objectives, including strategies identified using guidance in Section 2.3.4, are proposed to the certification authority.	●	●	○	○	Hardware Verification Plan	10.1.4	HC2	HC2	HC2	HC2
6.2.1	Evidence is provided that the hardware implementation meets the requirements.	●	●	○	○	Hardware Verification Plan	10.1.4	HC2	HC2	HC2	HC2
6.2.3	Evidence is provided that the hardware implementation meets the requirements.	●	●	○	○	Hardware Review and Analysis Procedures	10.4.2	HC1	HC1	NA	NA
6.2.4	Evidence is provided that the hardware implementation meets the requirements.	●	●	○	○	Hardware Test Procedures	10.4.4	HC1	HC1	HC2	HC2

Blank – Satisfaction of objective is at applicant's discretion.

● The objective should be satisfied with independence (intellectual, not organisational).

○ The objective should be satisfied.

Each of the outputs (or deliverables) also has configuration management objectives:

HC1: Requires Configuration Identification, Baseline Traceability, Change Control, Retrieval, Protection against Unauthorised Changes, Data Retention, Baselines, Problem Reporting, Change Review, Media Selection/Refreshing/Duplication and Release.

HC2: Does not require Baselines, Problem Reporting, Change Review, Media Selection/Refreshing/Duplication and Release.

9.2.4.3 *Verification deliverables*

From the guiding standards we can extract verification deliverables as follows:

- The system wide verification deliverables from ARP4754A (Appendix A) can be extracted into [Table 9.13](#).
- The S/W verification deliverables from RTCA/DO-178C (Appendix A) can be extracted into [Table 9.14](#).
- The CEHW verification deliverables from RTCA/DO-254C can be extracted into [Table 9.15](#).

9.2.5 *Supporting process*

[Fig. 9.1](#) shows that the V&V activities are supported by a number of processes, which include:

- The Safety Assessment Process.
- The Configuration Management Process.
- The Process Assurance process.
- Certification and Regulatory Authority Coordination.

The following subsections explore the outputs required from each of these processes as a function of the allocated DAL.

9.2.5.1 *The Safety Assessment process*

The Safety Assessment process assigns the DAL as explored in Step 1 ([Section 9.2.1](#)) mentioned earlier. The development assurance outputs from the Safety Assessment process is summarised in [Table 9.16](#).

9.2.5.2 *The Configuration Management process*

Configuration control of documentation and design artefacts (hardware and S/W) is essential to the control of a certification baseline. Configuration Management is therefore both a system development as well as a certification activity and a configuration baseline should be established as soon as compliance substantiation is first desired. Traceability of the final proposed Configuration Baseline is a necessary element of demonstrating Development Assurance [ARP4754A, paragraph 5.6].

Configuration should therefore be managed throughout all lifecycle phases to ensure that:

- All configuration items are identified (e.g., part numbers and revision numbers) in accordance with the Configuration Management Plan (CMP).
- Configuration baselines are established, and the ancestry of each baseline is both captured and accurate.
- Problem reporting, change control, change review and configuration status accounting are established to ensure that all sources of change have an instrument to manage their incorporation in a controlled and reviewed manner.
- Processes for archive and retrieval are followed, per those specified in the CMP.
- Configuration management of tools and the lifecycle environment (i.e., development and V&V environments) are established.
- Load control is established to ensure that only approved versions of S/W and CEHW are loaded onto the lifecycle environment and aircraft.

Table 9.13 SAE ARP4754A verification deliverables

Objective		Applicability by FDAL					Output/associated evidence		Configuration control category by FDAL				
No.	Description	A	B	C	D	E	Data item	Refs	A	B	C	D	E
2.7	Appropriate item, system and aircraft integrations are performed.	R	R	R	A	N	Verification Summary	4.6.3 4.6.4	②	②	②	②	
5.2	Verification demonstrates intended function and confidence of no unintended function impacts to safety.	R*	R	R	A	N	Verification Results	5.5.5.2	②	②	②	②	
5.3	Product implementation complies with aircraft, and system requirements.	R*	R	R	A	N	Verification Results	5.5.2	②	②	②	②	
5.4	Safety requirements are verified.	R*	R*	R	A	N	Verification Procedures and Results (ASA, SSA)	5.5.1 5.5.5.3	②	②	②	②	
5.5	Verification compliance substantiation is included.	R	R	R	A	N	Verification Matrix	5.5.6.3	②	②	②	②	
5.5	Verification compliance substantiation is included.	R	R	R	A	N	Verification Summary	5.5.6.4	②	②	②	②	
5.6	Assessment of deficiencies and their related impact on safety is identified.	R	R	R	A	N	Verification Summary	5.5.6.4	②	②	②	②	
5.6	Assessment of deficiencies and their related impact on safety is identified.	R	R	R	A	N	Problem Reports	5.5.6.4	②	②	②	②	

A=as negotiated for certification.

N=not required for certification.

R=recommended for certification.

R*=recommended for certification with process independence.

Each of the outputs (or deliverables) also has configuration management objectives:

- ① Requires Configuration Identification, Change Control, Configuration Index, Archiving/Retrieval, Configuration Baseline Establishment and Problem Reporting.
- ② Does not require Configuration Baseline Establishment and Problem Reporting.

Table 9.14 RTCA/DO-178C verification deliverables

Objective		Applicability by IDAL				Output/associated evidence		Configuration control category by IDAL			
No.	Description	A	B	C	D	Data item	Refs	A	B	C	D
A-1	S/W plans comply with this document.	○	○	○		S/W Verification Results	11.14	②	②	②	
A-1	Development and revision of S/W plans are coordinated.	○	○	○		S/W Verification Results	11.14	②	②	②	
A-2	High-level requirements are developed.	○	○	○	○	Traceability Data	11.21	①	①	①	①
A-2	Low-level requirements are developed.	○	○	○		Traceability Data	11.21	①	①	①	
A-2	Source Code is developed.	○	○	○		Source code	11.11	①	①	①	
A-2	Source Code is developed.	○	○	○		Traceability Data	11.21	①	①	①	
A-2	Executable Object Code and Parameter Data Item Files, if any, are produced and loaded in the target computer.	○	○	○	○	Executable Object Code	11.12	①	①	①	①
A-2	Executable Object Code and Parameter Data Item Files, if any, are produced and loaded in the target computer.	○	○	○	○	Parameter Data Item File	11.22	①	①	①	①
A-3	High-level requirements comply with system requirements.	●	●	○	○	S/W Verification Results	11.14	②	②	②	②
A-3	High-level requirements comply with system requirements.	●	●	○	○	Traceability Data*	11.21	②	②	②	②
A-3	High-level requirements are accurate and consistent.	●	●	○	○	S/W Verification Results	11.14	②	②	②	②
A-3	High-level requirements are compatible with target computer.	○	○			S/W Verification Results	11.14	②	②		
A-3	High-level requirements are verifiable.	○	○	○		S/W Verification Results	11.14	②	②	②	

A-3	High-level requirements conform to standards.	○	○	○		S/W Verification Results	11.14	②	②	②	
A-3	High-level requirements are traceable to system requirements.	○	○	○	○	S/W Verification Results	11.14	②	②	②	②
A-3	High-level requirements are traceable to system requirements.	○	○	○	○	Traceability Data*	11.21	②	②	②	②
A-3	Algorithms are accurate.	●	●	○		S/W Verification Results	11.14	②	②	②	
A-4	Low-level requirements comply with high-level requirements.	●	●	○		S/W Verification Results	11.14	②	②	②	
A-4	Low-level requirements comply with high-level requirements.	●	●	○		Traceability Data*	11.21	②	②	②	
A-4	Low-level requirements are accurate and consistent.	●	●	○		S/W Verification Results	11.14	②	②	②	
A-4	Low-level requirements are compatible with target computer.	○	○			S/W Verification Results	11.14	②	②		
A-4	Low-level requirements are verifiable.	○	○			S/W Verification Results	11.14	②	②		
A-4	Low-level requirements conform to standards.	○	○	○		S/W Verification Results	11.14	②	②	②	
A-4	Low-level requirements are traceable to high-level requirements.	○	○	○		S/W Verification Results	11.14	②	②	②	
A-4	Low-level requirements are traceable to high-level requirements.	○	○	○		Traceability Data*	11.21	②	②	②	
A-4	Algorithms are accurate.	●	●	○		S/W Verification Results	11.14	②	②	②	
A-4	S/W architecture is compatible with high-level requirements.	●	○	○		S/W Verification Results	11.14	②	②	②	

Continued

Table 9.14 Continued

Objective		Applicability by IDAL				Output/associated evidence		Configuration control category by IDAL			
No.	Description	A	B	C	D	Data item	Refs	A	B	C	D
A-4	S/W architecture is consistent.	●	○	○		S/W Verification Results	11.14	②	②	②	
A-4	S/W architecture is compatible with target computer.	○	○			S/W Verification Results	11.14	②	②		
A-4	S/W architecture is verifiable.	○	○			S/W Verification Results	11.14	②	②		
A-4	S/W architecture conforms to standards.	○	○	○		S/W Verification Results	11.14	②	②	②	
A-4	S/W architecture conforms to standards.	○	○	○		S/W Design Standards*	11.7	①	①	②	
A-4	S/W partitioning integrity is confirmed.	●	○	○	○	S/W Verification Results	11.14	②	②	②	②
A-5	Source code complies with low-level requirements.	●	●	○		S/W Verification Results	11.14	②	②	②	
A-5	Source code complies with low-level requirements.	●	●	○		Traceability Data*	11.21	②	②	②	
A-5	Source Code complies with S/W architecture.	●	○	○		S/W Verification Results	11.14	②	②	②	
A-5	Source Code is verifiable.	○	○			S/W Verification Results	11.14				
A-5	Source Code conforms to standards.	○	○	○		S/W Verification Results	11.14	②	②	②	
A-5	Source Code is traceable to low-level requirements.	○	○	○		S/W Verification Results	11.14	②	②	②	

A-5	Source Code is traceable to low-level requirements.	○	○	○		Traceability Data*	11.21	②	②	②	
A-5	Source Code is accurate and consistent.	●	○	○		S/W Verification Results	11.14	②	②	②	
A-5	Output of S/W integration process is complete and correct.	○	○	○		S/W Verification Results	11.14	②	②	②	
A-5	Parameter Data Item File is correct and complete.	●	●	○	○	S/W Verification Cases and Procedures	11.13	②	②	②	②
A-5	Parameter Data Item File is correct and complete.	●	●	○	○	S/W Verification Results	11.14	②	②	②	②
A-5	Verification of Parameter Data Item File is achieved.	●	●	○		S/W Verification Results	11.14	②	②	②	
A-6	Executable Object Code complies with high-level requirements.	○	○	○	○	S/W Verification Cases and Procedures	11.13	①	①	②	②
A-6	Executable Object Code complies with high-level requirements.	○	○	○	○	S/W Verification Results	11.14	②	②	②	②
A-6	Executable Object Code complies with high-level requirements.	○	○	○	○	Traceability Data*	11.21	①	①	②	②
A-6	Executable Object Code is robust with high-level requirements.	○	○	○	○	S/W Verification Cases and Procedures*	11.13	①	①	②	②
A-6	Executable Object Code is robust with high-level requirements.	○	○	○	○	S/W Verification Results	11.14	②	②	②	②
A-6	Executable Object Code is robust with high-level requirements.	○	○	○	○	Traceability Data	11.21	①	①	②	②
A-6	Executable Object Code complies with low-level requirements.	●	●	○		S/W Verification Cases and Procedures	11.13	①	①	②	

Continued

Table 9.14 Continued

Objective		Applicability by IDAL				Output/associated evidence		Configuration control category by IDAL			
No.	Description	A	B	C	D	Data item	Refs	A	B	C	D
A-6	Executable Object Code complies with low-level requirements.	●	●	○		S/W Verification Results	11.14	②	②	②	
A-6	Executable Object Code complies with low-level requirements.	●	●	○		Traceability Data	11.21	①	①	②	
A-6	Executable Object Code is robust with low-level requirements.	●	○	○		S/W Verification Cases and Procedures	11.13	①	①	②	
A-6	Executable Object Code is robust with low-level requirements.	●	○	○		S/W Verification Results	11.14	②	②	②	
A-6	Executable Object Code is robust with low-level requirements.	●	○	○		Traceability Data	11.21	①	①	②	
A-6	Executable Object Code is compatible with target computer.	○	○	○	○	S/W Verification Cases and Procedures	11.13	①	①	②	②
A-6	Executable Object Code is compatible with target computer.	○	○	○	○	S/W Verification Results	11.14	②	②	②	②
A-7	Test procedures are correct.	●	○	○		S/W Verification Results	11.14	②	②	②	
A-7	Test procedures are correct.	●	○	○		S/W Verification Cases and Procedures*	11.13	①	①	②	
A-7	Test results are correct and discrepancies explained.	●	○	○		S/W Verification Results	11.14	②	②	②	
A-7	Test results are correct and discrepancies explained.	●	○	○		Problem Reports*	11.17	②	②	②	
A-7	Test coverage of high-level requirements is achieved.	●	○	○	○	S/W Verification Results	11.14	②	②	②	②
A-7	Test coverage of high-level requirements is achieved.	●	○	○	○	Traceability Data*	11.21	①	①	②	②

A-7	Test coverage of low-level requirements is achieved.	●	○	○		S/W Verification Results	11.14	②	②	②	
A-7	Test coverage of low-level requirements is achieved.	●	○	○		Traceability Data*	11.21	①	①	②	
A-7	Test coverage of S/W structure (modified condition/decision coverage) is achieved.	●				S/W Verification Results	11.14	②			
A-7	Test coverage of S/W structure (decision coverage) is achieved.	●	●			S/W Verification Results	11.14	②	②		
A-7	Test coverage of S/W structure (statement coverage) is achieved.	●	●	○		S/W Verification Results	11.14	②	②	②	
A-7	Test coverage of S/W structure (data coupling and control coupling) is achieved.	●	●	○		S/W Verification Results	11.14	②	②	②	
A-7	Verification of additional code that cannot be traced to Source Code, is achieved.	●				S/W Verification Results	11.14	②			

Blank – Satisfaction of objective is at applicant's discretion.

● The objective should be satisfied with independence (intellectual, not organisational).

○ The objective should be satisfied.

* Data Items not explicitly listed in the tables in DO-178C, but are implied by the main body text.

Each of the outputs (or deliverables) also has configuration management objectives:

① Requires Configuration Identification, Traceability, Change Control, Retrieval, Protection against Unauthorised Changes, Data Retention, Baselines, Problem Reporting, Change Review, Configuration Status Accounting, Media Selection/Refreshing/Duplication and Release.

② Does not require Baselines, Problem Reporting, Change Review, Configuration Status Accounting, Media Selection/Refreshing/Duplication and Release.

Table 9.15 RTCA/DO-254 verification deliverables

Objective		Applicability by IDAL				Output/associated evidence		Configuration control by IDAL			
No.	Description	A	B	C	D	Data item	Refs	A	B	C	D
5.3.1	The detailed design is developed from the hardware item requirements and conceptual design data.	●	●	○	○	Detailed Design Data	10.3.2.2	HC1	HC1	HC1	HC1
5.3.1	The detailed design is developed from the hardware item requirements and conceptual design data.	●	●	○	○	Top-Level Drawing	10.3.2.2.1	HC1	HC1	HC1	HC1
5.3.1	The detailed design is developed from the hardware item requirements and conceptual design data.	●	●	○	○	Assembly Drawings	10.3.2.2.2.2	HC1	HC1	HC1	HC1
5.3.1	The detailed design is developed from the hardware item requirements and conceptual design data.	●	●	○	○	Hardware/S/W Interface Data	10.3.2.2.4	HC1	HC1	HC1	HC1
5.3.1	Requirement omissions or errors are provided to the appropriate processes for resolution.	●	●	○	○	Problem Reports	10.6	HC2	HC2	HC2	HC2
5.4.1	A hardware item is produced which implements the hardware detailed design using representative manufacturing processes.	●	●	○	○	Detailed Design Data	10.3.2.2	HC1	HC1	HC1	HC1
5.4.1	A hardware item is produced which implements the hardware detailed design using representative manufacturing processes.	●	●	○	○	Top-Level Drawing	10.3.2.2.1	HC1	HC1	HC1	HC1

5.4.1	A hardware item is produced which implements the hardware detailed design using representative manufacturing processes.	●	●	○	○	Assembly Drawings	10.3.2.2.2.2	HC1	HC1	HC1	HC1
5.4.1	A hardware item is produced which implements the hardware detailed design using representative manufacturing processes.	●	●	○	○	Installation Control Drawings	10.3.2.2.3	HC1	HC1	HC1	HC1
5.4.1	The hardware item implementation, assembly and installation data is complete.	●	●	○	○	Detailed Design Data	10.3.2.2	HC1	HC1	HC1	HC1
5.4.1	The hardware item implementation, assembly and installation data is complete.	●	●	○	○	Top-Level Drawing	10.3.2.2.1	HC1	HC1	HC1	HC1
5.4.1	The hardware item implementation, assembly and installation data is complete.	●	●	○	○	Assembly Drawings	10.3.2.2.2.2	HC1	HC1	HC1	HC1
5.4.1	The hardware item implementation, assembly and installation data is complete.	●	●	○	○	Installation Control Drawings	10.3.2.2.3	HC1	HC1	HC1	HC1
5.4.1	Derived requirements are fed back to the detailed design process or other appropriate processes.	●	●	○	○	Hardware Requirements	10.3.1	HC1	HC1	HC1	HC1
5.4.1	Requirement omissions and errors are provided to the appropriate processes for resolution.	●	●	○	○	Problem Reports	10.6	HC2	HC2	HC2	HC2
5.5.1	Derived requirements are fed back to the implementation process or other appropriate processes.	●	●	○	○	Hardware Requirements	10.3.1	HC1	HC1	HC1	HC1

Continued

Table 9.15 Continued

Objective		Applicability by IDAL				Output/associated evidence		Configuration control by IDAL			
No.	Description	A	B	C	D	Data item	Refs	A	B	C	D
5.5.1	Derived requirements are fed back to the implementation process or other appropriate processes.	●	●	○	○	Hardware Acceptance Test Criteria	10.5	HC2	HC2	HC2	HC2
6.2.2	Evidence is provided that the hardware implementation meets the requirements.	●	●	○	○	Hardware Requirements	10.3.1	HC1	HC1	HC1	HC1
6.1.3	Evidence is provided that the hardware implementation meets the requirements.	●	●	○	○	Hardware Traceability Data	10.4.1	HC2	HC2	HC2	HC2
6.2.5	Traceability is established between hardware requirements, the implementation, and the verification procedures and results.	●	●	○	○	Hardware Traceability Data	10.4.1	HC2	HC2	HC2	HC2

Blank – Satisfaction of objective is at applicant’s discretion.
● The objective should be satisfied with independence (intellectual, not organisational).
○ The objective should be satisfied.

Each of the outputs (or deliverables) also has configuration management objectives:
HC1: Requires Configuration Identification, Baseline Traceability, Change Control, Retrieval, Protection against Unauthorised Changes, Data Retention, Baselines, Problem Reporting, Change Review, Media Selection/Refreshing/Duplication and Release.
HC2: Does not require Baselines, Problem Reporting, Change Review, Media Selection/Refreshing/Duplication and Release.

Table 9.16 SAE ARP4754A safety assessment deliverables

Objective		Applicability by FDAL					Output/associated evidence		Configuration control category by FDAL				
No.	Description	A	B	C	D	E	Data item	Refs	A	B	C	D	E
1.1	System development and integral processes are defined.	R	R	R	R	N	Safety Program Plan	3.1 5.1.5 App B	②	②	②	②	
3.1	The aircraft/system functional hazard assessment is performed.	R*	R*	R	R	R	Aircraft FHA System FHA	5.1.1 5.2.3 5.2.4	①	①	①	①	①
3.2	The preliminary aircraft safety assessment is performed.	R*	R*	R	A	N	Preliminary Aircraft Safety Assessment (PASA)	5.1.2 5.2.3 5.2.4	①	①	①	①	
3.3	The preliminary system safety assessment is performed.	R*	R*	R	A	N	Preliminary System Safety Assessment (PSSA)	5.1.2 5.1.6 5.2.3 5.2.4	①	①	①	②	
3.4	The common cause analyses are performed.	R	R	A	N	N	Particular Risk Assessment (PRA) Common Mode Analysis (CMA) Zonal Safety Analysis (ZSA)	5.1.4	①	①	①		
3.5	The aircraft safety assessment is performed.	R*	R*	R	A	N	Aircraft Safety Assessment (ASA)	5.1.3 5.1.6	①	①	①		
3.6	The system safety assessment is performed.	R*	R*	R	A	N	System Safety Assessment (SSA)	5.1.3 5.1.6	①	①	①	①	

A=as negotiated for certification.

N=not required for certification.

R=recommended for certification.

R*=recommended for certification with process independence.

Each of the outputs (or deliverables) also has configuration management objectives:

① Requires Configuration Identification, Change Control, Configuration Index, Archiving/Retrieval, Configuration Baseline Establishment and Problem Reporting.

② Does not require Configuration Baseline Establishment and Problem Reporting.

[ARP4754A](#), DO-178C (see Annex B9.1) and DO-254 all use the concept of Control Categories (CC) for specifying the extent of configuration control over the lifecycle data items.

- CC1 is the most onerous configuration management obligation. It requires rigorous tracking of each change made to the item, explicit version control supported usually by tools, recording of status accounting for each change.
- CC2 is a more general set of configuration management controls. It is less rigorous than CC1, with changes being grouped into version releases, and the release process used to control the configuration identification.

In practice, most reputable developers operate two levels of CM system, which broadly mirror the CC1 and CC2 requirements. For example, many companies might:

- Use the CC2 requirements for document and report configuration management (possibly using tools such as Sharepoint, Atlassian Jira or Objective).
- Use the CC1 requirements for configuration control of S/W source code and S/W test data (using tools such as Serena dimensions, SVN or Rationale Clearquest/Clearcase).

The development assurance outputs from the Configuration Management process is summarised in [Tables 9.17–9.19](#). Each of the outputs (or deliverables) in [Tables 9.1–9.24](#) is allocated a Configuration Control category, which defines the measure of rigour of configuration and change control applicable to that output.

9.2.5.3 *Process Assurance*

Process Assurance looks at the processes used to create the hardware or S/W. The objectives are to [ARP4754A paragraph 5.7.2]:

- Ensure the necessary plans are developed, and then maintained
- Ensure development activities and processes are conducted as planned
- Establish evidence of adherence to the plans.

Accordingly, Process Assurance should have a level of independence from the development process and is typically performed by the QA Department, who provide the Senior Management Team (SMT) with objective feedback regarding compliance to approved plans, procedures, standards, and analyses.

Process assurance activities are performed throughout the lifecycle, including product conception, design, implementation, operation, and maintenance. It aims to detect, record, evaluate, approve, track and resolve deviations from approved plans and procedures. For each lifecycle phase, process assurance makes sure that planning is performed, that the plan is followed, and that the products of each phase are correct and complete.

The development assurance outputs from Process Assurance is summarised in [Tables 9.20–9.22](#).

Table 9.17 **SAE ARP4754A** configuration management deliverables

Objective		Applicability by FDAL					Output/associated evidence		Configuration control category by IDAL				
No.	Description	A	B	C	D	E	Data item	Refs	A	B	C	D	E
1.1	System development and integral processes are defined.	R	R	R	R	A	Configuration Management Plan	5.6.2.1	②	②	②	②	
6.1	Configuration items are identified.	R	R	R	A	N	CM Records	5.6.2.2	②	②	②	②	
6.2	Configuration baselines and derivatives are established.	R	R	R	A	N	Configuration Baseline Records	5.6.2.3	①	①	②	②	
6.3	Problem reporting, change control, change review and configuration status accounting are established.	R	R	R	R	N	Problem reports	5.6.2.4	②	②	③	②	
6.3	Problem reporting, change control, change review, and configuration status accounting are established.	R	R	R	R	N	CM Records	5.6.2.4	②	②	③	②	
6.4	Archive and retrieval are established.	R	R	R	R	N	CM Records	5.6.2.5	②	②	③	②	
8.1	Compliance substantiation is provided.	R	R	R	A	N	Configuration Index	5.8.4.2	①	①	②	②	

A=as negotiated for certification.

N=not required for certification.

R=recommended for certification.

R*=recommended for certification with process independence.

Each of the outputs (or deliverables) also has configuration management objectives:

① Requires Configuration Identification, Change Control, Configuration Index, Archiving/Retrieval, Configuration Baseline Establishment and Problem Reporting.

② Does not require Configuration Baseline Establishment and Problem Reporting.

Table 9.18 RTCA/DO-0178C S/W configuration management deliverables

Objective		Applicability by IDAL				Output/associated evidence		Configuration control by IDAL			
No.	Description	A	B	C	D	Data item	Refs	A	B	C	D
A-1	The activities of the S/W lifecycle processes are defined.	○	○	○	○	S/W Configuration Management Plan	11.4	①	①	②	②
A-1	The S/W lifecycle(s), including the interrelationships between the processes, their sequencing, feedback mechanisms and transition criteria, is defined.	○	○	○		S/W Configuration Management Plan	11.4	①	①	②	
A-1	S/W lifecycle environment is selected and defined.	○	○	○		S/W Configuration Management Plan	11.4	①	①	②	
A-1	Additional considerations are addressed.	○	○	○	○	S/W Configuration Management Plan	11.4	①	①	②	②
A-8	Configuration items are identified.	○	○	○	○	S/W Configuration Management Records	11.18	②	②	②	②
A-8	Baselines and traceability are established.	○	○	○	○	S/W Configuration Index	11.16	①	①	①	①
A-8	Baselines and traceability are established.	○	○	○	○	S/W Configuration Management Records	11.18	②	②	②	②
A-8	Problem reporting, change control, change review and configuration status accounting are established.	○	○	○	○	Problem Reports	11.17	②	②	②	②
A-8	Problem reporting, change control, change review and configuration status accounting are established.	○	○	○	○	S/W Configuration Management Records	11.18	②	②	②	②

A-8	Archive, retrieval and release are established.	○	○	○	○	S/W Configuration Management Records	11.18	②	②	②	②
A-8	S/W load control is established.	○	○	○	○	S/W Configuration Management Records	11.18	②	②	②	②
A-8	S/W lifecycle environment control is established.	○	○	○	○	S/W Life Cycle Environment Configuration Index	11.15	①	①	①	②
A-8	S/W lifecycle environment control is established.	○	○	○	○	S/W Configuration Management Records	11.18	②	②	②	②
A-9	Assurance is obtained that S/W plans and standards are developed and reviewed for compliance with this document for consistency.	●	●	●		S/W Configuration Management Plan*	11.4	①	①	②	
A-9	Assurance is obtained that S/W lifecycle processes comply with approved plans.	●	●	●	●	S/W Configuration Management Plan*	11.4	①	①	②	②

Blank – Satisfaction of objective is at applicant's discretion.

● The objective should be satisfied with independence (intellectual, not organisational).

○ The objective should be satisfied.

* Data Items are not explicitly listed in the tables in DO-178C, but are implied by the main body text.

Each of the outputs (or deliverables) also has configuration management objectives:

① Requires Configuration Identification, Traceability, Change Control, Retrieval, Protection against Unauthorised Changes, Data Retention, Baselines, Problem Reporting, Change Review, Configuration Status Accounting, Media Selection/Refreshing/Duplication and Release.

② Does not require Baselines, Problem Reporting, Change Review, Configuration Status Accounting, Media Selection/ Refreshing/Duplication and Release.

Table 9.19 RTCA/DO-254 complex electronic H/W Configuration Management Deliverables

Objective		Applicability by IDAL				Output / Associated Evidence		Configuration Control Category by IDAL			
No.	Description	A	B	C	D	Data Item	Refer	A	B	C	D
4.1	The hardware design life cycle processes are defined.	●	●	○	○	Hardware Configuration Management Plan	10.1.5	HC1	HC1	HC2	HC2
4.1	Standards are selected and defined.	●	●	○	○	Hardware Archive Standards	10.2.4	HC2	HC2	NA	NA
4.1	The hardware development and verification environments are selected or defined.	●	●	○	○	Hardware Configuration Management Plan	10.1.5	HC1	HC1	HC2	HC2
4.1	The means of compliance of the hardware design assurance objectives, including strategies identified using guidance in Section 2.3.4, are proposed to the certification authority.	●	●	○	○	Hardware Configuration Management Plan	10.1.5	HC1	HC1	HC2	HC2
5.1.1	Requirements are identified, defined and documented. This includes allocated requirements from the PSSA and derived requirements from the hardware safety assessment.	●	●	○	○	Hardware Archive Standards	10.2.4	HC2	HC2	NA	NA
5.5.1	A baseline is established that includes all design and manufacturing data needed to support the consistent replication of the hardware item.	●	●	○	○	Hardware Archive Standards	10.2.4	HC2	HC2	NA	NA
5.5.1	A baseline is established that includes all design and manufacturing data needed to support the consistent replication of the hardware item.	●	●	○	○	Hardware Requirements	10.3.1	HC1	HC1	HC1	HC1

5.5.1	A baseline is established that includes all design and manufacturing data needed to support the consistent replication of the hardware item.	●	●	○	○	Top-Level Drawing	10.3.2.2.1	HC1	HC1	HC1	HC1
5.5.1	A baseline is established that includes all design and manufacturing data needed to support the consistent replication of the hardware item.	●	●	○	○	Assembly Drawings	10.3.2.2.2.2	HC1	HC1	HC1	HC1
5.5.1	A baseline is established that includes all design and manufacturing data needed to support the consistent replication of the hardware item.	●	●	○	○	Installation Control Drawings	10.3.2.2.3	HC1	HC1	HC1	HC1
5.5.1	A baseline is established that includes all design and manufacturing data needed to support the consistent replication of the hardware item.	●	●	○	○	Hardware/S/W Interface Data	10.3.2.2.4	HC1	HC1	HC1	HC1
5.5.1	A baseline is established that includes all design and manufacturing data needed to support the consistent replication of the hardware item.	●	●	○	○	Hardware Configuration Management Records	10.7	HC2	HC2	HC2	HC2
5.5.1	Manufacturing requirements related to safety are identified and documented and manufacturing controls are established.	●	●	○	○	Hardware Requirements	10.3.1	HC1	HC1	HC1	HC1
7.1	Configuration items are uniquely identified and documented.	●	●	○	○	Hardware Archive Standards	10.2.4	HC2	HC2	NA	NA
7.1	Configuration items are uniquely identified and documented.	●	●	○	○	Hardware Configuration Management Records	10.7	HC2	HC2	HC2	HC2

Continued

Table 9.19 Continued

Objective		Applicability by IDAL				Output / Associated Evidence		Configuration Control Category by IDAL			
No.	Description	A	B	C	D	Data Item	Refer	A	B	C	D
7.1	Consistent and accurate replication of configuration items is ensured.	●	●	○	○	Hardware Archive Standards	10.2.4	HC2	HC2	NA	NA
7.1	Consistent and accurate replication of configuration items is ensured.	●	●	○	○	Hardware Configuration Management Records	10.7	HC2	HC2	HC2	HC2
7.1	Consistent and accurate replication of configuration items is ensured.	●	●	○	○	Hardware Process Assurance Records	10.8	HC2	HC2	HC2	NA
7.1	A controlled method of identifying and tracking modification to configuration items is provided.	●	●	○	○	Hardware Configuration Management Plan	10.1.5	HC1	HC1	HC2	HC2
7.1	A controlled method of identifying and tracking modification to configuration items is provided.	●	●	○	○	Hardware Configuration Management Records	10.7	HC2	HC2	HC2	HC2
7.1	A controlled method of identifying and tracking modification to configuration items is provided.	●	●	○	○	Problem Reports	10.6	HC2	HC2	HC2	HC2

Blank - Satisfaction of objective is at applicant's discretion

● - The objective should be satisfied with independence (intellectual, not organisational)

○ - The objective should be satisfied

Each of the outputs (or deliverables) also has configuration management objectives:

HC1 : Requires Configuration Identification, Baseline Traceability, Change Control, Retrieval, Protection against Unauthorised Changes, Data Retention, Baselines, Problem Reporting, Change Review, Media Selection/Refreshing/ Duplication, Release.

HC2: Does not require Baselines, Problem Reporting, Change Review, Media Selection/Refreshing/ Duplication, Release.

Table 9.20 **SAE ARP4754A** process assurance deliverables

Objective		Applicability by FDAL					Output/associated evidence		Configuration control category by FDAL				
No.	Description	A	B	C	D	E	Data item	Refs	A	B	C	D	E
1.1	System development and integral processes are defined	R	R	R	R	N	Process Assurance Plan	5.7.2	②	②	②	②	
7.1	Assurance is obtained that necessary plans are developed and maintained for all aspects of system certification	R*	R*	R*	R	N	Evidence of Process Assurance	5.7.3	②	②	②	②	
7.2	Development activities and processes are conducted in accordance with those plans	R*	R*	R*	R	N	Evidence of Process Assurance	5.7.4	②	②	②	②	

A=as negotiated for certification.

N=not required for certification.

R=recommended for certification.

R*=recommended for certification with process independence.

Each of the outputs (or deliverables) also has configuration management objectives:

- ① Requires Configuration Identification, Change Control, Configuration Index, Archiving/Retrieval, Configuration Baseline Establishment and Problem Reporting.
- ② Does not require Configuration Baseline Establishment and Problem Reporting.

Table 9.21 RTCA/DO-178C S/W process assurance

Objective		Applicability by IDAL				Output/associated evidence		Configuration control by IDAL			
No.	Description	A	B	C	D	Data item	Refs	A	B	C	D
A-1	The activities of the S/W lifecycle processes are defined.	○	○	○	○	S/W Quality Assurance Plan	11.5	①	①	②	②
A-1	The S/W lifecycle(s), including the inter-relationships between the processes, their sequencing, feedback mechanisms, and transition criteria, is defined.	○	○	○		S/W Quality Assurance Plan	11.5	①	①	②	
A-1	S/W lifecycle environment is selected and defined.	○	○	○		S/W Quality Assurance Plan	11.5	①	①	②	
A-1	Additional considerations are addressed.	○	○	○	○	S/W Quality Assurance Plan	11.5	①	①	②	②
A-9	Assurance is obtained that S/W plans and standards are developed and reviewed for compliance with this document for consistency.	●	●	●		S/W Quality Assurance Records	11.19	②	②	②	
A-9	Assurance is obtained that S/W plans and standards are developed and reviewed for compliance with this document for consistency.	●	●	●		S/W Verification Plan*	11.3	①	①	②	
A-9	Assurance is obtained that S/W lifecycle processes comply with approved plans.	●	●	●	●	S/W Quality Assurance Records	11.19	②	②	②	②

A-9	Assurance is obtained that S/W lifecycle processes comply with approved plans.	●	●	●	●	Plan for S/W Aspects of Certification*	11.1	①	①	①	①
A-9	Assurance is obtained that S/W lifecycle processes comply with approved plans.	●	●	●	●	S/W Development Plan*	11.2	①	①	②	②
A-9	Assurance is obtained that S/W lifecycle processes comply with approved plans.	●	●	●	●	S/W Verification Plan*	11.3	①	①	②	②
A-9	Assurance is obtained that S/W lifecycle processes comply with approved S/W standards.	●	●	●		S/W Quality Assurance Records	11.19	②	②	②	
A-9	Assurance is obtained that S/W lifecycle processes comply with approved S/W standards.	●	●	●		S/W Requirements Standards*	11.6	①	①	②	
A-9	Assurance is obtained that S/W lifecycle processes comply with approved S/W standards.	●	●	●		S/W Design Standards	11.7	①	①	②	
A-9	Assurance is obtained that S/W lifecycle processes comply with approved S/W standards.	●	●	●		S/W Coding Standards*	11.8	①	①	②	
A-9	Assurance is obtained that transition criteria for the S/W lifecycle processes are satisfied.	●	●	●		S/W Quality Assurance Records	11.19	②	②	②	
A-9	Assurance is obtained that transition criteria for the S/W lifecycle processes are satisfied.	●	●	●		Plan for S/W Aspects of Certification*	11.1	①	①	①	

Continued

Table 9.21 Continued

Objective		Applicability by IDAL				Output/associated evidence		Configuration control by IDAL			
No.	Description	A	B	C	D	Data item	Refs	A	B	C	D
A-9	Assurance is obtained that transition criteria for the S/W lifecycle processes are satisfied.	●	●	●		S/W Development Plan*	11.2	①	①	②	
A-9	Assurance is obtained that transition criteria for the S/W lifecycle processes are satisfied.	●	●	●		S/W Verification Plan*	11.3	①	①	②	
A-9	Assurance is obtained that S/W conformity review is conducted.	●	●	●	●	S/W Quality Assurance Records	11.19	②	②	②	②

Blank – Satisfaction of objective is at applicant's discretion.

● The objective should be satisfied with independence (intellectual, not organisational).

○ The objective should be satisfied.

* Data items are not explicitly listed in the tables in DO-178C, but are implied by the main body text.

Each of the outputs (or deliverables) also has configuration management objectives:

① Requires Configuration Identification, Traceability, Change Control, Retrieval, Protection against Unauthorised Changes, Data Retention, Baselines, Problem Reporting, Change Review, Configuration Status Accounting, Media Selection/Refreshing/Duplication, and Release.

② Does not require Baselines, Problem Reporting, Change Review, Configuration Status Accounting, Media Selection/Refreshing/Duplication, and Release.

Table 9.22 RTCA/DO-254 CEHW process assurance

Objective		Applicability by IDAL				Output/associated evidence		Configuration control by IDAL			
No.	Description	A	B	C	D	Data item	Refs	A	B	C	D
4.1	The hardware design lifecycle processes are defined.	●	●	○	○	Hardware Process Assurance Plan	10.1.6	HC2	HC2	NA	NA
4.1	The means of compliance of the hardware design assurance objectives, including strategies identified using guidance in Section 2.3.4, are proposed to the certification authority.	●	●	○	○	Hardware Process Assurance Plan	10.1.6	HC2	HC2	NA	NA
8.1	Life-cycle processes comply with the approved plans.	●	●	○	○	Hardware Process Assurance Plan	10.1.6	HC2	HC2	NA	NA
8.1	Life-cycle processes comply with the approved plans.	●	●	○	○	Hardware Process Assurance Records	10.8	HC2	HC2	HC2	NA
8.1	Life-cycle processes comply with the approved plans.	●	●	○	○	Hardware Accomplishment Summary	10.9	HC1	HC1	HC1	HC1
8.1	Hardware design lifecycle data produced complies with the approved plans.	●	●	○	○	Hardware Process Assurance Plan	10.1.6	HC2	HC2	NA	NA
8.1	Hardware design lifecycle data produced complies with the approved plans.	●	●	○	○	Hardware Process Assurance Records	10.8	HC2	HC2	HC2	NA

Continued

Table 9.22 Continued

Objective		Applicability by IDAL				Output/associated evidence		Configuration control by IDAL			
No.	Description	A	B	C	D	Data item	Refs	A	B	C	D
8.1	The hardware item used for conformance assessment is built to comply with the associated lifecycle data.	●	●	○	○	Hardware Process Assurance Plan	10.1.6	HC2	HC2	NA	NA
8.1	The hardware item used for conformance assessment is built to comply with the associated lifecycle data.	●	●	○	○	Hardware Process Assurance Records	10.8	HC2	HC2	HC2	NA

Blank – Satisfaction of objective is at applicant's discretion.

● The objective should be satisfied with independence (intellectual, not organisational).

○ The objective should be satisfied.

Each of the outputs (or deliverables) also has configuration management objectives.

HC1: Requires Configuration Identification, Baseline Traceability, Change Control, Retrieval, Protection against Unauthorised Changes, Data Retention, Baselines, Problem Reporting, Change Review, Media Selection/Refreshing/Duplication, and Release.

HC2: Does not require Baselines, Problem Reporting, Change Review, Media Selection/Refreshing/Duplication, and Release.

9.2.5.4 *Certification and Regulatory Authority coordination*

The objective of the certification process is to substantiate that the aircraft and its systems comply with applicable requirements. Planning and coordination with the Regulatory Authority are vital to:

- Establish communication and understanding with the certification authority at the commencement of the project.
- Propose the means of compliance for each objective to the certification authority through certification plans, detailed plans and standards. The level of review of these plans and standards by the certification authority will vary depending on the criticality of the S/W and the previous experience of the developer with working on similar systems with the certification authority (refer to FAA [Order 8110.49](#) and [CM-SWCEH-002](#)).
- Agree the means of compliance for each objective with the certification authority.
- Provide compliance substantiation to the certification authority through the provision of lifecycle evidence for audit purposes at certification authority nominated milestones within the project, and for final certification.

Certification authority liaison intends to establish unambiguous consensus between the certification authority and applicant/developer regarding provision of evidence for demonstrating requirements validity, satisfaction and traceability. It provides early visibility to the applicant/developer as to whether their proposed approach to satisfying DAL objectives will be acceptable to the certification authority. It also provides the certification authority the opportunity to discourage inappropriate approaches to satisfying DAL objectives for new or novel developments for which limited policy or guidance may yet be published.

The development assurance outputs from Certification Authority Liaison is summarised in [Tables 9.23–9.25](#), however this could be extended to include any of the outputs (as negotiated) from Step 1 to Step 4.

Table 9.23 SAE ARP4754A certification authority liaison

Objective		Applicability by Development Assurance Level					Output/associated evidence		Control category				
No.	Description	A	B	C	D	E	Data item	Refs	A	B	C	D	E
1.1	System development and integral processes are defined	R	R	R	R		Certification Plan	5.8.1 5.8.4.1	①	①	①	①	①
8.1	Compliance substantiation is provided	R	R	R	A	N	Certification Summary	5.8.3	①	①	①	②	

A=as negotiated for certification.

N=not required for certification.

R=recommended for certification.

R*=recommended for certification with process independence.

Each of the outputs (or deliverables) also has configuration management objectives:

- ① Requires Configuration Identification, Change Control, Configuration Index, Archiving/Retrieval, Configuration Baseline Establishment and Problem Reporting.
- ② Does not require Configuration Baseline Establishment and Problem Reporting.

Table 9.24 RTCA/DO-178C certification authority liaison

Objective		Applicability by Development Assurance Level				Output/associated evidence		Configuration control category against DAL			
No.	Description	A	B	C	D	Data item	Refs	A	B	C	D
A-1	The activities of the S/W lifecycle processes are defined.	○	○	○	○	Plans for S/W Aspects of Certification	11.1	①	①	①	①
A-1	S/W lifecycle environment is selected and defined.	○	○	○		Plans for S/W Aspects of Certification	11.1	①	①	①	
A-9	Assurance is obtained that S/W plans and standards are developed and reviewed for compliance with this document for consistency.	●	●	●		Plan for S/W Aspects of Certification*	11.1	①	①	①	
A-10	Communication and understanding between the applicant and the certification authority is established.	○	○	○	○	Plan for S/W Aspects of Certification	11.1	①	①	①	①
A-10	The means of compliance is proposed and agreement with the Plans for Aspects of Certification is obtained.	○	○	○	○	Plan for S/W Aspects of Certification	11.1	①	①	①	①
A-10	Compliance substantiation is provided.	○	○	○	○	S/W Accomplish Summary	11.2	①	①	①	①
A-10	Compliance substantiation is provided.	○	○	○	○	S/W Configuration Index	11.16	①	①	①	①

Blank – Satisfaction of objective is at applicant's discretion.

● The objective should be satisfied with independence (intellectual, not organisational).

○ The objective should be satisfied.

* Data Items are not explicitly listed in the tables in DO-178C, but are implied by the main body text.

Each of the outputs (or deliverables) also has configuration management objectives.

① Requires Configuration Identification, Traceability, Change Control, Retrieval, Protection against Unauthorised Changes, Data Retention, Baselines, Problem Reporting, Change Review, Configuration Status Accounting, Media Selection/Refreshing/Duplication, and Release.

② Does not require Baselines, Problem Reporting, Change Review, Configuration Status Accounting, Media Selection/Refreshing/Duplication, and Release.

Table 9.25 RTCA/DO-254 certification authority liaison

Objective		Applicability by Development Assurance Level				Output/associated evidence		Configuration control category against DAL			
No.	Description	A	B	C	D	Data item	Refs	A	B	C	D
4.1	The hardware design lifecycle processes are defined.	●	●	○	○	Plan for Hardware Aspects of Certification	10.1.1	HC1	HC1	HC1	HC1
8.1	Hardware design lifecycle data produced complies with the approved plans.	●	●	○	○	Hardware Accomplishment Summary	10.9	HC1	HC1	HC1	HC1
8.1	The hardware item used for conformance assessment is built to comply with the associated lifecycle data.	●	●	○	○	Hardware Accomplishment Summary	10.9	HC1	HC1	HC1	HC1

Blank – Satisfaction of objective is at applicant's discretion.

● The objective should be satisfied with independence (intellectual, not organisational).

○ The objective should be satisfied.

Each of the outputs (or deliverables) also has configuration management objectives.

HC1: Requires Configuration Identification, Baseline Traceability, Change Control, Retrieval, Protection against Unauthorised Changes, Data Retention, Baselines, Problem Reporting, Change Review, Media Selection/Refreshing/Duplication and Release.

HC2: Does not require Baselines, Problem Reporting, Change Review, Media Selection/Refreshing/ Duplication and Release.

9.3 The Case Study

In Section 2.3, we defined a safety strategy for a modification programme where an aircraft's attitude and altitude systems (see Section 1.3) are upgraded.

With reference to Fig. 2.5, this section explores that part of the strategy duplicated in (Fig. 9.2).

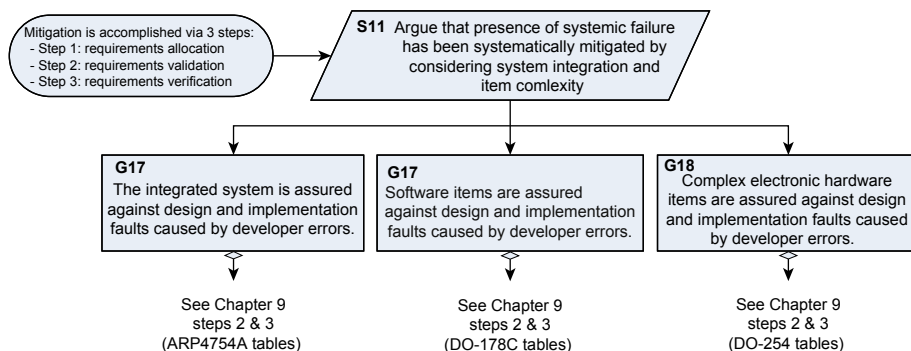


Figure 9.2 Safety strategy.

9.3.1 Step 1: Start-up planning

If we assume that the Design Organisation has mature Process Assurance and Configuration Management procedures (i.e., not project specific), then with reference to Section 9.2.1 the following high-level plans are required in Step 1:

- With reference to Table 9.1, for Level A and Level B functions, the following planning data items are required for the system (preferably at each level of integration):
 - Certification Plan with the objectives:
 - System development and integral processes are defined.
 - Transition criteria and interrelationship among processes are defined.
 - Safety Program Plan with the objectives:
 - System development and integral processes are defined.
 - Transition criteria and interrelationship among processes are defined.
 - Configuration Management Plan.
 - Process Assurance Plan.
- With reference to Table 9.2, for Level A and Level B functions, the following planning data items are required for S/W:
 - Plans for S/W Aspects of Certification with the objectives:
 - The activities of the S/W lifecycle processes are defined.
 - The S/W lifecycle(s), including the interrelationships between the processes, their sequencing, feedback mechanisms and transition criteria, is defined.
 - S/W lifecycle environment is selected and defined.
 - Additional considerations are addressed.

- S/W Development Plan (SDP) with the objectives:
 - The activities of the S/W lifecycle processes are defined.
 - The S/W lifecycle(s), including the interrelationships between the processes, their sequencing, feedback mechanisms and transition criteria, is defined.
 - S/W lifecycle environment is selected and defined.
 - Additional considerations are addressed.
- S/W Requirements Standards with the objective:
 - S/W development standards are defined.
- S/W Design Standards with the objective:
 - S/W development standards are defined.
- S/W Coding Standards with the objectives:
 - S/W development standards are defined.
 - Source Code conforms to standards.
- With reference to [Table 9.3](#), for Level A and Level B functions, the following planning data items are required for S/W:
 - Plan for Hardware Aspects of Certification with the objectives:
 - The hardware design lifecycle processes are defined.
 - Standards are selected and defined.
 - The hardware development and verification environments are selected or defined.
 - The means of compliance of the hardware design assurance objectives, including strategies identified using guidance in Section 2.3.4, are proposed to the certification authority.
 - Hardware Design Plan with the objective:
 - The hardware design lifecycle processes are defined.
 - Standards are selected and defined.
 - The hardware development and verification environments are selected or defined.
 - The means of compliance of the hardware design assurance objectives, including strategies identified, are proposed to the certification authority.
 - Requirements Standards with the objective:
 - Standards are selected and defined.
 - Hardware Design Standards with the objective:
 - Standards are selected and defined.
 - V&V Standards with the objective:
 - Standards are selected and defined.

9.3.2 Step 2: requirements allocation

9.3.2.1 Assign the Development Assurance level

For the purposes of applying the theory of [Section 9.2](#) mentioned earlier to this case study, we focus on the DAL allocation of Figs 4.3 and 4.5. In these FTAs, we have not yet assigned IDALs to the development of the PFDs, and Table 4.1 provides us with the following options:

- Fig. 4.3 has allocated FDAL B to the development of both PFDs, so Table 4.1 allows us the following IDAL options:
 - One PFD at Level B.
 - Both PFDs at Level C.

- For either option, Level B is required for the process to ensure that the two functions are indeed independent.¹⁹
- Fig. 4.5 has allocated FDAL A to the development of both PFDs, so Table 4.1 allows us the following IDAL options:
 - One PFD at Level A.
 - Both PFDs at Level B.
 - For either option, Level A is required for the process to ensure that the two functions are indeed independent.

It is most likely that the PFDs are identical components, so developing both to IDAL A is the most likely scenario.

9.3.2.2 Requirements allocation

With reference to [Section 9.2.2.2](#), this is where we ensure that the integrated altitude display system has requirements allocated to each level of abstraction (refer to Fig. 1.3).

With reference to Figs 1.8 and 1.9, if we specifically look at the components making up the altitude display system, then we will probably find that the PFD is an ETSO item (see [ETSO-C10b](#)). If we then purchase an item with ETSO approval, we do need to tailor our requirements allocation approach by considering three aspects:

- Determine if the behaviours of the selected ETSO item are compatible with the allocated system requirements.
- Determine whether the additional behaviours of the ETSO'd attitude indicator are compatible with the system's safety objectives.
- Acknowledge that the ETSO does not specify the DAL (so two different suppliers may have ETSO-approved PFDs, but developed to different DALs).

To achieve the first two aspects, the system engineer and safety practitioner will need to obtain technical information from the ETSO holder on the behaviour of their component. Whilst software requirements specifications and design descriptions are rarely available (due to intellectual property constraints), the integrator should at least be able to access the technical data on integration and the operator manuals in order to discern the behaviors of the system. Remember though, the focus should not just be the functional behaviors but also the behavior of the system under failure conditions.²⁰

9.3.3 Step 3: requirements validation

With reference to [Section 9.2.2](#), validation is all about asking ourselves: '*Are we building the right thing?*' Therefore any evidence we produce that answers that

¹⁹ For any parts of the system where the information comes together from the two functions, the part of the system that combines them should be to the higher assurance level (e.g., the cross-check logic, if provided, between the two PFDs).

²⁰ One possible way of focussing the assessment is to conduct a black-box SHARD (which is like a S/W HAZOP) that examines the failure modes of data flows based on a standardised taxonomy of failure conditions.

question (i.e., with respect to high-level S/W/hardware requirements, low-level S/W/hardware requirements and implementation of derived behaviours) is considered to be validation evidence. This could be anything from reviews of requirements against system requirements, through to test cases that test a requirement in a realistic scenario and therefore provide validation evidence of the requirement, not just verification evidence of the implementation against the requirement.

The following subsections summarise the Validation Data Items²¹ required:

9.3.3.1 Validation planning

With reference to [Table 9.4](#), for Level A and Level B functions, the following planning data items are required for the system solution:

- Validation Plan (ideally at each level of integration), meeting the following objectives:
 - System development and integral processes are defined.
 - Transition criteria and interrelationship among processes are defined.
- Preliminary Aircraft Safety Assessment:
 - containing (or referring to) the Functional Hazard Assessments, which also contains the list of aircraft-level functions;
 - capturing the independence requirements in functions, systems and items.
- Preliminary SSA:
 - containing (or referring to) the Functional Hazard Assessments;
 - capturing the independence requirements in functions, systems and items.

With reference to [Table 9.5](#), for Level A and Level B S/W, the following planning data items are required:

- Plan for S/W Aspects of Certification (PSAC), the intent of which is to obtain assurance that S/W plans and standards are developed and reviewed for compliance with this document for consistency. The objective of the PSAC is to provide Assurance is obtained that software plans and standards are developed and reviewed for compliance with this document for consistency.

In very simple terms, the PSAC explains how the applicant is going to satisfy each and every objective, by declaring what evidence they will produce for each objective, and how. The points listed below provide the supporting info on 'how':

 - The activities of the S/W lifecycle processes are defined.
 - The S/W lifecycle(s), including the interrelationships between the processes, their sequencing, feedback mechanisms and transition criteria, is defined.
 - S/W lifecycle environment is selected and defined.
 - Additional considerations are addressed.
- S/W Development plan (SDP), which has the same objectives as for the PSAC, all be it more focussed on the development process than the final certification process.

²¹ We need to remember that Project Managers and Engineering Managers manage/sign-off Data Items (which do need to meet predefined objectives). If we cannot allocate objectives to the Data Items, then we risk them signing off a deliverable which fails to meet an objective which they are not aware of. It is therefore useful to look at the 'Objectives' in the tables contained in Section 9.2 from the 'Data Item' or 'deliverable' perspective.

Sometimes the SDP will be included within the PSAC, all as one document, particularly where development scope is very limited.

Using the hierarchy in Fig. 1.3, these items are only required at Level 2.

With reference to [Table 9.6](#), for Level A and Level B CEHW, the following planning data items are required:

- H/W Validation Plan, meeting the following objective(s):
 - The hardware development and verification environments are selected or defined.
 - The means of compliance of the hardware design assurance objectives, including strategies identified using guidance in Section 2.3.4, are proposed to the certification authority.
 - Derived hardware requirements against which the hardware item is to be verified are correct and complete.
- Hardware Test Procedures, showing that:
 - Derived hardware requirements against which the hardware item is to be verified are correct and complete.
 - Derived requirements are evaluated for impact on safety.
- Hardware Review and Analysis Procedures showing that Derived requirements are evaluated for impact on safety.

Using the hierarchy in Fig. 1.3, these items are only required at Level 2.

9.3.3.2 *Validation deliverables*

With reference to [Table 9.7](#), for Level A and Level B functions, the following deliverables are required as evidence of validation accomplishment at each level of system integration (refer to Fig. 1.3):

- Specification at each level of system abstraction containing:
 - list of Functions
 - list of Requirements
 - design Description
- Validation Results
- Validation Summary, including a Validation Matrix.

With reference to [Table 9.8](#), the following S/W (Level 2 in Fig. 1.3) validation evidence is required:

- S/W Requirements Data, which needs to show that:
 - High-level requirements are developed.
 - High-level requirements conform to standards.
 - Derived high-level requirements are defined and provided to the system processes, including the SSA process.
 - Low-level requirements conform to design standards.
- Design Description, which needs to show that:
 - S/W architecture is developed.
 - Low-level requirements are developed and conform to standards.
 - Derived low-level requirements are defined and provided to the system processes, including the SSA process.

With reference to [Table 9.9](#), for Level A and Level B CEHW, the following validation evidence is required:

- Hardware Requirements, meeting the following objective(s):
 - Requirements are identified, defined and documented. This includes allocated requirements from the Preliminary System Safety Assessment (PSSA) and derived requirements from the hardware safety assessment.
 - Derived requirements are fed back to the detailed design process and implementation process (or other appropriate processes).
 - Derived hardware requirements against which the hardware item is to be verified are correct and complete.
 - Derived requirements are evaluated for impact on safety.
 - The hardware item conceptual design is developed consistent with its requirements.
- Problem Reports, meeting the following objective(s):
 - Requirement omissions and errors are provided to the appropriate processes for resolution.
 - Errors and omissions are provided to the appropriate processes for resolution.
- Hardware Acceptance Test Criteria for the derived requirements which have been fed back to the implementation process (or other appropriate processes).
- Hardware Traceability Data for the derived hardware requirements against which the hardware item is to be verified are correct and complete.
- Hardware Review, Analysis and Test Procedures, meeting the following objectives:
 - Derived hardware requirements against which the hardware item is to be verified are correct and complete.
 - Derived requirements are evaluated for impact on safety.
- Hardware Review, Analysis and Test Results showing the results of the execution of these Procedures.

Using the hierarchy in Fig. 1.3, these items are only required at Level 2.

9.3.4 Step 4: requirements verification

With reference to [Section 9.2.3](#), verification is all about asking ourselves: ‘*Did we build the thing right?*’ The following subsections summarise the validation data items required to answer this question.

9.3.4.1 Verification planning

With reference to [Table 9.10](#), for Level A and Level B functions, the following planning data items are required for the system solution:

- Verification Plan, meeting the following objectives:
 - System development and integral processes are defined.
 - Transition criteria and interrelationship among processes are defined.
- Verification Procedures, meeting the following objectives:
 - Test or demonstration procedures are correct.
 - Verification demonstrates intended function and confidence of no unintended function impacts to safety.
 - Product implementation complies with aircraft and system requirements.

With reference to [Table 9.11](#), for Level A and Level B S/W, a S/W Verification Plan is required, meeting the following objectives:

- The activities of the S/W lifecycle processes are defined.
- The S/W lifecycle(s), including the interrelationships between the processes, their sequencing, feedback mechanisms and transition criteria, is defined.
- S/W lifecycle environment is selected and defined.
- Additional considerations are addressed.

With reference to [Table 9.12](#), for Level A and Level B CEHW, the following planning data items are required:

- Hardware Verification Plan, meeting the following objectives:
 - The hardware design lifecycle processes are defined.
 - The hardware development and verification environments are selected or defined.
 - The means of compliance of the hardware Development Assurance objectives, including strategies identified, are proposed to the certification authority.
 - Verification Standards are selected and defined.
- Hardware Review and Analysis Procedures, which is to provide the evidence that the hardware implementation meets the requirements.
- Hardware Test Procedures, which is to provide the evidence that the hardware implementation meets the requirements.

9.3.4.2 Verification deliverables

With reference to [Table 9.13](#), the following deliverables are required as evidence of verification accomplishment at each level of system integration (refer to Fig. 1.3):

- Verification Summary supported by a Verification Matrix, meeting the following objectives:
 - Appropriate item, system and aircraft integrations are performed.
 - Verification compliance substantiation is included.
 - Assessment of deficiencies (including Problem Reports) and their related impact on safety is identified.
- Verification Procedures and Results which have shown that:
 - The test or demonstration procedures are correct.
 - Intended function and confidence of no unintended function impacts to safety.
 - Product implementation complies with aircraft and system requirements.
 - Safety requirements are verified.

With reference to [Table 9.14](#), the following S/W Verification Results need to demonstrate:

- Compliance to the S/W Plans.
- Development and revision of S/W plans are coordinated.
- All requirements (high and low level) are traceable, accurate, consistent and comply (i.e., are verifiable) with system requirements.
- Algorithms are accurate.
- S/W architecture is:
 - Consistent and compatible with high-level requirements.
 - Compatible with target computer.

- Verifiable.
- Conforms with Standards.
- Confirmed to have partitioning integrity.
- S/W Code is:
 - Traceable and in compliance with low-level requirements.
 - In compliance with S/W architecture.
 - Conforms with Standards.
- Source Code:
 - Is developed, validated and verified.
 - Complies with low-level requirements.
 - Complies with S/W architecture.
 - Conforms to standards.
 - Is accurate and consistent.
- Output of S/W integration process is complete and correct.
- Parameter Data Item File is correct and complete.
- Verification of Parameter Data Item File is achieved.
- Executable Object Code:
 - complies with high-level requirements.
 - complies with low-level requirements.
 - is robust with low-level requirements.
 - is compatible with target computer.
- Test procedures are correct and discrepancies explained.
- Test coverage:
 - of high-level requirements is achieved;
 - of low-level requirements is achieved;
 - of S/W structure (modified condition/decision coverage) is achieved.

With reference to [Table 9.14](#), the following CEHW verification evidence is required:

- Detailed Design Data for the produced hardware item, including:
 - Top-Level Drawing
 - Assembly Drawings
 - Installation Control Drawings
- Hardware Requirements (including all design and manufacturing data) have been baselined.
- Hardware Review, Analysis and Test Results verify:
 - Accomplishment of all derived (including safety) requirements.
 - Compliance with the Verification Plan (including meeting the Acceptance Test Criteria).
- Hardware Traceability Data, showing that traceability is established between hardware requirements, the implementation and the verification procedures and results.
- Problem Reports, showing that all omissions and errors are fed back to the appropriate processes for resolution.

9.3.5 Supporting process

9.3.5.1 The Safety Assessment Process

The data items in [Table 9.15](#) should be a standard output from the process defined in Fig. 1.2. As this book is all about the Safety Assessment Process (see Table 1.1), it is

assumed that the SSA is driving the need for Development Assurance requirements and compliance is demonstrated via Table 1.1.

9.3.5.2 Configuration management process

With reference to [Table 9.16](#), the following deliverables are required as evidence of verification accomplishment at each level of system integration (refer to Fig. 1.3):

- CMP showing that system development and integral processes are defined.
- CM Records showing that:
 - Configuration items are identified.
 - Problem reporting, change control, change review and configuration status accounting are established.
 - Archive and retrieval are established.
- Configuration Baseline Records showing that configuration baselines and derivatives are established.
- Configuration Index showing that compliance substantiation is provided.
- Problem reports showing that problem reporting, change control, change review and configuration status accounting are established.

With reference to [Table 9.17](#), at black-box level, the following S/W verification evidence is required:

- S/W Configuration Management Plan showing that:
 - the activities of the S/W lifecycle processes are defined.
 - the S/W lifecycle(s), including the interrelationships between the processes, their sequencing, feedback mechanisms and transition criteria, is defined.
 - S/W lifecycle environment is selected and defined.
 - Assurance is obtained that:
 - S/W lifecycle processes comply with approved plans.
 - S/W plans and standards are developed and reviewed for compliance with this document for consistency.
 - Additional considerations are addressed.
- S/W Configuration Management Records showing that:
 - Configuration items are identified.
 - Baselines and traceability are established.
 - Archive, retrieval and release are established.
 - S/W load control is established.
 - S/W lifecycle environment control is established.
- S/W Configuration Index showing that:
 - Baselines and traceability are established.
 - S/W lifecycle environment control is established.
- Problem Reports showing that problem reporting, change control, change review and configuration status accounting are established.

With reference to [Table 9.18](#), at black-box level, the following CEH verification evidence are required:

- Hardware Configuration Management Plan showing that:
 - the hardware design lifecycle processes are defined.
 - the hardware development and verification environments are selected or defined.

- the means of compliance of the hardware design assurance objectives, including strategies identified using guidance in RTCA DO-254 Section 2.3.4, are proposed to the certification authority.
- a controlled method of identifying and tracking modification to configuration items is provided.
- Hardware Archive Standards showing that:
 - Standards are selected and defined.
 - Requirements are identified, defined and documented. This includes allocated requirements from the PSSA and derived requirements from the hardware safety assessment.
 - A baseline is established that includes all design and manufacturing data needed to support the consistent replication of the hardware item.
 - Configuration items are uniquely identified and documented.
 - Consistent and accurate replication of configuration items is ensured.
- Hardware Requirements showing that:
 - A baseline is established that includes all design and manufacturing data needed to support the consistent replication of the hardware item.
 - Manufacturing requirements related to safety are identified and documented and manufacturing controls are established.
- Top-Level Drawing and Assembly Drawings all showing that a baseline is established that includes all design and manufacturing data needed to support the consistent replication of the hardware item.
- Installation Control Drawings and Hardware/S/W Interface Data showing that a baseline is established that includes all design and manufacturing data needed to support the consistent replication of the hardware item.
- Hardware Configuration Management Records showing that:
 - A baseline is established that includes all design and manufacturing data needed to support the consistent replication of the hardware item.
 - Configuration items are uniquely identified and documented.
 - Consistent and accurate replication of configuration items is ensured.
 - A controlled method of identifying and tracking modification to configuration items is provided.
- Hardware Process Assurance Records showing that consistent and accurate replication of configuration items is ensured.
- Problem Reports showing a controlled method of identifying and tracking modification to configuration items is provided.

9.3.5.3 Process Assurance

With reference to [Section 9.2.4.3](#), Process Assurance asks questions such as:

- *‘Were the plans followed, or just put on the shelf and ignored?’*
- *‘Does the development process meet any required standards?’*
- *‘Are best practices used to develop the product?’*

With reference to [Table 9.19](#), the following deliverables are required as evidence of process assurance at each level of system integration (refer to [Fig. 1.3](#)):

- A Process Assurance Plan
- Evidence of Process Assurance

With reference to [Table 9.20](#), the following deliverables are required as evidence of S/W development process assurance:

- S/W Quality Assurance Plan, meeting the following objectives:
 - The activities of the S/W lifecycle processes are defined.
 - The S/W lifecycle(s), including the interrelationships between the processes, their sequencing, feedback mechanisms and transition criteria, is defined.
 - S/W lifecycle environment is selected and defined.
 - Additional considerations are addressed.
 - Assurance is obtained that S/W plans and standards are developed and reviewed for compliance with this document for consistency.
- S/W Quality Assurance Records meeting the following objectives:
 - Assurance is obtained that S/W lifecycle processes comply with approved plans.
 - Assurance is obtained that S/W lifecycle processes comply with approved S/W standards.
 - Assurance is obtained that transition criteria for the S/W lifecycle processes are satisfied.
- Assurance is obtained that S/W lifecycle processes comply with the following approved plans.
 - Plan for S/W Aspects of Certification
 - S/W Coding Standards
 - S/W Design Standards
 - S/W Development Plan
 - S/W Requirements Standards
 - S/W Verification Plan

With reference to [Table 9.21](#), the following deliverables are required as evidence of complex hardware development process assurance:

- Hardware Process Assurance Plan showing that:
 - The hardware design lifecycle processes are defined.
 - The means of compliance of the hardware design assurance objectives, including strategies identified, are proposed to the certification authority.
 - Lifecycle processes comply with the approved plans.
 - Hardware design lifecycle data produced complies with the approved plans.
 - The hardware item used for conformance assessment is built to comply with the associated lifecycle data.
- Hardware Process Assurance Records showing that:
 - Life-cycle processes comply with the approved plans.
 - Hardware design lifecycle data produced complies with the approved plans.
 - The hardware item used for conformance assessment is built to comply with the associated lifecycle data.
- Hardware Accomplishment Summary showing that lifecycle processes comply with the approved plans.

9.3.5.4 Certification and Regulatory Authority coordination

As discussed in Section 9.2.4.4, planning and coordination with the Regulatory Authority are vital to efficiently and effectively achieve eventual certification. [Tables 9.22–9.24](#), provide a list of data items against this topic, however all the other tables

(i.e., [Table 9.1–9.21](#)) also contain data items which need to be available whenever the Regulatory Authority needs to see it.

- Note: *Once the DAL has been allocated (see Section 9.3.11), the certification authority liaison process) should commence, usually through the development, approval and agreement of either a PSAC or equivalent document. Usually, additional documents such as the SDP, S/W Requirements Standards, S/W Design Standards, S/W Coding Standards, and so on need to be prepared to support agreement by the certification authority with the PSAC. The PSAC will outline the proposed means of satisfaction, and associated evidence for each applicable Level A objective; in this case, all 66 Objectives, including the independence requirements. The PSAC and supporting documents will be developed by the S/W team, however the systems and safety engineers should be part of the review process of these document to assure the S/W approach accords with system-level expectations.*

9.4 Discussion

9.4.1 Development Assurance and the Validation and Verification lifecycle

In [Section 9.2](#), we attempted to allocate the ARP4754A, RTCA/DO-178C and RTCA/DO-254 outputs to each of the suggested four steps in [Fig. 9.1](#). The reader might have gathered by now that this allocation is not that easy and is very subjective.²² The reasons for this include:

- The lack of continuity in the three standards:
 - [SAE ARP4754A](#) was only updated in 2010, when the relationship with the other two standards was made explicit for the first time. It is likely that this document will be revised to become ARP 4754B by 2018.
 - RTCA/DO-178C has evolved over a number of years and has not yet been updated to interface more optimally with ARP 4754A.
 - RTCA/DO-254 has a format very different to 0178C, especially in the fact that the Objective Tables in the annexes of the other two documents are absent (although it can be derived from the embedded narrative within the main body of the standard).
- Development Assurance concerns itself with both Validation and Verification (V&V), but the split between the two is not easy to make explicit using either ‘Objectives’, nor using the ‘Data Items’. The reason for this is because the majority of ‘Objectives’ to some extent address both V&V, and the ‘Data Items’ contain evidence from both V&V activities. For instance, when we run a test case under DO-178C, we are assessing both V&V. The choice as to how we then interpret the results of that test case depends on the results of the V&V assessment:
 - some results will drive us to seek a verification solution (i.e., fix an implementation);
 - other results will drive us to seek a validation solution (i.e., better define or even change the requirements).

The difficulty with breaking the objectives out of the ARP4754A, RTCA/DO-178C and RTCA/DO-254 groups is that it makes these relationships a little less explicit, especially when we consider the reality of the development process. For instance, during S/W

²² See www.aircraftsystemsafety.com (the page supporting this book) for an up-to-date version of these tables in MS Excel format. These can be downloaded and re-allocated to suit individual needs and perspectives.

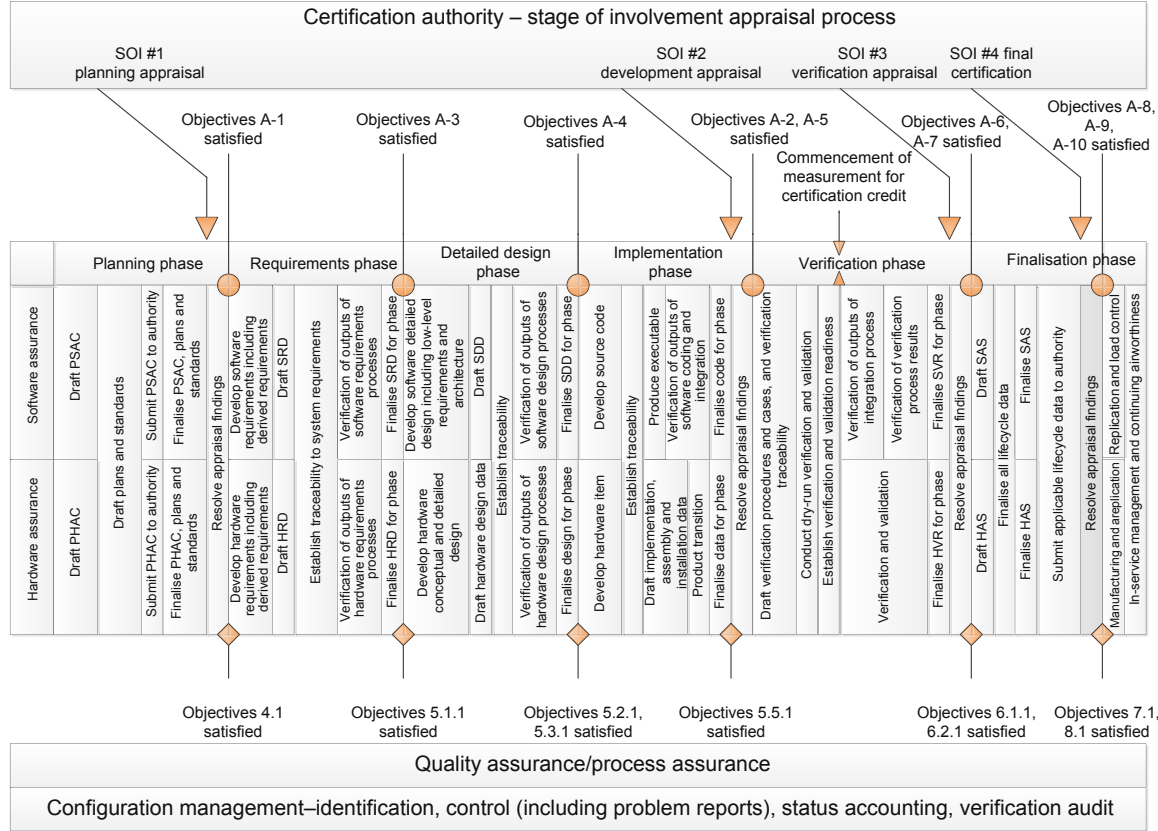


Figure 9.3 Example of RTCA/DO-178B/C and RTCA/DO-254 lifecycles and objective satisfaction.

development for many successful programs, a significant proportion of validation is completed pre-CDR, with the good intention to focus on verification activities post CDR. However, reality shows that most programs still conduct a significant amount on effort doing validation post CDR for S/W requirements. This is probably one reason why standards like DO-178C and DO-254 use such integrated models (e.g., see Fig. 6.1 in DO-178C), which appears very complex to those not intimately familiar with the standards.

If we consider RTCA/DO-178C and RTCA/DO-254 only, then Fig. 9.3 shows the typical progression of activities for both S/W and Hardware Assurance. The following key information is represented:

- Interactions with the Certification Authority and their S/W Approval Guidelines.
- Milestones where applicable Objectives should be satisfied by the lifecycle data being produced.
- The general sequence of activities that should be undertaken.

9.4.2 Development Assurance and the System Development Process

Development Assurance processes do not exist in isolation of system development processes. There are important information flows between them as illustrated in Fig. 9.4. These information flows are important to be understood, so that the developers understand their inputs, outputs and dependencies:

- The information which flows from the System Development Process to S/W/Hardware Design Life Cycle Process includes:
 - System, design and safety requirements allocated to S/W and hardware (refer to Fig. 1.3, Table 4.1).
 - DAL for each function, along with its associated requirements and failure conditions, if applicable (refer to Chapters on ARP4754A System Assurance and Fault Tree Analysis).
 - Hardware/S/W interface description, so that dependencies with the S/W lifecycle processes may be understood.

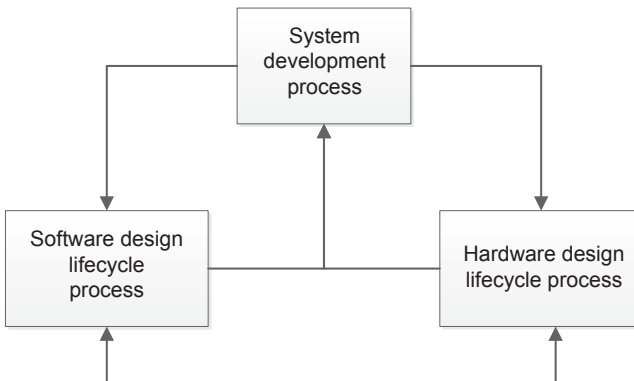


Figure 9.4 System development information flows.

- Requirements for safety strategies and design constraints, such as external interfaces, partitioning requirements, testability, design methods and hardware architectures.
- Requirements for system verification activities to be performed by S/W or hardware-level verification.
- Evidence of the acceptability of any lifecycle data provided by S/W or hardware processes to the system processes, including evaluations of derived requirements for impact on SSA and system requirements and issues raised by S/W or hardware processes in relation to system requirements allocated to S/W or hardware.
- Evidence for S/W or hardware verification activities performed by system lifecycle processes.
- Installation, ergonomic and environmental requirements allocated to S/W or hardware.
- Integration problem reports that may have an impact on requirements. These may arise as a result of activities, such as system verification, generation of system requirements or SSA.
- The information which flows from S/W/Hardware Design Life Cycle Process to System Development Process includes:
 - Implementation of the requirements, such as designs, drawings, schematics, code and parts lists.
 - Software or Hardware derived requirements that may have an impact on any allocated requirement.
 - Implementation architecture, including partitioning and fault containment boundaries.
 - Evidence of any required system V&V activities performed during the S/W or hardware design lifecycle.
 - Product safety analysis data, such as:
 - Probabilities and failure rates for designated hardware functional failures of concern to the SSA process.
 - Common mode fault analysis.
 - Isolation boundaries and generic fault mitigation strategies.
 - Latency analysis data relevant to system requirements. Examples are provisions for fault monitoring, fault detection intervals and undetectable faults.
 - Performance, timing and accuracy characteristics.
 - Limitations for use.
 - Lifecycle data to support integration of the S/W or hardware into the system.
 - Requirements for S/W or hardware verification activities to be performed by system-level verification.
 - Assumptions and analysis methods regarding installation requirements and environmental conditions necessary for the analyses to be valid.
 - Problem or change reports that may have an impact on system, S/W or allocated hardware requirements, and identified incompatibilities between the hardware and S/W.
 - Configuration identification and any configuration status accounting constraints.
- The information flow between Hardware Design Life Cycle Process and S/W Life Cycle Process includes:
 - Derived requirements needed for hardware/S/W integration, such as definition of protocols, timing constraints, and addressing schemes for the interface between hardware and S/W.
 - Instances where hardware and S/W verification activities require coordination.
 - Identified incompatibilities between the hardware and the S/W, which may be part of a reporting and corrective action system.
 - Safety assessment data that should also be made available to system processes.

Development Assurance should be carried out across the entire project lifecycle. While elements of it can be achieved retrospectively, this can be problematic as problems experienced downstream (e.g., during system integration verification) are symptoms of neglect upstream (e.g., requirements validation). Upstream problems can only be solved upstream. For economical reasons, the ability to influence a system's characteristics diminishes very rapidly as the system proceeds from one phase of its lifecycle to the next.

While we are mentioning economics, no discussion of the application of different level of DAL can be complete without considering the cost and programme implications. [Hilderman \(2009\)](#) provides a very useful guide, which can be summarised as follows:

- Consider Level E certification to be the baseline costs.
- Level D certification would add about 5% to this baseline cost.
- Level C certification would add about 30% to Level D cost.
- Level B certification would add about 15% to Level C cost.
- Level E certification would add about 5% to Level B cost.

9.5 Conclusions

This chapter has examined the general principles of the application of Development Assurance across system, S/W and electronic hardware development. It has summarised that Development Assurance is a methodology for providing confidence in the behaviours and properties of a system or item, based on setting objectives and configuration control requirements for the lifecycle evidence, and providing a process for establishing that the evidence satisfies the applicable objectives. The purpose of Development Assurance is to provide evidence that systematic failures do not unacceptably contribute to aircraft-level failure modes.

The objective approach within ARP4754A, RTCA/DO-178C and RTCA/DO-254 has been summarised, and related to the general principles of Development Assurance, which provides confidence, through the provision of relevant evidence, that:

- the system could have the right behaviours (systematically) with respect to safety objectives under normal and failure modes of system operation;
- unacceptable errors leading to undesirable behaviours were not introduced in development that might impact safety objectives.

To achieve this confidence, Development Assurance involves the application of techniques and methods, commensurate with the worst credible failure, to generate specific evidence with respect to attributes of the lifecycle and product of the following:

- Requirements Validity (i.e., *Does the equipment have the right behaviours?*)
- Requirements Satisfaction (i.e., *Are the required behaviours implemented in the product?*)
- Requirements Traceability (i.e., *Are the behaviours of the equipment fully accounted for, and are there any additional behaviours which could lead to a hazard?*)
- Non-Interference (i.e., *Are any of the behaviours necessary for safety interfered with by other behaviours of the equipment?*)

- Configuration Consistency (i.e., *Do the evidence produced throughout the lifecycle have traceability to the delivered product?*)
- Design Integrity (i.e., *Have reasonable steps been employed to adopt techniques and methods that contribute to design integrity rather than providing opportunities for vulnerabilities to be introduced that might be a source of counter evidence to requirements validity or satisfaction?*)

Development Assurance standards do not provide complete prescription of the activities and evidence required to demonstrate that the system is acceptably safe. This is both an advantage and limitation, with the advantage being that developers are afforded the flexibility to propose how they will satisfy the objectives (i.e., using technique, methods, tools and approaches they are comfortable with). The following subsections summarise a number of other advantages and disadvantages of the Development Assurance approach.

9.5.1 Advantages

- Objectives provide outcomes, leaving the design organisation some flexibility to select methods and tools that suit their specific needs (as we have done in Sections 9.2.1–9.2.5).
- Provides a systematised framework (driven by robust requirements V&V) for reasoning that the equipment will execute in the operational context with an acceptable level of safety. The systematised framework provides confidence to certification authorities and developers that the most important properties have been addressed with suitable evidence.
- Evidence based, including both product and process evidence, providing both confidence in the product as well as in the evidence that supports the product.
- Avoids a ‘process only’ compliance methodology (where the assumption made is that good process=good product), which has been shown to be problematic and often not true. By making sure the objectives relate to product properties of the S/W (such as requirements, source code and so on), this gives product output perspective in addition to the process controls.
- Widely accepted in the international aviation community:
 - Based on the methodology put in place in the original issue of DO-178B (circa 1992), with many years of industrial practice on various systems. DO-178B put in place a new way of using objectives and evidence, and when this was critically reviewed by SC-205 (to write DO-178C) the approach was upheld, with only some refinement. The success of the objective methodology was reaffirmed in the structure of DO-254 (c.2000) and ARP4754A (in 2010).
 - Was the result of consensus across a large number of certification authority and industry representatives, meaning that industry has been widely engaged in the establishment of these standards for Development Assurance.

9.5.2 Limitations

Development Assurance of critical systems is an onerous undertaking and is therefore often questioned by those that either do not understand the DAL objectives or the underlying arguments behind the standard assurance frameworks which is being applied:

- The lack of complete prescription of the activities and evidence required to demonstrate that system is acceptably safe is sometimes seen as a limitation when developers are looking to the standards for complete guidance on how to execute the requirements. Development

Assurance standards are by no means at the pinnacle of their evolution; however, in the absence of systematised framework for Development Assurance, developers of systems would be without a structured approach for demonstrating that their systems are acceptably safe.

- The approach is not always applied consistently by the design organisation, sometimes leading to rework when the design organisation engages with their certification authority. Stakeholder expectations often vary.
- The approach does require a thorough understanding of the objectives and the interrelationships of the objectives, which is not necessarily obvious from reading the standards in isolation without a reasonable level of training in the application of the standards and the associated certification authority guidance.
- The achievement of the IDAL does not imply achievement of the allocated Failure Probability Objective, and thus a failure probability cannot be claimed (or used in higher-level analysis) for DAL items within fault trees.

References

- Order 8110.49, 2003. Software Approval Guidelines. Federal Aviation Administration, United States Department of Transportation, Washington, DC.
- ASSC, January 2009. IPT Guidance for Acquisition of Systems With Complex Programmable Hardware Using DO-254, Issue 2, Defence Equipment and Support Safety Engineering. www.ASSCOnline.co.uk.
- Avizienis, A., Laprie, J., Randell, B., Landwehr, C., January–June 2004. Basic concepts and taxonomy of dependable and secure computing. *IEEE Transactions on Dependable and Secure Computing* 1 (1).
- Barnes, J., 2002. High Integrity Software: The SPARK Approach to Safety and Security. Addison-Wesley, Great Britain.
- CM-SWCEH-002, 2011. Software & Complex Electronic Hardware Selection, Certification Memorandum, Issue 01. EASA, Head of Certification Experts Department.
- CAST-27, June 2006. Clarification on the Use of RTCA Document DO-254 and EUROCEA Document ED-80, Design Assurance Guidance for Airborne Electronic Hardware. FAA Position Paper.
- Dupay, A., Leveson, N., October 2000. An Empirical Evaluation of the MCDC Coverage Criterion on the HETE-2 Satellite Software. Proceedings of the Digital Aviation Systems Conference, Philadelphia, USA.
- ETSO-C10b, 2003. Aircraft Altimeter Pressure Actuated Sensitive Type, European Technical Standard Order. EASA.
- Hilderman, V., DO-0178B, 2009. Costs versus Benefits, HighRely Whitepaper. Morten Ave, Arizona, USA. http://www.cems.uwe.ac.uk/~a2-lenz/n-gunton/hrwp_do_178b_cost_benefit.pdf.
- Hammett, R., 2001. Design by Extrapolation – an Evolution of Fault Tolerance Avionics. The Charles Stark Draper Laboratory, Cambridge, Massachusetts.
- Hayhurst, K., Veerhusen, D., 2001. In: A Practical Approach to Modified Condition Decision Coverage, 20th Digital Avionics Systems Conference, Daytona Beach, Florida, USA, 14–18 October 2001.
- Hitt, E.F., Mulcare, D., 2001. Fault tolerance avionics, Chapter 28. In: R Spitzer, C. (Ed.), *Digital Avionics Handbook*. CRC Press.

- Jackson, K., 1988. MASCOT with Other Methods, Software Technology Centre. SD-Scicon, Camberley, UK.
- Kerrel, T., Ferrell, U., 2001. RTCA DO-178/EUROCAE ED12B, Chapter 27. In: The Avionics Handbook. CRC Press LLC. http://www.davi.ws/avionics/TheAvionicsHandbook_Cap_27.pdf.
- Knight, J.C., Graydon, P.J., 2007. Engineering, Communication, and Safety, Department of Computer Science. University of Virginia.
- Leveson, N., 1995. Safeware: System Safety and Computers, Reading, Mass Addison Wesley.
- McDermid, J., 2012. Safety Critical Software, Article Published in the Encyclopaedia of Aerospace Engineering. John Wiley & Sons, Ltd. http://www-module.cs.york.ac.uk/casa/R_eae506.pdf.
- Marks, P., February 9, 2008. Flight of the Software Bugs, in New Scientist, pp. 26–28.
- Pumfrey, D.J., 1999. The Principled Design of Computer System Safety Analyses, Department of Computer Science. University of York.
- SAE ARP4754A, 2010. Guidelines for Development of Civil Aircraft and Systems. SAE International, Warrendale, PA.
- Software System Safety Handbook, 2010. Joint Services Computer Resources Management Group: U.S. Navy, U.S. Army, and the U.S. Air Force, Joint Services Software Safety Committee, Joint Services System Safety Panel. Electronic Industries Association, G-48 Committee. <https://acc.dau.mil/CommunityBrowser.aspx?id=683698>.
- Salmon, C., Lee, C., 2006. The Certification of Software Containing Software Developed Using RTCA/DO-178B, Avionics Systems Standardisation Committee. Ref. ASSC/12/0013 Issue 3, ERA Report 2006-0036 Issue 3, ERA Project 7D0134809. ERA Technology Ltd, UK.
- Vilkomir, S.A., Bowen, J.P., 2002. From MCDC to RCDC: Formalisation and Analysis of Control Flow Testing Criteria. South Bank University, UK.
- Weaver, R.A., 2003. The Safety of Software – Constructing and Assuring Arguments. Department of Computer Science, University of York.

Further reading

- AC20-152, June 30, 2005. Design Assurance Guidance for Airborne Electronic Hardware. Federal Aviation Administration.
- ANM-03-117-09, January 14, 2004. Policy Statement on Guidance for Determination of System, Hardware, and Software Development Assurance Levels on Transport Category Airplanes. FAA Memorandum.
- CAST-28, December 2006. Frequently Asked Question on the Use of RTCA Document DO-254 and EUROCAE Document ED-80, Design Assurance Guidance for Airborne Electronic Hardware. FAA Position Paper.
- CAST-29, February 2007. Use of COST Graphical Processors (CPG) in Airborne Display Systems. FAA Position Paper.
- CAST-30, August 2007. Simple Electronic Hardware and RTCA Document DO-254 and EUROCAE Document ED-80, Design Assurance Guidance for Airborne Electronic Hardware. FAA Position Paper.
- Lange, M. Understanding DO-254 and Solutions to Facilitate Compliance, Mentor Graphics Article. www.mentor.com.
- Software Considerations in Airborne Systems and Equipment Certification, 2011. RTCA Inc., Washington, DC (RTCA/DO-178C).
- Supporting Information for DO-178C and DO-278A, 2001. RTCA Inc., Washington, DC.
- Design Assurance Guidance for Airborne Electronic Hardware, 2000. RTCA Inc., Washington, DC.

ANNEXES

Annex A: specific approach to hardware Development Assurance

(A1) Simple or complex hardware

RTCA/DO-254 uses the notion of Simple or Complex hardware in order to determine whether the objectives of RTCA/DO-254 need to be applied in full. In general terms, a hardware item is classified as simple if, and only if, a comprehensive combination of deterministic tests and analyses appropriate to the DAL can ensure correct functional performance under all foreseeable operating conditions with no anomalous behaviour. To be deterministic, the hardware item needs to have a manageably small spectrum of input data, states and modes, such that effectively all combinations of these can be tested or analysed within a feasible timeframe. If this is not the case, then the item should be classified as complex.

(A2) Development Assurance methods

RTCA/DO-254 defines criteria for establishing what Development Assurance methods should be applied. The key criterion (in addition to the Simple or Complex decision discussed earlier) is if the function or subfunction (see next section):

- is Level A/B (see the following discussion)
- is Level C/D (in which case the more onerous development assurance methods may be used for Level C/D but are not required) or
- has no safety effect (in which case no further action is needed).

For Level A/B then the following Development Assurance methods are permitted:

- **Architectural Mitigation:** Architectural design features, such as dissimilar implementation, redundancy, monitors, isolation, partitioning and command/authority limits, can be specifically employed to mitigate or contain the adverse effects of hardware design and implementation errors. Architectural mitigation is performed by identifying Functional Failure Paths (FFPs) (see the following section) associated with a proposed hardware implementation, and then analysing design options to identify and propose design features and strategies that mitigate the effects of these FFPs.
- **Product Service Experience:** Service experience (i.e., previous or current usage of the component) may be used to substantiate Development Assurance for previously developed hardware and for COTS components, where change is not introduced within this application. Note that in this context, this is direct product service history evidence, and not colloquial here say evidence. Data from non-airborne applications is not specifically excluded under RTCA/DO-254, although often these applications lack the necessary failure reporting data in order to be able to assess the product service experience. RTCA/DO-254 defines criteria on how to evaluate product service history. It should be noted that this is an onerous process, and many product service history cases do not have sufficient data in order to successfully evaluate product service history. Therefore, the architectural mitigation or advance verification methods must be used.

- **Advanced Verification Methods:** RTCA/DO-254 defines (but does limit to) three advance verification methods which may be used to satisfy the objectives for Level A and B hardware. These are Elemental Analysis, Safety-Specific Analysis and Formal Methods, and are summarised as follows:
 - *Elemental Analysis.* Elemental analysis provides a measurement of the completeness of the hardware verification from a bottom-up perspective. Every functional element within the FFP is identified and verified using verification test cases that meet the verification objectives of Section 6.1. The analysis may also identify areas of concern that need to be addressed by other appropriate means.
 - *Safety-Specific Analysis.* This strategy focuses on exposing and correcting the design errors that could adversely affect the hardware outputs from a system-safety perspective. Applicable safety-sensitive portions of the hardware input space and output space are analytically determined. The sensitive portions of the hardware input space are stimulated, and the output space is observed not only for the safety-sensitive intended-function requirements verification, but also for anomalous behaviours. The methods of output space observation are identified in advance, by analysis that is accomplished using traditional safety analysis techniques.
 - *Formal Methods.* Formal Methods employ techniques from formal logic and discrete mathematics for the specification, design and verification of computer systems. These techniques may be used to substantiate the reasoning employed in various processes of the hardware design lifecycle.

(A3) Functional Failure Path analysis

RTCA/DO-254 defines Functional Failure Path (FFP) as the specific set of interdependent circuits that could cause a particular anomalous behaviour in the hardware that implements the function or in the hardware that is dependent upon the function. FFP Analysis (FFPA) is used to iteratively decompose the hardware functions into a hierarchy of subfunction to determine if it will be possible to fulfil completely the objectives of RTCA/DO-254 for each subfunction. If the assurance lifecycle data available or expected to be available is complete, correct and acceptable per the RTCA/DO-254 objectives and guidance, then no further decomposition is necessary. If it is not, then decomposition continues until such a stage as the FFP feasibly maps to one of the Development Assurance methods (and associated data set) as described in the previous section. For FFPs that are not Levels A or B, their interrelationships with the Level A or B FFPs should be evaluated using an F-FMEA, common mode analysis or dependency diagram to ensure that the Level A and B FFPs cannot be adversely impacted by the FFPs which are not Level A or B.

The process can be summarised as illustrated in [Fig. 9A.1](#).

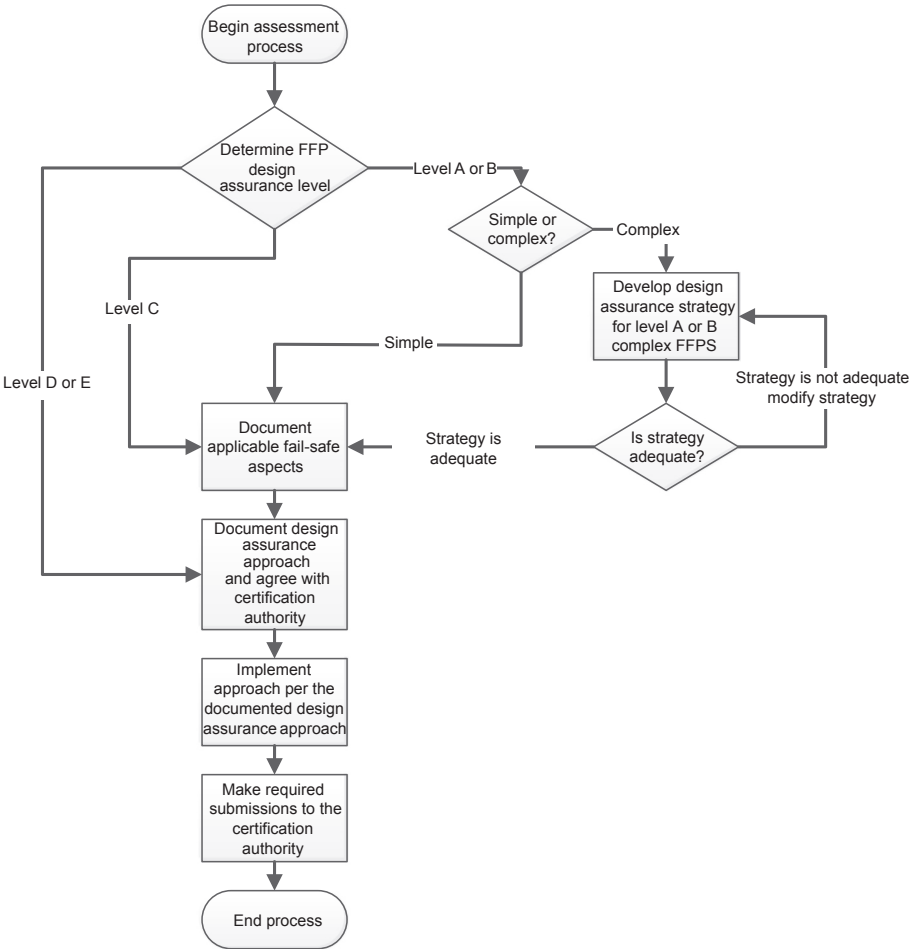


Figure 9A.1 RTCA/DO-254 FFP assessment process.

(A4) Objectives and lifecycle data outputs

Table 9A.1 shows each RTCA/DO-254 objective mapped to the corresponding lifecycle data (evidence) required to satisfy it. This table is not explicitly included in RTCA/DO-254, however it has been produced consistent with the RTCA/DO-178C table format in order to show the relationship between objectives and lifecycle data, and the synergies that exist between the standards. RTCA/DO-254 provides further definition of the lifecycle data outputs/evidence, which the reader should refer to understand the information requirements of the lifecycle data outputs.

The factors that distinguish hardware DALs are the level of independence required for Levels A and B, which is not required for Levels C and D (as specified by the Applicability by Development Assurance-Level columns), and the degree of control

Table 9A.1 RTCA/DO-254 objectives

	Objective	Applicability by Design Assurance Level				Output		Control category			
		A	B	C	D	Data item	Refs	A	B	C	D
4.1	Hardware planning										
(1)	The hardware design lifecycle processes are defined.	●	●	○	○	Plan for Hardware Aspects of Certification	10.1.1	HC1	HC1	HC1	HC1
						Hardware Design Plan	10.1.2	HC2	HC2	HC2	NA
						Hardware Validation Plan	10.1.3	HC2	HC2	HC2	NA
						Hardware Verification Plan	10.1.4	HC2	HC2	HC2	HC2
						Hardware Configuration Management Plan	10.1.5	HC1	HC1	HC2	HC2
						Hardware Process Assurance Plan	10.1.6	HC2	HC2	NA	NA
(2)	Standards are selected and defined.	●	●	○	○	As for (1), and:					
						Requirements Standards	10.2.1	HC2	HC2	NA	NA
						Hardware Design Standards	10.2.2	HC2	HC2	NA	NA
						Verification and Validation Standards	10.2.3	HC2	HC2	NA	NA
						Requirements Standards	10.2.1	HC2	HC2	NA	NA
						Hardware Design Standards	10.2.2	HC2	HC2	NA	NA
						Validation and Verification Standards	10.2.3	HC2	HC2	NA	NA
						Hardware Archive Standards	10.2.4	HC2	HC2	NA	NA

Continued

Table 9A.1 Continued

	Objective	Applicability by Design Assurance Level				Output		Control category			
	Description	A	B	C	D	Data item	Refs	A	B	C	D
(3)	The hardware development and verification environments are selected or defined.	●	●	○	○	Plan for Hardware Aspects of Certification	10.1.1S	HC1	HC1	HC1	HC1
						Hardware Design Plan	10.1.2	HC2	HC2	HC2	NA
						Hardware Validation Plan	10.1.3	HC2	HC2	HC2	NA
						Hardware Verification Plan	10.1.4S	HC2	HC2	HC2	HC2
						Hardware Configuration Management Plan	10.1.5	HC1	HC1	HC2	HC2
(4)	The means of compliance of the hardware design assurance objectives, including strategies identified using guidance in Section 2.3.4, are proposed to the certification authority.	●	●	○	○	Plan for Hardware Aspects of Certification	10.1.1S	HC1	HC1	HC1	HC1
						Hardware Design Plan	10.1.2	HC2	HC2	HC2	NA
						Hardware Validation Plan	10.1.3	HC2	HC2	HC2	NA
						Hardware Verification Plan	10.1.4S	HC2	HC2	HC2	HC2
						Hardware Configuration Management Plan	10.1.5	HC1	HC1	HC2	HC2
						Hardware Process Assurance Plan	10.1.6	HC2	HC2	NA	NA
5.1.1	Requirements Capture										
(1)	Requirements are identified, defined and documented. This includes allocated requirements from the PSSA and derived requirements from the hardware safety assessment.	●	●	○	○	Hardware Archive Standards	10.2.4	HC2	HC2	NA	NA
						Hardware Requirements	10.3.1	HC1	HC1	HC1	HC1

(2)	Derived requirements produced are fed back to the appropriate process.	●	●	○	○	Hardware Requirements	10.3.1	HC1	HC1	HC1	HC1
(3)	Requirement omissions and errors are provided to the appropriate process for resolution.	●	●	○	○	Problem Reports	10.6	HC2	HC2	HC2	HC2
5.2.1	Conceptual Design										
(1)	The hardware item conceptual design is developed consistent with its requirements.	●	●	○	○	Conceptual Design Data	10.3.2.1	HC2	HC2	NA	NA
(2)	Derived requirements produced are fed back to the requirements capture or other appropriate processes.	●	●	○	○	Hardware Requirements	10.3.1	HC1	HC1	HC1	HC1
(3)	Requirement omissions and errors are provided to the appropriate processes for resolution.	●	●	○	○	Problem Reports	10.6	HC2	HC2	HC2	HC2
5.3.1	Detail Design										
(1)	The detailed design is developed from the hardware item requirements and conceptual design data.	●	●	○	○	Detailed Design Data	10.3.2.2	HC1	HC1	HC1	HC1
						Top-Level Drawing	10.3.2.2.1S	HC1	HC1	HC1	HC1
						Assembly Drawings	10.3.2.2.2.2	HC1	HC1	HC1	HC1
						Hardware/S/W Interface Data	10.3.2.2.4	HC1	HC1	HC1	HC1
(2)	Derived requirements are fed back to the conceptual design process or other appropriate processes.	●	●	○	○	Hardware Requirements	10.3.1	HC1	HC1	HC1	HC1
(3)	Requirement omissions or errors are provided to the appropriate processes for resolution.	●	●	○	○	Problem Reports	10.6	HC2	HC2	HC2	HC2

Continued

Table 9A.1 Continued

	Objective	Applicability by Design Assurance Level				Output		Control category			
	Description	A	B	C	D	Data item	Refs	A	B	C	D
5.4.1	Implementation										
(1)	A hardware item is produced which implements the hardware detailed design using representative manufacturing processes.	●	●	○	○	Detailed Design Data	10.3.2.2	HC1	HC1	HC1	HC1
						Top-Level Drawing	10.3.2.2.1S	HC1	HC1	HC1	HC1
						Assembly Drawings	10.3.2.2.2.2	HC1	HC1	HC1	HC1
						Installation Control Drawings	10.3.2.2.3	HC1	HC1	HC1	HC1
(2)	The hardware item implementation, assembly and installation data is complete.	●	●	○	○	Detailed Design Data	10.3.2.2	HC1	HC1	HC1	HC1
						Top-Level Drawing	10.3.2.2.1S	HC1	HC1	HC1	HC1
						Assembly Drawings	10.3.2.2.2.2	HC1	HC1	HC1	HC1
						Installation Control Drawings	10.3.2.2.3	HC1	HC1	HC1	HC1
(3)	Derived requirements are fed back to the detailed design process or other appropriate processes.	●	●	○	○	Hardware Requirements	10.3.1	HC1	HC1	HC1	HC1
(4)	Requirement omissions and errors are provided to the appropriate processes for resolution.	●	●	○	○	Problem Reports	10.6	HC2	HC2	HC2	HC2

5.5.1	Product Transition										
(1)	A baseline is established that includes all design and manufacturing data needed to support the consistent replication of the hardware item.	●	●	○	○	Hardware Archive Standards	10.2.4	HC2	HC2	NA	NA
						Hardware Requirements	10.3.1	HC1	HC1	HC1	HC1
						Top-Level Drawing	10.3.2.2.1S	HC1	HC1	HC1	HC1
						Assembly Drawings	10.3.2.2.2.2	HC1	HC1	HC1	HC1
						Installation Control Drawings	10.3.2.2.3	HC1	HC1	HC1	HC1
						Hardware/S/W Interface Data	10.3.2.2.4	HC1	HC1	HC1	HC1
						Hardware Configuration Management Records	10.7	HC2	HC2	HC2	HC2
(2)	Manufacturing requirements related to safety are identified and documented and manufacturing controls are established.	●	●	○	○	Hardware Requirements	10.3.1	HC1	HC1	HC1	HC1
(3)	Derived requirements are fed back to the implementation process or other appropriate processes.	●	●	○	○	Hardware Requirements	10.3.1	HC1	HC1	HC1	HC1
						Hardware Acceptance Test Criteria	10.5	HC2	HC2	HC2	HC2
(4)	Errors and omissions are provided to the appropriate processes for resolution.	●	●	○	○	Problem Reports	10.6	HC2	HC2	HC2	HC2

Continued

Table 9A.1 Continued

Objective		Applicability by Design Assurance Level				Output		Control category			
Description		A	B	C	D	Data item	Refs	A	B	C	D
6.1.1	Validation										
(1)	Derived hardware requirements against which the hardware item is to be verified are correct and complete.					Hardware Validation Plan	10.1.3	HC2	HC2	HC2	NA
						Hardware Requirements	10.3.1	HC1	HC1	HC1	HC1
						Hardware Traceability Data	10.4.1	HC2	HC2	HC2	HC2
						Hardware Review and Analysis Procedures	10.4.2	HC1	HC1	NA	NA
						Hardware Review and Analysis Results	10.4.3	HC2	HC2	HC2	HC2
						Hardware Test Procedures	10.4.4	HC1	HC1	HC2	HC2
						Hardware Test Results	10.4.5	HC2	HC2	HC2	HC2
(2)	Derived requirements are evaluated for impact on safety.	●	●	○	○	Hardware Requirements	10.3.1	HC1	HC1	HC1	HC1
						Hardware Review and Analysis Procedures	10.4.2	HC1	HC1	NA	NA
						Hardware Review and Analysis Results	10.4.3	HC2	HC2	HC2	HC2
						Hardware Test Procedures	10.4.4	HC1	HC1	HC2	HC2
						Hardware Test Results	10.4.5	HC2	HC2	HC2	HC2
(3)	Omissions and errors are fed back to the appropriate processes for resolution.	●	●	○	○	Problem Reports	10.6	HC2	HC2	HC2	HC2

6.2.1	Verification										
(1)	Evidence is provided that the hardware implementation meets the requirements.					Hardware Verification Plan	10.1.4	HC2	HC2	HC2	HC2
						Hardware Requirements	10.3.1	HC1	HC1	HC1	HC1
						Hardware Traceability Data	10.4.1	HC2	HC2	HC2	HC2
						Hardware Review and Analysis Procedures	10.4.2	HC1	HC1	NA	NA
						Hardware Review and Analysis Results	10.4.3	HC2	HC2	HC2	HC2
						Hardware Test Procedures	10.4.4	HC1	HC1	HC2	HC2
						Hardware Test Results	10.4.5	HC2	HC2	HC2	HC2
(2)	Traceability is established between hardware requirements, the implementation, and the verification procedures and results.	●	●	○	○	Hardware Traceability Data	10.4.1	HC2	HC2	HC2	HC2
(3)	Acceptance test criteria are identified, can be implemented and are consistent with the hardware design assurance levels of the hardware functions.	●	●	○	○	Hardware Acceptance Test Criteria	10.5	HC2	HC2	HC2	HC2
(4)	Omissions and errors are fed back to the appropriate processes for resolution.	●	●	○	○	Problem Reports	10.6	HC2	HC2	HC2	HC2
7.1	Configuration Management										
(1)	Configuration items are uniquely identified and documented.	●	●	○	○	Hardware Archive Standards	10.2.4	HC2	HC2	NA	NA
						Hardware Configuration Management Records	10.7	HC2	HC2	HC2	HC2

Continued

Table 9A.1 Continued

	Objective	Applicability by Design Assurance Level				Output		Control category			
		A	B	C	D	Data item	Refs	A	B	C	D
(2)	Consistent and accurate replication of configuration items is ensured.	●	●	○	○	Hardware Archive Standards	10.2.4	HC2	HC2	NA	NA
						Hardware Configuration Management Records	10.7	HC2	HC2	HC2	HC2
						Hardware Process Assurance Records	10.8	HC2	HC2	HC2	NA
(3)	A controlled method of identifying and tracking modification to configuration items is provided.	●	●	○	○	Hardware Configuration Management Plan	10.1.5	HC1	HC1	HC2	HC2
						Hardware Configuration Management Records	10.7	HC2	HC2	HC2	HC2
						Problem Reports	10.6	HC2	HC2	HC2	HC2
8.1	Process Assurance										
(1)	Lifecycle processes comply with the approved plans.	●	●	○	○	Hardware Process Assurance Plan	10.1.6	HC2	HC2	NA	NA
						Hardware Process Assurance Records	10.8	HC2	HC2	HC2	NA
						Hardware Accomplishment Summary	10.9	HC1	HC1	HC1	HC1
(2)	Hardware design lifecycle data produced complies with the approved plans.	●	●	○	○	Hardware Process Assurance Plan	10.1.6	HC2	HC2	NA	NA
						Hardware Process Assurance Records	10.8	HC2	HC2	HC2	NA
						Hardware Accomplishment Summary	10.9	HC1	HC1	HC1	HC1
(3)	The hardware item used for conformance assessment is built to comply with the associated lifecycle data.	●	●	○	○	Hardware Process Assurance Plan	10.1.6	HC2	HC2	NA	NA
						Hardware Process Assurance Records	10.8	HC2	HC2	HC2	NA
						Hardware Accomplishment Summary	10.9	HC1	HC1	HC1	HC1

applied to the lifecycle data (as specified by the Control Category columns). For Level A and B functions, the data will need to address the Development Assurance methods established using the process described in the previous section, whereas Level C and D functions have more simplified analysis.

In RTCA/DO-254:

- For verification, independence is achieved by evaluation of the technical correctness of the data by means, either someone or something, other than those used to produce the data.
- For process assurance, independence is achieved by evaluation of process compliance by means, either someone or something, other than those used to perform the process. Independence is intellectual independence, such as another individual, and not departmental or company independence (although these may provide a means of achieving the independence). There are several means, for example, of establishing independence in satisfying an objective, based on the objective lifecycle phase, as follows:
 - Requirements, designs, implementation or documentation are reviewed by another individual. This individual should be intellectually independent on the development of the lifecycle data.
 - Test cases or procedures are developed by another individual.
 - Test cases or procedures developed by the designer are reviewed by another individual.
 - An analysis performed by the designer is reviewed by another individual or a review team.
 - A different test is performed that confirms the results of testing by the designer, such as a test during flight test confirms a hardware item test or S/W verification tests, developed independently and performed on the target hardware item, confirm the results of testing by the designer.
 - Test or analysis results are verified by a qualified tool.

Annex B: specific approach to S/W Development Assurance

(B1) Types of S/W assurance objectives

S/W assurance objectives are usually defined with respect to either S/W lifecycle data, products or processes. [Table 9B.1](#) identifies the types of objectives applicable to lifecycle data (e.g., requirements, design, code), lifecycle processes (e.g., refining requirements into design, design into code) and verification processes (e.g., testing) using the objectives terminology from RTCA/DO-178C.

Table 9B.1 Objectives of S/W lifecycle products and processes

Objectives relevant to S/W lifecycle products and processes			
Requirements, design/ architecture and coding S/W lifecycle products	Translation process – requirements to design to code	Verification	Integral processes (Configuration Management and Quality Assurance)
Developed/Defined Consistent Accurate Verifiable Conform to Standards Compatible with Target Computer	Traceable Compliant Compatible with S/W Architecture	Completeness Extensiveness Correctness Deficiencies Identified Coverage of Requirements Coverage of Design Coverage of Code Coverage of Control and Data Coupling	Items are identified Traceability established Problem reporting change control established Load control established Lifecycle environment control is established Processes comply with plans and standards Process transition criteria satisfied Conformity review conducted

For each objective, the developer is expected to propose evidence that would show that the objective is satisfied with respect to the product or process to which it relates.

(B2) S/W assurance objectives versus the Development Assurance level

With reference to Section 9.2.1.1, the highest DAL is achieved through the applicability of all identified S/W assurance objectives. For lower levels, there is systematically less specific emphasis on:

- verification of refined requirements and detailed design requirements;
- verification of the S/W architecture;
- completeness and extensiveness of testing and test coverage;
- configuration control of S/W lifecycle process artefacts;

- independence of processes used to produce S/W lifecycle process artefacts;
- overlapping and complementary S/W verification objectives;
- verification objectives with less direct effect requirements validity and satisfaction such as conformance to standards.

For RTCA/DO-178C, this results in the following number of objectives being applicable at each S/W level (see related discussion at the end of [Section 9.3](#)):

- Level A requires that all 136 objectives (i.e., A-1 to A-10) are met, with some additional independence requirements in satisfying some of the objectives.
- Level B requires that 133 objectives are met, also with some independence requirements in satisfying the objectives.
- Level C requires that 126 objectives are met.
- Level D requires that only 53 objectives are met.

In specific terms, [Table 9B.2](#) summarises which objectives are required to be satisfied in RTCA/DO-178C and the level to which they become applicable.

From [Table 9B.2](#) it is evident that the higher S/W levels set out to provide evidence of a broader range of behaviours of the S/W, both intended and unintended. Examining each S/W level in turn provides the following insights:

- Level D relates to a limited set of S/W assurance objectives and aims to provide evidence that the intended behaviours are satisfied under relatively normal operating conditions, but provides limited evidence of unintended behaviours and behaviours under error, fault or failure conditions. Hence its suitability is for minor failure conditions only.
- The significant threshold in objectives under RTCA/DO-178C is from Level D to Level C. This is because the objectives fundamental to assuring the validity and satisfaction of S/W requirements, including safety requirements, are incorporated at Level C. This is achieved by the following:
 - Correctness, completeness, consistency and traceability is established between systems requirements, high-level S/W requirements, low-level requirements and source code.
 - All behaviours documented in requirements at these different levels of abstraction are verified, including with respect to robustness criteria.
 - Any code or requirements which do not meet these criteria are accounted for and resolved.
- Beyond this, Levels A and B are targeted at addressing key sources of error in addressing the fundamental objectives (already incorporated at Level C), either with additional rigour, or through complementary objectives/activities. There are no significant gaps in the principles behind the objectives of Level C that Levels A and B address which would be required to address fundamental safety concerns. In simple terms, if the objectives of Level C have been comprehensively addressed, then often the objectives of Level A or B will merely provide additional evidence to assist with the trustworthiness of evidence presented, rather than further substantiating the relevance and coverage of claims (at least implicitly) being made. For systems with catastrophic and hazardous hazards, it seems intuitive that evidence presented should indeed be trustworthy, and be subject to complementary activities and reviews.
- The objectives of Level B assure that development staff did not make errors (independent assessment) in the critical aspects of the development relating to requirements, including S/W safety requirements, being valid and satisfied.
- Further to this, Level A objectives assure (more independence) that development staff did not make errors in further additional areas of the development which are known to be the most challenging, and therefore most likely to have been subject to error and the

Table 9B.2 Cumulative objectives of RTCA/DO-178C by S/W level

Level	Additional objective topics beyond lower levels
D	<ul style="list-style-type: none"> • Planning Process (processes and activities defined) • Configuration Management • S/W Quality Assurance (processes comply with plans and standards, conformity review) • High-Level Requirements (developed, defined, accurate, consistent, comply with system requirements, traceable, coverage) • Low-Level Requirements (developed, defined) • Source Code (developed) • Executable Object Code (integrated and compatible, complies and robust) • Verification (coverage of high-level requirements) • Certification Liaison • Tool Qualification • Partitioning Integrity
C	<p>Level D plus:</p> <ul style="list-style-type: none"> • Planning Process (process transition criteria, lifecycle environment, development standards, compliance) • High-Level Requirements (verifiable, conform to standards, algorithms accurate) • Low-Level Requirements (accurate, consistent, comply with high-level requirements, traceable, coverage, algorithms accurate) • S/W Architecture (compatible with high-level requirements, consistent, verifiable, conforms to standards) • Source Code (complied with low-level requirements and S/W architecture, conforms to standards, traceable, accurate and consistent) • Executable Object Code (complies and robust with low-level requirements) • Verification (procedures, results, coverage of low-level requirements, statement coverage, data coupling and control coupling)
B	<p>Level C plus</p> <ul style="list-style-type: none"> • Independence (statement coverage, decision coverage, data and control coupling, executable object code and source code complies with low-level requirements, high- and low-level requirements compliance, accuracy and consistency) • High-Level Requirements (compatible with target computer) • Low-Level Requirements (compatible with target computer, verifiable) • S/W Architecture (compatible with target computer, verifiable) • Source Code (verifiable) • Verification (decision coverage) • Lifecycle Transition Criteria
A	<p>Level B plus</p> <ul style="list-style-type: none"> • Independence (modified condition decision coverage, executable object code robustness with low-level requirements, source code complied, with S/W architecture, accuracy and consistency, partitioning integrity, S/W architecture is compatible with high-level requirements and consistent) • Verification (modified condition decision coverage)

introduction of faults. The differences between verification completion criteria (statement coverage, decision coverage and MCDC (Modified Condition Decision Coverage)) are in some respects almost irrelevant. If the requirements have been completely specified, and the verification of these requirements has completely addressed normal and robust cases, then actually this analysis would likely find few or no problems.

(B3) S/W requirements

S/W requirements are the means by which the behaviours intended for the S/W are documented for communication to other lifecycle processes (e.g., design, code and verification), the S/W developers and the tools (e.g., DOORS and 3SL Cradle) supporting those processes. Therefore, the precision and accuracy to which requirements are recorded is fundamental to successful implementation in the S/W design, and is particularly important for S/W safety requirements such as those identified in Step 2 of Chapter 9.

S/W requirements may be documented using formal methods (i.e., using mathematical notation), semiformal notation (i.e., a mix of natural language with mathematical notation), natural language or a combination thereof. Several notations are also available that are coupled with specific requirements identification methods [e.g., Consortium Requirements Engineering (CORE) or Structured Systems Analysis and Design Method (SSADM) as described by [Jackson \(1988\)](#)].

S/W requirements may be specified at various levels of abstraction, depending on the extent to which the behaviours of the S/W need to be systematically accounted for. The following sections describe the abstractions at which S/W requirements are specified, how this relates to requirements traceability and which S/W assurance objectives apply to S/W requirements:

- *High-Level S/W Requirements* are S/W requirements developed and defined from analysis of system requirements, safety-related requirements and system architecture; they are predominantly specified at an abstraction that is usually independent of the S/W architecture and target computer. High-Level S/W Requirements are the first level of requirements specified when capturing system or subsystem requirements allocated to S/W.
- *Low-Level/Detailed Design Requirements* are S/W requirements translated and refined from high-level requirements in the context of the S/W architecture and target computer and include derived requirements and additional design constraints. Low-Level Requirements are produced from S/W design activities such as modelling of the S/W architecture, specification of control and data flows, determination of internal S/W states and state transitions and so on. Most importantly though, low-level requirements should be specified at a level of refinement at which source code can be directly implemented without further information. Low-Level Requirements are covered in further detail in Section 3.6.
- *Abstract-Level S/W Requirements*. As S/W developers today rarely produce a design refinement of low-level requirements directly from high-level requirements in only one step, the concept of abstract-level requirements is also useful. Abstract-Level S/W Requirements take on the properties of both high- and low-level requirements, but not in totality. Therefore, there are additional refinements required from abstract-level requirements to produce the complete set of low-level requirements from which source code can be directly implemented without further information. Abstract-Level Requirements are common place when methods such as model-based S/W engineering and formal methods are applied.

(B4) S/W assurance objectives pertaining to S/W requirements

Table 9B.3 summarises the S/W assurance objectives applicable to high-level requirements and identifies the types of evidence normally used to show satisfaction of the applicable objectives.

Table 9B.3 High-level requirements S/W assurance objectives

High-level requirements S/W assurance objectives	Evidence
Developed/Defined	Results of review processes such as structured Requirements Walkthroughs and Inspections of the completed S/W Requirements Specification (SRS) documentation.
Accurate and Consistent	Results of a combination of analysis techniques and reviews/inspections, including: <ul style="list-style-type: none"> • safety analysis, • S/W safety analysis, • modelling and simulation, • formal method proofs that show internal consistency and completeness, • structured requirements walkthroughs and inspections.
Verifiable	Results of a combination of analysis techniques and reviews/inspections, including: <ul style="list-style-type: none"> • structured requirements walkthroughs and inspections, against the requirements standard verifiability criteria, • formal method proofs that show determinism, • verification results of the requirement (note: some developers and certification authorities adopt the approach that if the requirement was verified, this implies it is verifiable) – the limitation with this approach is that the inference is made via an assumption of the extensiveness of verification (and is implicit), rather than specific evidence in this regard.
Conform to Standards	Results of a combination of analysis techniques and reviews/inspections, including: <ul style="list-style-type: none"> • automated tool checkers against requirements standards, • modelling tool results that show conformance of the model to the modelling notation, • structured requirements inspections and reviews against requirements standards or formal notations.
Compatible with Target Computer	Results of a combination of analysis techniques and reviews/inspections, including: <ul style="list-style-type: none"> • structure requirements walkthroughs, inspections and reviews against target computer compatibility requirements such as word sizes, I/O implementation, cache, memory and other resource usage and so on. • modelling results, where the properties of the target computer pertinent to high-level requirements have been modelled and analysed within the model.

(B5) S/W design

S/W design is the outcome of the translation and refinement of S/W high-level requirements in the context of the target computer and associated implementation constraints. Put simply, S/W design is the structure of the S/W to implement the S/W requirements.

S/W design is produced from a process (usually involving more than one step) of using several (and sometimes many) S/W design techniques and methods to systematise the translation and refinement of high-level requirements into the low-level/detailed design requirements. Various abstract design interpretations (e.g., models) may be produced and refined as additional implementation considerations and constraints are analysed, and these may be documented as abstract-level requirements.

The S/W design is sufficiently complete when low-level and detailed design requirements are specified at a level of refinement at which source code can be directly implemented without further information. This is a key outcome of the S/W design process. If additional design assumptions need to be made in coding process to be able to implement the relevant requirements, then it is less likely that these design decisions will be subject to the same rigour as those considered in the design process. Therefore, it is very important that the design process fully establish a suitable end point as part of its completion and lifecycle transition criteria. This is not to say that S/W coding cannot start before the S/W design process is finished; it simply implies some controls on the coding process until such a stage as the design process as adequately accounted for the behaviours the S/W needs to have.

S/W design is fundamentally important to both requirements validity and requirements satisfaction. The remainder of this section describes how S/W design achieves these outcomes, by identifying the S/W assurance objectives applicable to low-level/detailed design requirements, and identifying the importance of design methods and notation, S/W architecture, partitioning (an important property of architecture for S/W assurance), real-time systems and S/W fault tolerance in S/W design.

B5.1 *S/W Assurance Objectives pertaining to S/W Design:* [Table 9B.4](#) summarises the S/W assurance objectives applicable to low-level/detailed design requirements (see [Fig. 9.3](#)) and identifies the types of evidence normally used to show satisfaction of the objectives.

B5.2 *S/W Design Methods and Notation:* There is a plethora of S/W design methods and notation described by the S/W literature. [Jackson \(1988\)](#), although somewhat outdated now, identified a nonexhaustive list of design notations and their relationship to stages in the S/W design process.

- Design specification (Yourdon, JSD, SSADM, MASCOT);
- Logical design (Yourdon, JSD, SSADM, MASCOT).
- Physical Design (Yourdon, JSD, SSADM, MASCOT).
- Implementation (JSD, MASCOT).
- Testing (MASCOT3).

Table 9B.4 Low-level/detailed design requirements S/W assurance objectives

Low-level/detailed design requirements S/W assurance objectives	Evidence
Developed/Defined	Results of review processes such as structured walk-throughs and Inspections of the completed S/W Design Description (SDD) documentation.
Accurate and Consistent	Results of a combination of analysis techniques and reviews/inspections, including: <ul style="list-style-type: none"> • safety analysis, • S/W safety analysis, • modelling and simulation, • formal method proofs that show internal consistency and completeness, • structured design walk-throughs and inspections.
Verifiable	Results of a combination of analysis techniques and reviews/inspections, including: <ul style="list-style-type: none"> • structured design walkthroughs and inspections against the design notation standard verifiability criteria; • formal method proofs that show determinism; • verification results of the requirement/design element (note: some developers and certification authorities adopt the approach that if the requirement/design was verified, this implies it is verifiable) – the limitation with this approach is that the inference is made via an assumption about the extensiveness of verification (and is implicit), rather than specific evidence in this regard.
Conform to Standards	Results of a combination of analysis techniques and reviews/inspections, including: <ul style="list-style-type: none"> • automated tool checkers against design standards, • modelling tool results that show conformance of the model to the modelling notation, • structured design inspections and reviews against design standards or formal notations.
Compatible with Target Computer	Results of a combination of analysis techniques and reviews/inspections, including: <ul style="list-style-type: none"> • structure design walk-throughs, inspections and reviews against target computer compatibility requirements such as word sizes, I/O implementation, cache, memory and other resource usage and so on; • modelling results, where the properties of the target computer pertinent to low-level and detailed design requirements have been modelled and analysed within the model.
Traceable to Higher-Level Requirements	Traceability tables or matrices between Low-Level/Detailed Design Requirements and High-Level Requirements.

Table 9B.4 Continued

Low-level/detailed design requirements S/W assurance objectives	Evidence
Compliant with Higher Level Requirements	Results of a combination of analysis techniques and reviews/inspections, including: <ul style="list-style-type: none">• design refinement tools,• modelling and simulation,• formal method proofs that show internal consistency and technical agreement,• structured walkthroughs and inspections.
Compatible with S/W Architecture	Results of a combination of analysis techniques and reviews/inspections, including: <ul style="list-style-type: none">• modelling and simulation of the properties of the S/W architecture including the executive and scheduler, resource allocation, timing constraints and so on;• design analysis of dependencies between the function and the S/W architectural properties;• structured walk-throughs and inspections.

Other common design methods and notations include:

- Unified Modelling Language (UML).
- Architecture Analysis and Design Language (AADL) – standardised by SAE.

Research programs around the world are continually developing new and refined design notations to address limitations in existing notations for application in a specific domain, or for application across domains. Therefore it is not possible to provide a list of all design notations and methods out there.

The important thing for the systems engineer or system safety practitioner to understand is that there is no silver bullet when it comes to S/W design. Usually no single notation (or set of diagrams) provides complete coverage of all aspects of the design process or resultant product. For example, the list provided by [Jackson \(1988\)](#) is classified according to design specification, logical design, physical design, implementation and testing, and no single method provides complete coverage of all these design aspects. Furthermore, no single notation is capable of modelling the necessary behaviours of all aspects of the system. Therefore there are two factors to be mindful of:

- select design methods and notation that are compatible with the S/W safety analysis the developer intends to produce;
- be prepared to use multiple notations to ensure that the behaviours of the system that are important for safety are accurately modelled and can be subjected to analysis.

It is common place for several design notations to be chosen, each with the purpose of modelling, analysing and refining the behaviours of the system with respect to a specific design perspective. The totality of these design notations (and associated results) provides completeness in design perspectives.

B5.3 *S/W Architecture*: S/W architecture can be thought of as the enabling platform on which the functional and safety requirements can be implemented and thus provided by a system. S/W architecture is also inherently important to the behaviour of a system under failure conditions.

While it is relatively easy to describe S/W architecture in these general terms, in specific terms a S/W engineer would probably consider S/W architecture to mean one or more of the following structures or attributes:

- Control and Data Flows, States and State Transitions, Services
- Partitioning – Containment and Mediation
 - Virtual Machine, MMU, Monitors, Wrappers, Classes, Encapsulation
 - Static/Dynamic Scheduling, Timing Constraints, S/W Fault Isolation
- Interface Contracts and Safety Contracts
 - Preconditions, Invariants, Post-conditions
 - I/O, Task to Task, Layer to Layer, Class to Class
- Exception Handlers
 - Hardened Kernel, Robust Data Structures and Audit Routines, Run Time Assertions
- Multiversion S/W
 - N-Version Program, Distinct and Dissimilar S/W
- Recovery Blocks
 - Deadline Mechanism, Dissimilar Backup S/W

From the perspective of a systems engineer or system safety practitioner, the importance of developing a robust S/W architecture is as important as a robust systems architecture when it comes to meeting safety objectives. Actually, the two are almost inseparable, as there will always be requirements on the S/W to support behaviours of the system architecture, and there will be requirements on the system architecture to resolve (through detection and handling) S/W errors, faults and failures that propagate outside the S/W boundary. Some of these most important aspects of S/W architecture are described in further detail in the following sections on partitioning, real-time systems and S/W fault tolerance. While these seem like quite detailed topics, it is important that the systems engineering and safety practitioner understand the dependencies between the system architecture and these S/W architectural properties.

RTCA/DO-178C recognises the importance of S/W architecture and identifies specific S/W assurance objectives related to its definition, implementation and verification. [Table 9B.5](#) summarises the S/W assurance objectives applicable to S/W architecture and identifies the types of evidence normally used to show satisfaction of the objectives.

B5.4 *Partitioning*: As avionics manufacturers strive to reduce aircraft weight, power requirements, wiring, cooling and so on, the trend is to consolidate many different S/W components of different criticality onto common computing resources (hardware, operating system, I/O subsystem and so on). While the advantages of doing this are obvious, the avionics manufacturers must be careful to ensure that the different S/W components do not interfere with each other in such a way that it leads to a hazard. The mechanism by which this is achieved is called partitioning. The purpose of partitioning is to prevent the failure effects of one S/W component affecting another S/W component in a way that leads to a behaviour that is not appropriate with respect to safety.

Table 9B.5 S/W architecture S/W assurance objectives

S/W architecture S/W assurance objectives	Evidence
Developed	Results of review processes such as structured walkthroughs and inspections of the completed S/W Design Description (SDD) documentation architectural sections
Consistent	Results of a combination of analysis techniques and reviews/inspections, including: <ul style="list-style-type: none">• safety analysis,• S/W safety analysis,• modelling and simulation,• formal method proofs that show internal consistency and completeness,• structured design walk-throughs and inspections.
Verifiable	Results of a combination of analysis techniques and reviews/inspections, including: <ul style="list-style-type: none">• structured design walkthroughs and inspections against the design notation standard verifiability criteria;• formal method proofs that show determinism;• verification results of the architectural element (note: some developers and certification authorities adopt the approach that if the architectural element was verified, this implies it is verifiable) – the limitation with this approach is that the inference is made via an assumption about the extensiveness of verification (and is implicit), rather than specific evidence in this regard.
Conform to Standards	Results of a combination of analysis techniques and reviews/inspections, including: <ul style="list-style-type: none">• automated tool checkers against architectural design standards (e.g., partitioning and IMA) and patterns;• modelling tool results that show conformance of the model to the architectural modelling notation;• structured architectural design inspections and reviews against design standards or formal notations.
Compatible with Target Computer	Results of a combination of analysis techniques and reviews/inspections, including: <ul style="list-style-type: none">• structure architectural design walk-throughs, inspections and reviews against target computer compatibility requirements such as word sizes, I/O implementation, cache, memory and other resource usage and so on;• modelling results, where the properties of the target computer pertinent to architectural design requirements have been modelled and analysed within the model.
Partitioning Integrity is Confirmed	Results of a combination of analysis techniques and reviews/inspections, including: <ul style="list-style-type: none">• temporary and spatial interference analysis;• ARINC 653 analysis and verification results;• structure architectural design walkthroughs, inspections and reviews against partitioning criteria.

In broad terms, there are two types of partitioning, and these are as follows:

- *physical partitioning* – S/W components are hosted on different hardware such as the federated architectures of the 1960–70s;
- *logical partitioning* – S/W components are hosted on common computing resources such as the integrated avionics architectures of the 1990s and beyond.

Since physical partitioning should be well known to most aircraft practitioners, and is straightforward to providing evidence demonstrating that partitioning is achieved, it shall not be considered further in this chapter. Logical partitioning however requires significant consideration by systems and S/W designers and will be discussed further.

To achieve logical partitioning on a common computing resource, two distinct, but related factors must be addressed. These are:

- *spatial partitioning* – S/W in one partition cannot affect the code or data of S/W in another partition, and is usually achieved through use of mechanisms such as a memory management unit, CPU protected and user modes, S/W fault isolation and so on;
- *temporal partitioning* – cannot affect the service received from shared resources, and is usually achieved through relevant real-time system static or dynamic scheduling mechanisms and associated analysis (see Section 3.6.5).

Partitioning (see also [Section 9.2.2.1](#)) can also be used as a mechanism to assign less onerous S/W levels to less critical elements of the S/W that need to operate in conjunction with S/W of a higher criticality.

The requirements for the partitioning mechanisms should be determined as part of the design process and are typically derived from the results of partitioning analysis. The general approach for conducting partitioning analysis is as follows:

- Identify all S/W components hosted on common computing resources.
- Identify all potential interaction paths between components, including intended communication between components and contamination of shared resources (unintended communication). The following types of shared resources should be evaluated:
 - memory
 - time
 - kernel services (i.e., semaphores, queues, timers)
 - interrupts
 - processors
 - registers
 - caches (e.g., data and instruction)
 - subsystems (e.g., Floating Point Unit (FPU) and computation)
- Classify each potential interaction path as either spatial or temporal (or both).
- Identify architectural, design, or procedural mechanisms that either prevent the interaction or ensure that interaction is acceptable. These mechanisms may be either:
 - containment – no behavioural interference, effect prevented outside of boundary mechanism or

- mediation – acceptable behaviour interference, the behaviour can escape but the effect is controlled and is acceptable with respect to safety objectives.
- Undertake design iterations until all interactions are contained or mediated by appropriate mechanisms.

There are numerous different mechanisms suitable for addressing aspects of containment and mediation. Typically more than one of these is required to provide acceptable partitioning. Example containment and mediation mechanisms are as follows:

<ul style="list-style-type: none">• Containment<ul style="list-style-type: none">• virtual machine• memory management unit• cache management (flush)• execution time monitors• data wrappers• S/W run instruction run-time evaluation	<ul style="list-style-type: none">• Mediation<ul style="list-style-type: none">• shutdown monitors• degraded modes• procedures• temporal analysis based on static/dynamic scheduling approach
--	--

B5.5 Real-Time Systems: A significant contributor to temporal mediation (discussed with respect to partitioning in Section B5.4 earlier) is the type of real-time system architecture adopted in the design. Real-time systems are broadly categorised as either statically scheduled or dynamically scheduled. The basic difference is that statically scheduled systems are scheduled prior to run time, and have their timing properties hard-coded into the system design. The order of execution is fixed. Dynamically scheduled systems are scheduled at run-time based on particular task attributes such as priority, deadlines, execution time, rate and so on. Each time the S/W runs, and depending on the scheduling conditions, the order of execution will vary (hopefully within some constraints laid down by the developers that still permit system-level timing requirement to be met).

The most common types of real-time systems are as follows:

- cyclic executive (statically scheduled);
- interrupts (dynamically initiated, interrupt service routines are statically scheduled);
- cyclic executive + interrupts (dynamically initiated, each frame is statically scheduled);
- cooperative multitasking (dynamically scheduled) – individual processes are responsible for relinquishing control pre-emptive priority-based multitasking (dynamically scheduled):
 - fixed priority scheduling – scheduler runs the highest priority process that is ready to run;
 - rate-monotonic scheduling – processes with shorter period are given higher priority;
 - earliest deadline first (dynamic) – process with soonest deadline is given the highest priority.

While each different real-time system architecture provides the designer benefits and limitations, the most important aspect of the real-time system architecture for safety systems is the ability to show that the S/W is schedulable. This is particularly important for any functions associated with safety requirements. [Table 9B.6](#) summarises the percentage of processor utilisation that can be consumed while still guaranteeing schedulability, and provides a qualitative indication of the difficulty of the schedulability analysis.

Table 9B.6 Real-time system utilisation and schedulability analysis

Real-time system		Maximum % of processor utilisation to assure schedulability	Relative difficulty of schedulability analysis
Cyclic Executive		100%	Easy – static
Cyclic Executive + Interrupts (frame initiation)		100%	Easy – static frames
Cooperative Multitasking		100%	Moderate if all tasks are known Difficult if tasks are not yet determined
Pre-emptive Priority Based Multitasking	Fixed priority scheduling	Utilisations are possible up to 100%	Moderate for two tasks Difficult for three tasks Very difficult for greater than three tasks
	Rate-Monotonic Scheduling (RMS)	<69%	Easy – schedulability guaranteed at <69% utilisation (refer to http://www.sei.cmu.edu/ reports/91tr006.pdf for further information)
		Utilisations are possible up to 100%	Moderate for two tasks Difficult for three tasks Very difficult for greater than three tasks
	Earliest Deadline First (EDF)	100%	Difficult, and result may not be deterministic

Many of the partitioning mechanisms described in this section are provided as part of most modern real-time operating systems (RTOS) targeted as the avionics domain. Examples of these commercially available RTOS include Green Hills Integrity RTOS, LynxWorks LynxOS-178, Windriver VxWork 178B, to name some of the more frequently used products. Avionics designers are encouraged to use these types of RTOSs as they substantially reduce the burden on the design team in conducting the partitioning analysis, as this data can be purchased from the RTOS developer to support certification. Also, several avionics developers have their own bespoke RTOS implementations that have similar features, where they have established the partitioning analysis themselves.

B5.6 S/W Fault Tolerance: A key factor to providing an acceptably safe system architecture (i.e., robust against both random and systematic faults and failures) is fault tolerance. Fault tolerance is the ability for a system to detect an error, fault

or failure condition (as defined by Avizienis et al., 2004) and then undertake some level of reconfiguration to prevent the fault or localised failure propagating to a system hazard. Fault tolerance is employed to control the system's behaviour when exposed to many classes of both known (or predicted) faults, or unknown sources of errors, faults and failures. Fault tolerance is also inherent in the fail-safe design concept detailed by FAA Advisory Circular AC25.1309-1A (refer to Kritzing (2006), Chapter 7). The fail-safe design concept is as follows:

In any system or subsystem, the failure of any single element, component or connection during any one flight (brake release through ground deceleration to stop) should be assumed, regardless of its probability. Such single failures should not prevent continued safe flight and landing, or significantly reduce the capability of the airplane or the ability of the crew to cope with the resulting failure condition. Subsequent failures during the same flight, whether detected or latent, and combinations thereof, should also be assumed, unless their joint probability with the first failure is shown to be extremely improbable.

The fail-safe design concept implies the application of fault-tolerant design approaches including:

- Redundancy or Backup Systems, Monitors
- Isolation of Systems, Components and Elements
- Designed Failure Effect Limits
- Designed Failure Path
- Fault and Error Tolerance

Fault tolerance inherently underpins the application of each of these approaches. Fault tolerance exists in several different guises depending on the level of system abstraction. Broadly, though, fault tolerance mechanisms can be classified as one of the following:

- system-level fault tolerance – mechanisms usually provided at a system or line replaceable unit (LRU) level to provide tolerance to subsystem faults (noting that the subsystem fault may be caused by a factor external to the system);
- hardware-implemented fault tolerance – implementation of system-level fault tolerance mechanisms by hardware;
- S/W-implemented fault tolerance – implementation of system fault tolerance mechanisms by S/W;
- S/W fault tolerance – mechanisms provided at S/W level for containing or mediating S/W errors, faults and failures.

Specifically, two of these fault-tolerant classifications directly relate to S/W:

- S/W-implemented fault tolerance: The systems engineer or system safety practitioner should be fully cognisant of the S/W-implemented fault tolerance mechanisms. These should be largely driven by system-level fault tolerance requirements and feed to the S/W team for implementation.
- S/W fault tolerance: These mechanisms may be less obvious to the systems engineer or safety practitioner. Although they should still have an understanding of the S/W fault coverage of the mechanisms employed as they will need to ensure that classes of S/W faults that propagate to the subsystem or system architecture level are appropriately detected and handled.

Table 9B.7 summarises commonly used fault tolerance mechanisms in fault-tolerant systems as sourced from [Hitt and Mulcare \(2001\)](#) and [Hammett \(2001\)](#). Notably, many of these are implemented by S/W, or shape the architecture of the S/W.

Table 9B.7 Commonly used fault tolerance mechanisms

System-level fault tolerance	Hardware-implemented fault tolerance	S/W-implemented fault tolerance	S/W fault tolerance
<ul style="list-style-type: none"> • Simplex, no fault tolerance • Simplex, with disengagement features • Dual standby • Self-checking pair (single or dual) • Self-checking pair with simplex fault down • Triple modular redundancy <ul style="list-style-type: none"> • fault down to self-checking pair or fault down to simplex 	<ul style="list-style-type: none"> • Redundancy • Dissimilar Hardware • Distinct Hardware • Command/ Monitors • Voter Comparators • Watchdog Timers <ul style="list-style-type: none"> • Middle Value Selection • Two-thirds Majority Vote 	<ul style="list-style-type: none"> • Error Detection – recognition of the incidence of a fault <ul style="list-style-type: none"> • Replication Checks • Timing Checks • Reversal Check (Analytical Redundancy) • Coding Checks • Reasonableness Checks • Structural Checks • Diagnostic checks • Damage Confinement/ Fault Containment – restriction of the scope of effects of a fault • Damage Assessment – diagnosis of the locus of a fault • Error Recovery – restoration of a restartable service • Service Continuation – sustained delivery of system services • Fault Treatment – repair of a fault • Distributed Fault Tolerance 	<ul style="list-style-type: none"> • Multiversion S/W <ul style="list-style-type: none"> • N-version program • Cranfield Algorithm for Fault Tolerance (CRAFT) • Distinct and Dissimilar S/W • Recovery Blocks <ul style="list-style-type: none"> • Deadline mechanism • Dissimilar Backup S/W • Exception Handlers • Hardened Kernels • Robust Data Structures and Audit Routines • Run Time Assertion • Hybrid Multiversion S/W and Recovery Block Techniques <ul style="list-style-type: none"> • Tandem • Consensus Recovery Block

(B6) S/W coding

Source code is the outcome of the coding process which involves implementation of S/W low-level/detailed design requirements and S/W architecture using source languages, such as assembly language and/or a higher-level language (e.g., C, C++, Ada). Source code is the lowest-level abstraction of the S/W implementation that is both human readable (programming language) and machine readable (by a compiler or assembler). As source code is machine readable, it may also be automatically analysed using static code analysis tools.

While S/W coding is almost entirely the purview of the S/W developer, there are several factors that the systems engineer and safety practitioner should be aware of. Specifically, be aware of the factors affecting the introduction of faults during the coding process and how this might impact subsequent V&V activities.

The remainder of this section describes the factors applicable to S/W coding by:

- identifying the importance of programming language selection in providing analysable S/W source code;
- in constraining the classes of faults introduced;
- identifying the S/W assurance objectives applicable to S/W coding.

B6.1 Programming Language: There are many different programming languages available to the systems and S/W developer for developing avionics systems. They can be broadly classified according to their type, as follows:

<ul style="list-style-type: none">• Declarative• Imperative• Functional• Procedural	<ul style="list-style-type: none">• Functional Object-oriented• Object based• Object oriented• Logical
--	---

Although almost any programming language can be used to implement a set of S/W behaviours, inevitably some are better suited than others for a whole host of reasons. For example, when it comes to real-time avionics systems, languages such as C (procedural), C++ (procedural and object oriented) and Ada (procedural, object based) are the most widely adopted languages.

When choosing a language to implement a system, the developers have to balance many factors, such as:

- Availability of tools to support the target computer.
- Availability of compile-time and run-time checking.
- Functional characteristics, logical soundness and simplicity, expressive power.
- Expression of exceptions or error codes.
- Bounded space and time requirements.
- Portability.
- Security features.
- Standardisation.

- Availability of ‘safer’ language subsets.
- Availability and quality of support tools.
- Expertise available within the development team.

While all these factors may be important to the management of the S/W team, several factors are most important when determining a programming language’s suitability for safety-related aircraft applications. The programming language is central to the role of translating the intended design into executable code to run on the target system. Thus, the chosen programming language for a given application should seek to minimise the likelihood that language-related errors are introduced into the implementation commensurate with the safety risk. The following factors should be considered as part of the language justification.

- The language definition should be documented in a recognised standard, and compiler implementations should comply with the standard.
- A language subset and suitable enforcement tools should be considered for all S/W that is safety related. All languages have features that, if not controlled properly, can lead to problems. For example: programmers may be prone to making errors when using the features; compilers may be prone to poor, inconsistent or incorrect implementation of the features; programs written using the features may be more difficult to analyse, test or prove; and the features may introduce implementation dependencies, reducing portability.
- The language should effectively support the application domain(s) of interest. Poor support for the application domain may make the development of an unambiguous solution difficult. This can also have an impact on the safety and performance characteristics of the code.

To illustrate some of these points, let us consider the following examples:

- Languages such as C have inherently weak typing and rely substantially on the use of pointers (which are essential a reference to a memory location in which data may be held). These features when applied without certain constraints introduce a substantial opportunity of non-deterministic behaviours to exist within the S/W. They also limit the extensiveness to which analysis tools can examine the appropriateness of the behaviours. Languages such as Ada, which are strongly typed, provide a much greater control of determinism, and are far more analysable. For these reasons, Ada is a preferred language for critical avionics systems over languages such as C and C++.
- Another related example though that illustrates why language choice is not as black and white as just choosing Ada is as follows. Languages such as C have fairly minimal run-time machines, and as such, provided that compiler optimisations are disabled, establishing source to object code traceability is relatively straightforward. Source to object code traceability allows explicit analysis and accounting for all behaviours of the binary code, not just the source code, and is required for RTCA/DO-178C Level A. Unfortunately, languages such as Ada, for which there are many additional analysability benefits (including tools built on formal approaches such as the Spark Ada toolset) over C and similar languages, have quite substantial run-time machines, and thus present numerous challenges, albeit not insurmountable, in constructing source to object code traceability. Hence it has been possible to observe favouritism for languages such as C and C++ for developments under DO-178C Level A, with respect to this objective.

B6.2 Programming Language: S/W Assurance Objectives pertaining to S/W Coding.

[Table 9B.8](#) summarises the S/W assurance objectives applicable to S/W coding, and identifies the types of evidence normally used to show satisfaction of the objectives.

Table 9B.8 S/W coding S/W assurance objectives

S/W coding (source code) S/W assurance objectives	Evidence
Complies with Low-Level Requirements	<p>Results of a combination of analysis techniques and reviews/inspections, including:</p> <ul style="list-style-type: none"> • code generation/refinement tools, • formal method proofs that show internal consistency and technical agreement, • static code analysis results that show technical agreement to requirements annotations, • structured code/requirements walk-throughs and inspections.
Complies with S/W Architecture	<p>Results of a combination of analysis techniques and reviews/inspections, including:</p> <ul style="list-style-type: none"> • code generation/refinement tools, • static code analysis results that show technical agreement to requirement annotations, • structured code/requirement walkthroughs and inspections.
Verifiable	<p>Results of a combination of analysis techniques and reviews/inspections, including:</p> <ul style="list-style-type: none"> • structured code walkthroughs and inspections against the coding standard verifiability criteria; • static code analysis that shows determinism; • verification results of the code against requirements (note: some developers and certification authorities adopt the approach that if the requirement/design was verified, this implies it is verifiable, and the code was also) – the limitation with this approach is that the inference is made via an assumption about the extensiveness of verification (and is implicit), rather than specific evidence in this regard.
Conforms to Standards	<p>Results of a combination of analysis techniques and reviews/inspections, including:</p> <ul style="list-style-type: none"> • automated tool checkers against coding standards, • compiler results, • structured code inspections and reviews against coding standards.
Traceable to Low-Level Requirements	Traceability tables or matrices between source code components/units/function/procedures and Low-Level/Detailed Design Requirements
Accurate and Consistent	<p>Results of a combination of analysis techniques and reviews/inspections, including:</p> <ul style="list-style-type: none"> • S/W safety analysis, • static code analysis, • compile-time analysis, • structured code walkthroughs and inspections.

(B7) S/W Verification and Validation

S/W verification is the activity of determining if the behaviours expressed in high-level requirements, low-level/detailed design requirements (and any other abstract-level requirements), and the source code are correctly implemented in the S/W product. It is achieved through a combination of analysis and test activities. S/W verification evidence is one of the main pieces of evidence used to show requirements satisfaction (as discussed in [Chapter 4](#), Step 3).

S/W validation, on the other hand, is the culmination of all activities of determining if the behaviours expressed in the high-level requirements, low-level/detailed design requirements (and any other abstract-level requirements), and the source code are actually the right (i.e., valid) behaviours. Therefore, satisfying each of the S/W assurance objectives applicable to S/W requirements, S/W design and S/W coding contributes evidence towards requirements validity (as discussed in [Chapter 4](#), Step 2).

In the following sections, we consider in more specific terms the analytic and empirical techniques and methods that contribute to requirements validity by:

- Describing the main approaches to V&V of S/W.
- Identifying the role of S/W reviews and inspections, formal methods, static code analysis and testing in S/W V&V.
- Identifying the S/W assurance objectives applicable to V&V.

B7.1 S/W Reviews and Inspections: Most safety practitioners and systems engineers will be familiar with the basic concept of reviews, and could reasonably infer what these mean for S/W. However, to understand S/W review and inspections' role in satisfying S/W assurance objectives and how this relates to safety, it is important to understand the types of reviews and inspections that might be used (to a lesser extent for validation but especially for validation) and the effectiveness of different review and inspection strategies.

There is a range of evidence that shows that life-cycle reviews and inspections have been shown to remove up to 95% of introduced errors. As reviews are an upfront effort, they provide a mechanism to correct errors at the current stage of the lifecycle where they are cheaper to correct. They also avoid placing additional pressure on later lifecycle verification activities to detect and correct the bulk of errors. There are also strengths and weaknesses of review, somewhat dependent on the type of review also. For example, desktop reviews are inherently good at syntactical type errors and other edit-level problems. However, they may not be effective at identifying information that is missing, either in totality or in part. The issue being that many reviews are quite good as assessing the information that is there, and detecting gross exclusions; but not so good at reviewing information which is not there.

In simple terms the effectiveness of reviews and inspection varies with the approach, or combination of approaches, employed. Review and inspection techniques range in increasing order of effectiveness from:

- ad-hoc review;
- pass around;
- desk check or peer review;

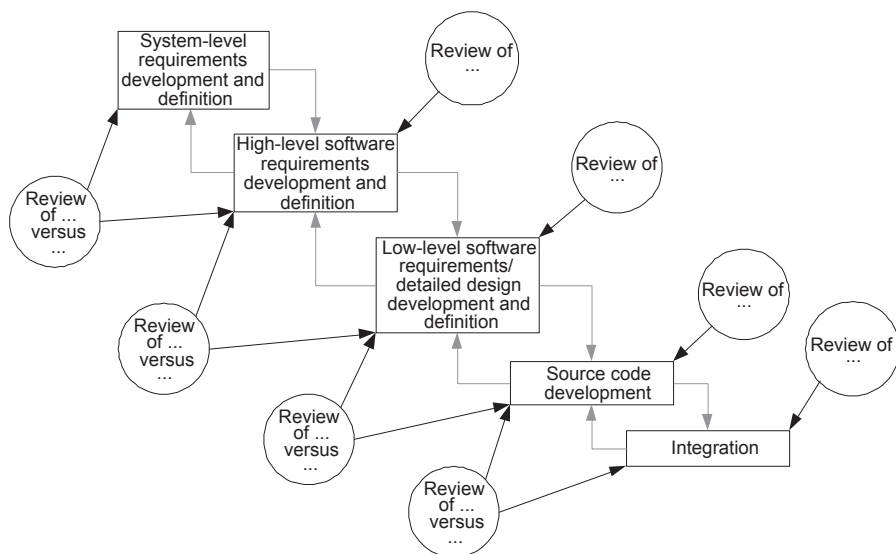


Figure 9B.1 Role of reviews and inspections in the S/W lifecycle.

- walkthrough (e.g., code walkthrough);
- team review;
- S/W inspections (e.g., fagan inspections).

Fig. 9B.1 shows the types of reviews that are normally conducted as part of providing evidence that S/W assurance objectives have been achieved. They are either a review of a standalone product (e.g., Review of S/W Requirements Specification, or review of some piece of analysis or verification evidence) or a review of one product with respect to another (e.g., Review of S/W Design Description against a S/W Requirements Specification to determine if they technically agree).

It is also important to recognise that reviews are not a substitute for structured analysis techniques. Because of the strengths and weaknesses of reviews identified in the previous sections, the reviews' greatest strength is in improving the trustworthiness of the information contained within the evidence article, rather than as a means for undertaking the fly analysis. Unfortunately, it has become common place in the industry for many of the RTCA/DO-178C objectives to be satisfied solely by reviews, rather than review of the results of analysis. There is considerable scope for improvement in the industry and certification authorities that allow such a means of compliance.

While reviews and inspections can be very effective at identifying errors, their effectiveness is also limited by the extent to which specific criteria are evaluated in the review. Thus most assurance standards define objectives based around specific review criteria. For example, the following review criteria are used by RTCA/DO-178C:

- Traceable: Are requirements/design/S/W architecture/code traceable between system-level requirements and the implementation? Downwards traceability implies there should be no childless parents. Upwards traceability implies there should be no orphans.

- **Compliance:** Do the requirements/S/W architecture/code satisfy the required system functional, performance and safety requirements?
- **Accuracy/Consistency:** Are the requirements/S/W architecture/code not in conflict with each other; and are they accurate and unambiguous?
- **Conformance to Standards:** Does the requirements/design/S/W architecture/code meet the required standards?
- **Compatible:** Are there any conflicts with the hardware/S/W features of the target computer (resources – bus loading/memory, response times, I/O hardware, synchronisation, interrupts)?
- **Verifiable:** Can the requirements/design/S/W architecture/code be verified using the proposed means (i.e., is a test possible, analysis possible?); does the proposed verification means provide adequate coverage of normal and robust properties? Proper test results generally imply verifiability – but by then it is usually too late to find out that the requirement is not verifiable.

Note that many of these attributes are not solely satisfiable by reviews and inspections. Some of them lend themselves very well to being satisfied through detailed analysis. [Table 9B.9](#) provides a summary of the strengths and weaknesses of the different S/W safety analysis techniques.

Table 9B.9 S/W safety analysis strengths and weaknesses

S/W safety analysis technique	Strengths	Weaknesses
SwFFA	Directly assesses S/W functional failures. Considers both the loss and malfunction of S/W functions. Can be directly used for assignment of S/W levels.	Limited consideration of S/W design and architecture. Limited consideration of the target computer.
SwFTA	Articulates S/W contribution to selected top-level failure modes, usually a hazardous output condition of the S/W. May be performed various levels of design abstraction, although most commonly applied at the source code/detailed design level. Existing pattern templates exist for common constructs such as assignment, procedure calls, loops, case, if-then-else and so on. Provides explicit insight into the weakest precondition. Substantial similarity to FTA, and so may be easily understood by safety practitioners.	S/W architecture only analysed to the extent that it is modelled in each fault tree. Analysis can be exceedingly cumbersome way to represent the data dependency analysis of the code. Architectural contributions are dispersed between large numbers of separate fault trees, and therefore it may be difficult to establish a holistic view of the fault tolerance of the S/W architecture. Difficult to provide assurance that all relevant features of the S/W architecture are modelled in the collective set of fault trees.

Table 9B.9 Continued

S/W safety analysis technique	Strengths	Weaknesses
SwFMEA/ SwFMECA	<p>Identifies potential S/W failure modes and provides a framework for describing potential causal factors and effects.</p> <p>Encourages design (functional or call tree) level modelling of S/W design as the basis for the analysis.</p> <p>Substantial similarity to FMEA, and so may be easily understood by safety practitioners.</p>	<p>No taxonomy of potential S/W failure conditions for evaluating the design representation.</p> <p>Technique does not directly provide for identifying relationships between S/W failure modes as effects tend to be expressed in terms of immediate effects on the S/W.</p> <p>While system-level effects are expressed, the technique provides for limited traceability between cascading failure modes and system hazards.</p> <p>Target computer behaviours are often not considered in the same FMEA set as the S/W.</p> <p>Less systematic than SwHAZOP/ CHAZOP or SHARD.</p>
SwHAZOP/ CHAZOP	<p>Provides an arguably complete taxonomy of potential failure modes for analysis.</p> <p>Encourages design-level modelling (e.g., MASCOT or other similar technique) of S/W design and architecture as the basis for the analysis.</p> <p>Directly assesses S/W communications (control and data flows) and provision of services against potential S/W failure modes.</p> <p>Provides a framework for articulating causal factors, including the relationship to other potential S/W failure modes, effects and relationship to system hazards.</p>	<p>Target computer behaviours may be difficult to model in the design or architectural model depending upon the notation chosen.</p> <p>Substantially more complex S/W failure mode taxonomy than SHARD technique.</p>

Continued

Table 9B.9 Continued

S/W safety analysis technique	Strengths	Weaknesses
SHARD	<p>Provides an arguably complete taxonomy of potential failure modes for analysis.</p> <p>Encourages design-level modelling (e.g., MASCOT or other similar technique) of S/W design and architecture as the basis for the analysis.</p> <p>Directly assesses S/W communications (control and data flows) and provision of services against potential S/W failure modes.</p> <p>Provides a framework for articulating causal factors, including the relationship to other potential S/W failure modes, effects and relationship to system hazards.</p> <p>Provides a framework for identifying and allocating S/W safety requirements for the absence or handling of S/W failure modes.</p>	<p>Target computer behaviours may be difficult to model in the design or architectural model depending upon the notation chosen.</p>
Markov Analysis	<p>Provides good modelling of states and state transitions, although is not specific to S/W.</p> <p>Well-established mathematical technique.</p>	<p>Is not a technique for deriving S/W safety requirements directly – these are inferred from assessments of undesired state transitions.</p> <p>Target computer and S/W architectural behaviours are difficult to model using purely state notation.</p> <p>Probabilities assigned to state transitions are problematic with respect to state diagrams describing S/W behaviours, and so probabilities are usually only useful for external events.</p>
Petri Net Analysis	<p>Provides good modelling of timing properties and dependences between S/W components, although the technique is not specific to S/W.</p>	<p>Is not a technique for deriving S/W safety requirements directly – these are inferred following critical analysis of dependencies modelled in the petri net.</p>
S/W Sneak Analysis	<p>Systematically considered all control and data flows in the S/W design and implementation.</p> <p>Encourages design-level modelling (e.g., MASCOT or other similar technique) of S/W design and architecture as the basis for the analysis.</p> <p>Provides a framework for articulating causal factors.</p>	<p>Failure mode taxonomy is typically limited to omission-type failure modes. Malfunction failure modes may not be considered.</p> <p>Target computer behaviours may be difficult to model in the design or architectural model depending upon the notation chosen.</p>

Table 9B.9 Continued

S/W safety analysis technique	Strengths	Weaknesses
FPTN	<p>Is a method of studying and expressing failure behaviour of complex systems.</p> <p>Uses a modular and hierarchical diagrammatic notation to express input and output failures of modules, as well as internal handling mechanisms for failures.</p>	<p>No detailed method is published.</p> <p>Method is only suitable for summarising failure information that has been derived using other techniques.</p>
LISA	<p>Direct consideration of the operating system with respect to potential S/W failures relevant to violations of partitioning arrangements between S/W functions.</p> <p>Direct consideration of the effects of the target computer on potential S/W failure modes.</p> <p>Direct consideration of computing resources.</p> <p>Uses the same taxonomy as SHARD, but for low-level interactions.</p>	<p>Limited consideration of S/W functional requirements with respect to system objectives.</p> <p>Focuses only on corruption of one S/W component with respect to control or data flows to/from other S/W components.</p>
What-If Analysis	<p>Permits targeted evaluation of known scenarios that would likely be hazardous if S/W failed.</p>	<p>No complete taxonomy of S/W failure modes for identifying potential failure conditions (i.e., the 'what'), and therefore it is difficult to determine when a sufficiently complete set of failure modes has been considered.</p>
SCPA	<p>Identifies S/W functions, control and data flows associated with potentially hazardous effects of S/W behaviour.</p> <p>Encourages design (functional or call tree) level modelling of S/W design as the basis for the analysis.</p>	<p>Potentially large numbers of paths through S/W may contribute to this analysis being extremely time-consuming.</p> <p>No complete taxonomy of S/W failure modes for identifying potential deviations to behaviours considered in path analysis.</p>

Table 9B.10 Formal methods terminology, techniques and methods

Formal methods terminology, techniques and methods	
Formal Specification	Used to provide a formalised description of the system, at the desired level of abstraction.
Formal Development	Refinement from the formalised specification to the computer program.
Formal Verification	Proofs of properties of the specification and proofs of properties of the refined implementation against the specification.
Extent of Application	English Specification (limited) Semiformal Specification (targeted) Formal Specification (complete, within the constraints of the formal language)
Types of Formal Languages	Denotational Semantics – expresses the behaviours of the system using the theory of domains. Operational Semantics – expresses the system as a sequence of actions based on some form of defined computational model. Axiomatic Semantics – expresses the behaviour of the system in terms of preconditions and post conditions which should hold true for each system task.
Type of Proofs	Human-directed Proof – mandraulic, and time-consuming for all but the most trivial system, and also prone to human error. Automated Proof – use of various tool sets to undertake automated proofs.
Types of Automation	Theorem Provers – fully formal machine-checked proofs, in which the theorem prover attempts to produce a formal proof, given a description of the system, a set of logical axioms and a set of inference rules. Model Checkers – automated proof of model against the specification, in which the model checker verifies certain properties by means of a search of possible states of a system.
Common methods	Z Notation Vienna Development Method (VDM) B Method Abstract State Machines (ASM) Abstract Machine Notation (AMN) and many others.

(B8) Formal methods

Formal methods involve the use of formal mathematical logic, discrete mathematics and computer languages (including a formalised grammar and vocabulary) to provide evidence that the system is complete and correct with respect to its requirements, and a determination of which code, S/W requirements or S/W architecture satisfy the next higher level of S/W requirements DO-178C.

The goal of applying formal methods is to prevent and eliminate requirements, design and code errors through mathematical analysis of the products of the S/W development processes. [Table 9B.10](#) presents a summary of common formal method techniques.

In their most thorough application, formal methods could be equivalent to exhaustive analysis of a system with respect to its requirements, within the constraints to which the target hardware and operating environment have been accurately modelled. However, these constraints are the ones that limit the applicability of formal methods. The following subsections provide some insight into their applicability, and why formal methods on their own are not sufficient to assure the safety of avionics S/W.

B8.1 *Application to the Problem Domain:* Formal methods are not yet universally applicable to the problem domains and technologies used in critical systems. They may also only be partially applicable to a problem scope. Formal methods are also based on formal languages that are closed, in that they have no inherent real-world meaning (Knight, 2007). A formal language is a collection of symbols and a set of rules for manipulating them. However, to be useful, they are still required to be linked to real-world concepts and objects using natural language, a medium that cannot be formally reasoned about or verified (Knight, 2007). In addition, formal language semantics are usually specified around a particular class of problems, and so one formal language is usually insufficient to describe the complete set of required behaviours of most avionics systems.

The general view of the formal methods community is that a model should be specified using a formal language (such as, e.g., ‘Z notation’ or ‘B format’) and then through a process called refinement, refined into implementation (source code) while proving certain properties of interest still hold true. Unfortunately, the types of models that can be built are very much constrained by the formal language that describes them. There are a number of challenges associated with describing real-time avionics systems using such languages, and therefore it may not be possible to apply formal methods to these types of problems in their entirety. However, formal methods are often well suited to small parts of the problem or implementation.

B8.2 *Formal Methods and Safety:* Because of the constraints on formal methods relating to target hardware and operating environment, formal methods do not address a number of significant sources of error that contribute to the safety of systems. In fact, there is little evidence that formal methods would have prevented some of the aircraft accidents and incidents attributable to S/W. Hence, the underlying message here is that formal methods alone are not the silver bullet to safe systems. Safety-related S/W errors arise most often from discrepancies between the documented requirement specifications and the requirements needed for correct (and safe) functioning of the system; and misunderstandings about the S/W’s interface with the rest of the system. S/W-related accidents have occurred when the S/W satisfied its specification and when the operational reliability of the system was very high. This is because the requirements specify behaviour that is not safe from a system perspective, requirements do not specify some particular behaviour, or the S/W has unintended (and unsafe) behaviours beyond what is specified in the requirements.

Formal methods allow us to build an unambiguous model of the system, within certain constraints. They allow us to determine if the model is consistent with itself, and allow us to determine if the implementation conforms to the model. However, formal methods tell us very little about whether the model is right, or safe, unless those properties have been captured in the model. In addition they do not tell us what these properties should be.

Unfortunately, most of the real-world interfaces and environmental parameters (particular for avionics systems) cannot be captured in formal methods at this time, or are prohibitively time-consuming to do so. For example, the time taken to build these detailed models may be better directed to analysis of safety properties of the system, using a range of S/W safety analysis techniques. Furthermore, the focus of formal methods is on correctness versus the specification, rather than on assuring that the S/W provides appropriate behaviours tolerant to the occurrence of faults and failures. This attribute of formal methods can often lead to its efforts being misdirected especially with respect to safety. Therefore, to capitalise the benefits of formal methods, improved approaches are required which bring together the system and S/W fault tolerance perspective provided by system and S/W safety analysis with the opportunities a formal refinement provides for showing that an implementation complies with and is consistent with its specified behaviours. In the interim, developers will have to continue to apply a diversity of techniques, of which formal methods may be simply one element, to arrive at an acceptably safe system.

B8.3 *Complementary to Testing:* Because there are limitations to the extent to which formal methods can address behaviour on real hardware, in the target environment, formal methods are complementary to testing. Formal methods also do not provide assurance that these real-world behaviours are appropriate with respect to safety. Testing is still required to demonstrate real-world behaviours, on real hardware, in the target environment. On the other hand, it is acknowledged that ‘testing alone is a completely hopeless way of showing that S/W does not contain errors’ (Marks, 2008). Therefore, the ideal approach is a complementary combination of evidence types: analysis and testing.

B8.4 *Static Code Analysis:* Static code analysis is the analysis of the source code before it is executed. Techniques include:

- control flow analysis,
- data flow analysis,
- symbolic execution,
- checking the source against a formal mathematical specification,
- checking conformance against a coding standard or language subset.

Barnes (2002) suggests that by using a combination of static analysis techniques, a variety of properties can be guaranteed about a computer program. An example of a well-known tool that performs such analysis is the SPARK Examiner (for Ada). Other tools include PC-Lint, Splint, Parasoft’s Static Code Analysis toolset and the Programming Research QA toolsets and related tools. There are many others.

In addition to those attributes of static analysis that relate directly to the refinement process and theorem proving, for which the discussion in the section on formal methods addresses, static analysis can be used to analyse S/W with respect to additional criteria (control flow analysis, data flow analysis, checking conformance against a coding standard or language subset, etc.) as mentioned earlier. Tools such as SPARK also permit the compliance with low-level requirements and traceability to low-level requirements to be established through the SPARK annotations and hypothesis checking.

It should be noted however that there are significant limitations with applying several important aspects of static analysis retrospectively (Salmon and Lee, 2006). This has

created significant challenges for some UK MoD programs accepting S/W from US vendors while trying to apply the framework of UK Defence Standard 00-55 and its associated prescriptions (now superseded by UK Defence Standard 00-56 Issue 4) (Salmon and Lee, 2006).

Developers are encouraged to adopt static code analysis for all programs that involve the new development of safety-critical or safety-related S/W. Static code analysis tools are widely available and affordable – some of them are even free! While static code analysis will not find all the errors most related to the safety of the S/W, there are several benefits that it provides. Static analysis does permit the programmers and testers to focus greater effort on those activities that relate to identifying requirements validity and satisfaction problems directly affecting the safety of the S/W. This prevents them becoming overwhelmed in code reviews and testing with identifying inadvertent implementation problems (conventional bugs), which static analysis tools readily detect. The focus of those reviews can be on important safety properties.

B8.4 S/W Testing: Most safety practitioners and systems engineers will be familiar with the basic concept of testing, and could reasonably infer what these means for S/W. However, to understand S/W testing's role in satisfying S/W assurance objectives and how this relates to safety, it is important to understand the types of testing that might be used, and the impact of different testing strategies on determining a defensible end point for testing while fully considering safety, cost and schedule.

Fig. 9B.2 shows the role of testing in the S/W lifecycle. It is an empirical evaluation of the executable object code against each of the other lifecycle products.

The effectiveness of testing varies with the approach, or combination of approaches, employed. Testing techniques (and supporting analyses) range in increasing order of combined effectiveness from:

- Ad-hoc testing.
- White-box testing – code structure-based test cases.
- Black-box testing – S/W requirements-based test cases.
- Black-box testing – S/W requirements/detailed design-based test cases.
- Robustness testing – S/W requirements/detailed design-based test cases.
- Requirements coverage analysis – to determine requirements-based testing shortfalls.
- Source code structural coverage analysis – to determine requirements-based testing shortfalls.
- Source to Object Code Traceability and Analysis.

B8.5 Requirements-Based Testing versus Structural Testing: There are two quite fundamentally different approaches to test case generation and testing – requirements-based testing and structural testing:

- Requirements-Based Testing is often referred to as black-box testing. The basic concept is that a S/W component is tested against the behaviours it is meant to have, which is usually specified in the S/W requirements defining the S/W components behaviour. Requirements-based tests are intended to explore intended and known functionality, and explore unknown/unintended functionality within the scope of the robustness of requirements only. Requirements-based testing provides no real insight into implementation (except when employed in conjunction with coverage analysis, which will be described shortly).

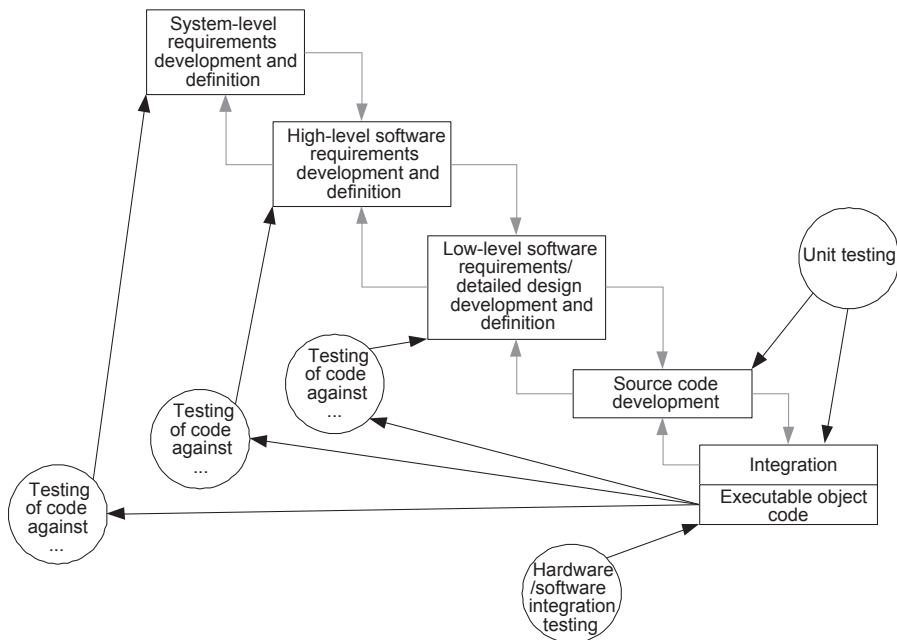


Figure 9B.2 Role of testing in the S/W lifecycle.

- Requirements-based tests are usually classified as either ‘normal’ (i.e., tests if the requirements are satisfied) or ‘robustness’ (i.e., tests if the requirements are robust, meaning not prone to violation by abnormal conditions such as fault and failure conditions caused by errors). Normal and robustness tests, including compatibility with hardware, consist of the following:

Normal	Robustness	Hardware compatibility tests
<ul style="list-style-type: none">• Coverage of equivalence classes and boundary values for variables (real, integer, boolean and so on)• multiple iterations of time-related functions (e.g., filters, integrators and delays)• Coverage of valid state transitions	<ul style="list-style-type: none">• Coverage of equivalence classes and boundary values for invalid values• System initialisation during abnormal conditions• Abnormal system initialisation• Failure modes of incoming data (e.g., sensor data)• Out of range loop count values• Protection mechanisms for exceeded time-frames• Arithmetic overflow/underflow, divide by zero• Coverage of invalid state transitions	<ul style="list-style-type: none">• Hardware transients and failures• Built-in test validation• Interface errors• Control loops/feedback loops• Resource management including time, interrupts, memory (stack, heap, variable RAM/ROM, registers and cache)• Field-loading operation• Protection boundaries <p>Note that some modern hardware includes features that make it difficult to comprehensively test – for example, cache.</p>

The limitation with requirements-based testing is that it provides no insight into the extent of S/W behaviours that have been exercised by the testing. Therefore, requirements-based testing, while an essential component of any aircraft S/W testing program, requires supplementation with structural coverage analysis to achieve its maximum effectiveness.

- Structural Testing is often referred to as white-box testing. The basic concept is that a piece of code is exercised by a tester based on tests derived from the tester's inspection or knowledge of the code's structure and implementation. This type of testing is very common in unit testing, particularly in the commercial information technology sector and for desktop application development.

The limitation with structural testing from an aircraft S/W certification perspective is that a test case that lacks traceability to a requirement specifying a required behaviour for the system only explores the suitability of S/W behaviours from the individual tester's perspective. For example, how useful is a test case without traceability to the basis of the test case – that is, the test case is actually exercising an intended behaviour, rather than merely exercising a behaviour that exists. Thus, while structural testing has its benefits, the benefits do not translate into a systematic understanding about the behaviours of the S/W unless it is complemented to requirements-based testing.

This is why in most S/W assurance standards, the objectives of the standard propose that:

- all test cases must be requirements-based; structural coverage analysis (i.e., how much of the implementation did the requirements based test cases exercise) is used to evaluate the sufficiency of requirements-based testing;
- structural coverage analysis shortfalls should be resolved using requirements-based test cases.

This combination of approaches provides the only reasonable combination of the two techniques that provides systematic and complete evaluation of the S/W behaviours. In addition, all certification authorities agree that there must a reasonable justification for the end point for testing. Most certification authorities agree that this end point is based on this reasoning regarding the use of requirements and structural testing. [Fig. 9B.2](#) summarises the approach.

B8.6 Test Coverage Analysis Requirements: The intent of specifying coverage objectives in S/W assurance standards is to ensure evidence is provided that all relevant behaviours of the S/W (executable object code) have been explored. The overall intent is to provide an unambiguous end point for testing.

Coverage objectives are usually specified from two perspectives:

- Requirements coverage, which includes demonstrating that:
 - test cases exist for each S/W requirement (high- and low-level/detailed design);
 - the test cases satisfy both the normal and robustness criteria.
- Structural coverage, which includes demonstrating:
 - Source code 'graph' coverage
 - *Statement coverage* implies that every code statement has been invoked at least once from requirements-based testing – all data flows exercised at least once.
 - *Decision coverage* (DC) implies that every entry and exit posting in the computer program has been evaluated, and evaluated for every possible outcome of every decision – all control and data flows exercised at least once.

Table 9B.11 DO-178C coverage requirements

S/W level	Requirements coverage	Structural coverage
D	High-Level Requirements only	No Structural Coverage requirements
C	High- and Low-Level Requirements	Statement
B	High- and Low-Level Requirements	Statement + Decision
A	High and Low-Level Requirements	Statement + Decision + MCDC

- *Modified condition/decision coverage* (MCDC) implies that both statement and decision coverage have been achieved and that each condition in a decision has been evaluated for all outcomes and each condition in a decision is shown to independently affect a decision’s outcome – all control and data flows exercised to the extent that control flows are influenced by data flows.
- Interactions
 - Control coupling: invocation behaviour.
 - Data coupling: communication behaviour.

Table 9B.11 shows how requirements coverage and structural coverage relate to the DO-178C S/W levels.

Fig. 9B.3 shows how the requirements-based and structural testing objectives combine in DO-178C to provide suitable coverage for S/W testing. The activity that provides the interface between requirements-based testing and structural testing is the structural coverage analysis. DO-178C states that the objectives of this analysis is to determine which code structure was exercised by the requirements-based testing procedures. Structural coverage analysis may reveal code structure (statements, decisions and control and data flows (note, some control flows only)) that has not been exercised during the requirements-based testing. DO-178C states that this may be the result of:

- Shortcomings in requirements-based test cases or procedures, in which case the requirements-based test cases should be supplemented or test procedures changed to provide the missing coverage. If there are significant shortcomings, then the method used to perform the coverage analysis may need to also be reviewed.
- Inadequacies in S/W requirements, in which case the S/W requirements should be modified and addition test cases developed and test procedures executed.
- Dead code, which should be removed and an analysis performed to assess the effect and the need for reverification.
- Deactivated code, in which case analysis and testing should show that the means by which such code could be inadvertently executed are prevented, isolated or eliminated.

Most coverage analysis is performed at the source code level. However, to detect possibly undesirable behaviour generated by the compiler (e.g., arrays bound checking, divide by zero recovery, etc.), structural coverage analysis is necessary at the object code level. This is only required in DO-178C for Level A.

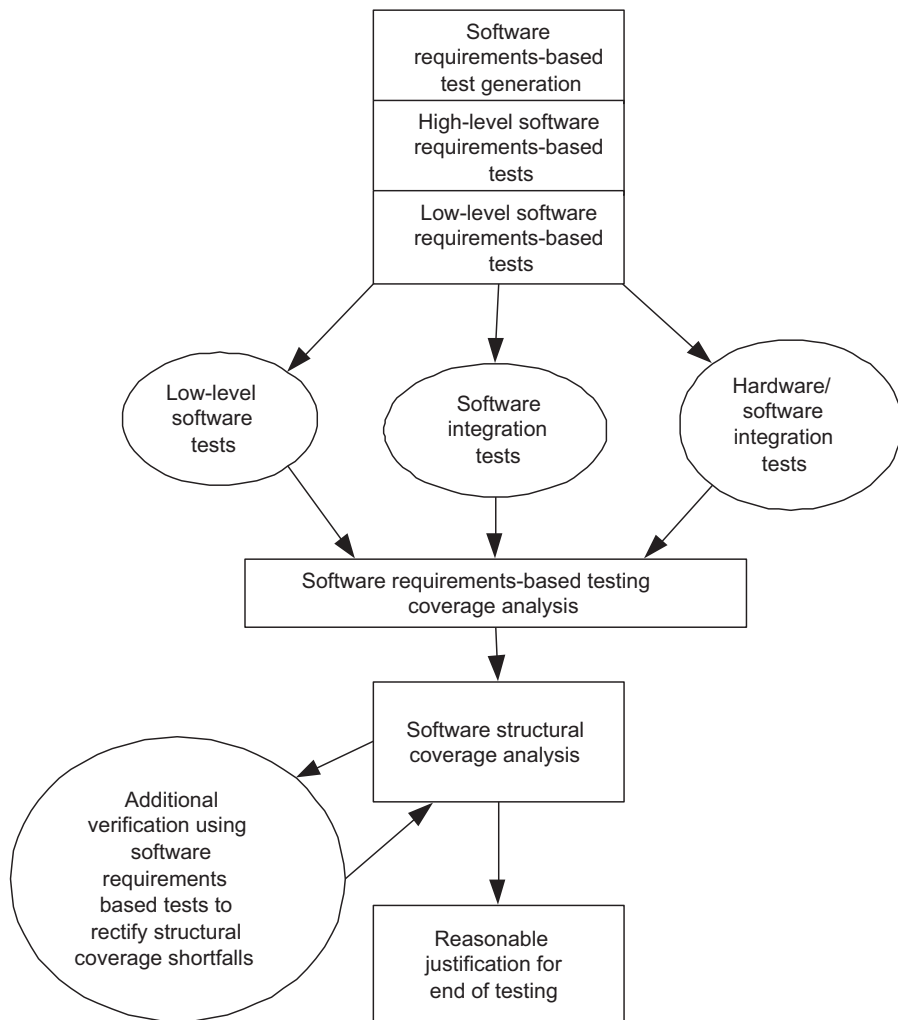


Figure 9B.3 Combining requirements-based and structural testing in DO-178C.

B8.7 Effectiveness of MCDC Testing: The MCDC objective of RTCA/DO-178C is arguably the most frequently and widely debated requirement of the DO-178C. The concept behind MCDC testing is to place the S/W in as many situations as possible to see if it behaves inappropriately. At a more detailed level, it attempts to exercise each data flow that directly affects a control flow within the S/W, in an effort to identify as many faults or errors as possible relating to the control (the logic implementing critical behaviours of the S/W), and the data that feeds these decision points.

Some studies (e.g., [Dupay and Leveson, 2000](#); Kapoor and Bowen, 2003; [Hayhurst and Veerhusen, 2001](#); [Vilkomir and Bowen, 2002](#)) have been carried out on the effectiveness of MCDC testing, and these studies generally confirm that MCDC testing

does find faults (requirements and implementation) that the other testing approaches within DO-178C do not find. However, other studies [such as the Qinetiq study referenced by Marks (2008)] found that there was ‘no significant difference’ between MCDC and the Decision Coverage objective at Level B. Furthermore, the perceived high cost (e.g., approximately 40% of the total testing effort) has contributed the widespread debate on the effectiveness of MCDC testing. It is the author’s experience that MCDC testing does find faults; but which faults, and how do they relate to the safety of the system is something that requires further assessment.

While finding more faults seems intuitively sensible, there is argument as to whether MCDC is the most effective (time and cost) way of finding these faults, and whether it finds the faults (requirements and implementation) most related to the safety of the system in question (Dupay and Leveson, 2000). Certainly, MCDC testing carried out in ignorance of the application domain and the knowledge of requirements most related to the safety of the system has reasonable likelihood that it may not actually identify the most critical faults.

For the most part, the objectives associated with identifying behaviours of the S/W most related to safety are included from Level C (the objective where it is first acknowledged that the S/W failure modes may be adverse to human safety). At Level C, testing should already be addressing the complete suite of:

- normal and robustness testing (of high and low-level S/W requirements – which infers testing of all S/W behaviours if the low-level requirements are adequate);
- error sources associated with the S/W operating in the target computer environment during S/W/hardware integration testing;
- interrelationships and interactions between S/W components, including adverse ones;
- known error sources in the implementation of low-level requirements.

Therefore, the fact that MCDC testing is not finding a plethora of additional faults is not actually the concern. If the normal and robustness testing has been comprehensive (included from Level C), then it is possible that the gap in MCDC coverage will be small and insignificant. It is the author’s opinion that much of the debate about MCDC is misdirected, and should instead be focussed on the adequacy of normal and robustness testing, including the analysis that underpins it, and variations in the application of this type of testing. MCDC is merely one of the cross-checks of the adequacy and comprehensiveness of these other types of analysis and testing. Arguable, it was also about the best practical approach available at the time RTCA/DO-178C was originally published.

Therefore, provided the faults most related to the safety of the system are being identified through the combination of analysis and verification techniques embodied by the objectives from Level C upwards to Level A, the objective of showing that the system is acceptably safe can be achieved, irrespective of the criticisms of the effectiveness of MCDC testing.

B8.8 S/W Assurance Objectives pertaining to S/W Verification: Table 9B.12 summarises the S/W assurance objectives applicable to high-level requirements and identifies the types of evidence normally used to show satisfaction of the objectives.

Table 9B.12 S/W verification S/W assurance objectives

S/W verification S/W assurance objectives	Evidence
Traceability	Traceability records in the form of matrices or tables that show traceability between each verification case and the high, low, detailed design or S/W architectural requirement to which it relates.
Completeness/Extensiveness	Results of reviews and inspection of test descriptions and test cases, including the degree to which requirements coverage is achieved and satisfaction of normal and robustness test criteria have been met. Results of reviews and inspection of test results. Tool generation of test cases against nominated test case criteria, such as normal and robustness criteria.
Correctness	Results of reviews and inspection of test descriptions and test cases.
Deficiencies Identified	Problem reports, including traceability to their identification, analysis, evaluation and treatments.
Coverage of Requirements	Results of requirements coverage analysis, and reviews/inspections of requirements coverage analysis.
Coverage of Design	Results of structural coverage analysis, and reviews/inspections of structural coverage analysis.
Coverage of Code	Results of structural coverage analysis, and reviews/inspections of code coverage analysis.
Coverage of Control and Data Coupling	Result of control and data coupling analysis, and reviews/inspections of control and data coupling analysis.

(B9) Integral processes

This section explores two of the integral processes of the S/W lifecycle: configuration management and S/W QA. While neither of these processes directly provide much information about the S/W product, they do contribute substantially to the trustworthiness of all of the other product evidence provided.

B9.1 Configuration Management: Recall from Section 9.3.4.3 that one of the objectives of S/W assurance was configuration consistency. Essentially, does the evidence produced throughout the S/W lifecycle have traceability to the delivered S/W product? For example, what is the relevance of a safety case claim made from S/W lifecycle data that cannot be traced or be shown to be consistent with the version of S/W being delivered? Clearly the relevance is low, and would not form a robust argument in the safety case.

S/W configuration management is the major factor in achieving configuration consistency. The purpose of S/W configuration management is to establish and maintain the integrity of S/W development lifecycle data throughout the S/W lifecycle.

It establishes that the evidence presented is consistent with the delivered executable object code and that any deficiencies identified in satisfying any other verification-related objectives are explicitly identified, and explained. This is achieved by establishing the traceability and consistency between the S/W and the S/W work products/S/W lifecycle data that document the development and verification of the S/W. S/W configuration management is an integral process, which means that if it is not applied from the commencement of S/W development, or it is lost throughout the development, S/W configuration management can be difficult to recover retrospectively.

The goals of configuration management activities are as follows:

- S/W configuration management activities are planned and repeatable;
- S/W work products are identified and controlled;
- changes to S/W work products are controlled;
- development and verification is only performed against relevant S/W baselines using relevant lifecycle products and data.

To achieve these goals, S/W configuration management is comprised of the following key elements:

- a repeatable process exists – for example, a S/W CMP;
- a library is established as a repository for S/W baselines;
- work products to be placed under configuration management are identified (configuration items) – particularly those that are fundamental to satisfying S/W assurance objectives;
- change requests and problem reports are initiated, recorded, tracked, reviewed and approved in accordance with a documented and repeatable procedure;
- changes to baselines are controlled in accordance with a documented and repeatable procedure;
- release of S/W products is controlled in accordance with a documented procedure – that is, S/W load control is established;
- status of configuration items is recorded in accordance with a documented procedure;
- reports summarising S/W configuration management activities and S/W baselines are available;
- S/W baseline audits are conducted according to a documented procedure;
- S/W lifecycle environment configuration is controlled.

Therefore, how are S/W work products and S/W lifecycle data controlled? The outputs of the S/W assurance lifecycle require different degrees of control depending on their role in satisfying S/W assurance objectives. S/W lifecycle data that is considered direct evidence to supporting satisfaction of the product centric assurance objectives requires the highest degree of confirmation management, while less robust configuration management control may be relevant to backing evidence supporting satisfaction of less significant process-related objectives. For example, RTCA/DO-178C defines two different degrees of configuration control:

- Control Category 1 (CC1), the more onerous configuration control requirements.
- Control Category 2 (CC2), which is considered the minimum acceptable configuration management practices on S/W lifecycle data.

<p>CC1 and CC2 both include requirements as follows:</p> <ul style="list-style-type: none"> • traceability, • change control – integrity and identification, • retrieval, • protection against unauthorised changes, • data retention. 	<p>CC1 also includes the following additional requirements:</p> <ul style="list-style-type: none"> • configuration identification; • baselines; • problem reporting; • change control – tracking; • change review; • configuration status accounting; • media selection, refreshing, duplication; • release.
---	--

The starting point for any robust configuration management system of S/W lifecycle data is a S/W CMP. The typical contents of a S/W CMP are as follows:

- management responsibilities and authorities (who?) for accomplishing the S/W configuration management controls and activities;
- activities (what?) to be performed in applying the configuration management process to the project;
- schedule (when?) coordination of configuration management activities with other S/W lifecycle activities;
- resources (how?) identifies the tools and physical and human resources required for configuration management;
- maintenance of how the plan will be kept current.

Despite the intended robustness of S/W configuration management controls, there are numerous challenges that conspire to erode the effectiveness of configuration management and may violate configuration consistency. These are as follows:

- large development teams, or geographically or time-dispersed teams – leading to difficulties in coordinating and ensuring consistency in configuration management activities as well as maintaining visibility of activities;
- parallel or concurrent development, multiple versions for different targets/platforms – leading to concurrent or overlapping lifecycle phases and products which may conspire against extent configuration tools or processes;
- mixture of COTS and bespoke S/W – leading to uncertainty among developers over the configuration management responsibility for particular elements of S/W;
- level of integration between S/W components – leading to increased sensitivity to subtleties to configuration variations and extensive opportunities to violate configuration consistency among V&V evidence;
- limited understanding on when configuration management activities should commence – leading to the retrospective application of configuration management controls part way through a program, with possible rework required to re-achieve configuration consistency;
- level of automation – leading to potential lack of visibility of key configuration management indicators.

Where these challenges present a reasonable likelihood of manifesting themselves in a development, then configuration management controls (i.e., tools and procedures) should be developed to treat them. Systems and S/W development teams should remain vigilant to the challenges of configuration management and should have respect for the

importance of configuration consistency as an objective of S/W assurance. Remember, a break down in configuration management will be looked upon very seriously by any certification authority.

B9.2 S/W Quality Assurance: S/W QA is the process for providing adequate assurance that the S/W products and processes in the lifecycle conform to their specified requirements and adhere to their established plans. It establishes that the evidence for demonstrating requirements validity, satisfaction and traceability was derived from processes agreed with the certification authority. Like S/W configuration management, S/W QA is an integral process and cannot be performed retrospectively. S/W quality cannot be built in at the end of a program.

S/W QA sets out to achieve that:

- Approved plans were followed – did they do what they agreed to do, remember that usually the basis for certification authority agreement is based on the plans submitted?
- Transition criteria for S/W lifecycle processes are satisfied – were relevant steps actually followed, and relevant issues resolved before progressing into later lifecycle phases?
- S/W conformity review – is the lifecycle process complete? Is the lifecycle data complete? Is the executable object code controlled and can it be regenerated using the S/W lifecycle data? Is there any evidence that the S/W product does not conform to its documentation?

S/W QA also typically requires independence to be effective and trustworthy to the certification authority. As such, there should usually be dedicated S/W quality assurance staff with an independent company reporting structure to the engineering and S/W teams. Quality assurance should already be a function of any developer's business; however, S/W QA requires QA staff who have an in-depth understanding of S/W assurance.

The starting point for any robust S/W QA program is a S/W QA plan. S/W QA planning should occur as early as possible in the S/W process, and should be presented to the certification authority for agreement prior to development. The typical contents of a S/W QA plan are as follows:

- Identification of who is to conduct QA activities and how are they independent (who?).
- Provision of specific compliance reports and corrective action (what?).
- Schedule (when?) of S/W QA activities coupled to the schedule of S/W work product development and S/W lifecycle process transition criteria.
- QA activities (how?) – process-specific evaluations (walkthroughs and design reviews), evaluation of project activities (configuration management and S/W development folders), evaluation of lifecycle work products (plans and data).
- QA methods and tools (how?) – including inspections, reviews and audits.

S/W QA provides assurance that the S/W products and processes in the lifecycle conform to their specified requirements and adhere to their established plans. It forms important backing evidence to Safety Assessment claims regarding the trustworthiness of evidence presented in the Safety Assessment.

(B10) S/W development artefacts

Chapter 9 has frequently referred to numerous products produced as part of the S/W lifecycle. To provide the safety practitioner and systems engineer a broad understanding

of the role of most common S/W lifecycle data elements, this section provides an overview of commonly²³ encountered work products.

B10.1 *Planning Documents.* These include:

- Plan for S/W Aspects of Certification or S/W Safety Plan – describes to the certification authority how the S/W assurance objectives are intended to be satisfied.
- SDP/S/W Management Plan – describes the objectives, standards and S/W lifecycles to be used in the S/W development processes and the activities that constitute them.
- S/W Verification Plan – describes the verification procedures intended to be used to satisfy the S/W assurance objectives related to verification.
- S/W Configuration Management Plan – describes who, how, what and when S/W configuration management activities will be conducted.
- S/W Quality Assurance Plan – describes who, how, what and when the S/W QA activities will be conducted.

B10.2 *Standards.* These include:

- S/W Requirements Standards – describe the methods, rules and tools to be used in developing high-level S/W requirements from the system requirements.
- S/W Design Standards – describe the methods, rules and tools to be used in developing the S/W architecture and low-level S/W requirements from the system requirements and high-level S/W requirements.
- S/W Coding Standards – describe the programming languages, methods, rules and tools used to code the S/W from the high- and low-level S/W requirements and S/W architecture description.

B10.3 *S/W Development and Verification Documents.* These include:

- S/W Requirements Specification – describes the high-level S/W requirements refined from the system-level requirements.
- S/W Design Description – describes the S/W architecture and low-level S/W requirements refined from high-level S/W requirements and system-level S/W requirements.
- S/W Source Code – consists of the code written in the source languages and the compiler instructions for generating the object code from the source code, and linking and loading data.
- S/W Verification Results – describe the results of the verification activities (reviews, analysis and testing).
- S/W Problem Reports/S/W Trouble Reports – identify and record the resolution to S/W product anomalous behaviour detected at any phase in the verification process, as well as process noncompliance with plans and standards, and any deficiencies detected in S/W lifecycle data from integral processes or other review activities.
- Integral Process Records – describe the results of S/W configuration management and QA process activities.

B10.4 *Accomplishments.* This includes:

- S/W Configuration Index – lists the configuration and version of the S/W that is covered by the S/W accomplishment summary.
- S/W Accomplishment Summary – describes how the plans for S/W aspects of certification (and associated plans) were executed and how the S/W assurance objectives were satisfied.

²³ For further information on the content of these work products or some of the alternate document names that these documents may be referred to as, the reader is referred to DO-178C, IEEE12207 (2008) or MIL-STD-498 (1994)/J-STD-16 (1995).

Annex C: differences between RTCA/DO-254 and RTCA/DO-178C

Unlike RTCA/DO-178C, in DO-254 all the objectives are applicable for each DAL:

- In RTCA/DO-178C, different objectives are required depending on the allocated DAL. For example at higher DALs, RTCA/DO-178C require greater fidelity in specification of design and implementation, along with more onerous criteria for verification coverage. Whereas in RTCA/DO-254, it is the Development Assurance methods that are proposed to satisfy each objective that are the primary differences between DALs.
- In RTCA/DO-254, the requirements for verification coverage are not captured as explicit objectives, and are instead subsumed within the guidance of the verification objectives, whereas in RTCA/DO-178B/C, they are specified as discrete objectives. The reason for this is that depending on the Development Assurance method chosen, and the technology involved, the unit of metric may be different, and therefore cannot be specified at the objective level.

RTCA/DO-254 provides explicit validation objectives, whereas RTCA/DO-178C includes the properties of validity within objectives that deal with accuracy, consistency and completeness.

RTCA/DO-254 does not have the concept of a target computer. The entire hardware is subject to Functional Failure Path Analyses (FFPA) and decomposed until an assurance strategy²⁴ can be applied to each decomposed FFP.

According to FAA guidance, micro-processors are excluded from RTCA/DO-254, and as such they are better treated as the target computer under RTCA/DO-178B/C, rather than as CEHW.

²⁴ Assurance strategy is deciding whether to exploit an architectural strategy such as using partitioning (in an FFP sense), or whether to assure the whole product to the highest assurance level.

Annex D: understanding errors, faults and failures

To understand the application of Design Assurance Principles, we need to explore some key terminology in the following subsections:

In the standards relating to civil aviation, the term ‘Failure Condition’ is defined as:

- *A condition having an effect on the aircraft and/or its occupants, either direct or consequential, which is caused or contributed to by one or more failures or errors, considering flight phase and relevant adverse operational or environmental conditions or external events [AMC 25.1309].*
- *The effect on the aircraft and its occupants both direct and consequential caused or contributed to by one or more failures, considering relevant adverse operational and environmental conditions. [RTCA/DO-178C].*

It can be seen that the civil definitions for ‘Failure Condition’ are defined in terms of ‘Failures’ and ‘Errors’. Several civil aviation definitions for the ‘Failure’ are as follows:

- *An occurrence which affects the operation of a component, part or element such that it can no longer function as intended (this includes both loss of function and malfunction). Note, errors may cause Failures, but are not considered Failures [AMC 25.1309].*
- *The inability of a system or system component to perform a required function within specified limits [RTCA/DO-178C].*

These definitions are in general agreement with the definition used in complex systems and computing, where a ‘Failure’ is defined as follows:

- *It is an event that occurs when the delivered service deviates from correct service. A service fails either because it does not comply with the functional specification, or because this specification did not adequately describe the system function. A service failure is a transition from correct service to incorrect service, that is, to not implementing the system function (Avizienis et al., 2004).*

In general, the deviation that causes the Failure is called an Error and the cause of an Error is a Fault. An ‘Error’ is:

- *A deviation of the state from the correct service state (Avizienis et al., 2004). In the context of a system in operation, an error might be an internal error state such as an overflow. In the context of the human processes conducting S/W development for example, an error is a mistake {by the developer} in requirements, design or code [RTCA/DO-178C].*

A ‘Fault’ is:

- *the adjudged or hypothesized cause of an error. Faults can be internal or external to a system. The prior presence of a vulnerability, i.e., an internal fault that enables an external fault to harm the system, is necessary for an external fault to cause an error and possibly subsequent failure(s) (Avizienis et al., 2004).*
- *In the context of a S/W product, a Fault is a manifestation of a {development process} error in S/W [RTCA/DO-178C].*

Avizienis et al. (2004) depict the relationship between fault, error and failure as shown in Fig. 9D.1.

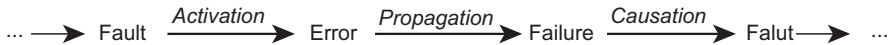


Figure 9D.1 Fault, error and failure relationships.

Some examples demonstrating the relationship between fault, error and failures, summarised from [Avizienis et al. \(2004\)](#), are as follows:

- **Requirements.** An *Error* by a requirements engineer leads to a *Failure* to adequately describe a function; that in turn results in a *Fault* in the documented specification.
- **Implementation.** An *Error* by a programmer leads to a *Failure* to write the correct instructions, that in turn results in a *Fault* in the written S/W;
- **Operation.** Upon activation, a latent *Fault* produces a S/W *Error*, and when this *Error* affects the delivered service, a *Failure* occurs.

Note that there is some variation in assumed definitions of these terms (failure, fault and error) between different standards and guidance.

Crew errors in the safety assessment

10

To err is human [Marcus Tullius Cicero, 106-43BC]– and pilots, in spite of pretences, are only human.

10.1 Introduction

10.1.1 Background

Human errors continue to dominate as a contributing factor in aircraft accidents (see [Annex A](#) to this chapter). A Boeing study (2001) found that flight crew errors are listed as the primary cause in 66% of accidents and that despite the introduction of protective devices or systems, this percentage has remained relatively unchanged in recent years. An FAA study (2002) into ‘Aeroplane Safety Assurance Processes’ concluded that ‘the processes used to determine and validate human responses to failure and methods to include human responses in safety assessments need to be improved’ and that ‘the industry challenge is to develop aeroplanes and procedures that are less likely to result in operator error and that are more tolerant to operator error when they do occur’.

Unfortunately, there is relatively little guidance (for instance, in AMC xx.1309, ARP 4761 and ARP 4754) to aid engineers and System Safety assessors in ensuring that human performance is adequately considered during the design of aircraft systems and during the development of training programs in support of such systems. An FAA Human Factors Team Report (1986) identified the following interrelated deficiencies in the current aviation system:

- Insufficient communicating and coordination (i.e. between designers and operators).
- Inadequate processes used for addressing human performance issues in design, training and regulatory functions.
- Insufficient criteria, methods and tools for design, training and evaluation.
- Insufficient knowledge and skill.
- Insufficient understanding and consideration of cultural differences in design, training, operations and evaluation.

If significant improvements are to be made in lowering accident rates, then clearly a far better understanding of the issues affecting human performance is required. Aircraft designers are therefore faced with the challenge of developing systems which are less error prone. This will require that designers proactively:

- Verify and validate how the flight crew responds to system failure conditions.
- Provide procedures which are more explicit and robust.
- Learn (and share) best industry practices in the interest of Continued Airworthiness.

The term ‘human error’ refers¹ to human actions or inactions outside the tolerances established by a system, potentially leading to undesired effects, even if no immediate consequences occur. Human error is an important consideration in complex safety critical systems, because it makes one of the most significant contributions to overall system safety (refer, inter alia, [Edwards, 1988](#)). In the design of such systems, it is therefore increasingly important that we understand how errors are made and what can be done to prevent (or reduce the probability) of their occurrence.

There are two major ways in which pilots can erroneously impact the function (see [Chapter 3](#)) of an aircraft or aircraft system:

- Incorrect actions with no system malfunction

Example: Air Inter Flight 148, Strasbourg 1992

The A340 crashed during an approach to landing. The Bureau d’Enquêtes et d’Analyses pour la Sécurité de l’Aviation Civile (BEAS) believe that Flight 148 crashed because the pilots inadvertently left the autopilot set in Vertical Speed mode (instead of Flight Path Angle mode) then entered ‘33’ for ‘3.3-degree descent angle’, which for the autopilot meant a descent rate of 3300 feet per minute.

The BEAS noted that the display of selected information was ambiguous on the control panel and inconsistent with the displayed information on the primary flight display – a classic Human Factors (HF) issue.

In the aftermath of the accident, Airbus modified the interface of the autopilot so that a vertical speed setting would be displayed as a four-digit number, preventing confusion with the Flight Path Angle mode.

<http://aviation-safety.net/database/record.php?id=19920120-0>.

- Incorrect response after a system malfunction

Example: British Midland Flight 92, Kegworth 1989

The 737-400 crashed 19 min after suffered a partial fan blade loss in the number 1 engine (left side) at flight level (FL) 92. Severe vibration was accompanied by engine surges and fumes in the cockpit.

The crew misinterpreted data (which was correctly displayed) and hastily decided that the number 2 engine (right side) was faulty and throttled it back. The pilots

¹ See <http://www.uscg.mil/hq/g-m/risk/e-guidelines/html/vol2/01/v2-01-03.htm>). For more information on human error in the Coast Guard, see the document entitled ‘Human Error and Marine Safety’ in the General Resources Directory in Volume 4.

Example: British Midland Flight 92, Kegworth 1989—cont'd

believed that the fault originated from the number 1 engine, since earlier models of the 737 ventilated the flight-deck from the right, and they were unaware that the -400 used a different system.

Number 1 engine then burst into flames and the aircraft crashed. Of the 126 people aboard, 47 died and 74 sustained serious injuries.

<http://aviation-safety.net/database/record.php?id=19920120-0>.

These, and other, errors may be characterised by the following descriptions:

- Errors of omission (i.e. failure to perform a task or step)
- Errors of commission (i.e. performing a task or step incorrectly) as in the following:
 - Selection error (e.g. selects wrong display/device/setting)
 - Extraneous act (i.e. inessential or unrelated or irrelevant to the topic or matter at hand)
 - Sequence error (i.e. too soon or too late)
 - Timing error (i.e. too short or too long)
 - Quantitative error (i.e. too little, too much, too fast)

All of the above can result in system misdiagnosis, which is known as a Cognitive Task Error. At best, this action delays the correct response; at worst, it compounds the problem. A more proactive approach is therefore required, and the wise consideration of the human role, performance and frailties in overall system performance, is thus a fertile area to explore in the System Safety Assessment (SSA).

10.1.2 Aim of this chapter

The aim of this chapter is to explore the issues surrounding crew performance and how this integrates with the content of a typical CS/FAR 25.1309 System Safety Assessment.

10.1.3 Objectives of this chapter

The objectives of this chapter are to:

- Provide an overview of how errors are made and how their probability and/or impact can be reduced.
- Propose a simple methodology to consider flight crew error as another failure mode in the SSA process.

10.1.4 Scope of the chapter

The scope of this chapter is limited to considering flight crew errors/mistakes only for the purposes of completing a typical CS25.1309 Safety Assessment.

The scope of this chapter does not extend to the full remit of HF and excludes proving compliance to CS/FAR25.1302.

The scope of this chapter is restricted to unintentional errors² only and does not extend to intentional³ errors or sabotage.

The scope of this chapter does not include system resilience provided by the fail safe design philosophy. See Chapter 7 in Kritzinger (2006) for more information on this.

10.2 System Safety Assessment process to mitigate crew errors

10.2.1 The process

The goal of a SSA is to make systems successful by enhancing performance, reliability and safety. This includes considering the operating crew around which such a system is designed. Mitigating the possibility of crew error must therefore be a systematic and proactive activity that is introduced early in the system design cycle. An integrated approach is therefore required to provide a balanced development of both the technical and human aspects of equipment acquisition (MoD, 2000, p. 6). This approach must provide a process that ensures the application of scientific knowledge about human characteristics through the specification, design and evaluation of systems.

Within the scope of this chapter then, let us consider the regulatory requirements which drive the integration of crew errors into the Safety Assessment process.

CS25.1309(c) at Amendment 17 requires:

“Information concerning unsafe system operating conditions must be provided to the crew to enable them to take appropriate corrective action. A warning indication must be provided if immediate corrective action is required. Systems and controls, including indications and annunciations must be designed to minimise crew errors, which could create additional hazards”.

There are various ways to approach this complicated task, and the illustration in Fig. 10.1 attempts to simplify the basic approach of how to mitigate crew errors as part of the system development life cycle. Most of these steps fall within the remit of the design team (with input from the HF specialist). The System Safety Engineer's key input is at Step 3, but each step will be explained below as any neglect upstream will make downstream certification efforts increasingly more difficult.

² Unintentional Error: An action committed or omitted accidentally, with no prior thought.

³ Intentional Errors (also referred to as violations). An intentional error does not include sabotage. The difference is in the motive. It concerns an action committed or omitted deliberately, because of a perception that there is a better or equally effective way to perform the task or step. Not following prescribed procedures is often a shortcut that may not be recognised as a mistake until other conditions arise that result in a noticeable problem.

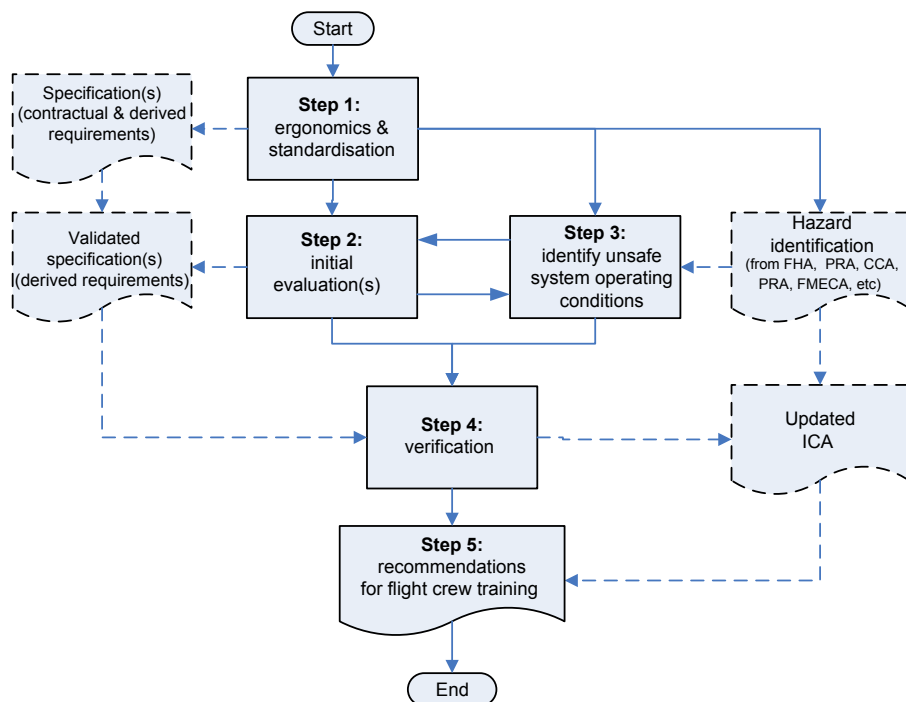


Figure 10.1 Process to mitigate crew errors.

10.2.2 Step 1: ergonomics and standardisation

Step 1 is a proactive, requirements driven activity which, within the context of this chapter, is aimed at ensuring that ‘Systems and controls, including indications and annunciations must be designed to minimise crew errors, which could create additional hazards’ (CS25.1309). The output of Step 1 is a Requirements Specification (refer Fig. 1.3).

Ergonomics⁴ is about designing for people in terms of their physical and mental abilities with due consideration of their limitations. In the late 1950s, aircraft authorities started to define guiding design principles under the banner of ‘ergonomics’. For operational safety, this initially meant little more than to ensure than an average (50 percentile) pilot could reach and operate the appropriate levers and buttons without physical contortions. The late 1980s then saw terminology such as ‘Man–Machine Interface’ and ‘Human Centred Engineering’ becoming common currency in the engineering vocabulary. These approaches made great strides in optimising the

⁴ The term ‘ergonomics’ derives from the Greek words ‘ergon’ (work) and ‘nomos’ (natural law). It is defined as ‘the study of the efficiency of persons in their working environment’. In some States, the term ergonomics is used strictly to refer to the study of human–machine system design issues. In CAP 719, the term ergonomics is used in a broader context, synonymous with the term Human Factors, and, therefore, including human performance and behaviour.

relationship between the person (e.g. the pilot) and their working environment (e.g. the cockpit) by using two approaches:

- ‘Fitting the person to the job’ (i.e. person selection and training). This topic is explored in Step 5.)
- ‘Fitting the job to the person’ (i.e. designing for the person). This is what ergonomics is all about. If a system requires a human operator, then the system hardware and software should be designed around the capacity of the operator.

Standardisation came about as lessons were learnt (both in certification processes as well as from unfortunate incidents and accidents). The guidance surrounding ergonomics thus became more extensive and prescriptive (e.g. standardisation of warning cues; position of key displays within a cone of view,⁵ etc.) In his paper on Cockpit Ergonomics, [Harris et al. \(2002\)](#) highlight one of the central principles of HF as follows:

- *Perception*: The pilot must first gain information from the displays in the cockpit. This information must be unambiguous, correct, in the right format and at the right time to facilitate a decision.
- *Decision*: The decision must be implanted in the cockpit via controls and this leads to action.
- *Action*: A good control will translate the operator’s intentions effectively, efficiently (with minimum of effort) and must provide feedback.
- *Feedback*: Finally, the pilot must obtain unambiguous knowledge of the result of their action to determine the appropriateness of this action.

The simple illustration in [Fig. 10.2](#) illustrates the key issues to consider, each of which are discussed in the following subsections.

10.2.2.1 Step 1a: design for physical attributes

These early ergonomic evaluations are meant to ensure that the design team adequately considers basic physical characteristics and control interfaces. Areas to concentrate on include positioning, reach, field of view, knob/button size, tactile perception, labelling, system usability, etc. Aircraft controls supplement aircraft displays in communicating to the pilot ([Jarrett, 2005](#)). It provides a two-way interaction between the aircraft and the crew. Controls should be easy to reach and be positioned appropriately

⁵ Because human senses are adapted for use on the ground, navigating by sensory input alone during flight can be dangerous: sensory input does not always accurately reflect the movement of the aircraft, causing sensory illusions. These illusions can be extremely dangerous for pilots. Fluid in the inner ear reacts only to rate of change, not a sustained change. For example, if a pilot initiates a banking left turn, the inner ear will detect the roll into the turn, but if the turn is held constant, the inner ear will compensate and rather quickly, although inaccurately, report to the brain that it has returned to level flight. Coriolis illusion involves the simultaneous stimulation of two semicircular canals and is associated with a sudden tilting (forward or backwards) of the pilot’s head while the aircraft is turning. This can occur when tilting the head down (to look at an approach chart or to write on the knee pad), or up (to look at an overhead instrument or switch) or sideways. This can produce an overpowering sensation that the aircraft is rolling, pitching and yawing all at the same time, which can be compared with the sensation of rolling down a hillside. This illusion can make the pilot quickly become disoriented and lose control of the aircraft.

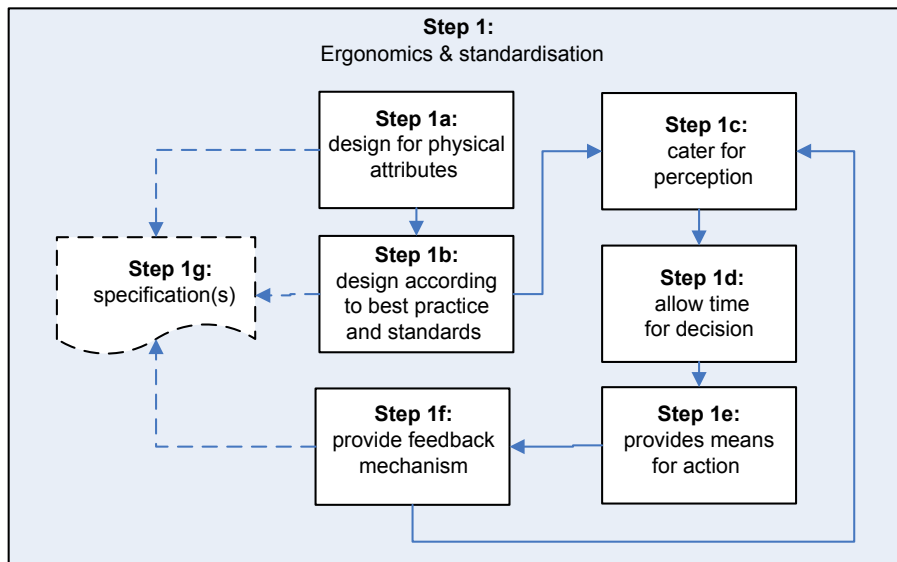


Figure 10.2 Design for ergonomics and standardisation.

in accordance to their usage. Controls which are used frequently should be positioned in a more prominent position. Controls should move in the natural sense and controls that complement each other or frequently used in conjunction of each other should be grouped together if possible.

10.2.2.2 Step 1b: design according to best practice and standards⁶

This step is closely related to Step 1a, but focuses more on lessons learned and best practices on topics such as instrument layout and LRU display characteristics.

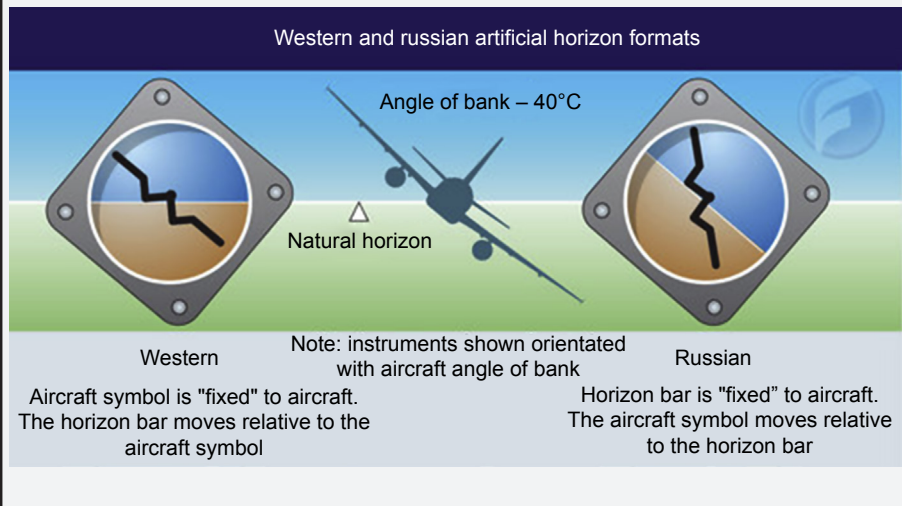
With reference to Fig. 1.1:

- At aircraft and system level, the areas to focus on include intended function and standard cockpit configuration.
- At subsystem and competent level, the areas to concentrate on include display appearance, indication of status, symbology, colour palette, menu structure, menu depth and menu complexity. In some instances, the regulatory authorities enforce standardisation via Technical Standard Orders (TSO). The Aeroflot-Nord 737 example below shows an avoidable accident if ETSO-C4c was adopted universally.

⁶ The importance of the standardisation of panel layout relates to safety, since there are numerous reports of errors arising from inconsistent panel layouts, involving inadvertent reversion to an operating practice appropriate to an aircraft flown previously (CAP719 para 6.2).

Aeroflot-Nord 737-500 (2008)

In 2008 an Aeroflot-Nord pilot flying a Boeing 737-500 crashed on approach to Perm, Russia, because of disorientation. The official report stated the accident was at least partly caused by the fact that the Western and Russian artificial horizons, also known as Attitude Director Indicator, operate on a completely different psychology and that the captain had spent most of his flying life using the Russian model.



HF specialists usually rely on several sources of information to guide their involvement in the design process, including previous published research, data compendiums, HF standards and more general principles and guidelines. For instance:

- *Data Compendiums*: As the field of HF has matured, many people have emphasised the need for sources of information to support HF aspects of system design (e.g. [Boff et al., 1991](#); [Rogers and Armstrong, 1977](#); [Rogers and Pegden, 1977](#)). Such information is being developed in several forms. One form consists of condensed and categorised databases, with information such as tables and formulas of human capabilities. An example is the four-volume publication by [Boff and Lincoln \(1988\)](#), *Engineering Data Compendium: Human Perception and Performance*, which is also published on CD-ROM under the title 'Computer-Aided Systems Human Engineering' (CASHE).
- *HF Design Standards*: Another form of information to support design is engineering or HF design standards. Standards are precise recommendations that relate to very specific areas or topics. One of the commonly used standards in HF is the military standard [MIL-STD-1472F](#). This standard provides detailed requirements for areas such as controls, visual and audio displays, labelling, anthropometry, workspace design, environmental factors and designing for maintenance, hazards and safety. Other standards include the ANSI/HFES-100 VDT standard, and the ANSI/HFES-200 design standard for software ergonomics ([Reed and Billingsley, 1996](#)).

- *HF Principles and Guidelines*: There are many situations where answers to design problems cannot be found in the existing standards. For example, if a designer is trying to decide where to place the controls on a camera, there will be no standard in current publications to answer these questions. The designer must look to more abstract principles and guidelines for this information. There are a number of books that catalogue HF techniques that can be used to make assessment on error and workload, e.g. [Stanton et al. \(2013\)](#).

There are numerous HF requirements and guidelines related to flight-deck displays and controls that are spread across a variety of regulatory documents (e.g. EASA Certification Specification and FAA Regulations, Technical Standard Orders and Advisory Circulars), industry standards (e.g. RTCA Minimum Operational Performance Standards (MOPS), SAE International Aerospace Recommended Practices) and guidance documents (e.g. the FAA's '[Human Factors Design Guidelines for Multifunction Displays](#)' and the DoD's '[Human Engineering Design Data Digest](#)'). One symptom of the relative youth of the field is the lack of centrality and organisation to these materials. Efforts are being made within the HF design community to (1) organise an electronic database to provide access to the existing principles and guidelines (however, this is a daunting task) and (2) to provide a single source of reference (e.g. see the FAA's '[Human Factors Considerations in the Design and Evaluation of Flight Deck Displays and Controls](#)'). Unfortunately, at this time, the HF practitioner must become familiar with the sources through regular literature reviews and attendance of the major conferences.

HF standards, principles and guidelines cover a wide range of topics, some more general than others. Additionally, because of the nature of how HF may impact different aspects of a display system or control, relevant guidance may be spread throughout any one document or across several documents. This is especially likely as new avionics systems offer functions and features beyond what was originally envisioned. Most guidelines require ([Woods et al., 1992](#)) careful consideration and application by designers, who must not only think through the implications of their design solutions, but also conduct a formal testing programme which would highlight system deficiencies early in the design life cycle.

10.2.2.3 Step 1c: cater for perception

The pilot needs to gain the right information (or feedback) at the right time from the displays in the cockpit. Feedback may be visual, via audio, tactile or haptic.⁷ Depending on the case, the feedback has different objectives:

- When the process of a situation change is initiated by the crew, the feedback is the acknowledgement by the machine of a crew command (e.g. when a new automation mode is engaged by the crew, the Fight Mode Annunciator confirms to the crew that the command has been received by the system).

⁷ Contrasting tactile versus haptic feedback: The wheel on top of the undercarriage lever would provide tactile feedback, whereas a stick shaker would provide haptic feedback.

- When the change originates from an aircraft system (e.g. failure condition) or from external conditions (e.g. wind shear), the feedback needs to get the crew's attention to:
 - orientate the understanding by the crew of the new situation;
 - orientate an adequate crew response and minimise potential crew error;
 - help the understanding of the new status of the aircraft after crew intervention.

Good decisions are a product of good information (Harris) and due consideration is needed on (1) how status information is communicated to pilot and (2) what information not to present to the crew and (3) prioritisation of the presentation of information. For more information, see paragraphs 3 and 4 in the FAA's 'Human Factors Considerations in the Design and Evaluation of Flight Deck Displays and Controls' and paragraph 5.1.1.4 in MIL-STD-1472F.

No discussion of crew perception is complete without considering the topic of Flight Deck Automation.⁸ Advances in technology have enabled increasingly sophisticated automation to be introduced to aircraft. Generally, this automation was added to accomplish worthy objectives (such as reducing crew workload, adding additional capability or increasing fuel economy), but the side effects are that crew are (1) often kept out of the loop (i.e. do not understand what the system is doing) and (2) are reduced to monitoring and interpreting functions only (and the human is not a reliable monitor over extended periods). History has shown that flight crews can become confused about the state of advanced automation (such as the autopilot, autothrottle and flight management computer). This condition is often referred to as either decreased mode awareness or automation surprise.

Example of accidents attributed to automation

1. On 26 April 1994, an A300-600 operated by China Airlines crashed at Nagoya, Japan, killing 264 people. Contributing to the accident were conflicting actions taken by the pilots and the autopilot.
2. On 20 December 1995, a B757 operated by American Airlines crashed near Cali, Columbia. Contributing to the accident were the pilots not understanding what the Flight Management System was doing.
3. On 25 February 2009, a B737-800 radio-altimeter malfunction caused the autothrottle and autopilot to diverge on approach to Schiphol. The pilots did not heed indications of significant decrease in airspeed until the stick shaker activated at final approach. The re-actions to the stall warning were uncoordinated and did not prevent stalling. There were 9 fatalities and 120 injuries

Flight crew errors typically occur when the crew does not perceive a problem and fails to correct the error in time to prevent the situation from deteriorating. A report by the FAA, 'The Interfaces between Flightcrews and Modern Flight Deck Systems' found

⁸ Automation refers to control of a process or system by a machine or electronic device. Each automated system required a different level of monitoring by the user. Some require extensive operator input and monitoring while others are almost completely independent.

that traditional methods of assessing safety are often insufficient to pinpoint automation vulnerabilities that may lead to an accident. Designers need to:

- Provide systems which support instrument displays with visual and tactile motion cues to minimise potential confusion about what functions are automated.
- Provide displays and cues which reinforce situational awareness (SA) and help keep the flight crew fully aware of changes occurring to the aeroplane's status and flight path during all phases of automated and manual flight.
- Consider the flight deck as a single system, as opposed to a collection of displays for multiple separate systems (such as hydraulic, electrical or pressurisation). Expertise should be applied towards matching the characteristics of these systems to those of humans, with due consideration to the job to be performed ([CAP 719 para 6](#)).
- Better communicate (see [Chapter 11](#)) the automated system principles, better understand flight crew use of automated systems, and systematically document skilled flight crew strategies (in the ICA) for using automation to make flight crew training more effective and efficient.

For more information, see [Endsley et al. \(2002\)](#).

10.2.2.4 Step 1d: allow time for decision

The designer needs to consider the time it takes to process the information presented and make a correct decision. Due to the time constraint, the depth of the analysis expected from the crew should be proportionally inversed to the time criticality of the expected reaction. The information needed by the flight crew should be presented in such a way as to assist the processing task, not only under normal circumstances, but also when performance is affected by stress (e.g. during failure conditions) or fatigue.

10.2.2.5 Step 1e: provide means for action

The decision needs to be implemented via the controls and switches in the cockpit. A good control will translate the operator's intentions effectively and efficiently ([Harris](#)). Three fundamental operational objectives apply to the design of warning, alerting and advisory systems ([CAP 719 para 6.5](#)):

- they should alert the crew and draw their attention,
- report the nature of the condition, and
- when possible, guide them to the appropriate corrective action.

[CAP 719](#) (paragraph 6.5) goes on to say that: 'System reliability is vital, since credibility will be lost if false warnings proliferate, as was the case with the first generation of ground proximity warning systems. In the event of a technical failure of the display system, the user should not be presented with unreliable information. Such information must be removed from sight or clearly flagged. For example, unreliable flight director command bars should disappear. Invalid guidance information which remained on display has been a factor in accidents'.

10.2.2.6 Step 1f: provide feedback mechanism

The pilot must be able to ascertain status of system condition (Step 1c), as well as feedback on any actions taken by them (so as to ascertain the appropriateness of the

decision). Menu structure, knob shape and labels are very important because pilots expect obvious and easy-to-recognise functions with a clear and distinct tactile feel. Pilots expect (Holland and Ryan, 2010) their actions to result in intuitive and obvious system responses. They also expect clearly displayed options with an obvious means of selection and a clear way to return to a standard or default condition.

ATR72 (registration TS-LBB), 6 Aug 2005

The aircraft ditched into the sea off Capo Gallo following the flameout of both engines. The aircraft broke into three pieces with 16 fatalities and 28 injured. The cause was attributed to human error by the mechanics who replaced the fuel gauge (the FQI) with one meant for an ATR42.

In this instance the fuel system indicator panel triggers an acoustic warning and a low level light.

Although not a recommendation in the accident report, it can be speculated that this feedback system (which can cause a Catastrophic functional failure, see Section 1.3.6) is vulnerable to a common mode failure (see [Chapter 6 Step 1a](#)) and should have been provided by independent means.

10.2.2.7 Step 1g: specification

Capture the results of all these activities (i.e. the requirements definition⁹) in the System Specification for validation in Step 4.

These requirements may be categorised in several interrelated ways, for instance:

- Customer requirements (often contractual and/or regulatory)
- Mission profile or scenario requirements
- Performance requirement
- Utilisation environments (i.e. how are the various system components to be used?)
- Effectiveness requirements (i.e. how effective or efficient must the system be in performing its mission?)
- Architectural requirements (i.e. identify the necessary systems architecture)
- Behavioural requirements (expected system and pilot behaviours)
- Functional requirements (necessary task, action or activity that must be accomplished)
- Functional requirements analysis will be used as the top level functions for functional analysis
- Design requirements (the 'build to', 'code to' and 'buy to' requirements for elements in the system)
- Derived requirements (requirements that are implied or transformed from higher-level requirement).

⁹ Requirements analysis is critical to the success of a project. The requirements should be documented, actionable, measurable, testable, traceable, related to identified business needs or opportunities and defined to a level of detail sufficient for system design.

For more information, see [DOT/FAA/AR-03/69](#) Table 4-1 (for a ‘Human Factors Requirements Template’) and [Chapter 6](#) (to ‘Formulate Human Factors in System Specifications’) and [MIL-STD-1472F](#) (specifically [Chapter 5](#) for ‘Detailed Requirements’).

10.2.3 Step 2: initial evaluations

Step 2 is all about checking that the evolving design concept is correct, complete and accomplishable (i.e. ‘Are we building the right thing which will minimise pilot error?’). The output of Step 2 is thus a validated Specification (refer [Fig. 1.3](#)).

There are many processes available to validate requirements, and the reader is encouraged to refer to Validation Process Model in [Fig. 12](#) of [SAE ARP4754A](#). The validation process can also be facilitated by a number of tools, such as:

- Task Analysis¹⁰: Describes tasks, information requirements, feedback, likely errors, detection methods and recovery procedures.
- Scenarios and performance targets: Developed as part of the system requirements definition and validation and a key stage for experienced operator input.
- Prototypes, mock-ups and models of the systems: Enables users to ‘test drive’ the system and spot problems which increase errors or reduce the effectiveness of detection and recovery. This is another key stage for experienced operator input.
- Developing procedures alongside the design: To check that all subsystems work together to support task performance. These procedures should also be assessed to identify risks associated with inaccurate or inappropriate application of the procedure.
- Defining training in parallel with the design: This aids understanding of likely required skill and experience levels of personnel. It also allows assessment of the extent to which additional training to improve mitigation of failure modes is achievable within training budgets. See [Chapter 11](#) for more information.

These early HF evaluations check basic physical characteristics (including control interfaces and display characteristics) and should be conducted at each level of integration:

- At component level, the focus is on the intended function, display appearance, symbology, colour palette, menu structure, menu depth and complexity, knob/button size, etc.
- At system integration level (e.g. see [Fig. 10.3](#)), the focus is on cockpit layout/configuration, labelling, system usability, ergonomics, etc.

In the absence of specific regulatory requirements, these evaluations do tend to be very subjective, so a diversity of input is required to settle on a negotiated solution.

As the system matures, mock-up and test rigs can be used to perform and evaluate distinctive tasks (e.g. entering a navigation frequency, changing a barometric setting or entering a simple flight plan). These are generally referred to as ‘part-task evaluations’.

¹⁰ Task analysis is a fundamental methodology in the assessment and reduction of human error. A wide variety of different task analysis methods exist, and it would be impracticable to describe all these techniques here. See *Task Analysis Techniques* by Embrey for a description of representative methodologies applicable to different types of task.



Figure 10.3 Example Cockpit for System Integration HF evaluation.

10.2.4 Step 3: identify unsafe system operating conditions

The activities in Step 2 are initially concerned with the workload of the pilots during normal equipment functionality and the probability of pilot error while operating these systems. The FAA points out (2002) that *‘a common assumption in the regulations, advisory materials and Safety Assessments, is that single failures, particularly when accompanied with a failure indication, can be detected by the flight crew and, through the use of redundant systems and alternate operating procedures, the flight can be safely continued’*. In the vast majority of cases, this assumption is correct. However, this is not always the case. Step 3 thus is aimed at extracting the predicted failure modes of the system and feeding this back into Step 2 so as assess the critical tasks required to recover the situation (which might include designing out the failure or the potential for the unsafe condition).

The illustration in Fig. 10.4 attempts to summarise the key processes in Step 3.

10.2.4.1 Step 3a: identify system failure conditions

This step involves the Safety Engineer highlighting to the Test Pilot and/or the HF Specialist all failure conditions identified via techniques such as the Functional Hazard Analysis (FHA) (Chapter 3), Failure Modes and Effects Analysis (FMEA) (Chapter 5), Common Mode Analysis (CMA) (Chapter 6), Particular Risk Analysis (PRA) (Chapter 7) and Zonal Safety Analysis (ZSA) (Chapter 8).

No discussion of crew responses to system failure is complete without considering the ‘Fail Safe’ design concept. A system is ‘fail safe’ if, in the event of a failure, the system or component automatically reverts to one of a small set of states known to be safe and thereafter operates in a highly restricted mode. This may involve complete loss of functionality, or reverting to back-up/redundant features. See AMC25.1309 (Amend 17, para 6b) and Kritzing (2006, Chapter 7) for more information on this topic.

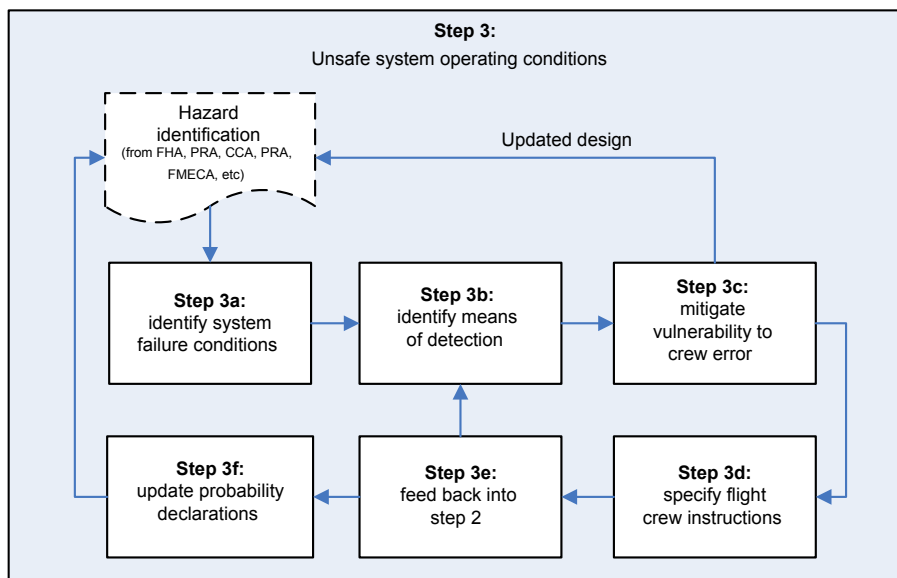


Figure 10.4 Identifying unsafe conditions.

10.2.4.2 Step 3b: identify means of detection

Working with the Design team, Test Pilot and HF Specialist, it is recommended that the System Safety Engineer extends each of the techniques¹¹ above to specifically address how the failure conditions will be detected by the flight crew. The required information will depend on the degree of urgency for recognition and corrective action by the crew. It should be in the standardised (refer Step 1b) form of:

1. A 'Warning', if immediate recognition and corrective or compensatory action by the crew is required.
2. A 'Caution', if immediate crew awareness is required and subsequent crew action may be required.
3. An 'Advisory', if crew awareness is required and subsequent crew action may be required.
4. A 'Message', in all other cases.

All of the above must be timely, obvious, clear and unambiguous.

With reference to the FHA in Chapter 3, the reliability of a failure monitoring and indication system must be compatible with the safety objectives associated with the system function for which it provides that indication (e.g. if the effects of failure and not annunciating is Catastrophic, then the combination of failure and not annunciating that failure must be Extremely Improbable). This approach can also be considered from the other direction too (e.g. if the risk is very low then there is no need to annunciate, which reduced workload and prioritises warnings.)

¹¹ For an example on how an FMEA can be extended, see Fig. 5.2 (Step 2c).

10.2.4.3 Step 3c: mitigate the vulnerability to crew error

The team now needs to consider the possibility of the crew making an error when reacting to a system failure. While it would be ideal to eliminate human error, this is not always realistic. After all, Marcus Tullius Cicero wrote¹² some 2000 years ago. ‘To err is human’. While traditional thinking has focussed on eliminating human error, contemporary thinking acknowledges that error is a way of life. Given the acceptance that human error may occur, the focus of today has become ‘how do we effectively manage error’ (Sumwalt and Thomas, 1999).

There are, therefore, two fundamental design approaches to mitigate the system’s vulnerability to human error¹³: reduce the probability and/or reduce the severity of the error:

- *Reduce the probability of the error*: The design of equipment (including the monitoring/feedback loop to its operators), procedures (e.g. how to deal with the occurrence) and training (e.g. ensure timely intervention) has a major influence on the likelihood of human errors.

If the probability of Human Error is to be quantified,¹⁴ then a table such as that in [Table 10.1](#) would require agreement with the applicable regulatory authority as it could be controversial (i.e. very subjective) and may need to be verified¹⁵ within the correct context.

In a large transport aircraft cockpit, where there will be a pilot and a copilot, the assessor might want/need to claim redundancy¹⁶ [e.g. via AND gates in the Fault tree analysis (FTA), refer [Chapter 4](#)] in the fact that the monitoring pilot will pick up errors made by the controlling/flying pilot (or vice versa when it comes to navigation or other aircraft settings). As required in traditional systems design, the monitor should point out to the controller that an error has been made, allowing the controller to correct the error. We say ‘should’ because it is of course possible that the error is flagged with the monitor and both pilots make a common mode error. In analysing this situation, we must therefore consider two probabilities (Harris et al.):

- Probability of controlling pilot making an error, and
- Probability that monitoring pilot will detect the error (and act).

Remember that the ‘AND’ gate above is dependent on the situation definitely not being work sharing (Harris et al.). If activities are distributed between the crew so that each crew

¹² Marcus Tullius Cicero said ‘*To err is human, but to persevere in error is only the act of a fool*’. [Latin: ‘*Cujusvis hominis est errare, nullius nisi insipientis in errore perseverare*’].

¹³ Note that the flight crew is often used as a means to mitigate a failure condition (i.e. stop it from propagating to a higher severity in the FHA). Human error is not included in probability analyses for 25.1309(b) compliance. Minimising human error is part of 25.1309(c) compliance.

¹⁴ Human errors are not easily quantified, and conservative assumptions may well be necessary.

¹⁵ Verification (refer [Fig. 1.3](#)) can be accomplished via some sort of human reliability analysis based on specific task analysis or test data.

¹⁶ Research has shown that once an error is committed, it is difficult to catch (trap) your own error. Other people or systems are more likely to catch your error. Therefore, redundancy in the cockpit through CRM is a strong defence to ensure that errors are trapped (Sumwalt & Thomas).

Table 10.1 Example human error probabilities¹⁷

Nature of task	Error probability
General omission error (e.g. skipping a necessary step), when there is no warning alarm or display	1×10^{-2}
Errors of omission (e.g. skipping a necessary step or failing to communicate) when the actions are embedded in a well-rehearsed procedure	3×10^{-3}
General error of commission (e.g. performing correct step on wrong item, performing step incorrectly on right item, performing correct step at wrong time)	3×10^{-3}
Simple arithmetic errors with self-checking	3×10^{-2}
General error of supervision	1×10^{-1}
Handover/changeover error	1×10^{-1}
General decision error rate for high stress levels	0.2–0.3
Failure to act correctly in reasonable time after the onset of a high stress condition	0.3–1

member is doing something different, then there will be no monitoring and the probability of error will not be reduced to the extent that would apply to independent, redundant systems. So, rather than benefiting from the mathematics we fall back to individual pilot error probability.

- *Reduce the severity of the error:* The focus here is to acknowledge that an error is possible and therefore the aim is to (1) reduce the consequence of such error (i.e. make the system error tolerant), (2) to provide systems or instructions on how to recover from that error in a timely fashion and (3) to consider improving system degradation rates (i.e. offering the crew more time to assess the problem and respond accordingly).

Example: Alaska Airlines MD-83 Flight 261, January 2000

The screw jack assembly failed due to inadequate maintenance, resulting in the loss of 88 lives. Even though this was a predictable failure, the designers seemed to have considered it so improbable as not to provide adequate flight crew reference cards with effective recovery instructions/limitations.

Continued

¹⁷ Sourced from Draft IEC 300-3-8 (Dependability Management), DEF STAN 00-56 Iss 2 Part 2 Table 4 and ACJ25.1309. This guidance was subsequently removed in DEF STAN 00-56 Iss3 and AMC25.1309. Examples obtained from Clemens, P.L. Human Factors and Operator Errors, J.E. Jacobs presentation second edition, Feb 2002. Human error probabilities are also contained in WASH-1400 (NUREG-75/014); 'Reactor Safety Study—An Assessment of Accident Risks in U.S. Commercial Nuclear Power Plants,' 1975.

Example: Alaska Airlines MD-83 Flight 261, January 2000—cont'd

The FAA concluded that the flight crew's use of the autopilot while the horizontal stabiliser was jammed was not appropriate. The crew should not have tried to troubleshoot the system by using the autopilot and trim motors. Once a stable aeroplane-landing configuration was obtained, landing should have followed immediately.

<http://lessonslearned.faa.gov/>.

Both these approaches may require design changes, which will then need to be reflected in updated Safety Assessment deliverables (e.g. FMEA, FTA, CMA, etc.). These design changes could be significant, so the intent would be to complete Step 3c as soon as practicably possible.

No discussion of crew error is complete without considering the hazard 'Loss of Situational Awareness', which studies have shown to be a leading causal factor in a review of 175 military aviation mishaps (Hartel et al., 1991) and a major causal factor in 88% of accidents associated with human error in a review of major aircraft accidents between 1989 and 1992 (Endsley, 2001).

Endsley (2000) defines the term 'Situation Awareness' as:

- the perception of the elements in the environment within a volume of time and space,
- the comprehension of their meaning, and
- the projection of their status in the near future.

Endsley (2000) furthermore provides the following subcategories of SA, which can be useful to consider when designing solutions which will reduce the probability of this type of error:

- Geographical Situational Awareness (e.g. own aircraft, other aircraft, terrain features, airports, cities, waypoints, navigation fixes, position relative to designated features, path to desired location, runway and taxiway assignments, path to desired location, climb/descent points, etc.)
- Spatial/Temporal Situational Awareness (e.g. attitude, altitude, heading, velocity, vertical speed, vertical acceleration (i.e. G's), flight path, actual values relative to assigned projected flight path, etc.)
- System Situational Awareness (e.g. system status; functioning and settings for radio, altimeter, transponders, flight modes and automation; deviations from correct settings; ATC communications; present fuel; impact of degradations and settings on performance; time and distance available on fuel; etc.)
- Environmental Situational Awareness (e.g. weather formations and movement; temperature; icing; fog; turbulence; winds; sun; IFR/VFR conditions; areas to avoid; etc.)
- Tactical Situational Awareness (e.g. identification; tactical status type; capabilities; location; threat flight dynamics; own capabilities relative to threat; threat detections; threat launch capabilities; threat prioritisation; threat imminence and assignments; current and projected intentions, tactics, firing, manoeuvring; mission timing and status; confidence level of information; etc.)

10.2.4.4 Step 3d: provide flight crew instructions

Working with the flight crew (e.g. test pilots) and the HF Specialist, the safety engineer needs to coordinate the drafting of appropriate instructions on how to deal with system failures should they occur (regardless of their probability of occurrence). Under the EASA system, this issue of these instructions forms part of the ‘Instructions for Continued Airworthiness’ (ICA) (refer EASA Part 21 and C25 Appendix H).

A technique called the Procedural Event Analysis Tool (PEAT¹⁸) could be usefully applied here too.

10.2.4.5 Step 3e: feedback into Step 2

At Step 3d above we have identified additional critical tasks, which may also need to be subjected to verification, the results of which may lead to recommend changes to the:

- Means of Detection (Step 3b)
- Flight Crew Instructions (Step 3d)
- Design (taking us back to Step 1) to make the system more forgiving of crew errors and revocable (i.e. provide time/opportunity to recognise and correct errors).

10.2.4.6 Step 3f: update probability declarations

Add probability of crew error to any qualitative and or quantitative probability analyses¹⁹ (e.g. Human Hazard FTA, see [Chapter 4](#)), noting that the probability of crew error is likely to increase (see [Fig. 10.5](#)) in a stress environment.²⁰

In their paper on ‘A Systems Approach to Safety in a Multi-Crew Aircraft’, the authors discuss a useful model (tailored in [Fig. 10.6](#)) to assess the flow of events which will occur following an initial error made by the pilot.

¹⁸ PEAT is a structured, cognitively based analytic tool traditionally used to help airline safety officers investigate and analyse serious incidents involving flight-crew procedural deviations. The objective of PEAT is to help develop effective remedial measures to prevent the occurrence of future similar errors. PEAT assumes that there are reasons why the flight crew member failed to follow a procedure or made an error and that the error was not intentional. Based on this assumption, a trained investigator interviews the flight crew to collect detailed information about the procedural deviation and the contributing factors associated with it. The PEAT process relies on a nonpunitive approach to identify key contributing factors to crew decisions. Using this process, the airline safety officer would be able to provide recommendations aimed at controlling the effect of contributing factors.

¹⁹ Note that the flight crew is often used as a means to mitigate a failure condition (i.e. stop it from propagating to a higher severity in the FHA). Human error is not included in probability analyses for 25.1309(b) compliance. Minimising human error is part of 25.1309(c) compliance.

²⁰ Flight crews are the last defence against accidents such as controlled flight into terrain (CFIT), approach-and-landing accidents (ALA) and loss of control (LOC) incidents. Unfortunately skill, vigilance and conscientiousness – while essential for safety – are insufficient to prevent error –specially when crew are interrupted, when preoccupied with one of several concurrent tasks, or when forced to defer an action out of its normal sequence.

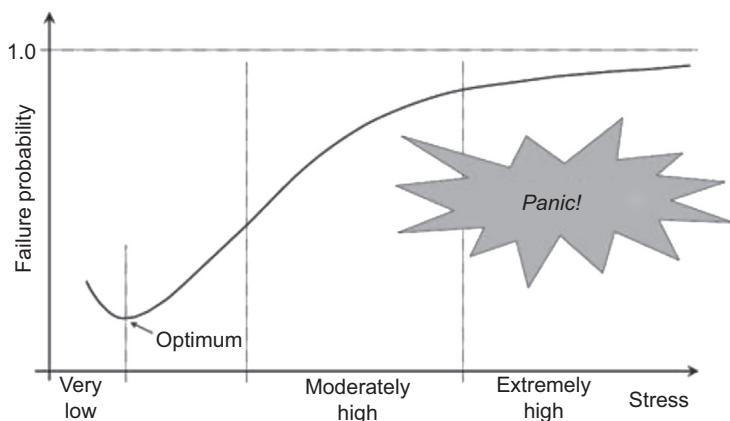


Figure 10.5 Crew error in a stress environment (Bell, as contained in Jacob's presentation 2002).

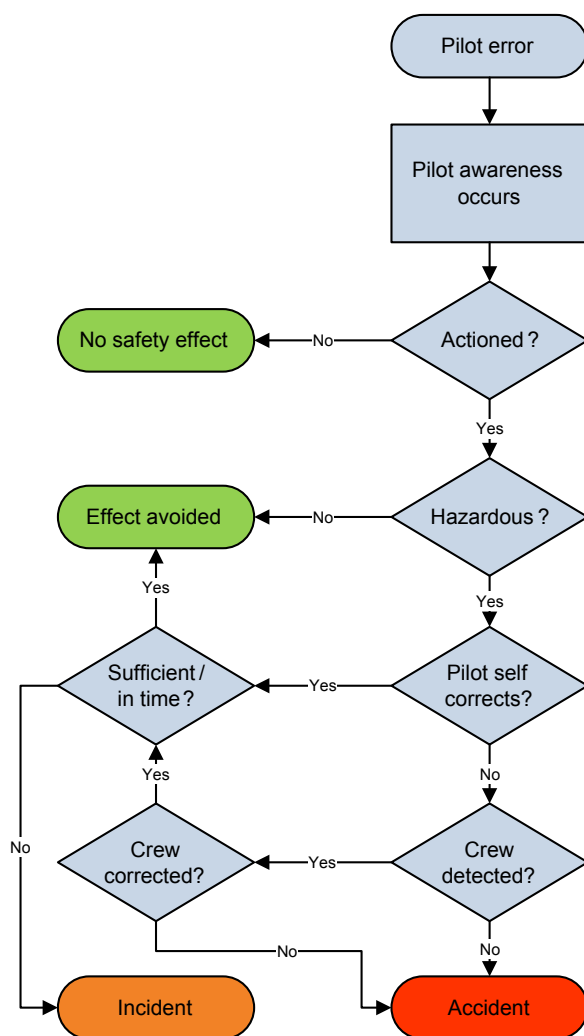


Figure 10.6 Pilot error in a multicrew cockpit.

10.2.5 Step 4: verification

Verification (refer Fig. 1.3) is the evaluation of an implementation of requirements to determine that they have been met (i.e. ‘Did we build the thing right?’).²¹ Finding #3 in the FAA study (2002) into ‘Aeroplane Safety Assurance Processes’ concluded that *‘A more robust...process that challenges the assumptions made in the safety analysis of flight critical functions is necessary in situations where a few flight critical failures (2 or 3) could result in catastrophic events’*.

Verification activities ensure that the physical specimens meet the requirements at each system level (refer Fig. 1.2). It is achieved by carrying out tests, simulations and/or analysis activities to provide evidence that:

- The system meets the requirements defined in the appropriate specifications
- The systems behaviour is in accordance with the design and that assumptions have not been optimistic.

For more mature design solutions, the emphasis in this phase is to support the full release (i.e. verified) of all Manuals and Flight Crew Reference Cards. Typical objectives to accomplish in the flight test phase include (tailored from [ER/TRI\(3320\)/06/0524](#)):

- *Assess Crew Workload*: Evaluate pilot performance and capacity, especially during failure scenarios as well as flying on standby instruments only.
- *Assess the potential for Distraction*: Evaluate the attention required to operate control and system input, including complexity, adequacy and time head down in the cockpit.
- *Assess the Human–Machine Interface (HMI)*: Confirm that information is presented clearly, unambiguously and is assessable. Buttons and bezels provide clear feedback and allow timely entry in all weather conditions (e.g. turbulence or day/night conditions). System delays do not confuse the pilots (or are clearly annunciated). Reversionary modes are easily assessable; clearly displaying which part of the system is in use. All system status indications are clear and unambiguous.
- *Assess Error Tolerance*: Ensure that detection and mitigation of interaction errors is supported (e.g. If FMS programming tasks were disrupted, or conducted under time pressure, the likelihood of errors is low).
- *Assess Automation*: Evaluate if the pilot is ‘in-the-loop’ (i.e. automaton behaviour is predictable, probability of automation surprises is low, understanding of automation transition and reversionary modes, etc.). Evaluate pilot authority over automation (e.g. switching back for automated system to manual operation is easily possible, including after failure conditions).
- *Assess Situational Awareness*: Evaluate ([Endsley, 2000](#)) Geographical SA (e.g. waypoints, navigation fixes, climb/descent points, etc.); Spatial/Temporal SA (e.g. attitude, altitude, projected flight path, etc.); System SA (e.g. system status, settings, impacts of degraded performance, etc.); Environmental SA (e.g. temperature, icing, IFR/VFR conditions, wind shear, etc.); and Tactical SA (capabilities and status, flight dynamics, mission timing and status, current and projected manoeuvring, threat detection and prioritisation).

²¹ A good question is, should it be ‘Did we build the right thing?’ or ‘Did we build the thing right?’ If we have validated that the requirements are correct, complete and fully traceable (i.e. validated), then verification is all about right ensuring that we have actually built it as specified.

A useful tool, prior to the flight testing phase, is the Operational Workload Analysis (OWA), which recognises that the likelihood of a human error in a task is directly related to the way the task itself is designed and the quality of the following key factors²²:

- workplace design (e.g. see Step 1),
- documentation (e.g. crew manuals and reference cards),
- operator competence (level of training, qualification, experience, etc.),
- the failure modes of the system (see Step 3).

The illustration in Fig. 10.7 attempts to summarise the key processes for a typical OWA. Most of these steps are completed by an HF specialist, with close cooperation and input from experienced flight crew (e.g. test pilots) and the System Safety Engineer(s).

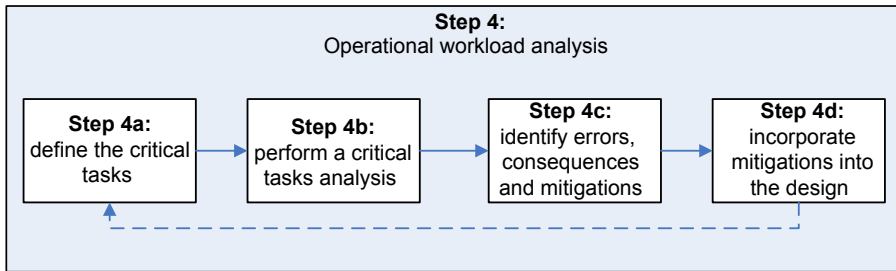


Figure 10.7 Operational workload analysis.²³

10.2.5.1 Step 4a: define the critical tasks

This involves considering each phase of flight and identifying those crew actions required for successful completion of that phase (e.g. transitioning from a landing mode to a go-around mode). Remember, if a procedure/task has not been thought of, then it has not been analysed.

10.2.5.2 Step 4b: perform a critical task analysis

Describe each critical task in terms of all the subtasks (or steps) needed to carry it out. In general, the more complex the system the more detailed the task analysis becomes. Task analysis encompasses many techniques for describing how people interact with systems. Generally, a task analysis is a way of systematically describing human interaction with a system to understand how to match the demands of the system to human capabilities. No one task analysis method²⁴ can provide a complete understanding of

²² Human Factors Briefing Note Nr 12, The Institute of Petroleum, London, 2003.

²³ Tailored from Human Factors Briefing Note Nr 12, The Institute of Petroleum, London, 2003.

²⁴ Kirwan and Ainsworth (1992) provide an exhaustive description of task analysis techniques.

the user or input to address all design decisions, but the following steps describe the basic elements of a task analysis:

- Define the analysis purpose and identify the type of data required,
- Collect task data,
- Summarise task data,
- Analyse task data.

Task analyses can be very time-consuming, so the process requires competent²⁵ oversight to ensure the effort is value adding, and as always, it is critical to focus the analysis on the end use of the data. A task analysis that has a broad scope and no specific focus may provide a comprehensive description of the system, but it may not be possible to complete with the available time and resource. Remember, one should always bear in mind that if a step or sequence of actions has not been considered, then it has certainly not been analysed.

10.2.5.3 Step 4c: identify errors, consequences and mitigations

Using the task analysis as a starting point, the System Safety Engineer can assist (via Step 3) the HF specialist to identify feasible types of error in the task and record the possible consequences of the error, and the safeguards and recovery mechanisms in place to prevent/detect/correct them. Be explicit in the recording of any assumptions (e.g. escalating or containing factors) which influence decisions made.

Should any of the following characteristics be present, then the team should pay particular attention to the cognitive components in conducting the analysis ([Gordon, 1995](#)):

- Complex decision making, problem solving, diagnosis, or reasoning
- Large amounts of conceptual knowledge needed to perform tasks
- Large and complex rule structures that are highly dependent on situational characteristics

There are various tools which can be used to assist (in isolation or in combination²⁶) with Step 3, for example:

- A Process FMEA (refer Table 5.7) or Human Error Mode and Effects Analysis (refer [SCF/SYS/A/108/4523](#)) can highlight the detectability and consequence of the human triggered Event following the classic bottom-up logic of the FMEA. It is typically applied to each step in a task (or process), with failure modes which include:
 - Action incorrect (including correct operation to wrong items, wrong operation to correct items and wrong operation to wrong item),
 - Action too early,
 - Action too late,
 - Action too briefly or too long,
 - No action (omission), etc.

²⁵ Because of the complexity of involved with activity, we cannot begin to give task analysis adequate coverage here. A good resource for Task Analyses is Kirwan and Ainsworth (1992), *A Guidebook to Task Analysis*, a book that describes 41 different methods for task analysis (with detailed examples). Cognitive task analyses are described in Seamster et al. (1997).

²⁶ For example, combine the FTA with an ETA to for a Bow Tie Analysis.

- An Event Tree Analysis (ETA), which explores all possible outcomes of an undesired event (i.e. the specific crew error of concern), because it is only when there is a derived architecture that the exact role of humans in the system becomes clear. By using FTA the potential role of critical Human Error is evident, as are the combinations of Failures and errors necessary to create Hazardous or Catastrophic systems' states.
- A Fault Tree Analysis (FTA), which can take the failure of a critical task and identify all subtasks and other causes contributing to that failure condition.

10.2.5.4 Step 4d: incorporate mitigations into the design

As a result of the investigations completed, the HF Specialist and the Safety Engineer must ensure that these derived requirements (refer Fig. 1.3) are captured in the System Specification(s) to ensure that they are traceable, validated (Step 2) and eventually verified (see Step 4).

The use of a flight simulator is a powerful tool to assist in verification activities; however, if these data are to be used as certification evidence, then the team will need to ensure that the simulator itself is verified to be a truly representative of the aircraft configuration and behaviour relative to the assessment.

Once the system has achieved some level of maturity and represents the end product, it is installed into an aircraft and the flight trials process can commence with a gradual expansion of the flight envelope under the stewardship of the SSA. This may represent the first time anyone evaluates display dynamics and system interfaces in the aircraft, so there is usually much to evaluate. Test pilots²⁷ will put the new aircraft system through its paces and then make recommendations for any design changes (e.g. recommending a stall warning system to be installed to aid the ordinary pilot). Although still very effective, this process tends to be reactive (i.e. post design) in its assessment, and therefore has limited mitigated options which must be explored at the earliest opportunity through early liaison between the designers and the test pilot.

The outputs of Step 5 include:

- An CS25.1309(c) compliant design.
- Updated Manuals and Flight Crew Reference Cards which are traceable to the safety decisions leading to their creation. With due consideration of the apt quote (by the American baseball player Yogi Berra 'I don't want to make the wrong mistake!'), such ICA (refer Section 11.2.3) must include explicit warning of those pilot errors which may cause a catastrophic or hazardous events.
- Recommendations for flight crew training (see Step 6), specifically those scenarios which may induce catastrophic and hazardous crew errors.

²⁷ In cases where the system is highly integrated, is complex and/or performs critical functions, the FAA uses a formal evaluation process (Holland et al.) that involves scenario-based evaluations by multiple FAA pilots. The process, termed Multiple Pilot System Usability Evaluation (MPSUE), has become a standard approach to evaluating complex avionics systems targeting general aviation aircraft.

10.2.6 Step 5: recommendations for crew training

A current trend in the aviation industry is increasing release of pilots from direct control of their aircraft. Computers are entering more and more as intermediaries in a supervisory control relation (Sheridan, 1987). The pilot is becoming a ‘flight manager’ supervising not one, but many computers. These computers are becoming intelligent subordinates to which the pilot gives high-level (macro) commands and specified goals, constraints, and contextual information. These subordinate computers then perform the direct control, executing the tasks as requested and reporting back to the pilot as to whether the goal has been achieved or whether there are any discrepancies. In assuming this new supervisory role, Sheridan (1987) remarks that the pilot undertakes five functions:

- Planning what to ask the computer to do,
- Teaching (i.e. commanding, programming) the computer,
- Monitoring its performance, detecting and diagnosing failures if they occur,
- Intervening to take over control directly if and when necessary and maintaining and repairing the semiautomatic systems,
- Learning from experience.

These functions can be reduced to the following categories of information processing (as coined by Rasmussen, 1986): skill-based, rule-based and knowledge-based.

Pilots successfully manage equipment malfunctions as threats that occur in normal operations.

However, Finding 4 in the FAA’s ‘Operational Use of Flights Management Systems’ concludes that insufficient system knowledge, flight crew procedure or understanding of aircraft state may decrease pilots’ ability to respond to failure situations. This is a particular concern for failure situations which do not have allied procedures or checklists, or where the procedures or checklists do not apply completely.

The FAA study (2002) into ‘Aeroplane Safety Assurance Processes’ concluded that procedures have to be more explicit and more robust with respect to the range of skills and techniques of operations. Training for today’s airline pilots includes many lessons learned from HF research. One topic which receives considerable attention is that of threat and error management (TEM). TEM recognises (Thomas et al., 2010) that even when flights are planned and aircraft are operated by trained and professional pilots in collaboration with dispatchers, mechanics, flight attendants, and others, human beings still make mistakes, especially when the environment presents challenges. The idea behind TEM is to accept this reality and train pilots to recognise errors as quickly as possible and manage, or mitigate, their negative impact.

Accordingly, Step 6 is reserved to make recommendations for recurring and nonrecurring training for flight crew to ensure they are expected competent in the use of the system during normal and during failure conditions. These recommendations are dependent on the following type of tasks to be conducted in the SSA process:

- Analysis (refer Step 3) that shows that all potential human errors have been identified and mitigated (including conditions/events that increase the probability/impact of error), refer Step 3.

- Prioritisation of the safety significance of human errors, bearing in mind the potential for detection and recovery of errors. The process FMEA (refer Step 3d and Table 5.7) is useful in this regard.
- Definition of operational procedures and ‘human-in-the-loop’ assessment of selected scenarios (see Step 4) to provide evidence that human performance in defined operational scenarios using those procedures will be acceptable.
- Audit trail showing that HF issues have been identified, integrated into relevant project activities and included in the design of equipment, procedures and training to mitigate safety risks.

Test Pilot(s), System Safety Engineer(s) and the HF Specialist(s) therefore need to consolidate their recommendations for type training. This training could consist of simulations, drills, verification examinations, certifications, etc. For multicrew cockpits, this also extends to specifying crew roles, including supervision, monitoring and cross-checking of safety critical actions. For more information, see EASA Operation Suitability Data (OSD), specifically CS-SIMD²⁸ and CS-FCF.²⁹

10.3 The Case Study

In Section 2.3 we defined a safety strategy for a modification programme where an aircraft attitude and altitude system (see Section 1.3) are upgraded. Below is an example³⁰ of where crew error, in using such a system, caused an accident:

Faulty altimeter a factor in plane crash

The Turkish Airlines B737-800 was landing on automatic pilot when a problem caused it to slow abruptly far short of the runway, sending it plunging into a muddy field, less than a mile short of the runway at the Amsterdam airport. At 1950 ft the aeroplane’s left altimeter suddenly and mistakenly registered an altitude of 8 ft below sea level and passed the reading on to the automatic control system. According to conversation recorded between the plane’s captain, first officer and an extra first officer on the flight, the pilots noticed the faulty altimeter but did not consider it a problem and did not react. But the autopilot reduced thrust to the engines and the aircraft decelerating until, at a height of 450 ft, it was about to stall. Warning systems alerted the pilots, who responded, but too late to recover.

The Boeing 737-800’s flight recorders also showed false readings from the altimeter on two flights before the 25 February crash. The Dutch Safety Authority said it had issued a warning to Boeing during its investigation, asking the company to alert customers that when altimeters did not function properly ‘the automatic pilot and the gas system coupled to them may not be used for approach and landing’, said the chief investigator.

²⁸ Certification Specifications and Guidance Material for Simulator Data.

²⁹ CS-FCF Flight Crew Data.

³⁰ Extract from an article by T Sterling on <http://www.independent.co.uk/news/world/europe/faulty-altimeter-a-factor-in-plane-crash-1637733.html>.

The requirement which drives the content of this chapter is CS25.1309(c) (Amendment 17) which requires that:

Information concerning unsafe system operating conditions must be provided to the crew to enable them to take appropriate corrective action. A warning indication must be provided if immediate corrective action is required.

Systems and controls, including indications and annunciations must be designed to minimise crew errors, which could create additional hazards.

With reference to Fig. 2.4, this section will explore that part of the strategy duplicated in Fig. 10.8 below:

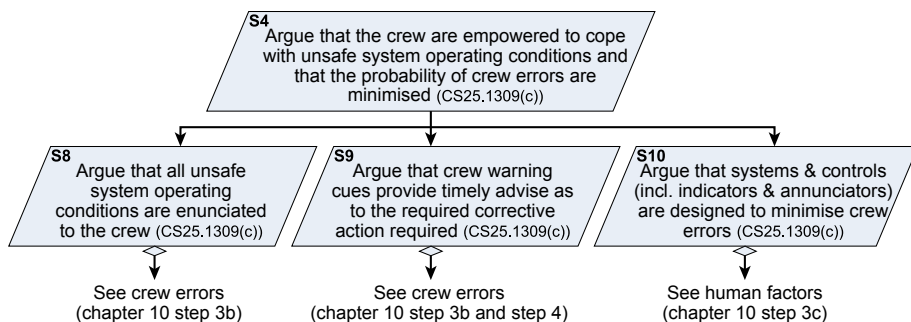


Figure 10.8 Safety strategy.

10.3.1 *Argue that systems and controls (incl. indicators & annunciators) are designed to minimise crew errors*

The accomplishment of this goal can be evidenced from the outputs of:

- Step 1g, where maximum use has been of ergonomics and standardisation,
- Step 3c, where the all sources of crew errors are mitigated,
- Step 4, where all requirements have been verified as accomplished in the actual system. This should include the reliability and integrity of any alerting systems (refer to para 4.4 in the FAA's *Human Factors Considerations in the Design and Evaluation of Flight Deck Displays and Controls*).

The safety assessor will need to highlight where this goal is not fully accomplished and how it will be mitigated in the ICA.

10.3.2 *Argue that unsafe system operating conditions are enunciated to the crew*

The accomplishment of this goal can be evidenced from the outputs of Step 3b, which ensures that unsafe system failure is communicated to the crew. For managing Display

failures, see para 5.3 in the FAA's *Human Factors Considerations in the Design and Evaluation of Flight Deck Displays and Controls*.

The safety assessor will need to highlight where this goal is not fully accomplished and how it will be mitigated in the ICA.

Table 10.2 proves a possible way to summarise the results of this part of the assessment into a format for use in the final SSA Report.

10.3.3 *Argue that crew warning cues provide timely advice as to the required corrective action require*

The accomplishment of this goal can be evidenced from the outputs of:

- Step 3b, which specifically identifies the way warning cues are detected by the crew following a failure condition.
- Step 4, where crew workload is factored into the corrective action response.

The safety assessor will need to highlight where this goal is not fully accomplished and how it will be mitigated in the ICA.

For more on flight crew alerts, see to para 4 in the FAA's *Human Factors Considerations in the Design and Evaluation of Flight Deck Displays and Controls*.

Table 10.2 can be used to capture the results of this part of the assessment, particularly to prove where the mitigation in the ICA has been verified (e.g. by simulator tests or flight tests).

10.4 Discussion

People and the jobs they do play an important safety role. Nowhere is this made more clear than in the study of aviation disasters, where, in more than two out of three cases, accident investigators are driven to conclude that human error played a major role (Edwards, 1988). These errors are not usually due to sudden illness, suicidal tendencies, wilful neglect or lack of basic abilities. More typically, they arise from temporary breakdown in skilled performance because, in many instances, system designers and managers have paid insufficient attention to human characteristics and skills, or not properly accounted for environmental stressors, workload and other reasonably foreseeable distractions.

The design of equipment (including the monitoring/feedback loop to its operators); procedures (e.g. how to deal with failure occurrence); and training (e.g. ensure timely intervention) has a major influence on the likelihood of an accident due to operator inability to cope with developing situation (such as too fast or intermittent flow of information, or out of reach/view location of controls).

Despite widespread awareness of the importance of HF in safety, it continues to play a key role in a majority of today's aircraft incidents and accidents. [Singer \(2002\)](#) correctly raises the concern that:

"The present methods of validating cockpit designs rely mostly on subjective statements and evaluations of a limited number of test pilots. This results in design solutions that have been approved for use without a realistic operational test and without objective or global agreed upon minimum acceptable performance levels. Unlike the technical system approval process that follows strict international standards and testing criteria, the HF interface evaluation lacks such standards. The evaluation and approval process today comprises several review phases with the appropriate feedback for change. The timing of the feedback is directly related to the effort and cost of each change. This constraint causes many manufacturers to defer design changes and instead apply "Band-Aid" fixes in the form of procedures, limitations and special training to overcome design weaknesses identified at the final stages of approval."

Many, if not most, products and systems are still designed and manufactured without adequate consideration of HF. Designers tend to focus primarily on the technology and its features without fully considering the use of the product from the human point of view. In a book that every engineer should read, [Norman \(1992\)](#) writes cogently:

"Why do we put up with the frustrations of everyday objects, with objects that we can't figure out how to use, with those neat plastic-wrapped packages that seem impossible to open, with doors that trap people, with washing machines and dryers that have become too confusing to use, with audio-stereo-television-video-cassette-recorders that claim in their advertisements to do everything, but that make it almost impossible to do anything?"

Poor design is common, and as our products become more technologically sophisticated, they frequently become more difficult to use. While HF is a difficult area, project experience suggests that waiting until the end of the project to find out how well the human-machine system meets the often loosely defined total system safety objectives is bad for both the customer and the supplier. Therefore, although the perfect solution may not exist, considerable risk reduction may be realised by defining safety targets for specific human actions. This enables iterative assessment of how well the proposed solutions may meet/have met the operational need, facilitate tradeoffs and enhance the integration of subsystems towards the overall system requirement.

Explicitly identifying and managing potential Human Errors is especially important during the product development life cycle. At the conceptual stages, designers have the greatest freedom and the cost of design and therefore design changes are minimal. As the design matures, design freedom is decreased and the subsequent costs associated

Table 10.2 Example crew error assessment summary

Failure/hazard					
ID	Source data	Description	Severity ^a	Probability ^b	Comments
1	System FHA ID4.1.1a	Loss of Primary Barometric Altitude Display (annunciated)	Hazardous	Extremely Remote	None
2	System FHA ID4.1.1a	Barometric Altitude Display Incorrect functioning (un-annunciated)	Catastrophic	Extremely Improbable	Although this failure mode is Extremely Improbable, there is nevertheless a probability of its occurrence
3	FMEA ID9	Port side Pitot Statics blocked (full or partial) or leaking	Catastrophic	Extremely Remote	Not a system failure, but an event caused by insects/bird strike
4	PRA ID1.1	Fire in avionics bay	Catastrophic	Extremely improbable	Although this event is Extremely Improbable, there is nevertheless a probability of its occurrence

selected failure modes only to illustrate the principle

^aSeverity in this column is based on the criteria in Table 3.2 and assumes crew will not make an error [25.1309(b)].

^bProbability in this column is based on the criteria in Table 3.3 and accounts for system failure probability only [25.1309(b)].

^cSeverity in this column is based on the criteria in Table 3.2 and assumes crew will make an error.

^dAlthough the probability in this column uses the same definitions as in Table 3.3, it is the additional probability of the crew error only. See Chapter 11 where this probability is added to the technical failure probability to provide for a risk which the operator needs to manage ion service (i.e. total system performance is a function of the H/W and S/W functioning correctly *and* the user performing the task correctly).

Crew error assessment						
Means of detection	Crew error	Severity of error ^c	Probability of error ^d	Justification	ICA status	Status
Pilots immediately aware of malfunction (either through failure flag or totally 'off-line') and will revert to use of standby display	None foreseen	N/A	N/A	N/A	N/A	Closed
Comparison with standby Altitude display EGPWS alert	Pilots may believe misleading instrument	Catastrophic	2×10^{-2}	2 pilots, each having a general omission error probability of 1×10^{-2} . Refer Table 10.1	TBD – To be compiled Note that one primary display receives pitot static data from same source as the standby display, so ensure pilots are able to diagnose correctly.	Open
DCU1 and DCU 2 discrepancy flagged on primary displays, but pilots are faced with 2 out of 3 displays showing the same incorrect data	See ID2	See ID2	See ID2	See ID2	See ID2	Open
Currently no smoke detection devises so detection will be due to <ul style="list-style-type: none"> • smoke/fumes entering the cockpit • multiple concurrent avionic failures 	Crew may take too long to respond	Catastrophic	TBD	TBD	TBD – need to provide fire suppression instructions in the Crew Reference Cards as well as warning to land ASAP	Open

with design changes increase. Fig. 10.9 illustrates that the ability to influence a system's characteristics diminishes rapidly as the system proceeds from one phase of its life cycle to the next. So it is important that this activity be conducted as early as possible in the development process because of the influence that it may have on system architecture.

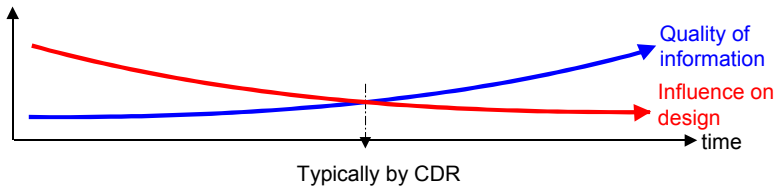


Figure 10.9 Safety influence versus product development life cycle.³¹

Process Management (*Cost Arguments and Evidence for Human Factors Integration, 2006*) is a critical activity to be carried out throughout the design process, ensuring efficient integration of HF activities into the design process. For example:

- It ensures that professionals with the right level of expertise are available when needed, including sufficient integration into the design team.
- It may draw on organisational mechanisms ensuring that HF expertise is considered and integrated into the design process.
- It is important to note that the way HF practitioners communicate their findings to design engineers is essential for the effective transmission of information, since it often involves aligning different perspectives on overlapping issues.

A well-functioning process ensures that emerging HF problems can be addressed quickly. Organisational barriers may need to be overcome such as:

- organisational fragmentation inhibiting collaboration;
- long-term pay-offs of HF effort conflicting with short-term savings;
- users (e.g. pilots) being difficult to access. This is especially true when seeking input from 'real world currently operating crews' as opposed to test pilots. The latter bring both HF advantages due to their wealth of experience and disadvantages for the same reason.

Other barriers may be of a perceptual nature (e.g. questioned credibility of specialists; users are seen as part of the problem, not part of the solution).

There is a need to avoid treating HF as stand-alone component. Human Factors Integration (HFI) needs to be recognised as an intrinsic element of system design. Finding 16 in the FAA's report on *Operational Use of Flights Management Systems* concludes that HF expertise has been increasingly incorporated into the design process at most manufacturers, but is still inconsistently applied at some manufacturers. Another report by the FAA (*The Interfaces Between Flightcrews and Modern Flight Deck Systems*) found that in most design processes:

- 'Automation design principles are often not defined, documented or distributed to appropriate design, test or training personnel.
- Some flight crew cognitive tasks are not comprehensively identified or considered in the design.

³¹ This illustrates that problems experienced downstream are symptoms of neglect upstream.

- *Flight crew information and feedback requirements are not always clearly identified or given high priority in making design trade-offs.*
- *During the design process, flight crew task allocation is not clearly identified either between flight crew members or between the automation and the flight crew. This can result in imbalances between tasks allocated to the pilot-not-flying versus pilot-flying.*
- *Designers sometimes make flight-deck and display design decisions based on subjective assessments in balancing flight test pilot input, chief pilot or project pilot input, operator input and economic input instead of being data (or service-history) driven.³²*
- *Flight test evaluation is able to address many human performance concerns, but cannot address them all. In some cases where it is considered too expensive to change the design, a procedure is developed to address the concern. An effort must be made to minimise this method of fixing vulnerabilities in the design. The concern here is that the 'fix' may mask the real problem, and if this operational procedure should be revised or eliminated sometime in the future, the original design problem may become a hazard.'*
- Furthermore, the FAA's report on [Operational Use of Flights Management Systems](#) concludes that HF specialists may not exist in some organisations or are called upon (either in-house or at another manufacturer) to resolve or mitigate crew-centred issues that are discovered late in the design schedule.

Although this chapter has attempted (in [Fig. 10.1](#)) to provide a basic approach in how to mitigate crew errors as part of the system development life cycle, there remains very much more research to be done. The FAA's report on [Operational Use of Flights Management Systems](#) makes the following relevant recommendations:

- Recommendation 5 addresses the 'Verification and Validation for Equipment Design' and recommends that 'Research should be conducted and implemented on processes and methods of verification and validation (includes validation of requirements) during the design of highly integrated systems that specifically address failures and failure effects resulting from the integration'.
- Recommendation 8 addresses the 'Design of Flight crew Procedures' and recommends 'For the near term, update guidance (e.g. Advisory Circular (AC) 120-71A) and develop recommended practices for design of SOPs based on manufacturer procedures, continuous feedback from operational experience and lessons learned. This guidance should be updated to reflect operational experience and research findings on a recurring basis. For the longer term, conduct research to understand and address when and why SOPs are not followed. The activities should place particular emphasis on monitoring, cross verification and appropriate allocation of tasks between pilot flying and pilot monitoring'. Monitoring is a skill and, like any other skill can be improved upon ([Sumwalt et al., 2002](#)).
- Recommendation 12 addresses 'Flight Deck Design Process and Resources' and recommends 'Ensure that appropriate HF expertise is integrated into the flight-deck design processing partnership with other disciplines with the goal of contributing to a human-centred design. To assist in this process, an accessible repository of references should be developed that identifies the core documents relevant to 'recommended practices' for human-centred flight-deck and equipment design. Early in the design process, designers should document their assumptions on how the equipment should be used in operation'.

³² Be data (or service history) driven is difficult to achieve when introducing new technologies.

10.5 Conclusions

The HF result of a design is only as good as the tools used and the methods applied during its assessment. A simple recipe for understanding HF design includes:

- Understanding the causes of error – and then design to minimise those causes.
- Making it possible to reverse actions – to undo them – or make it harder to do what cannot be reversed.
- Making it easier to discover the errors that do occur, and make them easier to correct.
- Changing the attitude towards error. Think of a user as attempting to do a task, getting there by imperfect approximations. Do not think of the user as making errors, think of the actions as approximations of what is required.

If a safety-related system requires human intervention in its operation, then HF aspects (e.g. clarity of data presentation and complexity of input) must be given adequate consideration. The human beings who operate and maintain safety-related systems must also be regarded as part of the system and they should be regarded as both lines of defence and potentially hazardous elements in the system.

Flight crews are the last defence against accidents such as controlled flight into terrain (CFIT), approach-and-landing accidents (ALA) and loss of control (LOC) incidents. Unfortunately, skill, vigilance and conscientiousness – while essential for safety – are insufficient to prevent error. This is especially so when crew are interrupted, when preoccupied with one of several concurrent tasks, or when forced to defer an action out of its normal sequence.

Fundamentally, safety is underpinned or undermined by human capabilities and limitations, and to err is human. Accident investigators have moved away from the position of regarding the phrase ‘pilot error’ as an appropriate explanatory cause. It is far more appropriate to ask why the error was made and why it was not detected and corrected in time, including the creation of barriers (or defensive layers) through the design of suitable interfaces, training, operational procedures etc.

10.5.1 Advantages

Human error is an important consideration in complex safety critical systems, because it makes the most significant contribution to overall system risk (Edwards, 1988). The goal of a HF assessment is to make systems successful by enhancing performance, satisfaction and safety. Clearly it is not practical or cost-effective to develop and/or test all possible combinations of conditions that could affect human performance. Nevertheless, a systematic and informed consideration of the human as part of the safety risk management process can provide significant risk reduction, even if all risks are unlikely to be fully alleviated. Explicitly identifying and managing human error risks throughout the product life cycle have the knock-on benefit of improving operational effectiveness.

Error reduction, however, is not the only approach to the problem of error. The second line of attack is directed towards the elimination of disastrous consequences of human error. The design of equipment (including the monitoring/feedback loop to its

operators), procedures (e.g. how to deal with the occurrence) and training (e.g. ensure timely intervention) has a major influence on the likelihood and result of human errors.

10.5.2 Limitations

Addressing the risks posed by human error presents challenges to traditional approaches to safety risk management. Many of the challenges arise because of the very nature of human error (e.g. the nature of the task, conditions of operation, human frailties such as emotions or fatigue, team factors such as supervision, etc.), which has numerous causes and can therefore be difficult to fully understand, predict, model or prevent.

The goal of a SSA is to make systems successful by enhancing performance, reliability and safety. Unfortunately many systems are still designed and manufactured without adequate consideration of the human's role in the system. Designers tend to focus primarily on the technology and its features without fully considering the use of the product from the human point of view. Even when designers attempt to consider it, they often complete the product design first and only then hand off the blueprint or prototype to a HF expert. The HF expert is then placed in the unenviable position of having to come back with criticisms of a design that a person or design team has spent much effort to develop. It is thus not hard to understand why engineers are less than thrilled to receive the results of a HF Assessment. They have invested in the design, clearly believe in the design, and are often reluctant to accept HF recommendations.

The process of bringing HF analysis in at the end of the product design phase inherently places everyone involved at odds with one another. In contrast, an integrated proactive approach can lead to systems which are less vulnerable to human error and save a company time and money in its certification.

References

- Bell, B.J. Human Error Evaluation and Human Reliability Analysis, American Institute of Chemical Engineers, as contained in Jacob's presentation 2002.
- Boff, K., Lincoln, T., 1988. Engineering Data Compendium: Human Perception and Performance, 4 Volumes. Wright-Patterson Air Force Base, OH: Armstrong Aerospace Medical Research Laboratory, AAMRL/NATO.
- Boff, K., Monk, D.L., Swierenga, S.J., Brown, C.E., Cody, W.T., 1991. Computer-aided human factors for systems designers. In: Proceedings of the Human Factors Society 35th Annual Meeting. Human Factors Society, Santa Monica, CA, pp. 332–336.
- CAP 719, Fundamental Human Factors Concepts, Safety Regulation Group, UK CAA, Gatwick. Cost Arguments and Evidence for Human Factors Integration. October 2006. Issue 1. Produced by Systems Engineering Assessment Ltd on Behalf of the MoD HFI DTC.
- Commercial Airplane Certification Process Study, March 2002. An Evaluation of Selected Aircraft Certification, Operations and Maintenance Processes. The Federal Aviation Authority. DOT/FAA/AM-01/17, October 2001. Human Factors Design Guidelines for Multifunction Displays. Office of Aerospace Medicine, Washington, DC 20591.

- DOT/FAA/AR-03/69, Human Factors Acquisition Job Aid, Federal Aviation Administration, Human Factors Research and Engineering Division.
- Edwards, E., 1988. Part 1: Introductory overview. In: Wieger, Nagel (Eds.), *Human Factors in Aviation*. Academic Press.
- Embrey, D., 2000. Task Analysis Techniques. Human Reliability Associates Ltd. downloaded from: <http://www.humanreliability.com/articles/Task%20Analysis%20Techniques.pdf> on 1/12/15.
- Endsley, M.R., Bolte, B., Jones, D.G., 2002. *Designing for Situational Awareness: An Approach to User-centered Design*. Taylor and Francis, London. ISBN:0-748-40967-X.
- Endsley, M.R., May 2001. Training for situational awareness. In: A Presentation to the Royal Aeronautical Society. SA Technologies Inc.
- Endsley, M.R., February 2000. Flight crews & older aircraft: in search of SA. In: A Presentation to the Royal Aeronautical Society. SA Technologies Inc.
- ER/TRI(3320)/06/0524, Dec 2009. RAF Tristar Human-machine Interface Report. Marshall Aerospace, Cambridge.
- Gordon, S.E. (1995). Cognitive task analysis using complementary elicitation methods. Proceedings of the Human Factors and Ergonomics Society 39th Annual Meeting.
- Harris, D., Howard, R., Cook, S., February 12, 2002. A Systems Approach to Safety in a Multi-crew Aircraft, an INCOSE UK Paper V3. Systems Engineering and Evaluation Centre, University of South Australia, Mason Lakes Campus, Adelaide, Australia.
- Harris, D. Cockpit ergonomics, A Paper Presented to the Human Factors Group, College of Aeronautics (Date Unknown).
- Hartel, Smith & Prince, Defining aircrew coordination: Searching mishaps for meaning. Paper presented at the Sixth International Symposium on Aviation Psychology, Columbus, OH, 1991.
- Human Factors Considerations in the Design and Evaluation of Flight Deck Displays and Controls, FAA Final Report Version 1, Federal Aviation Administration, Human Factors Division, Washington, downloaded from http://ntl.bts.gov/lib/50000/50700/50760/General_Guidance_Document_Nov_2013_v1.pdf on 1/12/15.
- Human Engineering Design Data Digest, Human Factors Standardisation, Department of Defense, Human Factors Engineering Technical Advisory Group.
- Jarrett, D., 2005. *Cockpit Engineering*. United Kingdom: Ashgate Publishing Ltd, Farnham.
- MIL-STD-1472F, 1998. Human Engineering Design Criteria for Military Systems, Equipment, and Facilities. U.S. Department of Defense, Washington, DC.
- MoD, 2000. Human Factors Integration: An Introductory Guide. HMSO, London, (as quoted in Stanton et al).
- Norman, D.A., 1992. *Turn Signals Are the Facial Expressions of Automobiles*. Addison-Wesley Publishing, Reading, MA.
- Operational Use of Flights Management Systems. September 2013. Report of the PARC/CAST Flight Deck Automation WG. FAA, Washington.
- Rasmussen, J., 1986. *Information Processing and Human-machine Interaction*. Elsevier North-Holland, Amsterdam.
- Reed, P., Billingsley, P., 1996. Software ergonomics comes of age: the ANSI/HFES-200 standard. In: Proceedings of the Human Factors and Ergonomics Society 40th Annual Meeting. Human Factors and Ergonomics Society, Santa Monica, CA, pp. 323–327.
- Rogers, T.G., Armstrong, R., 1977. Use of human engineering standards in design. *Human Factors* 19 (1), 15–23.
- Rogers, T.G., Pegden, C.D., 1977. Formatting and organizing of a human engineering standard. *Human Factors* 19 (1), 55–61.

- SCF/SYS/A/108/4523, 2001. Human Hazard Analysis: A Demonstrator for a Means of Compliance, a Technical Report by the System Centre of Competence. Airbus UK Ltd, Filton.
- Sheridan, T.B., 1987. Supervisory control. In: Salvendy, G. (Ed.), *Handbook of Human Factors/ergonomics*. Wiley, New York.
- Singer, G., March, 2002. *Methods for Validating Cockpit Design, the Best Tool for the Task*. Department of Aeronautic, Kungliga Tekniska Högskolan (KTH), Royal Institute of Technology, Stockholm. SE-100 44.
- Stanton, N., Salmon, P., Rafferty, L., Walker, G., Baber, C., Jenkins, D., 2013. *Human Factors Methods : A Practical Guide for Engineering and Design*. Ashgate Publishing.
- Statistical Summary of Commercial Jet Accidents Worldwide Operation, June 2001. The Boeing Company, 1959–2000, p. 21.
- Sumwalt, R., Thomas, R., Dismukes, K., November 4–7, 2002. Enhancing flight-crew monitoring skills can increase flight safety. In: 55th International Air Safety Seminar, Dublin, Ireland.
- The Interfaces Between Flightcrews and Modern Flight Deck Systems. June 18, 1986. FAA Human Factors Team Report.
- Thomas, R., Chidester, T., Hackworth, C., January/February 2010. The Importance of the Human Element. FAA Aviation News.
- Woods, D.D., Johannesen, L., Potter, S.S., 1992. The sophistry of guidelines: revisiting recipes for color use in human-computer interface design. In: *Proceedings of the Human Factors Society 36th Annual Meeting*. Human Factors Society, Santa Monica, CA, pp. 418–422.

Further reading

- Alexander, D.C., 1995. The economics of ergonomics: Part II. In: *Proceedings of the Human Factors and Ergonomics Society 39th Annual Meeting*. Human Factors and Ergonomics Society, Santa Monica, CA, pp. 1025–1027.
- DEF STAN 00-250, Human Factors for Designers of Systems, UK MoD.
- Dennison, T.W., Gawron, V.J., 1995. Tools and methods for human factors test and evaluation: mockups, physical and electronic human models, and simulation. In: *Proceedings of the Human Factors and Ergonomics Society 39th Annual Meeting*. Human Factors and Ergonomics Society, Santa Monica, CA, pp. 1228–1232.
- Fewins, A., Mitchell, K., Williams, T.C., 1992. Balancing automation and human action through task analysis. In: Kirwan, B., Ainsworth, L.K. (Eds.), *A Guide to Task Analysis*. Taylor & Francis, London, pp. 241–251.
- Gordon, S.E., 1995. Cognitive task analysis using complementary elicitation methods. In: *Proceedings of the Human Factors and Ergonomics Society 39th Annual Meeting*. Human Factors and Ergonomics Society, Santa Monica, CA, pp. 525–529.
- Gould, T.D., 1988. How to design usable systems. In: Helander, M. (Ed.), *Handbook of Human-computer Interaction*. Elsevier, The Netherlands, pp. 757–789.
- Hamilton, D.B., Bierbaum, C.R., 1990. Task Analysis/Workload (TAWL): a methodology for predicting operator workload. In: *Proceedings of the Human Factors Society 34th Annual Meeting*. Human Factors and Ergonomics Society, Santa Monica, CA, pp. 1117–1121.
- Holland, J., Ryan, W., Jan/Feb 2010. Factoring in the Human in Avionics Certification. FAA Aviation News, pp. 13–15.
- Kirwan, B., Ainsworth, L.K., 1992. *A Guide to Task Analysis*. Taylor & Francis, London.
- Riley, V., March 2005. Reducing Mode Errors Through Design. *Avionics Magazine*, p. 20. www.avionicsmagazine.com.

- Seamster, T.L., Redding, R.R., Kaempf, G.F., 1997. Applied Cognitive Task Analysis in Aviation. Ashgate Publishing, Brookfield, VT.
- Shappel, S., Wigman, D., Feb 2000. The Human Factors Analysis and Classification System—HFACS. DOT/FAA/QM-00/7. Office of Aviation Medicine, FAA, Washington.
- Sumwalt, R.L., Thomas, R.J., November 1999. Enhancing safety through error management, Air Line Pilots Association. In: A Paper Presented at the Joint Meeting (FSF, IFA and IATA), Enhancing Safety in the 21st Century, Rio de Janeiro, Brazil.
- Chidester, T.R., Hackworth, C.A., January/February 2010. The Importance of the Human Element. FAA Aviation News.
- Wiener, E.L., Nagel, D.C., 1988. Human Factors in Aviation. Academy Press.

Annex A to Chapter 10

Human-machine interface in aerospace

(selected extracts from a paper by Vahid Norouzalibeik)

A.1 Background

Safety is paramount in aviation. Evidence shows that human reliability is often not consistent in high-pressure situations with serious time constraints because, among other factors, humans' sensory systems are not designed for three-dimensional environments. It is also undeniable that there is more than one factor in the *chain of events* resulting in catastrophe, the user interface being an important aspect of it. With increasing demand in the industry, defensive layers continue to move from *reactive* to *proactive* and *preventive* thinking. Aviation psychology plays an important role in accident prevention and the human element must be carefully studied and managed.

People think computers will keep them from making mistakes. They're wrong. With computers you make mistakes faster.

Adam Osborne

Obviously, as one moves from a stand-alone computer to a computer system to a computerised system, the complexity of the interfaces increases and the designer must be concerned not only for computer-operator relationships, but also for computer-machine relationships. The interface is a medium transporting information from an output boundary of one system to the input boundary of another. The designer must also have some concept of what the intelligent human-computer interface should do in the new system. The improvements in the HMI were largely an undertaking of the designers, builders, and fliers of the machines.

With the introduction of the *Fly-By-Wire* system in commercial aviation in the late 1980s, *automation surprise* has been considered as one of the major contributing factors in aircraft incidents and accidents. In the wake of Air France's Airbus A330 crash in 2009 (where temporary blockage of the pitot tubes by icing started it off and the

irrational inputs on the side-sticks from the pilot made the situation fatally worse), flight-deck design has raised questions and concerns. Where a very complex work station such as an aircraft cockpit has to be designed, developing an increasing emphasis upon certain characteristics in the layout proves itself essential. While the overall design process focuses on several subordinate, parallel processes with the SHELL³³ model in mind, the designer must have some concept of what the intelligent human-computer interface should be able to do – and his understanding must be in place before design begins so as to ensure that the HMI design is proactively focussed on the accuracy, clarity and nonambiguity of information.

While the interface analysis techniques are used to assess the interface of a product or system in terms of usability, error, user-satisfaction and layout, the HMI encompasses displays, alarms and manual controls. Major aircraft manufacturers (such as Boeing and Airbus) often follow very different approaches to the very important aspect of HMI in their designs and crew interface during normal and emergency situations. Using accident case studies, this Annex aims to elaborating on a few of these preferences and differences, with their advantages and limitations.

A.2 Human-machine interface and the flight-deck

The human's interaction with the flight-deck displays has long been recognised as an issue. The analysis of many accidents indicates that deficiencies in the design started the chain of events leading either directly to the accident or inhibited recovery by making diagnosis of any system failures difficult. This means that these accidents might have been avoided.

We spent over fifty years on the hardware, which is now pretty reliable. Now it's time to work with people.

Donald Engen (ex-FAA)

The machine interface tasks fall into two clusters on the flight-deck: the actual control of the aircraft (either manually or through computer-based flight management systems) and adherence to established procedures for the conduct of flight. Having a horizon of expectation, users approach, utilise and become part of a system with goals they want to achieve. Realising that users' goals may vary and that their goals strongly influence what they perceive, designers should understand and keep those goals in mind. Although tradeoffs are unavoidable in design, cognitive aspects of task performance need to be sufficiently considered and adequately addressed in design. The HMI often becomes the determining factor in the event of an emergency, in which correct, timely decisions and execution make the difference between life and death.

If machine and man are to be matched to form an integral working unit, close attention must be paid to the area of contact between them – the display and control interface – to reconcile their fundamentally different characteristics. In human-computer interaction, 'WYSIWYG' (what you see is what you get) user interfaces are very good

³³ SHELL: Software, Hardware, Environment and Liveware.

examples of mediating environments that enable humans and machines to interact easily. Problems involved in man's communication with the complex computer are in many respects similar to those problems involved with his communication with another man who speaks an unfamiliar language. Translation proves itself essential in either situation in order for an effective communication to take place. Poor design of equipment could result in latent errors and failures (such as faulty maintenance and bad management decisions) produced by actions that are removed in space and time from those of the actual operator directly interfacing with the system.

Considering technical systems from an ergonomic point of view, HMIs have to be designed according to the user's capabilities, taking into account perception as well as cognition of the user interface. Perceptual ergonomics includes classic design factors that support human information processing during stimulus uptake, short-term sensory storage and the following perception. In doing so, sensory impressions are identified and by means of cortical signal processing assigned to concepts of the long-term memory that reflect former experiences and memories. A central design principle of cognitive ergonomics is the compatibility of a technical system with the mental model of the user. To minimise transformation efforts, it is useful to provide information in a form that complies with the user's mental model necessary to perform the task.

New technology does not *remove* human error; it *changes* it. The interfaces may look simple or appealing, but they can hide a lot of complexity since computers can undermine people's formation of accurate mental models of how the system processes. Computers are supposed to divest people of their work, but the demand to interact with computers often concentrates itself on exactly those times when there is already a lot to do; when other tasks or people are also competing for the operator's attention. Computers can make things invisible; hide interesting changes, events, or system anomalies. The characteristics of computer technology shape the way in which people assess, think, decide, act and coordinate, which in turn determines the reasons for their errors.

Cues that signal a problem are not always clear-cut. Poor interface design that does not provide adequate diagnostic information or action feedback can lead a crew astray. In the British Midland Airways flight BD092 accident in Kegworth on January 8, 1989 – the worst air accident in the United Kingdom in many decades – badly grouped and poorly displayed engine instrumentation in a B737-400 cockpit led to a catastrophic visual selection error, contributing to the flight crew shutting down the wrong engine. The Boeing 737-400 was the newest Boeing plane, introduced only four months earlier, and only 17 were in service worldwide when this crash happened. The plane had been delivered to the airline just 12 weeks prior and had flown 520 h. On the liveware side in the flight-deck, the Captain and the First Officer of flight 092 had only 23 and 53 h of experience, respectively, with the updated instrument panel in the B737-400. There is a CRT display at the centre of the B737-400 panel presenting primary and secondary data relating to operation of the twin CSM56-3C turbofan engines. By convention the engines are numbered from the left facing forward: No. 1 being the left and No. 2 being the right engine.

The pilots noticed a strong shaking of the aircraft and the No. 1 engine vibration indicator began to fluctuate. Having detected the engine malfunction, the Captain took over control of the aircraft from his First Officer, his first task being to disengage the autopilot. When the Captain asked his First Officer about the faulty engine, the First Officer, after some hesitation, replied that it was the No. 2 engine. Investigators determined that the pilot apparently shut down No. 2 engine (right engine) even though the plane was having trouble with its No.1 engine (left engine). Experts said it was highly unlikely the pilot could have confused the two engines, given the system of double checks between pilot and copilot and the cockpit layout. Neither pilot had had a chance to practice interpretation of engine problems on a simulator. The first time they saw abnormal engine indications was in a real flight resulting in 47 fatalities and many injuries.

A.3 Automation

Automation changes the nature of the human task, and there may be many control options. So it has become increasingly important to distinguish *direct control* from *supervisory control* and the forms it may take. Direct control means that a command is sent directly to the machine with no intervention by a computer. In supervisory control, the human command is sent to a computer and the computer processes the human command using its own programmed intelligence to determine the final command transmitted to the aircraft. One significant approach to the design of human-machine systems is one where the *cognitive work* is shared by humans and automation, aiming at a work environment where the crew is involved in all operations, maintains a reasonable workload level leading to high situation awareness at all times.

There is nothing remarkable about it. All one has to do is hit the right keys at the right time and the instrument plays itself.

Johann Sebastian Bach

The A320 entered airline service in March 1988, and just 3 months later the first one crashed on Sunday 26 June 1988. The chartered Air France Flight 296 crashed during a demonstration flight in an air show in Habsheim, France. Flying at low altitude, the crew engaged a digital pitch mode that provided a relatively slow thrust response to throttle movement. The cause of this accident and similar ones involved three major issues: cockpit crew error, crew interaction with automation, and controlled flight into terrain (CFIT). CFIT occurs when an airworthy aircraft is flown, under the control of a qualified pilot, into terrain (water or obstacles) with inadequate awareness on the part of the pilot of the impending collision.

At the controls were two senior Air France pilots, both with unblemished records, and between them more than 21,000 h of flying time. After years in command of Boeings, the captain was embracing this revolutionary design. Known as problems of cockpit crew interaction with automation or *automation surprise*, it is a scenario arising from the complexity of the computer control system coupled with a high cockpit workload.

Human error is suspended somewhere between the human and the engineered interfaces. The error could be neither fully human nor fully engineered. At the same time, mechanical ‘failures’ (providing identical switches located next to one another) get to express themselves in human action. One insight of early human factors work was that machinistic feature and human action are intertwined in ways that resist the neat, dualist, deconstructed disentanglement still favoured by investigations today.

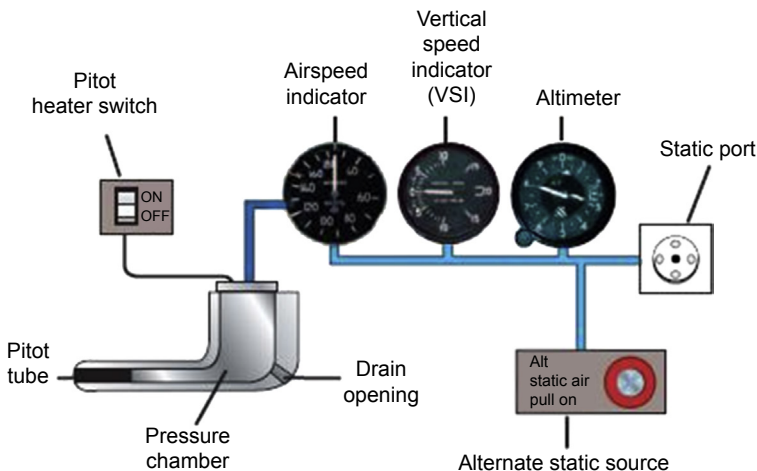
A.4 The pitot/static system

All aircraft utilise a pitot/static system to determine and display airspeed and altitude:

- An externally mounted pitot tube has a very small opening in the end facing into the air-stream that measures ram air pressure. Indicated airspeed is related to the difference between total pressure and static pressure, called dynamic pressure $= P_T - P_s = 0.5\rho V^2$.
- A static port, a very small opening on the outside of the fuselage, measures the ambient or nonram air pressure. Displayed altitude is directly related to static pressure, where 1" mercury decrease = 1000 ft.

If the pitot probes are blocked at takeoff, air will be trapped at runway elevation static pressure. The failure symptoms are as follows:

- At approach for takeoff (i.e. V_1 to V_R to V_2), the sensed dynamic pressure = 0, so airspeed indicator remains pegged at its lower stop
- However, at takeoff, the airspeed indicator will suddenly come to life because the static pressure starts to drop. The indicated airspeed continues to increase through the correct value as the aeroplane climbs, but continues until the V_{mo} can appear to be exceeded, triggering an overspeed warning.



If the pilot trusts the faulty airspeed indicator, there is grave danger of increasing pitch or reducing thrust or both to reduce the erroneous indicated airspeed. This could result in an aircraft stall.

If the static are blocked at takeoff, air will be trapped at runway elevation static pressure. The failure symptoms are as follows:

- During takeoff, both the altimeter and the airspeed indicator operate correctly.
- However, during flight, the:
 - Altimeter indication remains at the field elevation.
 - The sensed dynamic pressure (i.e. indicated airspeed) fails to increase as rapidly as it should during climb. If the aircraft actually climbs at a constant speed, the airspeed indication decays, eventually reaching near zero.

If the captain relies on the airspeed indicator, the typical response will be to reduce the pitch attitude to maintain the erroneous airspeed, possibly causing the aircraft to exceed its airspeed limitations. Overspeed warning may not operate if connected to the same erroneous airspeed source.

Incident and accident reports identified several reasons (all potential PRA or CMA candidates) for pitot-static failures:

- Pitot probe and/or static port covers not removed
- Pitot or static hoses disconnected
- Hoses leaking
- Water trapped in lines
- Pitot probes blocked by volcanic ash
- Radome damaged
- Aeroplane icing
- Pitot probes or static ports blocked by insects, etc.

The ability to predict the effects of a partial blockage is hindered when:

- anomalies occur during different phases of flight,
- the amount of blockage differs, or
- blockage corrects (or appears to) correct itself.

Paragraphs A5 to A8 below show how the HMI with the pitot-static system has contributed to many accidents.

A.5 Air France flight 447

On 31 May 2009, Air France 447, an Airbus A330-200, entered oceanic airspace two and half hours after takeoff for a 12 h flight to Paris. While flying at 35,000 ft, the pitot probes became blocked with ice crystals, leading to erroneous speed indications in the flight-deck which caused the autopilot and autothrottle to disengage, i.e. the normal Airbus stall prevention mechanisms were no longer active. At this point, there were two First Officers occupying the cockpit seats while the captain was taking his rest break. No particular briefing had been conducted by the captain to designate who would serve as pilot in command. The captain had discussed that the 32-year-old FO

(the least experienced pilot of the three with 800h on the A330) would be doing the landing and thus was to be the pilot flying at that point. The 37-year-old FO (with almost 4500h on the A330) was in the left seat. Shortly after the autopilot and auto-throttle disengaged, the stall warning triggered and the pilot flying made rapid roll control inputs, then pitched the aircraft up (to 11 degrees) and the aircraft climbed (to 37,000ft). In spite of attempts being made to reduce the pitch, the aircraft continued to climb. Shortly afterwards, the stall warning was triggered again and the thrust-levers were advanced but the pilot flying continued to make pitch-up inputs to the side-stick instead of pitch-down inputs required in a stall situation. By the time the captain reentered the flight-deck, the angle of attack was too steep (40degrees) and they were descending at 10,000 ft per minute. The pilot flying continued commanding a pitch-up, causing the excessive nose attitude that led to stall. The doomed aircraft failed to recover from stall and crashed into the sea, killing all 228 people onboard.

A.5.1 Control laws in the airbus A330

Depending upon the status of the fly-by-wire system, three sets of control laws are provided: Normal Law, Alternate Law and Direct Law. Under most circumstances, the aircraft is operated in Normal Law. Normal Law is designed to accommodate single system failures. Normal Law provides five different protections:

- High Angle of Attack Protection
- Load Factor Protection
- High Pitch Attitude Protection
- High Speed Protection
- Bank Angle Protection

When the pitot tubes clogged with ice crystals in Air France 447, the autopilot disengaged and the flight director bars disappeared. At this time the aeroplane's flight control law changed from Normal to Alternate, shutting down most of the built-in protections and increasing the sensitivity to roll inputs due to the loss of indicated airspeed.

A.5.2 Stall warning in the A330

A stall condition in Boeing designs is indicated by the shaking of both the Captain's and First Officer's control columns. When the aeroplane AOA equals or exceeds the computed trip value, the shakers are activated and remain activated until AOA is reduced below the trip value. Also in commercial designs by Boeing, pilots' control columns are interconnected by a torque tube assembly with a forward control quadrant mounted at each end. In the Airbus A330 the stall warning is a synthetic voice that says, 'STALL, STALL, STALL' accompanied by a cricket sound and a red master warning light in front of each pilot. Unlike the stick shaker stall warning found in many other transport aircraft, the stall warning in an A330 does not affect the side-stick's input or feel.

A.5.3 Threat and error management in the A330

Moments before AF447's impact a synthetic voice announced, 'DUAL INPUT' five times, indicating that both side-sticks were displaced from neutral. Nose-up and

nose-down pitch commands from the pilots *cancelled* each other out with one pilot *pushing* the stick forward while the other *pulling* it back. Threat and Error Management (TEM) plays a vital mitigating role in accidents. Weather, being the threat, mixed with errors made by the crew led to the AF447 catastrophe. The crew flew into an area of heavy weather where the pitot tubes (although electrically heated along with the static ports, pitot tubes, angle of attack and total air temperature probes to prevent ice formation) became FOD by ice. When ice crystals began to hit the plane, it encountered an updraught. Cruise speed and engine thrust were reduced. On the flight-deck confusion reigned. The understandable fears were caused by the sudden failures in the instruments, rapidly deteriorating situation and events. The modern, automated instrumentation and displays had never failed them before. Would they be able to analyse and comprehend their dilemma, coordinate a plan of action, and revert to basic flying skills quickly enough to avert disaster?

A.5.4 Stall recovery in the A330

In Boeing designs, this is accomplished by pushing the throttles to their furthest forward position, calling 'MAX THROTTLE', and levelling the wings if in a turn. If an indication of an impending stall is encountered at cruising altitude, it may be necessary to lower the pitch attitude below the horizon to trade altitude for airspeed. Stick shakers will be activated before the actual stall. There is sufficient margin to recover from stick shaker without stalling. Per the Flight Crew Operating Manual (FCOM) in alternate and direct laws in the Airbus A330, when an aural stall warning 'STALL, STALL, STALL' is heard at low speeds, the pilot must return to the normal operating speed by taking conventional actions with the controls:

- THRUST LEVERS...TOGA switch on autothrottle
- PITCH ATTITUDE...REDUCE to 10° below FL200 or 5° at or above FL200
- BANK ANGLE...ROLL WINGS LEVEL
- SPEEDBRAKES...CHECK RETRACTED

It is also emphasised that pilots should not deliberately fly the aircraft in alpha protection except for brief periods when maximum manoeuvring is required. If pilots enter alpha protection inadvertently, they should get out of it as quickly as possible by easing forward on the side-stick to reduce the angle of attack while simultaneously adding power (if alpha floor has not already been activated or has been cancelled).

A.6 Northwest airlines flight 6231 (1 December 1974)

The Boeing 727-251's pitot heads became blocked at an altitude of about 16,000 ft. The formation of ice on the pitot heads should have been prevented by electrical heating elements which are activated by the pitot heater switches located in the cockpit. Investigation and the aircraft wreckage proved that the heating system was never activated. Even when at stall, pilots continued to react primarily to the high rate of descent indications by continuing to pull back on the control column. The NTSB concluded that the probable cause of this accident was the loss of aircraft control because the

flight crew failed to address the high-angle-of-attack, low-speed stall and its descending spiral correctly.

A.7 Birgenair flight 301 (February 6, 1996)

As the Boeing 757-225 was climbing, indicated airspeed began to increase. Acting appropriately based on signals sensed by the air data computer, the centre autopilot increased pitch, and the auto-throttles reduced power to counter the ever-increasing airspeed. Before long, the captain's erroneous airspeed indication generated an over-speed warning. Shortly thereafter the stick shaker also activated, and without proper flight control inputs, the aircraft stalled. Fourteen nautical miles northeast of Puerto Plata, Birgenair 301 slammed into the Atlantic Ocean, the aircraft disintegrated, instantly killing all 189 people aboard. As stated in the report, 'Investigators concluded that the probable source of obstruction in the pitot system was mud and/or debris from a small insect that was introduced into the pitot tube during the time the aircraft was on the ground in Puerto Plata.'

A.8 Aeroperú flight 603 (2 October 1996)

Flight 301's problems were not over, only disguised. Only 8 months after the fatal crash of Birgenair 301 and 2400 miles to the southwest, routine maintenance was being completed on another Boeing 757-200 in preparation for its scheduled international flight. Thirty-one minutes after takeoff, Aeroperu 603 smashed into the sea in a very steep left bank, killing all 70 people on board. At impact, the captain's instruments showed an altitude of 9500 feet and an airspeed of 450 knots. Blocked static ports – taped by the maintenance staff and overlooked by the flight crew in their walk-around – caused completely erroneous airspeed and altitude information to be displayed on the pilots' instruments.

Both flight 301 and 603 accidents involved Boeing 757, foreign crews, overwater operations at night, flight guidance system anomalies shortly after takeoff, and both proved fatal to all passengers and crew members on board.

*Serendipity should not be ignored – Confucius
...but neither should vicissitudes*

11.1 Introduction

11.1.1 Background

The corporate universe is littered with obsolete documentation that does little but take up space in a filing cabinet. This documentation took a lot of time and effort to generate and sometimes even longer to perfect and get buy-in from all the stakeholders, but once issued, it only takes few seconds to be forgotten...consigned to a filing cabinet for potential future retrieval that never happens...unless something goes wrong.

The System Safety Assessment (SSA) is such a document. Generated to support Type Certification, it can be hugely challenging to compile and, in the process, it often makes assumptions and predictions on how the system will perform in service. The SSA also feeds many and varied documents (see [Section 11.2.3](#)) for continued realisation of the design safety goal. Unfortunately many Safety Assessments are considered to end their useful life at the close of the Type Certification process. These are seldom to be reexamined or used to justify future modification and/or maintenance decisions and only brought back to light when faced with the scrutiny of litigation after the unfortunate accident happens.

SSA activities do not stop post certification. There needs to be a proper interchange of information between the aircraft manufacturer and operators, so that:

- Modes of failure and critical failure rates which occur in service can be checked against the predictions. If either a particular failure mode or its effect has not been correctly predicted, it is important that the aircraft constructor should know so that he can consider whether the implication is serious.
- Alterations to check and maintenance periods can be substantiated by the analysis.
- A sound Minimum Equipment List (MEL)¹ can be maintained and amended according to experience.

What is needed is smooth handover from the Initial Airworthiness Phase to the Continuing Airworthiness Phase, with the ability to accomplish some of the outcomes listed in [Section 11.1.3](#). Certifying staff always need to remember that airworthiness is the delivery of the technical aspects of a societal expectation for safety,² not the delivery of safety in itself.

¹ See Kritzinger (2006) Chapter 11 for more on the MEL and MMEL.

² For more information on the societal expectation, see Kritzinger (2006) para 2.1.

11.1.2 Aim

The aim of this chapter is to look beyond the “Initial Airworthiness Phase” (leading up to certification) and discuss the SSA interface with the Safety Management System (SMS) in the ‘Continuing Airworthiness Phase’.

11.1.3 Objectives

The objectives of this chapter are to explore the ability for the SSA to remain ‘live’ so that:

- the Continuing Airworthiness Maintenance Organisation (CAMO) can challenge some of the operating and maintenance assumptions made in the SSA and translated to the Instructions for Continued Airworthiness (ICA);
- the Design Organisation can efficiently correct any unforeseen failure conditions as well as crew error and maintenance error vulnerabilities;
- the CAMO (and their contracted Maintenance Organisations) can benefit from the failure diagnostic data [(e.g. Fault Tree Analysis (FTAs))] generated in the SSA;
- future additions/upgrades/modification of the system or any of its constituent parts can be efficiently accomplished within the framework of previous SSA deliverables.

11.1.4 Scope

The scope of this chapter is limited to the activities conducted by design organisations who delivered a CS/FAR2x.1309 compliant SSA.

The scope of this chapter will touch on, but not extend to, the through-life³ Safety Case⁴ or the SMS.⁵

11.2 The Safety Assessment’s handshake with the Safety Management System

During the Safety Assessment process, various assumptions might have been made (e.g. MTBF as well as how the equipment will be operated and maintained); hazardous failure conditions would have been identified (e.g. loss of Altitude Display might be “Extremely Improbable”, but is still possible); and procedures would be prescribed to mitigate risks for through life safety (i.e. how to deal with failure scenarios when they do occur).

³ Regarding through-life safety, SAE ARP5150 correctly states that: ‘To improve safety during the complete aeroplane life cycle, it is not sufficient to assess the safety of the aeroplane only during its design phase. Ongoing aeroplane operations must be evaluated for safety (e.g., maintenance or operation procedures). The aeroplane is also evolving and changing during the “In-Service” phase (e.g., obsolescence, modifications). Differences exist or can develop between the assumptions made during the design phase and how the aeroplanes are actually operated and maintained. For these reasons, safety should be assessed also during the “In-Service” phase of the aeroplane life cycle. To do that, information must be collected, monitored and analysed. A large portion of this needed information may already exist in an organisation’s maintenance or warranty information databases’.

⁴ For more on Safety Case, refer Chapter 9 in Kritzinger (2006).

⁵ For more on SMS, refer Chapter 12 in Kritzinger (2006).

Fig. 11.1 provides a simple illustration of the role the Design Organisation plays in through life safety of their type-certificated products. The following subsections will explore some of these interfaces and how the flow of hazard information finds its way into the ICA.

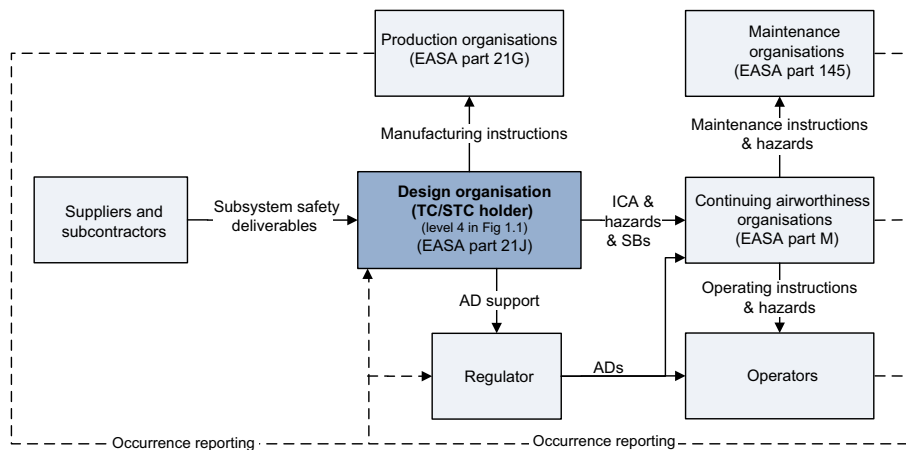


Figure 11.1 The design organisation's influence on the CAW phase.

11.2.1 Subsystem safety deliverables

In Fig. 1.3 we introduced the V&V Model of Systems Engineering. The right hand side of this model shows how we amalgamate and consolidate lower system level evidence to ultimately deliver a CS/FAR2x.1309 compliant SSA.

This concept of this process of amalgamating and consolidation is illustrated in Fig. 11.2, and an example of this process is demonstrated in the generation of the Failure Modes and Effects Summary discussed in Step 4 in Chapter 5. The ultimate aim is to summarise and consolidate the data, so that it can be presented in a format which is compatible with the SMS of those organisations (see Section 11.2.2) who are exposed to the hazards inherent in the design.⁶

11.2.2 Hazards

During the extensive work to generate the CS/FAR2x.1309 compliant equipment Safety Assessment, the Design Organisation would have built up a thorough understanding of many⁷ of the hazards which users (i.e. operators, manufacturers and maintainers) of the system might be exposed to. These users need to be provided with this information so that it can be managed within their individual SMSs (or, as the UK MoD refer to it, their through-life Safety Case.⁸

⁶ For more information, see para 6.2 and 6.3 in Kritzinger (2006).

⁷ The designers will identify many of the hazards, but not all – which provides insight into the importance of the Occurrence Reporting System (see Section 11.2.5).

⁸ For more on the Safety Case, see Kritzinger (2006) section 9.3.

This is a further extension of the process of amalgamating and consolidation as is illustrated in Fig. 11.2. For example, using the case study (refer Chapter 1):

- For the Operator: If we look at the Functional Hazard Analysis (Chapter 3), the Hazard Log entry might be ‘*Loss of Situational Awareness*’, of which the Altitude Display System is but one contributory cause.
- For the Maintainer: If we look at the Common Mode Analysis (CMA) (Chapter 6), the Hazard Log entry might be ‘*cross connection of safety critical systems*’, of which the ATT and ALT displays may be one of the vulnerabilities. The Design Organisation might also highlight OH&S-related hazards (e.g. working inside a fuel tank or with poisonous lubricants), but this would not be as a result of a 2x.1309 Safety Assessment (see the Preface to this book)
- For the Manufacturer: Hazards of concern to the manufacture’s SMS would primarily be focussed on OH&S-related hazards (e.g. working with hazardous materials⁹ or conducting electrical tests on newly manufactured equipment), but this would not be as a result of a 2x.1309 Safety Assessment (see the Preface to this book). However, the design might be vulnerable to errors in manufacturing process (see Table 6.1), and the manufacturer will need to be put in place control procedures with specific provisions for any critical parts [refer, inter alia, EASA Part 21.A.139(b)(1)].

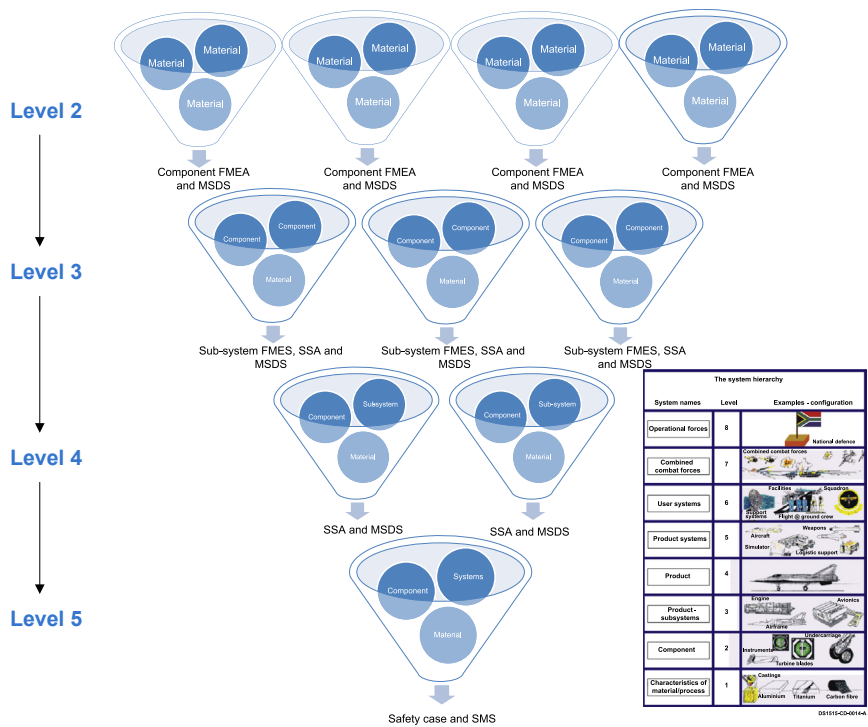


Figure 11.2 The amalgamating and consolidating of safety deliverables.

⁹ It is for this reason that suppliers should all be contracted to provide Material Safety Data Sheets (MSDS), which in Europe would need to be REACH compliant. REACH is a regulation of the European Union, adopted to improve the protection of human health and the environment from the risks that can be posed by chemicals, while enhancing the competitiveness of the EU chemicals industry. It also promotes alternative methods for the hazard assessment of substances to reduce the number of tests on animals.

11.2.3 Instructions for continued airworthiness

ICA are produced by design organisations as part of the product/part certification (e.g. see CS25 Appendix H). These ICA, if properly implemented by operators and maintainers, should ensure that the product/part remains airworthy during its intended life.

Example manuals/documents that can be considered to be part of the ICA are listed below, many of which will receive safety-related information from the Safety Assessment process:

- Airworthiness Directives (ADs) and Service Bulletins (SBs)
- Aircraft Flight Manual (AFM)
- Airworthiness Limitation Section (ALS), which may be a separate document (or part of another)
- Cabin Crew Operating Manual
- Flight Crew Operating Manual
- Configuration Maintenance and Procedures Document (linked to ETOPS operation)
- Certification Maintenance Requirements (CMR)
- Aircraft Maintenance Manual (AMM)
- Schedule Maintenance Program
- Component Maintenance Manuals (CMM) or Component Overhaul Manuals (COM)
- Illustrated Parts Catalogue (IPC)
- Wiring Diagram Manuals
- Weight and Balance Manuals
- Electrical Load Analysis
- Supplemental Structural Inspection Document (SSID)
- Nondestructive Testing Manual
- Key Safety Information (KSI).

Safety recommendations in the ICA may include:

- Flight crew procedures and checks. Account needs to be taken of warnings and indication given to the flight crew and the sequence of actions which need to be followed by them during normal operation as well as operation with failure conditions. In the case of some failure conditions, it will be assumed that the crew will take action within a given period depending on the urgency of the warning system or the behaviour of the aircraft (see [Chapter 10](#)). It is, therefore, important that the Flight/Operations/Maintenance Manuals are consistent with the assumptions and recommendations made in the Safety Assessment.

Pitot-static system example

[relating to [Chapter 10](#) (App A4) and the CMA in Table 6.3 (ID6.1)]

Erroneous airspeed and altitude indications caused by pitot and static system anomalies can confuse an unprepared flight crew. A crew's failure to respond correctly can result in an aeroplane accident or incident.

Good Flight Reference Cards are important for diagnostics and the following table might provide useful input:

Pitot-Static System Example—cont’d

Failure	Indicated airspeed	Indicated attitude
Pitot source blocked	Increased with alt gain. Decreases with alt loss.	Unaffected
Once static port blocked	Inaccurate during sideslipping. Very sensitive in turbulence.	
Both static sources blocked	Decreases with alt gain. Increases with alt loss.	Does not change with actual gain or loss of altitude
Both static and pitot sources blocked	All indications remain constant, regardless of actual changes in airspeed or altitude	

- Certification Maintenance Requirements¹⁰. The numerical calculations (e.g. see [Chapter 4](#)) performed to assess the probability of significant failure conditions may lead to the need for maintenance tasks to achieve the safety objectives. A CMR is a mandatory periodic task required for maintaining the safety of the aircraft. [AC25.19A](#) provides the following guidance regarding CMRs:
 - A CMR is a required periodic task established during the design certification of the aeroplane as an operating limitation of the type certification.
 - CMRs usually result from a formal, numerical analysis conducted to show compliance with Catastrophic or Hazardous failure conditions, and the following three considerations are appropriate:
 - CMRs are failure finding tasks (not preventative maintenance tasks) and exist solely to limit the exposure to otherwise hidden/dormant failures.
 - They need to be tracked individually, and accomplishment records have to be available for regulatory oversight.
 - The underlying goal of the system design is to minimise the CMRs, with none as the ideal situation¹¹.
- Reliability Monitoring: If the SSA relies heavily on unproven reliability claims (e.g. in the FTA), then the affected component’s reliability (i.e. MTBF) may need to be monitored in service and reevaluated at a defined point in time.
- Minimum Equipment required for dispatch of the aircraft. The regulatory authorities require that all equipment installed on an aeroplane in compliance with the Type Certification Basis must be operative. Experience has proven that, due to the various levels of redundancy designed into an aeroplane, operation of every system or installed component may not be necessary when the remaining operative equipment satisfies the airworthiness authorities (ARP5150, para 3.4.2). Therefore, certain conditional deviations from the original requirement are authorised to permit aeroplane dispatch. These ‘Dispatch Deviations’ identify:
 - what is required to operate the aeroplane in the various nonstandard configurations allowed by the type certification authority’s approved Master Minimum Equipment List (MMEL) and Configuration Deviation List (CDL), and

¹⁰ The FAA refers to this as Certification Check Requirement (CCR).

¹¹ Many operators regard CMRs as merely a means for the designer to address design deficiencies (such as hidden failures, high failure rates, insufficient redundancy, etc.), while the designers might argue that CMR will reduce cost/complexity/weight).

- by the Minimum Equipment List (MEL) approved by the operator's regulatory authority. The SSA used to show compliance with 25.1309, together with any other relevant information, should be naturally considered in the development of these lists. See Chapter 11 in Kritzinger (1996) for more information on the MEL.
- Training Requirements for those foreseeable failure scenarios which were mitigated in the SSA. It has to be remembered that although a failure event may be '*Extremely Improbable*', it does not mean that it cannot occur. Under controlled circumstances (such as during flight simulation training), failures should be induced to train the crew how to diagnose the failure and act appropriately for the unlikely event of when it does occur.

Pitot-static system example (continued)

[relating to [Chapter 10](#) (App A4) and the CMA in Table 6.3 (ID6.1)]

Diagnoses require knowledge of the aircraft systems. The crew can then determine which instruments are reliable and develop a strategy for recovery.

During pitot-static failures, typical reliable information includes:

- Pitch and roll indicators
- Engine thrust indication
- Radio Altitude
- Basic GPWS (EGPWS/terrain avoidance warning system may not)
- Stick Shaker
- Ground Speed (uses inertial information)
- Aeroplane position (uses inertial information).

During pitot-static failures, typical unreliable information includes:

- Autopilot
- Auto throttle
- Airspeed indication
- Altimeter
- Vertical speed
- Wind information
- Vertical navigation
- EPGWS
- Overspeed warning
- Windshear warnings.

Always remember that basic airmanship is vital:

- Rapid recognition is essential – the longer you deviate from intended flight path, the more difficult the recovery will be.
- Find or maintain favourable flying conditions – find visual references (e.g. daylight, climb to above cloudbase, call a 'chase plane').
- Train for unusual failure conditions so as to:
 - Recognise an unusual or suspect indication (i.e. cross-check instruments, maintain standard call-outs, etc.);
 - Maintain safe flight first (i.e. fly with the basic pitch and power setting), and only then diagnose (or do trouble shooting);
 - Err on the side of caution (e.g. shallower climb, excess speed, etc.);
 - Distinguish between reliable/unreliable information.

When providing the ICA, it is worthwhile reflecting on the following system safety design order of precedence (tailored from [MIL-STD-882E](#) para 4.3.4), which is listed in order of decreasing effectiveness:

- Eliminate hazards through design selection: Ideally, the hazard should be eliminated by selecting a design that removes the hazard altogether.
- Reduce risk through design alteration: If adopting an alternative design is not feasible, consider design changes that reduce the severity and/or the probability of the mishap potential caused by the hazard(s).
- Incorporate engineered features or devices: If mitigation of the risk through design alteration is not feasible, reduce the severity or the probability of the mishap potential caused by the hazard(s) using engineered features or devices. In general, engineered features actively interrupt the mishap sequence and devices reduce the risk of a mishap.
- Provide warning devices: If engineered features and devices are not feasible or do not adequately lower the severity or probability of the mishap potential caused by the hazard, then include detection and warning systems to alert personnel to the presence of a hazardous condition or occurrence of a hazardous event.
- Incorporate procedures in the ICA (including in the training curricula): Where design alternatives, design changes, and engineered features and devices are not feasible and warning devices cannot adequately mitigate the severity or probability of the mishap potential caused by the hazard, then incorporate procedures and training. Procedures¹² and training should include appropriate warnings and cautions.

11.2.4 *Manufacturing instructions*

Each Design Organisation needs to collaborate (refer, inter alia EASA 21.A.4 and 21.A.133) with Production (i.e. Manufacturing) Organisations to properly support the continued airworthiness of the product, part or appliance. This collaboration entails:

- The responsibilities of a Design Organisation, which include:
 - Providing correct and timely airworthiness data (e.g. drawings, material specifications, dimensional data, processes, surface treatments, shipping conditions, quality requirements, etc.);
 - Providing design approval/rejection for change requests (i.e. deviations, waivers and nonconcessions);
 - Instructions on how to handle, manufacture, install and test critical parts (i.e. those which can cause a Catastrophic or Hazardous failure condition).
- The responsibilities of the Production Organisation for:
 - Developing, where applicable, its own manufacturing data (in compliance with the airworthiness data package);
 - Not making any design change decisions without formal DO approval;

¹² Too often we find that hazards are mitigated by procedures only. Remember, if something can go wrong, then one day it will go wrong. Remember the 50-50-90 rule: Anytime you have a 50-50 chance of getting something right, there is a 90% probability you will get it wrong.

- Control procedures for critical parts. Human errors in production, which have the potential to lead to Catastrophic or Hazardous Failure Conditions, must be identified and mitigated;
- Conformance records (for future traceability of any issues of concern).

11.2.5 Occurrence reporting

Continuing airworthiness is the responsibility of the owner/operator. The most important element for the success of the continuing airworthiness function is a cooperative and efficient international system of data and information exchange between the owners, operators, maintainers, airworthiness authorities and equipment manufacturers. This is the purpose of the Occurrence Reporting System.

With regards to the equipment manufacturers, in service experience might prove that certain assumptions made during certification are longer be valid [GM21.A.3B(b)(1)], e.g.

- fatigue behaviour,
- modelling techniques used for Aircraft Flight Manual performances calculations,
- Systems Safety Analyses predictions (failure modes, effects and probabilities),
- the ability of the crew to apply procedures correctly,
- the ability to maintain the aircraft i.a.w. the ICA.

Each Design Organisation¹³ therefore has an obligation (e.g. see EASA 21.A.3A) to put in place a system for collecting, investigating and analysing reports of and information related to failures, malfunctions, defects or other occurrences which cause or might cause adverse effects on the Continuing Airworthiness of the product, part or appliance data.

Any occurrence which may result in an unsafe condition needs to be reported to the Regulator. For more information, see [EU Council Regulation 376/2014](#), EASA Part 21.A.3, [EASA Leaflet 8 in AMC20](#).

11.2.6 Service Bulletins and Airworthiness Directives

To correct any deficiencies found as a result of the Occurrence Reporting system, a Design Organisation might elect to issue a Service Bulletin (SB) to end users. From a legal perspective, the SBs are not mandatory in character (unless under AD cover letter).¹⁴

An AD is issued by the Regulator to end users. It mandates actions to be performed on an aircraft to restore an acceptable level of safety, when evidence shows that

¹³ Strictly speaking, this is applicable to each DO who is a holder of a type certificate, restricted type certificate, supplemental type certificate, European Technical Standard Order (ETSO) authorisation or major repair design approval [EASA 21.A.3.A].

¹⁴ For more information on the relationship between and AD and an SB, see CM-21.A-J-001.

the safety level of this aircraft may otherwise be compromised (i.e. this is an unsafe condition which is not an isolated event). It is mandatory, so an aircraft may lose its Certificate of Airworthiness if an AD is not implemented correctly or within the required timescale. On the back of the Occurrence Reporting System, the Design Organisation needs to provide the Regulator with all support needed for the issuance of the AD, which typically [EASA 21.A.3B(d)] contains the following information:

- an identification of the unsafe condition (see [Section 11.2.6.1](#) for more information);
- an identification of the affected aircraft;
- the action(s) required;
- the compliance time for the required action(s) (see [Section 11.2.6.1](#) for more information);
- the date of entry into force.

11.2.6.1 *The unsafe condition*

An ‘unsafe condition’ exists if there is factual evidence (from service experience, analysis or tests) that:

- a Hazardous event probability will exceed $1E-7$ (for CS25 aircraft), or
- a Catastrophic event probability will exceed $1E-9$ (for CS25 aircraft), or
- there is an unacceptable risk of serious/fatal injury to persons other than occupants, or
- design features intended to minimise the effects of survivable accidents are not, performing their intended function, or
- there is significant shortfall in flight performance/handling qualities [GM21.A.3B(b)(2.1.2.1)], or
- there is a deficiency in a principal structural element [see GM21.A.3B(b)(2.1.2.2)],
- there is one of the following deficiencies [GM21.A.3B(b)(2.1.2.2 to 2.1.2.4)]:
 - A deficiency effecting systems used during emergency evacuation, including, ELT and CVR or FDR.
 - A deficiency effecting systems involved in fire detection/protection or which are intended to minimise/retard the effects of fire/smoke.
 - A deficiency in the Lightning and HIRF protection of a system which could result in a Hazardous or Catastrophic failure.
 - A deficiency which could lead to a total loss of power or thrust due to common mode failure.

11.2.6.2 *The compliance time*

The compliance time for the required action(s) is linked to the safety criteria of 25.1309, and EASA GM 21.A.3B(d)(4) provides guidance on this ‘*rectification campaign*’. The following should help the assessor understand the logic of in this Guidance Material¹⁵:

¹⁵ While the main principles of this guidance below could be applied to small private aeroplanes, helicopters, etc., the numerical values chosen for illustration are appropriate to large aeroplanes for public transport.

- GM21.A.3B(d)(4) para 3.7: For large commuter transport aircraft, history¹⁶ has shown that the accident probability due to all technical causes is 1 in 10 million flying hours (i.e. $1\text{E-}7$ per flight hour).
 - If we assume that 75% (i.e. $7.5\text{E-}8$) is made of airworthy aircraft, and
 - the remaining 25% (i.e. $2.5\text{E-}8$) is for operating with a defect, and
 - history has shown that a total of 10 occasions might arise during the life of an individual aircraft when it needs to operate with a '*catastrophic failure condition*' defect. The allowable exposure time for 10 catastrophic defects is $2.5\text{E-}8$ and so the allowable exposure time per catastrophic defect is $2.5\text{E-}9$ per flight hour.
- GM21.A.3B(d)(4) (para 3.8 and 3.10): We can then create the first two columns in [Table 11.1](#) showing the flying time within which a defect should be corrected if we assume typical aircraft design life is 60,000 hours and a typical annual utilisation is 3000 hours.
- GM21.A.3B(d)(4) (para 3.11 and 3.12): To take into account large fleet size effect, the expected probability of a catastrophic event during the rectification period on the affected fleet is limited to 0.1. We can then add the last two columns in [Table 11.1](#) and illustrate it in [Fig. 11.3](#).
- GM21.A.3B(d)(4) (para 3.14 and 3.15): A similar approach is taken to cover the defects associated with '*Hazardous*' failure conditions, which has a probability of occurrence 2 orders of magnitude higher than '*Catastrophic*', i.e. $2.5\text{E-}7$. We can then create [Table 11.2](#) and [Fig. 11.4](#) if we:
 - assume a typical aircraft design life is 60,000 hours and a typical annual utilisation is 3000 hours,
 - and, to take into account large fleet size effect, the expected probability of hazardous event during the rectification period on the affected fleet shall not exceed 0.5.

¹⁶ See AMC25.1309 to CS25: '*Historical evidence indicated that the probability of a serious accident due to operational and airframe-related causes was approximately one per million hours of flight. Furthermore, about 10% of the total were attributed to Failure Conditions caused by the aeroplane's systems. It seems reasonable that serious accidents caused by systems should not be allowed a higher probability than this in new aeroplane designs. It is reasonable to expect that the probability of a serious accident from all such Failure Conditions be not greater than one per 10 million flight hours or 1×10^{-7} per flight hour for a newly designed aeroplane*'. This leads to an accident probability of 0.0000001 (10^{-7}) per hour for technical cause factors. Therefore, for transport category aircraft, most civil airworthiness authorities require that aircraft systems and associated components (considered separately and in relation to other systems) be designed in a manner such that the occurrence of any failure condition which would prevent the continued safe flight and landing of the aircraft should virtually never occur in the life of an aircraft type. The difficulty with this is that it is not possible to say whether the target has been met until all the systems on the aeroplane are collectively analysed numerically. A typical transport category aircraft type design has many individual systems that may influence the safe flight and landing of an aircraft. Without a full system safety analysis, it is difficult, if not impossible, to consider the contribution of each individual system to the overall accident rate. For most aircraft types, it is thus assumed that as many as 100 individual system failure conditions may exist which could prevent continued safe flight and Landing. The target allowable probability of 1×10^{-7} is thus apportioned equally among these Conditions, resulting in a probability allocation of not greater than 1×10^{-9} per flight hour to each. This upper probability limit establishes an approximate probability value for the term '*Extremely Improbable*'. Failure conditions having less severe effects could be relatively more likely to occur based on the principle that an inverse relationship should exist between the probability of an occurrence and the degree of hazard inherent in its effect as summarised in Table 3.3.

Table 11.1 CS25 Catastrophic failure condition table

Safety-related malfunction (per flight hour)	Average reaction time (hours)	Probability of accident	Affected fleet exposure
1E-9 (lower boundary set by 25.1309)	150,000	1.5E-4	1E+8
4E-8	3750	1.5E-4	2.5E+6
1E-7	1500	1.5E-4	1E+6
5E-7	300	1.5E-4	2E+5
1E-6	1500	1.5E-4	1E+5
2E-6 (upper boundary, set by para 3.11 and 3.12)	75	1.5E-4	5E+4
	= 60000 × 2.5E-9 ÷ Column 1	= Column 1 × Column 3	= 0.1 × Column 2 ÷ Column 3

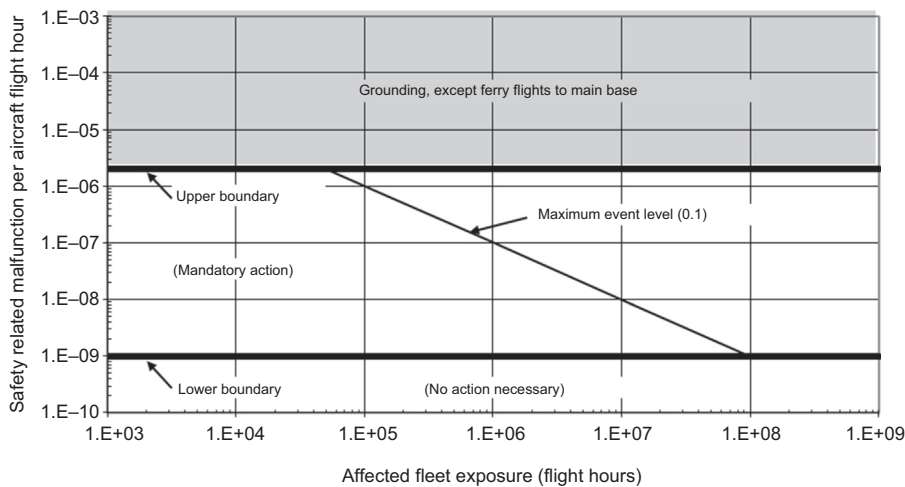


Figure 11.3 CS25 catastrophic failure condition visualisation chart.

- GM21.A.3B(d)(4) (para 4.1): The AD action(s) required to remedy “catastrophic” failure conditions without grounding aircraft:
 - Establish all possible alleviating actions (such as inspections, crew drills, route restrictions and other limitations).
 - Identify that part of the fleet, which is exposed to the residual risk (i.e. after compliance has been established with the above).
 - Compare the speed with which any suggested campaign (i.e. all required alleviating actions) will correct the deficiency with the time suggested in Table 11.1 (or the corresponding Fig. 11.3). Do not exceed 2E-6, except for specially authorised flights.

Similarly, these guidelines would be applicable for a rectification campaign to remedy a discovered defect associated to a *hazardous* failure condition without grounding the aircraft.

Table 11.2 CS25 hazardous failure condition table

Safety-related malfunction (per flight hour)	Average reaction time (h)	Probability of accident	Affected fleet exposure
1E-7	150,000	1.5E-2	5E+6
1E-6	15,000	1.5E-2	5E+5
1E-5	1500	1.5E-2	5E+4
5E-4	150	1.5E-2	5E+3
2E-4	75	1.5E-2	2.5E+3
	= 60000 × 2.5E-9 ÷ Column 1	= Column 1 × Column 3	= 0.5 × Column 2 ÷ Column 3

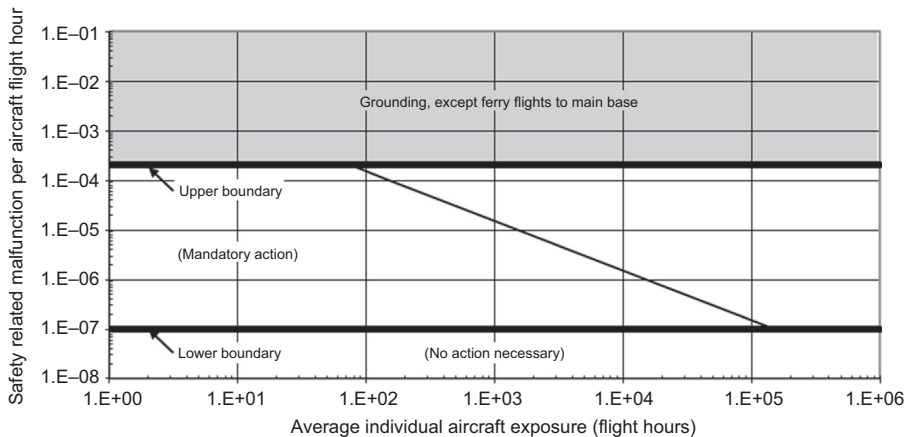


Figure 11.4 CS25 hazardous failure condition visualisation chart.

11.3 Discussion

Design Organisations are at the heart of safety because the design solution determines the manner in which the system is operated and maintained. There are many design deficiencies which require constant management and reevaluation in the Continuing Airworthiness, for instance:

- Most Human Factors interventions in a Maintenance Organisation can be traced back to the fact that the system was not designed to minimise the possibility of Human Error during maintenance (refer Table 6.1).
- Many errors made by flight crew could be better mitigated if they were more effectively assessed in the IAW phase (refer [Section 10.5](#)).

- Systems fail more frequently than expected and a mechanisms needs to put in place to challenge the reliability assumptions (e.g. MTBF¹⁷) on which failure probabilities in the SSA are based to prove compliance to requirements such as CS25.1309(b).
- Systems fail in ways not predicted by the designer's SSA. The causes of these failures need to be assessed and mitigated by a revision in the SSA and/or SB/AD action.

Ultimately, no matter how good our SSA work up-front is, there will always be unforeseen hazards that slip through the net. The overall aim of our design processes and in-service SMS are to make sure they are of low severity and/or probability and to have a resilient and robust means of spotting and correcting them before they become significant. To err is human, to manage it is smart.

11.4 Conclusion

The data and design familiarisation obtained during the certification process is critical to the success of the continuing airworthiness function. It enables the continual reassessment to be made, action to be taken to ensure the continued fitness to fly of the aircraft, and feedback through to the original maintenance and design standards.

In [Section 11.3](#) we stated that *'Design Organisations are at the heart of safety because the design solution determines the manner in which the system is operated and maintained'*. A useful technique which can be used to illustrate this principle (and drive the required Design Organisations behaviours in the 'handshake' with the user's SMS) is the Bow Tie Analyses (see [Fig. 11.5](#)), where:

- The SSA uses a top-down approach to identify functional failure conditions of concern, the probability of which can be determined by techniques such as the FTA.
- The residual¹⁸ functional failures are consolidated into a list of Hazards applicable to the user (refer [Section 11.2.2](#) above). This is the safety 'start point' for the user's SMS.
- The designer provides (via ICA, see [Section 11.2.3](#)) instruction and training requirements to help the user manage the consequences (i.e. probability and severity) of an uncontained hazard. A useful technique in this process is the Event Tree Analysis (see Kritzinger (2006) Annex A).

However, with reference to [Chapter 10](#) and Table 6.1, human performance is still the dominant factor in many accidents. This may be attributed to:

- Regulations (and therefore the resulting design techniques and Safety Assessments), which do not adequately address the subject of human error in design or in operations and maintenance.

¹⁷ Actual MTBF experienced will probably differ from predicted MTBFs used in FTA calculations during initial certification. Direct advantages are possible if the FTA can be updated with actual MTBF data: either action can be taken to improve safety or (if the predicted MTBFs were very conservative) maintenance intervals can be increased.

¹⁸ Residual means those not designed out of the system and which are possible to occur in service.

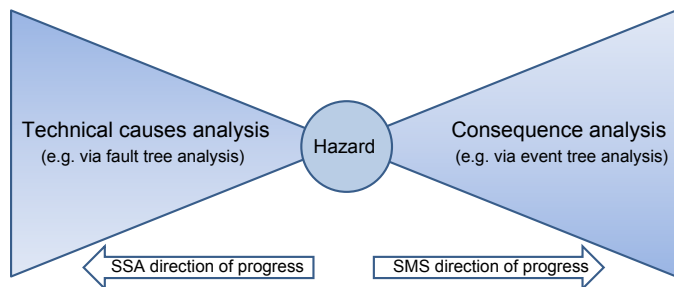


Figure 11.5 Bow tie analyses.

- The immature and inconsistent processes used to determine and validate human responses to failure identified in the Safety Assessment (see Step 3c in [Chapter 10](#)). There require extra vigilance in certification, entry to service and should be supported by in-service occurrence reporting [EASA Part 21.A.3A].
- There is no reliable process to ensure that assumptions made in the Safety Assessments are valid with respect to operations and maintenance activities and that operators are aware of these assumptions when developing their operations or maintenance procedures. It will always be necessary to make assumptions in Safety Analyses; however, where possible, those assumptions may need to be validated by actual experience. There is currently no organised program to periodically revisit design safety assumptions to ensure that they reflect the full range of environments and operations as the fleet ages.

References

- AC25.19A, March 2011. Certification Maintenance Requirements. US DOD, FAA, Washington.
- CM-21.A-J-001, June 2013. Service bulletins (SBs) related to airworthiness directives (ADs). EASA Certification Memorandum (1).
- EASA Part 21. Certification of aircraft and related products, parts and appliances, and of design and production organisations. In: Annex 1 of the Commission Regulation 748/2012. The European Commission.
- EASA AMC 20, November 2003. General Acceptable Means of Compliance for Airworthiness of Products, Parts and Appliances. EASA Decision No 2003/12/RM.
- MIL-STD-882E, February 2000. System Safety. US Department of Defense (DOD), Standard Practice.
- Regulation (EU) No 376/2014, April 2014. Reporting, Analysis and Follow-up of Occurrences in Civil Aviation. The European Parliament and of the Council.

Further reading

- SCF/SYS/A/108/4523, 2001. Human Hazard Analysis: A Demonstrator for a Means of Compliance, a Technical Report by the Systems Centre of Competence. Airbus UK Ltd., Filton.

Abbreviations

Abbreviation	Meaning (+ reference source if applicable)
ADC	Air Data Computer
ADI	Attitude Director Indicator
AD	Airworthiness Directive
AIAG	Automotive Industry Action Group
ALS	Airworthiness Limitation Section
APQP	Advanced Product Quality Planning
APU	Auxiliary Power Unit
ARP	Aircraft Recommended Practices
ASA	Aircraft Safety Assessment
ASIC	Application Specific Integrated Circuit
BIT	Built-in Testing
CAMO	Continuing Airworthiness Maintenance Organisation
CBIT	Continuous Built-in Testing
CCF	Common Cause Failures
CCR	Certification Check Requirement
CDR	Critical Design Review
CEH	Complex Electronic Hardware
CFIT	Controlled Flight into Terrain
CMA	Common Mode Analysis
CMR	Certification Maintenance Requirements
CMR	A Certification Maintenance Requirement is a required periodic maintenance task established through formal numerical analysis to show compliance with certification requirements (ARP 5150)
CofC	Certificate of Conformance
COTS	Consumed off the shelf
CS	Certification Specification
DAL	Development Assurance Level
DCD	Display Control Panel
DDP	Declaration of Design and Performance
DD	Dependence Diagram
DOA	Design Organisation Approval
DOD	Department of Defence
EASA	European Aviation Safety Agency

Abbreviation	Meaning (+ reference source if applicable)
EGPWS	Enhanced Ground Proximity Warning System
ETSO	European Technical Standard Order
EWIS	Electrical Wiring Integration System
FAA	Federal Aviation Authority
FAR	Federal Aviation Regulation
FFPA	Functional Failure Path Analysis
FHA	Functional Hazard Analysis
FL	Flight Level
FMEA	Failure Modes and Effects Analysis
FMECA	Failure Modes, Effects and Criticality Analysis
FPGA	Field Programmable Gate Arrays
FTA	Fault Tree Analysis
FTRR	Flight Test Readiness Review
GPWS	Ground Proximity Warning System
GSN	Goal Structured Notation
H/W	Hardware
HF	Human Factors
HFA	Human Factors Analysis
i.t.o.	In terms of
iASA	Interim Aircraft Safety Assessment
ICA	Instructions for Continuing Airworthiness
IFR	Instrument Flight Rules
INS	Inertial Navigation System
IRU	Inertial Reference Unit
iSSA	Interim System Safety Assessment
LOI	Level of Involvement
LOSA	Loss of Situational Awareness
LRU	Line Replaceable Unit
MA	Markov Analysis
MEL	Minimum Equipment List
MMEL	Master Minimum Equipment List
MSDS	Material Safety Data Sheets
MTBF	Mean Time between Failures
OS&S	Occupational Health and Safety
PASA	Preliminary Aircraft Safety Assessment
PBIT	Power-on Built-in Testing
PDR	Preliminary Design Review
PFD	Primary Flight Display
PLD	Programmable Logic Device

Abbreviation	Meaning (+ reference source if applicable)
PRA	Particular Risk Analysis
PSSA	Preliminary System Safety Assessment
RAT	Ram Air Turbine
REACH	Registration, Evaluation, Authorisation and Restriction of Chemicals
S/W	Software
SAE	Society of Automotive Engineers
SB	Service Bulletin
SRU	Shop Replaceable Unit
SSA	System Safety Assessment
SSWG	System Safety Working Group
STC	Supplemental Type Certificate
TC	Type Certificate
TCAS	Traffic Collision Avoidance System
TSO	Technical Standard Orders
VFR	Visual Flight Rules
ZSA	Zonal Safety Analysis
TOGA	Take-off, Go around

Definitions

Term	Definitions (+ reference source if applicable)
Airborne Electronic Hardware	Electronic airborne hardware includes line replaceable units, circuit board assemblies, application-specific integrated circuits, programmable logic devices, etc. [RTCA/DO-254].
Airworthiness	The condition of an aircraft, aircraft system, or component in which it operates in a safe manner to accomplish its intended function [ARP4754A].
Airworthiness Directive	A document issued or adopted by the Agency which mandates actions to be performed on an aircraft to restore an acceptable level of safety, when evidence shows that the safety level of this aircraft may otherwise be compromised. [Regulation (EC) 748/2012 21A.3B Airworthiness directives and Regulation (EC) 1702/2003 21A.3B Airworthiness directives].
Analysis	An evaluation based on decomposition into simpler elements [ARP4754A].
Application-Specific Integrated Circuits	ASSC (Jan 2009) advises that, in practice, the term Programmable Logic Device (PLD) is occasionally used with the intention of including ASIC devices. The main difference between a PLD and an ASIC is that a PLD is manufactured as a standard device which is then ‘programmed’ or configured for a specific application. A PLD may be programmed by placing the chip in a special ‘Device programmer’ or, increasingly, programmed without removal from the circuit board which is called ISP (In System Programming). An ASIC cannot be reprogrammed.
Application-Specific Integrated Circuit	Integrated Circuits which are developed to implement a function, including, but not limited to: gate arrays, standard cells and full custom devices encompassing linear, digital and mixed mode technologies. ASICs are mask-programmable components [EASA CM-SWCEH-001].
Assembly	A composite of subassemblies.
Assessment	An evaluation based upon engineering judgement [ARP4754A].
Assessment	An evaluation based on engineering judgement [ARP4754A].
Assumptions	Statements, principles and/or promises offered without proof [ARP4754A].
Assurance	The planned and systematic actions necessary to provide adequate confidence and evidence that a product or process satisfies given requirements [RTCA/DO-178B/ED-12B].

Term	Definitions (+ reference source if applicable)
Authority	The organisation or person responsible within the State/Country concerned with the certification of compliance with applicable requirements [RTCA/DO-178B/ED-12B].
Classification (failure condition)	A discreet scale allowing categorisation of the severity of the effects of a failure condition [RTCA/DO-178B/ED-12B].
Combustible	Refers to the ability of any solid, liquid or gaseous material to cause a fire to be sustained after removal of the ignition source.
Common Cause Analysis	Generic term encompassing zonal safety analysis, particular risk analysis and common mode analysis [RTCA/DO-178B/ED-12B].
Common Mode Error	An error which affects a number of elements otherwise considered to be independent [ARP4754A].
Common Mode Error	An error which affects a number of elements otherwise considered to be independent.
Complex Electronic Hardware	<p>All devices that are not simple are considered to be complex. See the definition of Simple Hardware [EASA CM-SWCEH-001].</p> <p>Complex Electronic Hardware: Includes custom micro-coded components (such as ASICs, PLDs and associated macro functions) and integrated technology components (such as hybrids and multichip modules).</p> <p>RTCA-DO-254 (para 1.6) advises that ‘hardware should be examined hierarchically at the levels of integrated circuit, board and LRU for complexity, including addressing functions that may not be testable, such as unused modes in multiple usage devices and potentially hidden states in sequential machines’. An item constructed entirely from simple items may itself be complex. Complexity is an attribute of systems or items which makes their operation difficult to comprehend. Increased system complexity is often caused by such items as sophisticated components and multiple interrelationships.</p>
Complex hardware item	<p>All items that are not simple are considered to be complex.</p> <p>See definition of simple hardware item. Source: RTCA/DO-254, Appendix C and Order 8110.105.</p>
Complex system	<p>A system is ‘complex’ when its operation, failure modes or failure effects are difficult to comprehend without the aid of analytical methods or structured assessment methods. FMEA and FTA are examples of such structured assessment methods. Increased system complexity is often caused by such items as sophisticated components and multiple interrelationships. For example, for these types of systems, a portion of the compliance may be shown by the use of DALs such as by processes in RTCA/DO-178B or RTCA/DO-254 or equivalent. See the definitions for ‘conventional’ and ‘simple’ for more information.</p>

Term	Definitions (+ reference source if applicable)
Complexity	<p>An attribute of functions, systems or items, which makes their operation, failure modes or failure effects difficult to comprehend without the aid of analytical methods [ARP4754A].</p> <p>An attribute of systems or items which makes their operation difficult to comprehend.</p> <p>Increased system complexity is often caused by such items as sophisticated components and multiple interrelationships [ARP 5150].</p>
Component	Any self-contained part, combination of parts, subassemblies or units that perform a distinctive function necessary to the operation of the system [RTCA/DO-178B/ED-12B].
Component	Any self-contained part, combination of parts, subassemblies or units that perform a distinctive function necessary to the operation of the system [ARP5754A p11].
Confirmed Failure	The detected inability of any piece of equipment to perform in accordance with its specification.
Continued Airworthiness	Means all tasks to be carried out to verify that the conditions under which a type certificate or a supplemental type certificate has been granted continue to be fulfilled at any time during its period of validity [EMAR 21]. For example tasks, see EASA Part 21: 21.A.3, 21.A.57, 21.A.119, 21.A.120, 21.A.61, and 21.A.265.
Continuing Airworthiness	All of the processes ensuring that, at any time in its operating life, the aircraft complies with the airworthiness requirements in force and is in a condition for safe operation [Regulation (EU) 1321/2014].
Conventional	An attribute of a system is considered to be conventional if it is the same as, or closely similar to, that of previously approved systems that are commonly used [AMJ25.1309 para 6.f].
Conventional system	A system is considered ‘conventional’ if its function, the technological means to implement its function, and its intended usage are all the same as, or closely similar to, that of previously approved systems that are commonly used. The systems that have established an adequate service history and the means of compliance for approval are generally accepted as ‘conventional’. Normally conventional and simple systems may be analysed by qualitative assessments as shown in Figure 3. See the definitions for complex and simple systems for more information.
Critical Part	The term ‘critical part’ is used in various EASA requirements, certification specifications (CS) and also in the draft EU–US bilateral; however, it is not always defined.

Term	Definitions (+ reference source if applicable)
	<p>A widely accepted definition does not seem to exist. There are currently basically three different EASA definitions:</p> <ul style="list-style-type: none"> • For rotorcraft [CS 27-29-VLR.602(a)]: A critical part is a part, the failure of which could have a catastrophic effect upon the rotorcraft, and for which critical characteristics have been identified which must be controlled to ensure the required level of integrity. • For engines, propellers and APUs [CS-E.510(c)]: It is recognised that the probability of Primary Failures of certain single elements cannot be sensibly estimated in numerical terms. If the Failure of such elements is likely to result in Hazardous Engine Effects, reliance must be placed on meeting the prescribed integrity specifications of CS-E 515 (engine critical parts) to support the objective of an Extremely Remote probability of Failure [similar for CS-P.150(c) and CS-P.160 and also for CS-APU.210(c) and CS-APU.150]. • In the draft EU–US bilateral: A ‘critical component’ is a part identified as critical by the design approval holder during the validation process, or otherwise by the exporting authority. Typically, such components include parts for which a replacement time, inspection interval, or related procedure is specified in the Airworthiness Limitations section or certification maintenance requirements of the manufacturer’s maintenance manual or Instructions for Continued Airworthiness. <p>Each of the above definitions should be used only within their own context and for their own purpose, i.e. the definition of the bilateral is only relevant for the automatic acceptance of PMA parts and repair design from the United States. Where the term ‘critical part’ is not defined the dictionary meaning of ‘critical’ should be used, i.e. crucial, decisive, important, etc.</p> <p>For the application of Part 21A.805, critical parts are those identified as such by the design approval holder, which for rotorcraft, engines, propellers and APUs as a minimum should be those using the definitions of the relevant CS (http://easa.europa.eu/the-agency/faqs/initial-airworthiness).</p>
Cut set	<p>A Fault Tree may have a number of cut sets, each of which being a set of basic events that together cause the top (undesired) event to occur. The minimum cut set is the one with the least amount of events that can cause the top event to occur. The critical path is the highest probability cut set (i.e. most probable cause of the top event) [Ericson (2005), Chapter 11].</p>
Defect	<p>State of an item consisting of the nonperformance of specified requirements by a characteristic of the item. A defect may, but need not, lead to a failure [ARP 5150].</p>
Derived Requirements	<p>Additional requirements resulting from the design or implementation decision during the development process which is not directly related to higher level requirement [RTCA/DO-178B/ED-12B].</p>

Term	Definitions (+ reference source if applicable)
Development Assurance	All those planned and systematic actions used to substantiate, at an adequate level of confidence, that errors in requirements, design and implementation have been identified and corrected such that the system satisfies the applicable certification basis [AMC 25.1309].
Development Error	A mistake in requirements determination, design or implementation [ARP4754A].
Electrical Wire Interconnection System (EWIS)	Is defined in CS-25 Book 1 Subpart H (CS-25.1701). As defined by that section EWIS refers to any wire, wiring device, or combination of these, including termination devices, installed in any area of the aircraft for the purpose of transmitting electrical energy, including data and signals between two or more intended termination points. For complete definition, see AMC 25.1701.
Enhanced Zonal Analysis Procedure (EZAP)	Refers to the logical process of developing maintenance and inspection instructions for an Electrical Wiring Interconnection System (EWIS). Refer to 03-02-I3007-02 for further definition.
Ergonomics	Ergonomics is the scientific discipline concerned with the understanding of interactions among human and other elements of the system, and the profession that applies theory, principles, data and methods to design to optimise human well-being and overall system performance (International Ergonomics Association).
Error	An omission or incorrect action by a crew member or maintenance personnel, or a mistake in requirements, design or implementation [AC23.1309E]. An occurrence arising as a result of an incorrect action or decision by personnel operating or maintaining a system [ARP5150]. A mistake in specification, design or implementation [ARP5150].
Event	An occurrence that is of interest for aeroplane safety [ARP5150].
External Event	An occurrence which has its origin distinct from the aeroplane, such as atmospheric conditions (e.g. wind gusts, temperature variations, icing, lightning strikes), runway conditions, cabin and baggage fires. The term is not intended to cover sabotage [ARP 5150].
Failure	An occurrence that affects the operation of a component, part or element such that it can no longer function as intended (this includes both loss of function and malfunction) [AC23.1309E]. Note: Errors may cause failures but are not considered failures.
Failure	An occurrence which affects the operation of a component, part or element such that it can no longer function as intended (this includes both loss of function and malfunction). Note: Errors may cause failures, but are not considered to be failures [AMC 25.1309].

Term	Definitions (+ reference source if applicable)
Failure condition	A condition with an effect on the aeroplane and its occupants, both direct and consequential, caused or contributed to by one or more failures, considering relevant adverse operation or environmental conditions. A Failure Condition is classified in accordance to the severity of its effects as defined in regulatory advisory circulars [ARP 5150].
Failure conditions	A condition having an effect on either the aeroplane or its occupants, or both, either direct or consequential, which is caused or contributed to by one or more failure or errors considering flight phase and relevant adverse operational or environmental conditions or external events [AC23.1309E].
Failure mode	A condition having an effect on the aircraft and its occupants, either direct or consequential, which is caused or contributed to by one ore more failures or error, considering flight phases and relevant adverse operating or environment conditions or external events [AMC25.1309].
Failure Rate	The gradient of the failure distribution function divided by the reliability distribution function at time t. If the failure distribution function is exponential, the failure rate is constant and the failure rate can be approximately calculated by dividing the number of failures within a hardware item population, by the total unit operating hours. Note: Failure rate could also be expressed in terms of failures per flight hour or per cycle [ARP 5150].
Fault	An abnormal undesirable state of a system or a system element* induced by a failure or by the presence of an improper command (or absence of a proper one). Note: All failures cause faults; not all faults are caused by failures. A system which has been shut down by safety features has not faulted [Clemens].
Field Programmable Gate Arrays	ASSC (Jan 2009) advises that there are 3 main families of FPGAs depending on how they are made: SRAM, Flash and antifuse. SRAM (Static RAM)-based FPGAs have been dominant, benefiting from the advances of CMOS technology to keep ahead in terms of speed and density. However, SRAM is volatile unlike Flash and Antifuse. Flash-based FPGAs are nonvolatile and reprogrammable but lack the speed and density of SRAM FPGAs by 1 or 2 years. Antifuse FPGAs are not reprogrammable but are nonvolatile and show greatest resistance to Single Event Upsets.
Functional Hazard Analysis	A Functional Hazard Analysis (FHA) is defined [SAE ARP4761 para 3.2] as a systematic, comprehensive examination of a system's functions to identify and classify potential Failure Conditions which the system can cause or contribute to, not only if it malfunctions or fails to function, but also in its normal response to unusual or abnormal external factors.

Term	Definitions (+ reference source if applicable)
Implementation	The act of creating a physical reality from a specification [ARP4754A].
Independence	A concept that minimises the likelihood of common mode errors and cascade failures between aircraft/system functions or items [ARP4754A].
	Separation of responsibilities that assured the accomplishment of objectives evaluation, e.g. validation activities not performed solely by the developer of the requirement of a system or item [ARP4754A].
Installation appraisal	A qualitative appraisal of the integrity and safety of the installation. Any deviations from normal industry-accepted installation practices should be evaluated [23.1309E].
Instructions for Continued Airworthiness (ICA)	Refers to the information documented in accordance with CS-25 Book 2, AMC – Appendices, AMC to Appendix H, H25.4(a)(3) and H25.5.
Interface	The interaction point(s) necessary to produce the desired/essential effects between system elements (interfaces transfer energy/information, maintain mechanical integrity, etc.).
Item	A Hardware of Software element having bounded and well defined interfaces [ARP5754A p12].
Level of Involvement	A selection of the compliance demonstration items that the Regulatory Authority will investigate and the depth of those investigations.
Line Replaceable Unit	A line replaceable unit (LRU), lower line replaceable unit (LLRU), line replaceable component (LRC) or line replaceable item (LRI) is a modular component that is designed to be replaced quickly at an operating location. An LRU is usually a sealed unit such as a radio or other auxiliary equipment. LRUs improve maintenance operations, because they can be stocked and replaced quickly from on-site inventory, restoring the system to service, while the failed (unserviceable) LRU is undergoing maintenance. Because they are modular, they also reduce system costs and increase quality, by centralising development across different platforms.
Malfunction	The occurrence of a condition whereby the operation of an item is outside of specified limits.
Master Minimum Equipment List	A list of equipment and functions which need not be operative for safe flight and landing based on stated compensating precautions created during aeroplane type certification [ARP5150].
Mean Time between Failure	A performance figure calculated by dividing the total unit flying hours (airborne) accrued in a period by the number of unit failures that occurred during the same period [ARP5150].
Minimum Equipment List	An approved list of items which may be inoperative for flight under specified conditions cooperatively created by operators and regulatory authorities [ARP5150].

Term	Definitions (+ reference source if applicable)
Partitioning	The mechanism used to separate portions of a system or item with sufficient independence such that a specific development assurance level can be substantiated within the partitioned item [ARP4754A].
Partitioning	<p>Partitioning is a means for providing isolation between components to contain and/or isolate faults. Partitioning between components may be achieved within the system architecturally by allocating unique target hardware and hardware resources to each component. Alternatively, partitioning may be achieved within the component architecture to allow multiple software or hardware items to run within the same hardware platform. Partitioning should ensure:</p> <ol style="list-style-type: none"> 1. A partitioned component should not be allowed to contaminate (spatially) another partitioned component's implementation code, input/output (I/O) or data storage areas. 2. A partitioned component should be allowed to consume shared resources (temporally) only during its scheduled period of execution. 3. Failures of software or hardware unique to a partitioned component should not cause adverse effects on other partitioned components. 4. Any software or hardware providing the underlying mechanisms of partitioning should have the same or higher DAL as the highest level assigned to any of the partitioned components. 5. Any software or hardware providing partitioning should be assessed by the system safety assessment process to ensure that it does not adversely affect safety.
Partitioning (Software)	Partitioning is a technique for providing isolation between functionally independent software components to contain and/or isolate faults and potentially reduce the effort of the software verification process. If protection by partitioning is provided, the software level for each partitioned component may be determined using the most severe failure condition category associated with that component [ABD0100.1.3 Issue E para 2.1.5].
Piece Part	Least fabricated item, not further reducible.
Programmable Logic Device	A component that is purchased as an electronic component and altered to perform an application-specific function. PLDs include, but are not limited to: Programmable Array Logic components (PAL), General Array Logic components, Field Programmable Gate Array (FPGA) components, and Erasable Programmable Logic Devices (EPLDs) [EASA CM-SWCEH-001].
Qualitative	Those analytical processes that assess system and aeroplane safety in an objective nonnumerical manner [AC23.1309E].
Qualitative	Those analytical processes that assess aeroplane safety in a subjective and nonnumerical manner [ARP5150].

Term	Definitions (+ reference source if applicable)
Quantitative	Those analytical processes that apply mathematical methods to assess the system and aeroplane safety [AC23.1309E].
Quantitative	Those analytical processes that apply statistical and data-based methods to assess aeroplane safety [ARP5150].
Redundancy	The presence of more than one independent means for accomplishing a given function. Each means of accomplishing the function need not be identical [AC23.1309E].
Redundancy	Multiple independent means incorporated to accomplish a given function [ARP5150].
Risks	The frequency (probability) of occurrence and the associated level of hazard. Perceived risk: As seen intuitively by individuals.
	Predicted risk: As predicted analytically from models structured from historical studies [ARP5150].
Shop Replaceable Unit	<p>A shop replaceable unit (SRU) or shop replaceable component (SRC) is a modular component of an aeroplane, ship or spacecraft that is designed to be replaced by a technician at a backshop. Repair at backshops is known as field-level maintenance or intermediate-level (I-level) maintenance.</p> <p>SRUs are similar in nature to line replaceable units (LRUs), but rather than being complete functional units, represent component functions, such as circuit card assemblies, of a larger LRU. SRUs are typically assigned logistics control numbers (LCNs) or work unit codes (WUCs) to manage logistics operations.</p> <p>SRUs can be stocked to allow for quick remove and replace (R&R) operations on their parent LRUs or LLRUs, while also allowing for more extended repair operations at the backshop.</p>
Simple Electronic Hardware	<p>A hardware device is considered simple only if a comprehensive combination of deterministic tests and analyses appropriate to the DAL/IDAL can ensure correct functional performance under all foreseeable operating conditions with no anomalous behaviour [EASA CM-SWCEH-001].</p> <ul style="list-style-type: none"> • Comment 1: For the purposes of this Guideline, this definition can be applied to airborne electronic hardware devices whose simplicity has been confirmed by a documented engineering analysis of the logic and the design. This analysis should be based on criteria denoting a measure of simplicity such as, for example, the number of states in the state machine, and hysteresis characteristics. • Comment 2: For the purposes of this Guideline, this definition can be applied to airborne electronic hardware devices whose logic is simple enough to comprehend without the aid of analytical tools. Some examples of Simple COTS could be: UART, A/D converters, D/A converters, PWM.

Term	Definitions (+ reference source if applicable)
Software	<p>Computer programs, procedures, rules and any associated documentation pertaining to the operation of a computer system [ARP5150].</p> <p>IEEE610.12 defines software as ‘computer programs, procedures, and possibly associated documentation and data pertaining to the operation of a computer system’, where a computer program is defined as ‘a combination of computer instructions and data definitions that enable computer hardware to perform computations or control functions’.</p> <p>In simple terms, software is a set of instructions for computer hardware along with its documentation.</p>
Software Partitioning	<p>A technique for providing isolation between functional independent software components to contain and/or isolate faults and potential reduce the effort of the software verification process. If protection by partitioning is provided, the software level for each partitioned components may be determined using the most severe failure condition category associated with that component.</p>
Subassembly	<p>A composite of piece parts.</p>
Subsystem	<p>A composite of assemblies whose functions are integrated to achieve a specific activity necessary for achieving a mission.</p>
System	<p>A composite of subsystems whose functions are integrated to achieve a mission/function (includes materials, tools, personnel, facilities, software, equipment).</p>
System	<p>A combination of interrelated items arranged to perform a specific function [ARP5754A p13].</p>
Systematic	<p>Using a fixed and organised plan (e.g. We have got to be a bit more systematic in the way that we approach this task). [Cambridge Dictionary].</p> <p>Adjective, done or acting according to a fixed plan or system; methodical [Oxford English Dictionary].</p>
Systematic Failure	<p>Systematic Failures are produced by design and implementation Faults caused by Errors made by Developers (i.e. humans or tools) during System development or manufacture, or by human Error during operation or maintenance.</p>
Systemic	<p>Adjective, relating to a system as a whole [Oxford English Dictionary].</p> <p>Systemic problem or change is a basic one, experienced by the whole of an organisation or a country and not just particular parts of it [Cambridge Dictionary].</p>
Technical Standard Orders	<p>A detailed airworthiness specification issued by the authorities (e.g. FAA or EASA) to ensure compliance with TSO requirements as a minimum performance standard for specified articles.</p>
Type Certification Basis	<p>All the Airworthiness Standards/Design Codes (e.g. 14 CFR Part 25 or CS-25) and the Operating Rules (e.g. 14 CFR Parts 91, 121 or CS-26) which the aircraft has been designed to comply with.</p>

Term	Definitions (+ reference source if applicable)
Unconfirmed Failure	If no component failure is detected, the equipment performs in accordance with specification.
Unscheduled removal	The removal of an item brought about as a result of a known or suspected fault and/or defect.
Validation	The determination that the requirements for a specific system level or correct and complete (refer, inter alia ARP4754A). The determination that the requirements for a product are sufficiently correct and complete. ‘Did I build the right house?’ (with respect to my original expectations) [ARP 510].
Verification	The evaluation of an implementation of requirements to determine that they have been met [ARP4754A]. The evaluation of an implementation to determine that applicable requirements are met. ‘Did I build the house right?’ (in accordance with the drawings) [ARP5150].
Vicissitude	The definition of vicissitude according to Oxford Dictionaries is: ‘A change of circumstances or fortune, typically one that is unwelcome or unpleasant’.
Zonal Inspection	A collective term comprising selected general visual inspections and visual checks that are applied to each aeroplane zone, defined by access and area, to check system and power plant installations and structure for security and general condition. Refer to 03-02-I3007-02 for further definition.
Zonal Inspection Program (ZIP)	A part of an aeroplane model’s overall maintenance program where the whole of the aeroplane is divided into zones. For each zone of the aeroplane, applicable maintenance instructions are identified.

Index

'Note: Page numbers followed by “f” indicate figures, “t” indicate tables, and “b” indicate boxes.'

A

Aeroplane Safety Assurance
Processes, 325
Aircraft level requirements, 178
Air Data Computer (ADC), 16
Architectural mitigation, 272
Automation, 345

C

Causal analysis, 89
CC. *See* Control Categories (CC)
Common Cause Failures (CCF), 67
Common Mode Analysis
advantages, 153
aim, 133–135
background, 133–135
case studies, 145–152
checklist, 146
identification, 145–146
independence validation, 152
qualitative/quantitative assessments, 152
recommend corrective/preventative
action, 152
vulnerabilities, 146
checklist, 143
AND events independence,
138t–142t, 143
example, 133b–134b
identification, 136–137
independence validation, 144–145
limitations, 154
objectives, 135
qualitative/quantitative assessments, 145
scope, 135
vulnerability to identify/verify
independence criteria, 143
Component external failure modes, 180
Component intrinsic hazards, 180, 181t–182t

Configuration Management Plan (CMP), 224
Configuration Management process, 237t
RTCA/DO-0178C, 236, 238t–239t
RTCA/DO-254, 236, 240t–242t
Containment/mediation mechanisms, 295,
295t
Control Categories (CC), 236
Crew errors, 354t
advantages, 358–359
Air Inter Flight 148, Strasbourg, 326b
British Midland Flight 92, Kegworth,
326b–327b
commission errors, 327
crew training, recommendations, 349–350
crew warning cues, 352
ergonomics, 329–337
identify unsafe system operating
conditions, 338–343
initial evaluations, 337
intentional errors, 328
limitations, 359
omission errors, 327
poor design, 353
safety influence vs. product development
life cycle, 353–356, 356f
standardisation, 329–337
systems and controls, 351
unintentional error, 328
unsafe system operating conditions,
351–352
verification, 345–348
Crew workload, 345
CS/FAR2X.1309, 9, 10f, 11t
CS25.1309, safety criteria, 11t

D

Decision coverage (DC), 313
Dependence Diagrams (DD), 89

- Design Review Based on Failure Mode (DRBFM) approach, 102
 - Development Assurance Level (DAL), 63
 - Development Assurance process, 198, 200
 - advantages, 269
 - errors, faults and failures, 323–324
 - guidance material pertaining
 - product and artefact, 195–196
 - RTCA/DO-254, 196–198
 - RTCA/DO-178C, 196
 - SAE ARP4754A, 196
 - Hardware Development Assurance
 - advanced verification methods, 273
 - architectural mitigation, 272
 - Functional Failure Path Analysis (FFPA), 273
 - objectives and lifecycle data outputs, 274–283, 275t–282t
 - product service experience, 272
 - simple/complex hardware, 272
 - limitations, 269–270
 - requirements allocation, 201–208, 254–255
 - requirements validation, 255–258, 264–266
 - validation accomplishment, 209–213
 - validation deliverables, 213
 - validation planning, 209
 - requirements verification, 258–260, 264–266
 - verification accomplishment, 220
 - verification deliverables, 224
 - verification planning, 220
 - RTCA/DO-254 vs. RTCA/DO-178C, 322
 - Software Development Assurance, 288
 - development artefacts, 320–321
 - formal methods, 308–316, 308t
 - integral processes, 317–320
 - objectives vs. DAL, 284–287, 286t
 - software coding, 299–300
 - software design, 289–298, 290t–291t
 - software requirements, 287
 - software validation, 302–304
 - software verification, 302–304
 - types, 284, 284t
 - start-up planning, 202t, 253–254
 - RTCA/DO-254, 201, 205t–206t
 - RTCA/DO-178C, 201, 203t–204t
 - supporting process, 224, 260–264
 - certification and regulatory authority
 - coordination, 247t–248t, 249, 250t–252t
 - Configuration Management process, 224–236
 - process assurance, 236, 240t–243t
 - Safety Assessment process, 224, 235t
 - system development process, 265f–266f, 266–268
 - Display Concentrator Unit (DCU), 16
 - Display Control Panel (DCP), 8
 - DOORS, 208
- E**
- Environmental situational awareness, 342
 - Ergonomics, 329–330
 - Error tolerance, 345
- F**
- Fail safe, 338
 - Failure Modes and Effects Summary (FMES), 47
 - Failure Modes & Effects Analysis (FMEA), 3, 47
 - advantages, 130
 - aim, 102
 - background, 101–102
 - case studies, 120–129
 - compensating actions, 119, 123
 - detection identify, 118–119
 - effect evaluation, 117–118
 - functional block diagram (FBD), 104
 - identification, 114–116, 123
 - limitations, 124t–128t, 131
 - objectives, 102–103
 - overview, 101–104
 - prioritisation, 119, 123, 124t–128t
 - process modelling, 105f, 106
 - scope, 103–104, 106–114, 121
 - severity analysis, 119, 123
 - software-induced errors, 116
 - Fault Tree Analysis (FTA)
 - advantages, 90–91
 - aim, 60
 - analyses, 74–75
 - background, 59–60
 - case studies, 73–88
 - Common Cause Failures (CCF), 67
 - component designers, 64

- defined, 59–61
 - development, 75–82
 - development assurance level (DAL), 63, 78
 - validating, 70–72, 71t
 - functional independence, 72
 - ground rules, 64–65
 - human factors, 63
 - human hazard validating, 72–73
 - Immediate-Necessary-Sufficient' (I-N-S)
 - concept, 65
 - item development independence, 72
 - limitations, 91
 - objectives, 61
 - Primary-Secondary-Command' (P-S-C)
 - concept, 66
 - process flow, 61, 62f
 - reliability, 63, 76–78
 - reliability validating, 68–70
 - scope, 61
 - State-of-the-System and State-of-the-Component (SS-SC) concept, 66
 - system integrators, 63
 - system level, 63
 - validation, 82–88
 - verification, 73, 88
 - FFPA. *See* Functional Failure Path Analysis (FFPA)
 - Formal methods, 273, 312f, 316
 - complementary to testing, 310
 - MCDC testing, 315
 - problem domains, 309
 - requirements-based testing vs. structural testing, 311
 - software testing, 311
 - static code analysis, 310
 - test coverage analysis requirements, 313
 - Functional Failure Path Analysis (FFPA), 273
 - Functional hazard analysis, 38
 - advantages, 56
 - aim of, 37–38
 - background, 37, 37f
 - case studies, 47–51
 - aircraft level steps, 47–49, 48f, 50t, 51f
 - system level steps, 49, 51, 52t–55t
 - defined, 38–40
 - failure effects, 41–42, 43t
 - functional failure modes, 41
 - limitations, 57
 - objectives, 38
 - Preliminary System Safety Assessment (PSSA), 45–46
 - safety target accomplishment, 42, 45f, 46–47
 - safety targets, 41–46
 - scope of, 38
 - severity, 42, 44t
- ## G
- Geographical situational awareness, 342
 - Goal Structuring Notation (GSN), 23
 - aim, 25
 - background, 23–24, 24f
 - case studies, 28–30
 - conducting, 25–28
 - objective, 26–27
 - objective contextualise, 27
 - process modelling, 26, 26f
 - strategy contextualise, 27
 - symbols, 25–26
 - overview, 23–25
 - scope, 25
- ## H
- Hardware Design Plan, 254
 - Hardware Development Assurance
 - advanced verification methods, 273
 - architectural mitigation, 272
 - Functional Failure Path Analysis (FFPA), 273
 - objectives and lifecycle data outputs, 274–283, 275t–282t
 - product service experience, 272
 - simple/complex hardware, 272
 - High-Level Software Requirements, 287
 - Human errors, 326. *See also* Crew errors
 - Human Factors Team Report, 325
 - Human-machine interface (HMI), 345
- ## I
- Immediate-Necessary-Sufficient (I-N-S)
 - concept, 65
 - Inertial Reference Unit (IRU), 13–14, 17
 - In-Service phase, 372
 - Integral processes
 - configuration management, 317
 - software QA, 320
 - Intentional errors, 328
 - Intrinsic hazard analysis checklist, 180, 181t–182t

L

Legacy system description, 12–14
 altitude, 12
 ARINC 429 data bus, 19–20, 19f
 attitude, 13–14, 13f
 electrical interface, 18–19, 18t, 19f
 general description, 12, 12f
 standby display, 17–18, 17f
 Line Replaceable Units (LRUs), 12
 Liquid Crystal Displays (LCDs), 14
 Logical partitioning, 294
 Low-Level Software Requirements, 287

M

Markov Analyses (MA), 89
 Material Safety Data Sheets (MSDS), 374
 Modified condition/decision coverage
 (MCDC), 314

O

Operational Workload Analysis (OWA), 346

P

Particular risk analysis
 accomplish recommendations, 163
 advantages, 173
 aim, 155
 background, 155
 case studies, 166–172
 hazard/event, 156
 limitations, 173
 modelling process, 156
 objectives, 155–156
 probability declarations, 163–166,
 164t–165t, 172
 safety target, 156
 scope, 156
 Partitioning, 292
 Part-task evaluations, 337
 Physical partitioning, 294
 Pilot error, multicrew cockpit, 343, 344f
 Preliminary Aircraft Safety Assessment
 (PASA), 8
 Preliminary Design Review (PDR), 137
 Preliminary System Safety Assessment
 (PSSA), 8, 178
 Primary barometric altitude display, 76–78,
 77f, 78t

Procedural Event Analysis Tool (PEAT), 343
 Process assurance, 236, 240t–243t
 Product service experience, 272
 Programming language, 299–300
 Proximity issues, zone 211/212, 186,
 187t–188t

R

Random failures, 193–194
 REACH, 374
 Real-time operating systems
 (RTOS), 296
 Real-time systems, 295, 296t
 Requirements Allocation, 207
 Requirements Engineering, 207
 RTCA/DO-178, 196
 RTCA/DO-178B, 198, 265f, 266, 269
 RTCA/DO-178C, 196–197, 265f
 RTCA/DO-254, 196–198, 265f, 266

S

SAE ARP4754A, 196
 SAE ARP4761, 178b, 180b, 185b
 Safety Management System (SMS)
 design organisation, 373, 373f
 hazards, 373–374
 In-Service phase, 372
 Instructions for Continued Airworthiness,
 375–378
 manufacturing instructions, 378–379
 Occurrence Reporting, 379
 SB/AD, 379–380
 compliance time, 380–382, 382f–383f,
 382t–383t
 unsafe condition, 380
 subsystem system safety deliverables,
 373, 374f, 382f–383f
 Safety-specific analysis, 273
 Safety strategy, 253f
 Senior Management Team (SMT), 236
 Situational awareness, 345
 Software
 architecture, 292, 293t
 coding, 299–300, 301t
 fault tolerance, 296–297, 298t
 level, 197
 lifecycle data, 198
 validation, 302–304
 verification, 302–304

- Software Coding Standards, 254
 - Software Design Standards, 254
 - Software Development Assurance, 288
 - development artefacts, 320–321
 - formal methods, 308–316, 308t
 - integral processes, 317–320
 - objectives *vs.* DAL, 284–287, 286t
 - software coding, 299–300
 - software design, 289–298, 290t–291t
 - software requirements, 287
 - software validation, 302–304
 - software verification, 302–304
 - types, 284, 284t
 - Software Development Plan (SDP), 254
 - Spatial partitioning, 294
 - Spatial/temporal situational awareness, 342
 - SSA. *See* System Safety Assessment (SSA)
 - process
 - Statement coverage, 313
 - State-of-the-System and State-of-the-Component (SS-SC) concept, 66
 - Static code analysis, 310
 - Supporting process, 224, 260–264
 - certification and regulatory authority
 - coordination, 247t–248t, 249, 250t–252t
 - Configuration Management process, 224–236
 - process assurance, 236, 240t–243t
 - Safety Assessment process, 224, 235t
 - Systematic failures, 194
 - System development information flows, 266, 266f
 - System Engineering, 207
 - Systemic deficiencies, design, 181
 - System Safety Assessment (SSA) process, 207, 329f, 383–385
 - aim, 1–2
 - background, 1
 - conducting
 - formulation, 6
 - implementation, 8
 - process modelling, 5, 5f
 - recommendations and limitations, 9
 - safety targets, 6–8
 - scope, 5–6
 - crew training, recommendations, 349–350
 - CS/FAR2X.1309, 9, 10f, 11t
 - ergonomics/standardisation
 - allow time for decision, 335
 - cater for perception, 333–335
 - data compendiums, 332
 - design for, 330, 331f
 - HF design standards, 332
 - HF principles and guidelines, 333
 - provide feedback mechanism, 335–336
 - provide means for action, 335
 - specification, 336–337
 - identify unsafe system operating
 - conditions, 339f, 343
 - flight crew instructions, 343
 - means of detection, 339
 - system failure conditions, 338
 - update probability declarations, 343, 344f
 - vulnerability to crew error, 340–342
 - initial evaluations, 337
 - intentional errors, 328
 - objectives, 2
 - Safety Management System (SMS)
 - Airworthiness Directives (ADs), 379–382
 - design organisation, 373, 373f
 - hazards, 373–374
 - In-Service phase, 372
 - Instructions for Continued Airworthiness, 375–378
 - manufacturing instructions, 378–379
 - Occurrence Reporting, 379
 - Service Bulletin (SB), 379–382
 - subsystem system safety deliverables, 373, 374f, 382f–383f
 - scope, 2–5
 - Type Certification, 371
 - unintentional error, 328
 - verification
 - critical tasks, 346–347
 - errors, consequences and mitigations, 347–348
 - flight test phase, 345
 - incorporate mitigations, design, 348
 - System Safety Working Group (SSWG), 143
 - System situational awareness, 342
- T**
- Tactical situational awareness, 342
 - Task analysis, 337

Temporal partitioning, 294
Threat and error management (TEM), 349
Time-dependent assessment, 89
Total Air Temperature (TAT)
 probe, 16–17
Traceability, 213
Turkish Airlines B737-800, 350b

U

Unintentional error, 328

V

Validation
 accomplishment, 209–213
 deliverables, 214t–215t
 RTCA/DO-254, 213, 217t–219t
 RTCA/DO-178C, 213, 216t
 planning, 210t
 RTCA/DO-254, 209, 212t
 RTCA/DO-178C, 209, 211t
Vector Cast's predicate table features, 208
Verification
 accomplishment, 220
 deliverables, 225t
 RTCA/DO-254, 224, 232t–234t
 RTCA/DO-178C, 224, 226t–231t

planning, 221t
 RTCA/DO-254, 220, 223t
 RTCA/DO-178C, 220, 222t

Z

Zonal Safety Analysis (ZSA), 177f, 184f
 advantages, 190
 basic installation, 176
 B707-300 taking off, Aden, 175b
 design and installation guidelines,
 177–178, 185–186
 event independence, 177
 inspection of each zone, 183–184
 interference between systems, 176
 JAL792 taking off, Shanghai, 175b–176b
 limitations, 190
 maintenance errors, 177
 potential proximity issues, 178, 186,
 187t–188t
 identify components, 180
 identify potential issues, 180–183
 zones, defined, 179, 179f
 ZSA report, 184, 186–190, 191f
Zonal safety analysis inspection sheet, 186,
 189t
ZSA. *See* Zonal Safety Analysis (ZSA)