SOFTWARE VERIFICATION RESEARCH CENTRE

SCHOOL OF INFORMATION TECHNOLOGY

THE UNIVERSITY OF QUEENSLAND

Queensland 4072
Australia

TECHNICAL REPORT

No. 99-30

A Survey of International Safety Standards

Axel Wabenhorst, Brenton Atchison

November 1999

Phone: +61 7 3365 1003
Fax: +61 7 3365 1533

A Survey of International Safety Standards

Axel Wabenhorst, Brenton Atchison

Software Verification Research Centre
The University of Queensland QLD 4072
Australia

email: {akw,brenton}@svrc.uq.edu.au

Abstract

This report presents a survey of international standards for computer-based safety-critical systems. Eleven standards are surveyed: the Australian Def(Aust) 5679; MIL-STD-882C; NATO STANAG 4404 and STANAG 4452; UK Def Stan 00-56, Def Stan 00-55 and Def Stan 00-54; avionics standards ARP4754, ARP4761 and RTCA/DO-178B; and the civilian standard IEC 61508. The standards are surveyed according to a wide range of attributes, including levels of prescription and tailoring; safety management issues such as agents, their responsibilities, and deliverables required; and technical issues such as development constraints, hazard analysis, risk assessment, implementation assurance, human factors and non-development items.

Keywords: safety-critical systems, safety assurance, safety standards, safety management

Table of Contents

1. Introduction

1.1 Scope

A safety-critical system is a system in which failure to function as expected could result in death or serious injury. Many standards have been written for the safety of computer-based systems in both the military and civilian sectors. A recent addition is the Australian Defence Force standard Def(Aust) 5679 [2].

This document presents a survey of international standards for safety-critical computer based systems with the intention of drawing comparisons to Def(Aust) 5679. Apart from Def(Aust) 5679, standards surveyed include the US military standard MIL-STD-882C [2], NATO standards STANAG 4404 [3] and STANAG 4452 [4], UK military standards Def Stan 00-56 [5], Def Stan 00-55 [6] and Def Stan 00-54 [7], civilian avionics standards ARP4754 [8], ARP4761 [9] and RTCA/DO-178B [10] and civilian standard IEC 61508 [11]. The scope and background of each standard is summarised in Section 2.1.

The standards are surveyed in accordance with a wide selection of attributes, including issues of usability, safety management and technical processes. The attributes are described in Section 2.2, and the standards are compared in Section 2.3 according to these attributes. The selected attributes and survey results expand on a previous comparison of standards undertaken by the Australian Defence Science and Technology Organisation [12]. Detailed survey results are provided in Sections 3, 4 and 5. References to the text of the standards are provided by footnotes for convenience.

In a separate report [13], conclusions are drawn about the relationship between Def(Aust) 5679 and other international standards and guidance is provided on how Def(Aust) 5679 might accommodate and contribute to other standards relating to safety critical computer-based systems.

1.2 Acronyms and Definitions

ARP	Aerospace Recommended Practice
CLSD	Component-Level System Design
CSR	Component Safety Requirement
DRACAS	Data Reporting, Analysis and Corrective Action System
E/E/PE	Electrical/Electronic/Programmable Electronic (IEC 61508)
FHA	Functional Hazard Assessment
FMEA	Failure Modes and Effects Analysis
FTA	Fault Tree Analysis
LOT	Level of Trust
MA	Managing Activity
NDI	Non-Development Item
PSSA	Preliminary System Safety Assessment
SIL	Safety Integrity Level
SSA	System Safety Assessment
SSMP	System Safety Management Plan
SSPP	System Safety Program Plan

1.3 Referenced Documents

1 Australian Department of Defence. Def(Aust) Standard 5679, The Procurement of Computer-Based Safety-Critical Systems, Army Technology Engineering Agency, October 1998.

2 US Department of Defense. Draft MIL-STD-882C: Standard Practice for System Safety Program Requirements, January 1996.

3 North Atlantic Treaty Organisation. NATO STANAG 4404: Safety Design Requirements and Guidelines for Munition Related Safety Critical Computing Systems, Edition 1, December 1996.

4 North Atlantic Treaty Organisation. NATO STANAG 4452: Safety Assessment of Munition-Related Computing Systems, September 1996.

5 UK Ministry of Defence. Def Stan 00-56: Safety Management Requirements for Defence Systems. December 1996.

6 UK Ministry of Defence. Def Stan 00-55: Requirements for Safety Related Software in Defence Equipment, August 1997.

7 UK Ministry of Defence. Interim Def Stan 00-54: Requirements for Safety Related Electronic Hardware in Defence Equipment, April 1999.

8 Society of Automotive Engineers. Aerospace Recommended Practice 4754: Certification Considerations for Highly-Integrated or Complex Aircraft Systems, November 1996.

9 Society of Automotive Engineers. Aerospace Recommended Practice 4761: Guidelines and Methods for Conducting the Safety Assessment Process on Civil Airborne Systems and Equipment, December 1996.

10 RTCA, Inc. RTCA/DO-178B: Software Considerations in Airborne Systems and Equipment Certification, December 1992.

11 International Electrotechnical Commission. IEC 61508: Functional Safety of Electrical/Electronic/ Programmable Electronic Safety-Related Systems, December 1997.

12 K.A. Eastaughffe, A. Cant, M.A. Ozols. A Critique of Standards for Safety Critical Computer-Based Systems. Proceedings of the Fourth International Software Standards Symposium (ISESS'99), Curitiba, Brazil, May 1999.

13 Axel Wabenhorst, Brenton Atchison. Comparison of Def(Aust) 5679 and International Safety Standards for Australian Defence Acquisition, September 1999.

2. Executive Summary

2.1 Safety Standards

The following safety standards are addressed by this survey.

Def(Aust) 5679 The Australian Defence Standard Def(Aust) 5679 [1] is a standard for the procurement of computer-based safety-critical systems, published by the Department of Defence in March 1999. It focuses on safety management and the phased production of safety assurance through the system development lifecycle, with emphasis on software and software-like processes. Assurance is delivered in the form of a safety case that provides auditable evidence of safety.

MIL-STD-882C The US Department of Defense Standard MIL-STD-882C [2] provides uniform requirements for defence system safety programs. It is well-established and emphasises the proactive management of safety issues and systematic hazard analysis. The standard is more general in scope than Def(Aust) 5679, in that it applies to the procurement of all systems, including chemical and mechanical systems. However, little special consideration is given to computer-based systems.

STANAG NATO STANAG 4404 [3] provides requirements and guidelines for the design and development of munition-related safety-critical computing systems. NATO STANAG 4452 [4] provides a general framework for assessing the safety of such systems, with an emphasis on hazard analysis and testing. It should be used in conjunction with NATO standard AOP-15 to obtain an overall system safety assessment.

Def Stan 00-56 UK Def Stan 00-56 [5] provides requirements and guidelines for the development of all defence systems. The standard applies to all systems engineering phases of the project lifecycle and all systems, not just computer-based ones.

Def Stans 00-55 & 00-54 UK Def Stan 00-55 [6] describes requirements and guidelines for procedures and technical practices in the development of safety-related software. The standard applies to all phases of the procurement lifecycle. Interim UK Def Stan 00-54 [7] describes requirements for the procurement of safety-related electronic hardware, with particular emphasis on the procedures required in various phases of the procurement lifecycle. Both standards are designed to be used in conjunction with Def Stan 00-56.

ARP The Society of Automotive Engineers provides two standards representing Aerospace Recommended Practice to guide the development of complex aircraft systems. ARP4754 [8] presents guidelines for the development of highly integrated or complex aircraft systems, with particular emphasis on electronic systems. While safety is a key concern, the advice covers the complete development process. The standard is designed for use with ARP4761 [9], which contains detailed guidance and examples of safety assessment procedures. These standards could be applied across application domains but some aspects are avionics specific.

DO-178B RTCA/DO-178B [10] provides guidelines on the production of software for airborne systems and equipment. The standard could be applied across application domains but some aspects are avionics specific.

IEC 61508 IEC 61508 [11] is a generic standard for electrical/electronic/programmable electronic safety-related systems. The standard may be used directly or tailored for a specific application domain. The standard is divided into seven parts. The parts contain, respectively, general requirements; hardware requirements; software requirements; definitions and abbreviations; examples of methods used to determine safety integrity levels; guidelines on satisfying hardware and software requirements; and an overview of techniques and measures. Some parts are in draft form but the standard is expected to be approved for use in 1999. The versions used for this report are expected to be close to the approved versions.

2.2 Attributes of Standards

The attributes selected for the survey of safety standards are categorised as being related to usability of the standard, management processes to be applied or technical processes and tasks.

Usability

Issues of usability are addressed in Section 3 and relate to the ease with which the standard can be understood and applied to a contract. Specific issues include:

- **Level of prescription:** the degree of prescription in the standard and the detail of the requirements to be satisfied for compliance. A related issue is the amount of guidance offered by the standard and the means by which this is separated from the requirements. (See Section 3.1.)

- **Tailoring and conformance:** the requirements for conformance and the ability to tailor the standard requirements for particular contracts. This includes selection and modification of requirements to be satisfied, as well as completing unspecified information. (See Section 3.2.)

Management Issues

The approaches by the different standards to management issues are covered in Section 4, and comprise the following:

- **Agents and responsibilities:** the different parties, or agents, in the system procurement process and their responsibilities, whether it be to specify system requirements, to provide evidence of safety assurance, or to review the safety assurance arguments. The role of certification bodies is also considered. (See Section 4.1.)

- **Deliverables** required to demonstrate safety of the system (Section 4.2). Such deliverables include management plans and technical data. Configuration management requirements for the deliverables are also considered.

- **Planning and control** of safety activities. (See Section 4.3.)

- **Project lifecycle:** the relationship between the system development lifecycle and the safety activities. Any assumptions about the project lifecycle are also recorded. (See Section 4.4.)

- **Post development activities**, including installation and commissioning, maintenance and modification. Particular attention is paid to post delivery changes. (See Section 4.5.)

Technical Issues

Technical issues are considered in Section 5, and comprise the following:

- **Development constraints:** the constraints on development deliverables and the methods used to produce them. (See Section 5.1.)

- The **hazard analysis** activities to be performed to identify the system hazards and safety-critical components. (See Section 5.2.)

- The **risk and integrity assessment** model used to determine the system risk and the degree of care required in the assurance activities. (See Section 5.3.)

- The means of achieving general **design assurance**. (See Section 5.4.)

- The methods of achieving **software assurance**. (See Section 5.5.)

- The methods of achieving **hardware assurance**. (See Section 5.6.)

- The consideration of **human factors** in the design and implementation of the system, and the skills and training of system users. (See Section 5.7.)

- The use of **Non-Development Items** (NDIs) in the construction of a safety critical system. (See Section 5.8.)

2.3 Comparison of Standards

A summary of the survey findings is presented in Table 2-1. This summary is a simplification, and should be considered only together with the detailed discussion in the following sections. Some of the findings are discussed below.

Usability Issues The standards vary in the way that they are intended for contractual use, from mandating all requirements to providing non-compulsory guidance. There is no provision for tailoring of requirements in Def(Aust) 5679, the UK Def Stans 00-55 and 00-54, or IEC 61508. The two UK Def Stans have a limited scope, dealing with software and hardware aspects of safety only, so tailoring makes less sense than for a system-level standard. IEC 61508 and Def(Aust) 5679 are system-level standards that aspire to be sufficiently generic so that tailoring is not required. In the other standards, where tailoring is permitted, the Customer and Developer are responsible for selecting requirements for an adequate safety program. MIL-STD-882C offers guidance on tailoring depending on project attributes, but the remaining standards do not.

Management Issues All standards recognise the importance of safety management and impose various levels of requirements. They define the responsibilities of the Customer and Contractor, with the Customer ultimately responsible for procuring a safe system. All military standards have provision for a safety management group to review progress of the safety activities. All military standards except the NATO STANAGs have provision for an independent auditor to oversee safety processes. In addition, Def(Aust) 5679, UK Def Stans 00-55 and 00-54 and IEC 61508 require independent reviews of the technical content of the tasks. The avionics standards require an independent Certification Authority to certify the safety of the system after considering the deliverables; however, there is no mention of reviews during development.

All standards except NATO STANAG 4404, require approval of a safety management plan before development begins. The types of other deliverables vary, but all are designed to show identification of hazards and evidence of their resolution. Def(Aust) 5679, the avionics standards and UK Def Stan 00-56 further require a high-level safety argument to supplement documentation of technical tasks.

An important management consideration for all standards is the integration of safety activities into the system development lifecycle. Consideration of safety issues as early as possible in the system development is much more effective in assuring safety than delaying consideration until after many of the design and development decisions have been made. All standards except MIL-STD-882C and the NATO STANAGs advocate a close relationship between safety activities and the development lifecycle.

Technical Issues While all standards follow a similar technical framework, the details of technical requirements vary substantially, particularly in the areas of risk assessment and assurance. All system-level standards require Preliminary Hazard Analysis and a component level hazard analysis. In addition, MIL-STD-882C, STANAG 4452 and the ARP standards recommend hazard analysis for system integration.

Most standards determine levels of acceptable risk from accident severity and likelihood. In Def(Aust) 5679 and the avionics standards, acceptable risk levels are prescribed, while other standards allow acceptable risk to be defined for each project, often allowing for various levels of risk tolerability. Since likelihood of complex design and software failure cannot be predicted, most standards employ some form of integrity level to measure design confidence. These are often mapped to numerical failure rates for the purpose of risk assessment.

Most standards define constraints on development processes and methods, such as architecture, design methods, programming languages and coding standards. Most constraints are on the content of development deliverables required to support safety assurance activities. In some cases, the form of the deliverables must allow the necessary analysis, such as formal proof. NATO STANAG 4404 differs by providing detailed guidance on system design and implementation.

All standards define safety assurance tasks, using various forms of verification and validation to demonstrate resolution of hazards. Where integrity levels are defined, the assurance effort increases with required integrity. In Def(Aust) 5679, the UK military standards and IEC 61508, additional assurance is achieved with increased rigour, requiring formal proof in the most extreme circumstances. DO-178B requires more verification evidence with more independence as integrity targets increase.

Table 2-1 Summary of Standards Survey

	Def(Aust) 5679	MIL-STD-882C	NATO STANAGs 4404 & 4452	UK Def Stan 00-56
Level of Prescription and Guidance	Requirements in bold font; guidance in normal font	Requirements only, but these are usually open to interpretation	4404: depends on application domain (table). 4452: separate requirements and guidance	Separate requirements and guidance documents
Tailoring and Conformance	No tailoring	Tailoring to application	Tailoring to application	Tailoring to application
Agents and Responsibilities[1]	Auditor, Evaluator, Safety Management Group	Tasks for Auditor, System Safety Working Group	4404: Configuration Control Boards. 4452: Task for System Safety Working Group	Independent Safety Auditor, Project Safety Committee
Deliverables[2]	Safety Case, safety review reports, evaluation and audit reports	Progress reports	4404: technical tasks only	Safety Case, audit reports
Safety Planning and Control[3]	audits and evaluations	audits	4404: no management plans; peer reviews and two-person rule	Audits, DRACAS
Project Lifecycle[4]	-	No lifecycle assumed	4404: one lifecycle that includes testing. 4452: no lifecycle assumed	-
Post Development Processes	Installation, maintenance, commissioning,	Failure analyses	4404: maintenance. 4452: modification	Modification
Development Constraints[5]	Design methods, programming languages	"design for minimum risk"	4404: very detailed requirements for architecture and coding	design methods, coding, tools
Hazard Analysis[6]	-	Integration, human factors, health	4404: no hazard analysis 4452: Integration, change	Change, operations & support, health
Risk and Integrity Assessment	Component SIL derived from accident severity & external probability, use of fault-tolerant design	Risk class derived from hazard severity & probability, must be acceptable	4452: based on Software Control Categories, but not used subsequently	Target SIL or probability derived from accident severity and protective measures
Design Assurance	Rigour depends on SIL; use of formal methods	No specific requirements	No specific requirements	No specific requirements
Software Assurance[7]	Rigour depends on SIL; use of formal methods	-	Static & dynamic analysis	Use of static & dynamic analysis depends on SIL
Hardware Assurance	Testing. Use of formal methods depends on SIL	Testing recommended	4404: none. 4452: required	Use of static & dynamic analysis depends on SIL
Human Factors	SILs achieved by operator training; procedures	Training; procedures; operator hazard analysis	4404: interface design requirements. 4452: procedures; hazard analysis	Estimation of operator failure rates; training; procedures
Non Development Items	Transfer assurance or build safety case	Depend on size; tailoring	4452: analysis & testing	Safety case

[1] in addition to Customer and Developer
[2] in addition to Safety Management Plan, Hazard Log, documentation of technical tasks
[3] deviating from default: Safety Management Plan, reviews
[4] deviating from system definition; design; implementation; post development activities; with parallel safety lifecycle
[5] in addition to development documentation required to perform analyses
[6] in addition to Preliminary Hazard Analysis and component-level Hazard Analysis
[7] in addition to testing

	UK Def Stans 00-55 & 00-54	ARP 4754 & 4761	RTCA/DO-178B	IEC 61508
Level of Prescription and Guidance	Separate requirements and guidance documents	Guidance only	Guidance only	Requirements in parts 1-4, guidance in small-font "notes" and parts 5-7
Tailoring and Conformance	No tailoring	Tailoring by selecting guidance	Tailoring by selecting guidance	No tailoring; derived sector standards recommended
Agents and Responsibilities[8]	As for Def Stan 00-56, plus V&V Team	Certification Authority	Certification Authority	Functional safety assessor
Deliverables[9]	00-55: Software quality, development and V&V plans. 00-54: design plan	Certification Plan, Certification Summary	Software Accomplishment Summary, quality and verification plans	Plans for safety assessment and validation
Safety Planning and Control[10]	As for 00-56	Certification Plan	Lifecycle phase transition criteria	-
Project Lifecycle[11]	One lifecycle, includes verification	-	-	No development lifecycle assumed; detailed safety lifecycle
Post Development Processes	Maintenance	Installation, maintenance, modification	Modification	Installation, commissioning, maintenance, modification
Development Constraints[12]	ISO 9001, ISO 9000-3. languages, tools, coding	None	None	Detailed design methods and architecture
Hazard Analysis[13]	No hazard analysis	Integration	No hazard analysis	Continuous, to identify emergent hazards
Risk and Integrity Assessment	SILs derived from Def Stan 00-56	Target Assurance Level & failure probability derived from failure severity and fault-tolerant design	Software Level depends on failure severity and fault-tolerant design	Target SIL and probability derived from accident severity and protective measures
Design Assurance	Rigour depends on SIL. 00-55: use formal methods	General techniques arise from Assurance Level	General requirements, depend on Software Level	Rigour depends on SIL; use of formal methods
Software Assurance[14]	Rigour depends on SIL; use of formal methods	None	Depends on Software Level	Rigour depends on SIL; use of formal methods
Hardware Assurance	Static & dynamic analysis	None	None	Use of static & dynamic analysis depends on SIL
Human Factors	00-55: procedures	None	None	Procedures; training
Non Development Items	Verification, validation, can use service history	New safety assessment; can use service history	New safety assessment; can use service history	Service history or verification & validation

[8] in addition to Customer and Developer
[9] in addition to Safety Management Plan, Hazard Log, documentation of technical tasks
[10] deviating from default: Safety Management Plan, reviews
[11] deviating from system definition; design; implementation; post development activities; with parallel safety lifecycle
[12] in addition to development documentation required to perform analyses
[13] in addition to Preliminary Hazard Analysis and component-level Hazard Analysis
[14] in addition to testing

3. Usability Issues

3.1 Level of Prescription and Guidance

Def(Aust) 5679 Def(Aust) 5679 provides a framework for safety management and assessment rather than detailed guidance. The standard requires the Developer to interpret requirements and provide assurance by reasoned arguments and evidence, rather than satisfaction of prescribed technical criteria. As a result, the Developer must invest effort in planning the structure of the Safety Case. This increases the difficulty of the technical evaluation of the delivered safety case.

Sections incorporate requirements, guidelines and notes of explanation, with paragraphs stating requirements distinguished by bold font. While the guidelines and notes are useful, additional external guidance is required to apply the standard.

MIL-STD-882C MIL-STD-882C defines a number of management and technical tasks to be performed to achieve safety. Compliance with the standard requires satisfaction of the task requirements. Some advice is provided in an appendix[15] on the interpretation and application of requirements.

STANAG STANAG 4404 contains both requirements and guidelines. Each subsection is designated as mandated, optional or not applicable, depending on the application domain[16]. Justification must be provided if an optional subsection is not complied with. STANAG 4452 contains descriptions of required analysis tasks, with guidance provided in an appendix.

Def Stan 00-56 In Def Stan 00-56, requirements and guidance are separated into two parts, with the same section headings in both parts. Established sector-specific design standards may be applied instead of the standard if the Independent Safety Auditor and the Customer agree[17].

Def Stans 00-55 & 00-54 In Def Stans 00-55 and 00-54, requirements and guidance are separated into two parts, with the same section headings in both parts. Guidance is given on the application of each requirement to different SILs[18]. In Def Stan 00-55, some requirements need not be satisfied for certain SILs, provided sufficient justification is given. In Def Stan 00-54, some requirements may be satisfied with less rigour for certain SILs. For simple systems verifiable by exhaustive testing, some requirements need not be complied with

ARP The ARP standards constitute guidelines that are not mandated by law. The standards recognise that there may be alternative methods of satisfying the recommendations[19]. However, it is difficult to identify which parts of the documents should be treated as mandatory requirements.

DO-178B DO-178B constitutes guidelines that are not mandated by law. It acknowledges that there may be alternative methods of satisfying the recommendations, although it claims to represent a consensus of the aviation community[20].

IEC 61508 IEC 61508 presents requirements on tasks to be performed during the system development life-cycle. The requirements are extensive and include details of acceptable design and assurance techniques to be applied. However, exemptions are possible for low-complexity systems, where the failure modes of each component are well defined and the behaviour of the system under fault conditions can be completely determined[21].

Guidance on the determination of safety integrity levels and the means of providing assurance is provided in parts

[15] Appendix A
[16] STANAG 4404 Table B-1 Annex B
[17] part 1 section 1.6 p4
[18] 00-55 part 2 annex D; 00-54 part 2 annex C
[19] ARP4754 p9
[20] p2
[21] part 1 section 4 p13

5 and 6 respectively. Additional information about the design and assurance techniques referenced is provided in part 7.

3.2 Tailoring and Conformance

Def(Aust) 5679 Def(Aust) 5679 is not intended to be tailored. Compliance with the standard requires satisfaction of all requirements, but many of these provide scope for interpretation. In very few cases, requirements may be modified if all stakeholders agree.

MIL-STD-882C MIL-STD-882C is designed to be tailored for application to a contract. The Customer and Developer should agree on the selection of tasks to be applied and the extent of their application. Advice is provided on tailoring details to be specified for each task and guidance is given as to which tasks should be allocated, depending on the expected level of risk and dollar resources available[22]. The extent of possible tailoring may improve the cost-effectiveness of application but places great responsibility on the Customer and may result in abuse.

STANAG STANAG 4404 requires tailoring appropriate to the application to be included in the development contract[23]. In STANAG 4452, eight analysis tasks are available for application. For small computing systems, Analysis Task 6 is recommended instead of Tasks 1 to 5[24].

Def Stan 00-56 Def Stan 00-56 requires tailoring appropriate to the application domain and the system under development[25].

Def Stans 00-55 & 00-54 There is no scope for tailoring in Def Stans 00-55 and 00-54.

ARP The ARP standards focus on fundamental principles, and recommend tailoring of the application of the standards in the contract between the certification organisation and the developer. It is recognised that systems generally require engineering judgment by the two parties, especially in the light of the rapid developments in systems engineering and the variety of systems applications[26].

DO-178B Since DO-178B constitutes recommendations only, tailoring is permitted implicitly, but the concept of tailoring is not mentioned explicitly.

IEC 61508 IEC 61508 requires satisfaction of all requirements, although some requirements explicitly require the use of sector standards for compliance, for example definitions of acceptable risk levels[27]. IEC 61508 can be applied directly but is also intended to provide a framework for the development of industry specific standards[28] and has already provided the basis for European railway safety standards.

[22] pA-13. Allocating tasks depending on funds available is a significant difference from Def(Aust) 5679.
[23] STANAG 4404 section 4 pp3-4
[24] STANAG 4452 section 5 p4, see also Analysis Task 6.
[25] part 1 section 1.4 p4
[26] ARP4754 p10, ARP4761 p4
[27] e.g. part 1 section 7.5.2.3 p29
[28] part 1 section 1.1 p8

4. Management Issues

4.1 Agents and Responsibilities

Def(Aust) 5679 Def(Aust) 5679 defines the agents to be involved in the system development. The Customer is the procurer of the system and has ultimate responsibility for the safety of the system, including adherence to the safety standard[29]. The Customer is also responsible for specifying operational requirements and the system environment. The Developer is responsible for delivering the system to the customer together with assurances of safety. The prime contractor must ensure that subcontractors meet applicable requirements[30]. Users may be involved in providing information about the operational context and must be suitably trained[31].

The Auditor and the Evaluator oversee the development of the system. The Auditor has responsibility for ensuring compliance with the procedural aspects of the standard, while the Evaluator checks the validity of the Safety Case. The process may also involve a Certifier. The Auditor and Evaluator are appointed by the Customer, and are both independent of the Developer.

A Safety Management Group, comprising representatives of the Customer, Developer, Auditor and possibly the Certifier, is created to review the process of compliance with the standard. In particular, the Safety Management Group reviews the deliverables described by Section 4.2, including the Safety Management Plan of Section 4.3.

MIL-STD-882C In MIL-STD-882C, the terms MA (Managing Activity) and Contractor refer to the Def(Aust) 5679 Customer and Developer respectively. The MA imposes system safety tasks on the Developer, and is the only party with the authority to approve any residual risk in the system under development. The System Safety Manager and System Safety Engineer are also defined[32]. There is scope for an audit program in Tasks 102 and 104. There is scope in Task 105 for System Safety Groups and System Safety Working Groups to undertake reviews of the process of compliance with the standard.

STANAG In STANAG 4404, the developer has responsibility for implementing the design requirements and showing that the overall system safety goal is achieved, subject to review by the appropriate safety authority[33]. A Software Configuration Control Board and a Hardware Configuration Control Board approve any software or hardware changes respectively once baselines have been established. These boards should have (at least) one member in common. One member of the Software Configuration Control Board has responsibility for evaluating software changes for their potential safety impact[34].

In STANAG 4452, the developer conducts the analysis and testing tasks, and establishes and documents the System Safety Program. The Managing Activity must approve any deviations from the hazard risk assessment process in Appendix A[35]. According to Analysis Tasks 1 and 2, the Managing Activity must also approve analysis techniques, methodologies and tools used by the developer. A System Safety Working Group is established by Analysis Task 1.

Def Stan 00-56 In Def Stan 00-56, agents and their responsibilities are specified in detail. The Contractor appoints a Project Manager with responsibility for all safety activities. This Project Manager appoints a Project Safety Engineer, who has responsibility for implementing the tasks in the Safety Programme Plan (see Section 4.2). The Project Safety Committee is chaired by the Project Safety Engineer and consists of representatives of the Contractor, subcontractors, and the Independent Safety Auditor. This committee is responsible for endorsing the tolerability of each risk and the output of the safety reviews, and specifies corrective action if necessary[36].

[29] p19
[30] p19
[31] p20
[32] section 1.2 p1, section 3.2.2 p5, section 3.2.8 p5, section 3.2.19 p6, section 3.2.23 p7
[33] STANAG 4404 section 4 p4
[34] STANAG 4404 section 6.1 pp5-6
[35] STANAG 4452 section 6 pp4-5
[36] part 1 section 4.3.3 p7, sections 5.3.1-5.3.3 pp11-12

An Independent Safety Auditor is appointed by the Contractor and the Customer's Project Manger if the Preliminary Hazard Analysis identifies risks of sufficient severity. The Auditor is concerned with the adherence of the Safety Programme Plan to the standard, and audits the documentation provided by the Contractor[37]. Any deviations from the standard by the Developer must be approved by the Independent Safety Auditor and the Customer's Project Manager. The Customer's Project Manager must approve the Safety Programme Plan and any subsequent changes[38].

The Contractor has responsibility for ensuring that subcontractors' activities are consistent with the Safety Programme Plan, and that items obtained from subcontractors enable the system to meet overall safety requirements as specified by the standard. Subcontractors must document their activities in a separate Safety Programme Plan[39].

Def Stans 00-55 & 00-54 In Def Stan 00-55, the Design Authority corresponds to the Developer of Def(Aust) 5679. The Design Authority has responsibility for safety management, including that of subcontractors. The Design Authority appoints a Software Design Authority, who in turn appoints a Software Project Manager, a V&V Team, and a Software Project Safety Engineer. The Design Authority must demonstrate to the customer that appointees have appropriate qualifications and authority[40].

The Software Project Manager is responsible for discharging the requirements of Def Stan 00-55. The Design Team specifies, designs and codes the software. The V&V Team, which must be independent of the Design Team, verifies and validates the software. The Software Project Safety Engineer ensures that safety activities are conducted according to the Software Safety Plan. An Independent Safety Auditor is appointed in accordance with Def Stan 00-56[41].

In Def Stan 00-54, agents and responsibilities are as for Def Stan 00-56. In particular, the V&V Team conducts or reviews design analysis, simulation and physical testing activities. Independence between the developer and reviewer is recommended for certain requirements at certain SILs[42].

ARP In the ARP standards, the Certification Authority is the organisation that defines certification requirements, conducts reviews of compliance with safety requirements, and certifies compliance with the requirements. The Applicant is the organisation that requires and provides evidence for certification[43]. Beyond this, the standards do not allocate responsibilities for compliance activities. However, ARP4754 recommends process assurance activities to ensure adequate communication between parties (see Section 4.3).

DO-178B In DO-178B, as in the ARP standards, the Certification Authority is the organisation that defines certification requirements, conducts reviews of compliance with safety requirements, and certifies compliance with the requirements. The Applicant is the organisation that requires and provides evidence for certification[44]. Beyond this, the standard does not allocate responsibilities for compliance activities. However, planning and review activities are recommended (see Section 4.3).

IEC 61508 IEC 61508 does not explicitly refer to agents or allocate responsibilities. However, the standard requires allocation of responsibilities to organisations or individuals to be made[45]. Personnel performing the Functional Safety Assessment (see Section 4.3) may need to be independent of the developers, depending on the integrity level of the system under development[46].

[37] part 1 section 5.3.4 pp12-14
[38] part 1 section 1.6 p4, section 5.2.4 p10
[39] part 1 section 5.7 p16
[40] 00-55 part 1 sections 12-14 p12
[41] 00-55 part 1 sections 15-19 pp12-14
[42] 00-54 part 1 section 9 p9; section 12.5.1 p13; part 2 annex C
[43] ARP4754 p7
[44] p45
[45] part 1 section 6.2.1 p16
[46] part 1 tables 4 and 5 p48

4.2 Deliverables

Def(Aust) 5679 Def(Aust) 5679 requires documentation of both the management process and compliance with the technical requirements of the standard.

A System Safety Management Plan (SSMP) (see Section 4.3) is submitted for approval before development begins. The SSMP includes a System Development Plan[47]. In addition, the submission of System Safety Review Reports and System Safety Evaluation Reports is required at regular intervals (see Section 4.3).

Assurance of safety is provided by a Safety Case, consisting of a number of reports. The reports document hazard and risk analysis activities (see Section 5) and component design and implementation assurance. System models and documentation are included where appropriate. In addition, a Hazard Log provides cross-references to records of hazards, critical functions and safety requirements, and their resolution. The detailed reports are summarised by a high level argument detailing the strategy through which safety is demonstrated[48].

Def(Aust) 5679 requires configuration management of all deliverables[49].

MIL-STD-882C MIL-STD-882C also requires documented evidence of safety management and the conduct of technical tasks. Data items are associated with each task, which define the structure and contents of the deliverables[50].

Task 102 requires a System Safety Program Plan (SSPP) (see Section 4.3) and Task 107 provides scope for the preparation of regular progress reports on system safety activity.

The results of the multiple hazard analysis tasks are documented in safety assessment reports, each of which contains details of system function, operation and safety engineering. Tasks 401 and 402 require the production of documentation assessing verification and compliance of safety specifications, and incorporating the techniques of Section 5. Task 106 provides scope for a Hazard Log similar to that in Def(Aust) 5679.

STANAG STANAG 4404 requires documentation supporting the implementation of design requirements[51]. STANAG 4452 requires documentation of the System Safety Program and each of the analysis and testing tasks conducted. Requirements traceability and a hazard log are mandated.

Def Stan 00-56 Def Stan 00-56 requires the establishment of a Safety Programme Plan (see Section 4.3). In addition, a Hazard Log is maintained for the three highest risk classes throughout the system lifecycle. The Hazard Log identifies hazards, associated risks and potential accidents, and documents progress on resolving risks. In addition, it references all analyses and reports produced during the safety program. Detailed requirements and guidance are given on the structure and content of the Hazard Log[52].

The Safety Case provides justification that the system is safe, and is constructed using information from the Hazard Log. The Safety Case must describe the system, its boundaries, and hazards and risks of the system together with their probabilities, and identify the safeguards in place to prevent accidents[53]. Guidelines are given on the evolution and structure of the Safety Case. The Independent Safety Audit is documented in an Independent Safety Audit Report[54].

Documentation of the Preliminary Hazard Listing, the Preliminary Hazard Analysis and the System Hazard Analysis is required. Detailed requirements are given concerning the contents of these deliverables. The Safety Criteria Report states the rationale used in the determination of accident risk classes and the corresponding system

[47] p24, p46
[48] p46
[49] p31
[50] Appendix D
[51] STANAG 4404 section 4 p4
[52] part 1 section 4.4.2 pp7-8, section 4.6 p9, section 5.8 pp16-18, part 2 section 5.8 pp19-22
[53] part 1 section 4.7 p9
[54] part 1 section 5.3.4.9 p13

function design rules and techniques. The Safety Compliance Assessment Report provides assurance that safety targets have been met[55].

Configuration management of documentation of data must meet the requirements of Def Stan 05-57, unless alternatives are agreed with the Customer. The configuration management system is identified in the Project Configuration Management Plan[56].

Def Stans 00-55 & 00-54 Def Stan 00-55 requires the production and maintenance of a Software Safety Plan (see Section 4.3).

The Software Safety Case is produced incrementally as part of the Safety Case of Def Stan 00-56 and provides reasoned justification that the software satisfies the safety aspects of the software requirements. Milestones in the production of the Software Safety Case are given. Supporting evidence for the Safety Case is provided in the Software Safety Records Log, which includes documentation of the technical tasks in Section 5. Other deliverables include a Software Quality Plan, a Software Development Plan, a Code of Design Practice, a Software Risk Management Plan, a Software Verification and Validation Plan, a Software Configuration Record and a Software Maintenance Plan[57]. Detailed guidance on the structure and contents of each deliverable is given in Annex B of Part 2.

Def Stan 00-55 requires all deliverables and software to be subject to configuration management, in accordance with Def Stan 05-57. Additional requirements and guidance are given[58].

A Safety Programme Plan, a Safety Case, and a Hazard Log are produced for Def Stan 00-54, and constitute parts of the deliverables of the same name in Def Stan 00-56. In addition, evidence for the Safety Case is accumulated in the Safety Records Log. Other deliverables include a Design Plan and a Maintenance Plan[59]. Further guidance on the structure and contents of each deliverable is given in Annex B of Part 2.

ARP ARP4754 suggests submission of a Certification Plan (see Section 4.3) outlining proposed activities for compliance with certification requirements. The Configuration Index identifies the physical elements comprising the system and their configuration, including interfaces with other elements. The Certification Summary outlines the results of certification activities and provides a high-level argument showing compliance with the requirements.

Other deliverables in addition to the minimum suggested above include a statement of functional and safety requirements; an architecture and design description, including failure containment and other safety features; a process assurance plan (see Section 4.3); and plans for and documentation of safety activities (see Section 5)[60]. The safety assessment procedures described by ARP4761 are documented in various safety assessment reports.

DO-178B DO-178B provides detailed guidelines on deliverables and their contents[61]. The master plan is the Plan for Software Aspects of Certification, which states how the Applicant proposes to comply with certification requirements. Other plans deal with configuration management, software development and verification, and quality assurance. Standards for the development of requirements, designs and code should also be defined.

The Software Accomplishment Summary demonstrates compliance with the Software Aspects of Certification. It references evidence in other deliverables including descriptions of requirements, design and code, verification procedures and results, configuration data and quality assurance records.

The degree of care required for configuration management of deliverables depends on the Software Level of the

[55] part 1 section 7.3.3-7.3.4 p26, section 7.5 pp31-33
[56] part 1 section 5.6 pp15-16
[57] 00-55 part 1 sections 7 & 8, pp7-10; section 20 p15; section 27 p18; annex B
[58] 00-55 part 1 section 25 pp17-18; part 2 section 25 pp25-26
[59] 00-54 part 1 sections 7 & 8 pp7-8; annex B
[60] A summary of all deliverables together with cross-references is provided on p18 of ARP4754.
[61] pp44-55

software (see Section 5.3)[62].

IEC 61508 IEC 61508 requires documentation for safety management and functional safety assessment. Outputs of each stage of the safety lifecycle are specified[63]. This includes descriptions of the system scope and environment; specification of safety requirements; plans for validation, installation, commissioning, operation, and maintenance; and outputs of the technical tasks in Section 5. The document structure is not mandated, but examples are given in Annex A of Part 1.

The documentation is subject to configuration management guidelines[64].

4.3 Safety Planning and Control

Def(Aust) 5679 Def(Aust) 5679 requires submission of a System Safety Management Plan (SSMP) by the developer before development begins, outlining the approach to be taken in order to comply with the safety requirements of the standard[65]. This includes

- a Safety Analysis Plan outlining the approach to safety and integration with the development processes. This should include a high-level argument describing the contribution of supporting documents[66];
- a description of subsystem dependencies and integration of safety analysis with configuration management; and
- a schedule of analysis, reviews and evaluations.

The submission of System Safety Review Reports is required at regular intervals (to be specified by the SSMP) to show compliance with the standard, and that all personnel involved in the development process have the skill and awareness to ensure compliance with the standard.

System Safety Evaluation Reports, made by an independent Evaluator, describe and evaluate steps taken to comply with the standard; a schedule for their delivery must be submitted for approval before development begins. Audits of safety activities are conducted by and independent auditor.

MIL-STD-882C In MIL-STD-882C, the developer is required to identify a management system for implementing system safety requirements, which must include mechanisms to monitor and assess system risks, and to eliminate such risks or minimise them to a level acceptable to the customer[67]. Task 102 provides scope for a detailed System Safety Program Plan (SSPP), to be agreed between the developer and the customer, to implement these requirements. Requirements are similar to those of the Def(Aust) 5679 SSMP, except that there is no explicit reference to a high-level argument. Particular requirements include descriptions of

- the relationships and chains of command within the development organisation;
- risk and hazard analysis and techniques to be used;
- analysis (including testing) techniques to be used; and
- training of users and incident reporting.

Task 103 includes extra requirements for the coordination of system safety management in the case where there are subcontractors.

Task 104 requires a program of safety reviews/audits to be formulated. In addition, safety progress summaries are produced periodically via Task 107.

---

[62] p39, Annex A
[63] part 1 table 1 pp22-24; part 2 table 1 p13; part 3 table 1 pp14-16
[64] part 1 pp14-15
[65] p24
[66] p46
[67] section 4.1.1

STANAG In STANAG 4404, at least two people must be familiar with the design, code, testing and operation of each software module[68]. Desk audits and peer reviews are required to help verify implementation of design requirements[69].

In STANAG 4452, the System Safety Program includes hazard tracking, software development plans, test plans, configuration management and quality evaluation plans. The developer must prepare evaluation criteria for the safety of the computing system and incorporate these into the quality evaluation plans[70]. Analysis Task 1 requires the development of plans for design reviews as required by the Managing Activity. The use of a safety requirements traceability matrix is mandated throughout the development.

Def Stan 00-56 In Def Stan 00-56, a Safety Programme Plan outlines the analytical and verification activities to be conducted in order to achieve the system safety requirements. The Plan contains the schedule and management structure for safety-related activities, and the safety requirements and targets. The Plan ascribes responsibilities to agents, including subcontractors[71]. Detailed guidance on the contents and structure of the Plan is given in part 2.

Safety Reviews are conducted by the Contractor as part of project design reviews, and are scheduled in the Safety Programme Plan. Detailed requirements are stated concerning the content of the Safety Reviews. Quality assurance activities for implementation of the Safety Programme Plan are conducted in accordance with Def Stan 05-91, unless alternatives are agreed with the Customer. Plans for such activities are documented in the Project Quality Plan[72].

An Independent Safety Audit is required for the two highest risk classes, and is described by an Audit Plan made by the Independent Safety Auditor[73]. A Data Reporting, Analysis and Corrective Action System (DRACAS) must be established in accordance with Def Stan 00-40 to review incidents arising during design, implementation, and in-service lifecycle phases[74].

The Independent Safety Auditor and safety program staff are required to have appropriate skills[75].

Def Stans 00-55 & 00-54 Def Stan 00-55 requires the production of a Software Safety Plan prior to the development of the software specification, showing the software planning and control measures to be employed. This plan must be updated at the commencement of each subsequent project phase. The initial version and any subsequent changes must be agreed with the customer. The Software Safety Plan should contain acceptance criteria for process data, justified against historical norms[76].

Def Stan 00-55 requires the conduct of software safety reviews as specified in the Software Safety Plan. These are carried out by the contractor, with the Independent Safety Auditor and customer invited to attend. The reviews consider the procurement of evidence for the Software Safety Case and recommend corrective action, with results documented in the Software Safety Records Log. The reviews approve any changes to the Software Safety Case, the Software Safety Plan and the Software Safety Records Log. Software Safety Audits are conducted by the Independent Safety Auditor according to a Software Safety Audit Plan, in accordance with Def Stan 00-56, with results recorded in a Software Safety Audit Report. As part of the DRACAS of Def Stan 00-56, any in-service anomalies in the operation of software must be recorded, and appropriate action undertaken to prevent an unsafe situation from occurring[77].

A Safety Programme Plan and an Audit Plan are produced for Def Stan 00-54, and constitute parts of the deliverables of the same name in Def Stan 00-56. Safety reviews are conducted in accordance with the Safety

[68] STANAG 4404 section 6.3 p6
[69] STANAG 4404 section 6.4 p6
[70] STANAG 4452 section 6 p5
[71] part 1 sections 4.2.2-4.3.2 p7, part 1 section 5.2 p10, part 1 section 6 p18
[72] part 1 Table 1 p8, part 1 sections 5.4-5.5 pp14-15
[73] part 1 Table 1 p8, part 1 section 5.3.4.2 p12
[74] part 1 section 4.5 p8, section 8 p33
[75] part 1 section 5.3.4.3 p12, section 5.3.5 p14
[76] 00-55 part 1 section 6 p7, section 7.4.6 p9
[77] 00-55 part 1 sections 10 & 11 pp10-11; section 42 pp37-38

Programme Plan. Safety reviews and safety audits are documented in the Safety Records Log. Design methods to be used are documented in the Design Plan and justified in the Safety Programme Plan. The Design Plan includes a V&V Plan, which identifies activities to be performed by the V&V team, including design analysis, simulation, and testing. The V&V team should review the results of formal analysis, simulations and physical tests. Safeguards against hazards in the development process are required, and the limitations of tools used identified in the Safety Records Log. As part of the DRACAS of Def Stan 00-56, any in-service anomalies in the operation of hardware must be recorded, and appropriate action undertaken to prevent an unsafe situation from occurring[78].

ARP ARP4754 recommends a general Certification Plan rather than a specific safety plan. Contents include:

- a functional and operational description of the system and the aircraft;
- summaries of the Functional Hazard Assessment and the Preliminary System Safety Assessment (Section 5.2);
- the proposed method of verifying compliance with certification requirements;
- a proposed schedule for deliverables; and
- the identification of personnel involved in certification activities.

ARP4754 suggests that the Applicant should submit, and obtain agreement on, plans for compliance with certification activities from the Certification Authority before the relevant development activities occur[79]. Process assurance activities, including reviews, are proposed in order to ensure that the necessary plans are developed and complied with.

DO-178B DO-178B suggests that the Applicant submit a Plan for Software Aspects of Certification to the Certification Authority for approval. This plan should provide timely guidance to personnel, and should state the Software Level to be satisfied by the software (see Section 5.3). Topics of guidance for the software planning process include:

- development standards, methods and tools;
- the coordination between software development and other processes, including safety activities;
- specific technical issues, including multiple version software and deactivated code; and
- provision for review of the plans as the project progresses[80].

Software quality assurance activities, including plans and reviews, are recommended to ensure that software standards are complied with, including the satisfaction of prerequisites for transition between software lifecycle processes.

IEC 61508 IEC 61508 requires the specification of technical and management activities that are necessary to achieve functional safety. The following should be specified[81]:

- the policy and strategy for achieving and evaluating safety[82];
- responsible persons and organisations;
- lifecycle phases to be applied;
- documentation structure;
- selected methods and techniques[83];

[78] 00-54 part 1 section 7 p7; section 8.6 p9; sections 12.3-12.4 pp12-13; section 12.7 p13; section 13.6 p15; section 15 p16
[79] ARP4754 p17
[80] pp15-18
[81] part 1 pp16-17
[82] part 1 pp45-48, part 2 p36

- functional safety assessment activities;
- procedures for issue resolution;
- staff competence[84];
- procedures for incident and operations analysis; and
- procedures for configuration management[85]

Additional plans for safety validation, installation and commissioning are required.

4.4 Project Lifecycle

Def(Aust) 5679 Def(Aust) 5679 is not prescriptive about the system development processes, but assumes a generic development lifecycle consisting of system definition and preliminary design, design development, implementation and post development activities, with revisions where necessary[86]. Development of the Safety Case is conducted in parallel with the system development lifecycle. While the exact relationship is to be specified by a management plan, a model of integration is proposed[87]. Any revisions made to part of one lifecycle must be reflected in corresponding parts of the parallel lifecycle.

MIL-STD-882C MIL-STD-882C does not require any particular system development lifecycle, and safety management and engineering tasks can be conducted at any time. However, it contains general guidance as to which tasks could be conducted at particular stages of a model system development lifecycle[88], including provisions for incorporating design changes. Detailed guidance is provided in Appendix B.

STANAG STANAG 4404 assumes a development lifecycle which includes conceptual design, preliminary design, detailed design, software coding and component building, unit or module testing, system and software integration testing, and modification and maintenance. Guidance is given on which of the design requirements are best performed at which stage of this development lifecycle[89].

STANAG 4452 assumes no particular project lifecycle, but each analysis or testing task gives guidance as to when that task should be conducted.

Def Stan 00-56 Def Stan 00-56 requires consideration of initiation, project definition, full development, design certification, production, in-service and disposal lifecycle phases. The safety program is planned, integrated and developed in conjunction with the system development[90]. Guidance is given on activities to be conducted in particular phases.

Def Stan 00-55 & 00-54 Def Stan 00-55 assumes a software development lifecycle consisting of the production of a software specification, the development of increasingly detailed software designs, coding, and testing and integrating the software. The Software Development Plan should describe these phases, their inputs and outputs, and the relationships between them, such as entry and exit criteria[91].

The development lifecycle of a custom circuit in Def Stan 00-54 includes a specification process, a development process and a verification process, and is documented in the Design Plan[92].

ARP ARP4754 assumes an iterative system development lifecycle which includes the specification of high-level

---

[83] part 2 p18
[84] Guidance on the qualifications of personnel is given in Annex B of Part 1.
[85] part 3 p10
[86] pp23-pp24
[87] p26
[88] Table 4 pA-11
[89] STANAG 4404 Table B-4 Annex B
[90] part 1 section 5.2 p10, section 10 p34
[91] 00-55 part 1 section 31.1.2 p22; part 2 section 32.1.1 p39
[92] 00-54 part 1 section 12.1.2 p11

functional requirements, the allocation of functions to systems, the development of the system architecture, the allocation of item requirements to hardware and software, and the system implementation. The safety assurance activities of ARP4761 are conducted in parallel with the system development[93], but the interaction is less structured than for Def(Aust) 5679. It is recognised that changes to the development should be reflected in a revision of relevant safety activity deliverables[94].

DO-178B DO-178B defines a software lifecycle to be performed within the overall system and safety lifecycle[95]. The development lifecycle phases include planning, requirements, design, coding and integration, although it is recognised that the phases might be applied iteratively. Other integral processes, including verification and quality assurance, are performed concurrently with the development lifecycle. It is recognised that particular stages of the software lifecycle may have transition criteria, to be specified in the plans for software development[96].

Safety-related information flowing from the system lifecycle to the software lifecycle includes system requirements allocated to software; software levels (see Section 5.3); design constraints and hardware definition. In particular, the system design determines the software safety requirements. In the opposite direction, information flow includes fault containment boundaries, identification and elimination of error sources, and software requirements and architecture. In particular, modifications to software need to be reflected in the system safety activities.

IEC 61508 IEC 61508 defines an overall safety lifecycle[97] comprising concept description; scope definition; hazard and risk analysis; allocation of safety requirements; design and development; integration; development of operational and maintenance procedures; safety validation; installation and commissioning; operation and maintenance; and decommissioning. More detailed lifecycles are provided for computer system and software development. A separate development lifecycle is not specified, although the lifecycles merge during the computer system and software development activities. For each stage of the lifecycle, detailed descriptions, inputs (or prerequisites) and outputs are given[98]. It is acknowledged that iteration is a vital part of the development process.

4.5 Post Development Processes

Def(Aust) 5679 Def(Aust) 5679 requires installation activities to be described in an installation plan and considered in the hazard analyses[99]. Commissioning tests are performed to demonstrate requirements after installation is complete.

Maintenance tasks must avoid violation of System Safety Requirements and modify the Safety Case where necessary. Special attention is drawn to compromising safety by overriding safety interlocks or modifying software.

Major system changes resulting from modification require production of a revised Safety Case[100].

MIL-STD-882C Appendix B of MIL-STD-882C describes tasks which could be performed during the operations and support phase. The tasks include evaluation of failure analyses and mishap investigations, review of procedures, monitoring results of field inspections or tests for deterioration of safety, and review of disposal plans[101].

STANAG In STANAG 4404, requirements applicable to the design and development phases are also applicable to the maintenance of software. Software patches are prohibited. A Software Configuration Control Board and a

[93] ARP4754 p14, p32. See also Appendix A.
[94] ARP4761 p12-13
[95] pp5-6, pp13-14
[96] p16
[97] part 1 pp18-21, part 3 p13
[98] part 1 pp22-24, part 2 p13, part 3 pp14-16
[99] section 10.1
[100] p40
[101] section 60.2.6, pB-7

Hardware Configuration Control Board approve any software or hardware changes respectively once baselines have been established, and configuration control is mandated[102].

In STANAG 4452, proposed design changes must be analysed for effects on safety-critical computing system functions[103]. Analysis Task 7, the Change Hazard Analysis, requires analysis of changes to software or requirements, and the results integrated into previously conducted analyses. Affected system documentation must be updated.

Def Stan 00-56 Def Stan 00-56 requires a Change Hazard Analysis in the event of system changes[104]. A new Safety Case is required when systems are modified, such as when functionality is added, technology is updated, or the system is used for a different purpose than originally envisaged[105].

Def Stans 00-55 & 00-54 Def Stan 00-55 requires software maintenance to be conducted according to a Software Maintenance Plan. Impact analysis must be conducted in order to assess any impact on safety and, in particular, determines the extent of required assurance activities for unchanged parts of the software. All changes to the software must be made according to the requirements of the standard and documented[106].

Def Stan 00-54 requires maintenance to be conducted according to a Maintenance Plan. Replacement of components that cause changes to the specification or performance of the hardware must be reflected in a redesign of the hardware according to the standard[107].

ARP ARP4754 identifies some typical installation assumptions and recommends that they be validated[108]. ARP4761 further recommends that requirements for installation design be derived during the Preliminary System Safety Assessment[109].

ARP4761 also recognises that some safety requirements will be allocated to maintenance tasks. ARP4754 requires that these are considered in the certification process and recommends validation of maintenance assumptions[110].

ARP4754 examines aircraft modification in detail[111] and considers introduction of new functions, replacement of systems, adaptation of existing systems to new aircraft types, and alteration of existing systems. Modifications generally require adherence to the guidelines of the standard. In particular, the existing safety assessment should be reviewed and necessary certification data compiled. Details of the technical arguments required are considered in Section 5.8.

DO-178B DO-178B requires modifications to software to be reviewed by analysis[112]. Analysis activities include review of the system safety assessment process and analysis of the modification impact, including data flow analysis, control flow analysis, timing analysis and traceability analysis. Areas affected by the changes should be reverified. If the software level is raised, the assurance activities should be reviewed for adequacy.

IEC 61508 IEC 61508 requires plans to be developed for installation and commissioning activities[113]. Plans must include the schedule, responsibilities, procedures, acceptance criteria and procedures for failure resolution. The activities must be conducted in accordance with the plans.

---

[102] STANAG 4404 section 6.1 pp5-6, section 16 p17
[103] STANAG 4452 Analysis Tasks 1 and 2
[104] part 1 Table 1 p8
[105] part 1 section 4.7 p9, part 2 section 4.7 p11
[106] 00-55 section 42 pp37-38
[107] 00-54 part 1 section 16 p16
[108] ARP4754 section 7.5.5
[109] ARP4761 Appendix B.3.3
[110] ARP4754 section 6.5, section 7.5.4
[111] ARP4754 section 11
[112] section 12.1.1
[113] part 1 p37, p39

A plan for maintenance and operation is also required[114]. The plan identifies routine actions and procedures to be carried out, including fault detection activities and safety audits, and documentation to be maintained (including records of incidents).

Modification and retrofit occur only under an authorised request and an impact analysis must be performed, including a revised hazard and risk analysis. All modifications that impinge on the functional safety of the system require a return to the relevant safety lifecycle phases[115].

[114] part 1 p35, pp40-42, part 2 p29, pp32-33, p54
[115] part 1 pp42-43, part 2 p35, part 3 pp29-30, p40

5. Technical Issues

5.1 Development Constraints

Def(Aust) 5679 Before development of the system begins, Def(Aust) 5679 stipulates that the Developer must submit a System Development Plan, detailing the design methods and techniques to be used. The Safety Working Group must agree to the plan.

The Customer, the Developer and the end users should agree on the requirements of the system to be developed, including the context in which the system should operate[116]. A subsequent System Functional Requirements Specification is required to support the Preliminary Hazard Analysis report[117].

A Component-Level System Design is required to support the System Hazard Analysis[118]. This describes the architecture of the system components and demonstrates how the components combine to achieve the system functions, using a structured approach, and formal modelling if appropriate. Safety-critical functions must be localised and isolated if possible. The use of software requires justification because of its likely complexity and relative unpredictability compared with physical systems.

Requirements for the design of the system include the use of structured design methods appropriate to the component being developed[119].

Custom hardware and software components must use structured design and development techniques and the design method must allow the assurance of safety in accordance with the required integrity level[120]. Software must be developed using sound software engineering principles and be subject to thorough testing[121]. The choice of programming language may be constrained by the safety requirements. Further implementation constraints specific to software and hardware are described in Section 5.5 and Section 5.6 respectively.

Human factors should be considered in the system development and some general guidance and requirements on operator interfaces is offered[122].

MIL-STD-882C MIL-STD-882C does not require the inclusion of design or specification requirements in the contract, but stipulates several general principles: design for minimum risk, incorporation of safety devices, and provision of warning devices[123]. For the severest hazard categories, sole reliance on safety and warning devices is prohibited[124].

Hazard analysis tasks 202 to 206 require a description of the physical and functional characteristics of the system and its components to support analyses. Several documents are required, including system requirements and design specifications, configuration item specifications, software requirements specifications and interface specifications. The methods used are not prescribed, but system block diagrams and functional flow diagrams are suggested.

STANAG STANAG 4404 contains detailed design constraints. Software must return hardware to a safe state when failure or unsafe conditions are detected. The system must be designed to perform under peak load conditions and return to a safe state when the safety kernel or other system components fail. Any battle shorts must be designed so that they cannot be activated inadvertently or without authorisation. Software must be designed for ease of maintenance. Safety-critical functions should be isolated from non-critical functions to the maximum extent practical, with the former implemented on a stand-alone computer if possible. Software patches

[116] p27
[117] p47
[118] pp51-52
[119] p29
[120] p28
[121] pp29-30
[122] p30
[123] section 4.4 p11
[124] section 4.6 p14

are prohibited[125].

There are requirements for the safe operation of the system on power-up initialisation, with the software performing a system-level check, and when power faults occur[126]. Detailed guidelines are given on the selection of CPUs. Requirements for the self-checking of software include the use of watchdog timers, memory checks, fault detection and isolation programs, and checks of testable safety-critical functions prior to performance of a related safety-critical operation[127]. Protection mechanisms are required to ensure load data integrity and to prevent unauthorised or inadvertent initiation of a safety-critical function sequence, or changes to software[128]. Constraints are given on the design of input-output interfaces[129].

In addition to the design constraints above, STANAG 4404 contains constraints on the software. Software design and code must be modular, with all modules having one entry and one exit point. Loops must have one entry point, and must exit to a single point outside the loop. Each safety-critical system function must have exactly one path leading to its execution. Unnecessary features, unused code and unused variables are prohibited. The use of halt or wait instructions within safety-critical code is prohibited. Files used for the storage of safety-critical data must be single-purpose and unique. Run-time boundary checks must be placed on arrays and indirect addresses when executing safety-critical functions. Unused memory must be initialised to a pattern which, if executed, causes the system to revert to a safe state. Variable naming requirements are given. The execution time of loops must be prevented from exceeding a maximum value. The results of a program should be independent of the duration of execution or the time of initiation of the execution[130].

STANAG 4452 requires the development of System Safety Design Requirements as part of Analysis Task 1. Design guidelines must be developed and implemented in order to reduce the risks identified in Analysis Task 1 to acceptable levels. Analysis Task 2 requires that code developers be provided with explicit safety-related coding recommendations. The number of safety-critical modules should be as low as possible with as little interaction with other modules as possible; this is verified by Analysis Task 3.

Def Stan 00-56 Def Stan 00-56 does not impose particular development constraints. However, design rules and techniques appropriate to each Safety Integrity Level must be determined prior to implementation of the system functions. These must be approved by the Project Safety Committee and the Safety Review, and the rationale for their choice must be recorded in the Safety Criteria Report[131]. Guidance is given on the appropriateness of formal specifications, structured design methods, coding standards, and the use of tools and compilers, for particular Safety Integrity Levels[132].

Def Stans 00-55 & 00-54 Def Stan 00-55 requires software to be developed according to the requirements of ISO 9001 and the guidance of ISO 9000-3. Software development planning should be conducted according to Def Stans 05-91 and 05-95. Risk analysis (relating to the success of the project, rather than safety) should also be conducted. A Code of Design Practice is required[133].

The choice of implementation language must be justified, and for the highest SILs must be high-level, strongly typed and block-structured, with a formally-defined syntax. Assembler language may be used in exceptional circumstances. Compilers must be validated, and all tools used must have sufficient safety assurance, commensurate with the reliance placed on the tool to develop safe software. Unreachable code may only remain in the application if it can be shown that the risks of leaving it are less than the risks of removing it. Detailed guidance is given on factors to be considered in coding[134].

[125] STANAG 4404 section 6.5 p6, section 7 pp6-8
[126] STANAG 4404 section 8 pp8-9
[127] STANAG 4404 sections 9-10 pp9-10
[128] STANAG 4404 section 11 pp10-11
[129] STANAG 4404 section 12 pp11-12
[130] STANAG 4404 sections 14-15 pp13-16, section 18 pp18-19
[131] part 1 section 7.3.3 p26
[132] part 2 table 3 pp36-37
[133] 00-55 part 1 section 20.1 p15; section 22.1 p16; section 23 p16; section 27 p18
[134] 00-55 part 1 sections 28-29 pp18-20; section 30.2 p20

In Def Stan 00-55, software diversity may be used for additional confidence in safety, and a discussion on risks and benefits is given[135].

Def Stan 00-54 requires hardware to be developed according to the requirements of ISO 9000. A Design Plan documents the development process, with individual attention paid to each custom circuit. The choice of physical implementation should be justified in the Safety Programme Plan[136].

ARP The ARP standards are intended to provide guidance on the engineering of complex aircraft systems. Before development of the system begins, ARP4754 recommends a Development Plan describing milestones of the development cycle, the organisational structure and the responsibilities of personnel.

ARP4754 provides guidance on the determination of captured and derived requirements, at the functional level, the system level, the item level and the hardware/software level[137].

An Architecture and Design Description is suggested to specify the high-level functionality of the system and gives sufficient detail to establish that this functionality will be achieved[138]. A number of architectural design techniques to improve safety is suggested.

DO-178B DO-178B suggests that system requirements include safety strategies and design constraints, including design methods such as partitioning, dissimilarity, redundancy or safety monitoring[139]. Software development plans should specify the prerequisites for transition between stages of the software development, and the methods, tools, programming languages and compilers to be used[140]. Software must be structured to assist the verification activities[141], but limited guidance on development is given otherwise. Particular emphasis is placed on traceability.

IEC 61508 IEC 61508 imposes a number of constraints on the development of computer-based systems[142] to enforce desirable development practices or assist assurance activities.

At commencement of development, a description of the system concept is required, as well as a definition of the system scope and structure. This is used as input to the initial hazard and risk analysis.

Specification of system and software safety requirements can be made independently of the development but the safety process merges with the development process during the computer and software design activities. Where high levels of integrity are required, computer-aided, semi-formal or formal safety requirements specifications are recommended[143].

In addition, combinations of hardware architecture and diagnostic coverage are mandated to achieve required levels of hardware reliability[144]. Extensive recommendations are made on design methods for systems to control design faults or hardware failures[145]. For high levels of integrity, semi-formal and formal design representations are recommended.

Requirements are also made of the software design, depending on the target safety integrity level of the software[146]. These include restrictions on software architecture and selection of software development tools. Suitable programming languages are highly recommended, including safe language subsets for higher assurance levels. Software design and coding standards are highly recommended, as well as modular development.

[135] 00-55 part 1 section 31 p21; part 2 section 31 pp37-38
[136] 00-54 part 1 section 10.2 p10; section 12.1.1 p11; section 12.6 p13
[137] ARP4754 pp21-24
[138] ARP4754 p20
[139] p6
[140] p17
[141] pp19-23
[142] part 1 pp29-30, part 2 pp14-16, part 3 pp17-19
[143] part 2 table B.1 p52; part 3 table A.1 p38
[144] part 2 section 7.4.5.4 p25
[145] part 2 section A.3 pp47-50, tables B.2 & B.3 pp53-54, table B.6 pp56-57
[146] part 3 sections 7.4.2-7.4.6 pp21-25 , tables A.2-A.4 pp38-39, table B.1 p42, tables B.7 & B.9 p44

Defensive programming is highly recommended for the higher integrity levels.

5.2 Hazard Analysis

Def(Aust) 5679 In Def(Aust) 5679, two hazard analysis tasks are performed with the intention of identifying component safety requirements.

Using the System Functional Requirements Specification, the Preliminary Hazard Analysis identifies possible accidents and the system hazards from which they might arise. A complementary list of System Safety Requirements is subsequently produced for consideration in the Component-Level System Design (CLSD)[147].

The system hazards of the Preliminary Hazard Analysis are systematically decomposed by a System Hazard Analysis using the CLSD to form component-level hazards. Component Safety Requirements (CSRs) are also specified and are independently checked to be same as the decomposition of the system safety requirements[148].

No techniques are mandated for the hazard analysis, although specific reference is made to event tree analysis and fault tree analysis.

MIL-STD-882C MIL-STD-882C provides detailed guidelines for Preliminary Hazard Analysis in Tasks 201 and 202. To begin, potential hazards are identified and compiled into a Preliminary Hazard List based on the system concept and past experience[149]. A more systematic analysis is then conducted by considering hazardous components, system interfaces, environmental constraints and potential malfunctions[150].

The Safety Requirements / Criteria Analysis (Task 203) relates the hazards of the Preliminary Hazard Analysis to the system design and identifies or develops design requirements to eliminate or reduce the hazards to an acceptable level. The analysis confirms that the safety requirements are satisfied by the design documentation, including operator procedures, or recommends changes to be made.

A Subsystem Hazard Analysis can be performed (Task 204), in which subsystem hazards are identified and analysed to be eliminated or reduced. The analysis shows that subsystem designs and implementations adhere to safety design criteria, system hazards are adequately controlled and no new hazards are introduced. A subsequent System Hazard Analysis is performed (Task 205) in which the whole system is analysed rather than the components. Special consideration is given to system interfaces and dependent subsystem failures.

There is also scope for human factors hazard analysis (Task 206), described in more detail in Section 5.7, and a health hazard analysis (Task 207) that considers hazards with long-term rather than immediate adverse health effects, for example due to chemical, biological or radioactive agents.

STANAG STANAG 4452 gives detailed hazard analysis requirements. A hazard tracking system is required throughout the development. The preliminary hazard analysis should be conducted in accordance with NATO standard AOP-15[151]. Analysis Task 1, the Computing System Requirements Hazard Analysis, determines design requirements that will eliminate or reduce the risk associated with system functions. A hazard category is assigned to each hazard, but a description of hazard categories is not given. Analysis Task 2, the Computing System Design Hazard Analysis, analyses the design and implementation of safety-critical functions in the computing system. The Interface Hazard Analysis of Analysis Task 4 is designed to ensure that hazards in one component are not propagated through the system. Analysis Task 7 is the Change Hazard Analysis, to be conducted when changes to the system are proposed.

Def Stan 00-56 Def Stan 00-56 requires a Preliminary Hazard Listing and Preliminary Hazard Analysis to be conducted. The Preliminary Hazard Listing identifies the main generic hazards and the accidents that they may

[147] p47
[148] p51
[149] p201-1
[150] p202-1
[151] STANAG 4452 section 6 p5

cause. The Preliminary Hazard Analysis evaluates the major hazards and accident sequences (together with probabilities) of the system by means of a HAZOP study in accordance with Def Stan 00-58, or a similar method agreed with the Customer. The hazards are assigned a preliminary probability target, and the system must be designed with these in mind[152].

A Hazard Log to track hazards and potential accidents must be established (see Section 4.2).

For the highest three risk classes, an iterative System Hazard Analysis must be conducted, using the definitions of functions and system components. This analysis includes Functional Analysis, Zonal Analysis, Component Failure Analysis, Operating and Support Hazard Analysis, and Occupational Health Hazard Analysis. Detailed requirements are given on contents[153].

In addition, a System Change Hazard Analysis is required if any changes are made to the system functions or components after design certification, or if any changes are made to the domain of operation. This requires changes to system safety program deliverables.

Def Stans 00-55 & 00-54 Since hazard analysis is a system-level activity, Def Stans 00-55 and 00-54 require hazard analysis to be conducted in accordance with Def Stan 00-56.

ARP ARP4754 provides a hazard analysis framework that is considered in more detail by ARP 4761.

An initial Functional Hazard Assessment (FHA) is recommended to identify failure conditions and their effects[154]. The assessment is carried out at two distinct levels for the aircraft and aircraft systems. The aircraft level FHA is a qualitative assessment of the basic functions of the aircraft, with the aim of classifying the failure conditions associated with these functions according to severity. The system level FHA is similar, but considers failures after functions have been allocated to systems by the design process. Analysis, such as fault tree analysis, is used to relate the functional failures at each level.

The FHA is used as input to the Preliminary System Safety Assessment (PSSA), which completes the list of failures, identifies system safety requirements and demonstrates how the proposed system architecture can reasonably be expected to meet these requirements. Hazard analysis techniques are used to relate system level hazards to failures of specific hardware or software units and safety requirements are derived for those units.

During and following unit development, a System Safety Assessment (SSA) is performed to integrate unit level analyses and evaluate safety of the implemented system. Evidence of the software development process is examined to ensure that derived software unit safety requirements are satisfied. Actual failure rates of hardware components are estimated and combined in accordance with the system design to estimate likelihoods of system and aircraft failures.

The safety assessments are supplemented by Common Cause Analysis, which is applied at all levels to determine and verify physical and functional independence requirements.

The avionics standards are quite specific about suitable hazard analysis techniques to apply and suggest Fault Tree Analysis, Dependence Diagrams or Markov Analysis and Failure Modes Effects Analysis. ARP4761 provides an extensive description of the techniques in appendices, including a fully worked example in Appendix A.

DO-178B Hazard analysis is a system-level activity, so it lies outside the scope of DO-178B.

IEC 61508 During concept definition, likely sources of hazards and information about them are identified. This is augmented by consideration of initiating events, external events and hazardous subsystems during the Overall Scope Definition. In this context, a hazard analysis is performed which identifies hazards, hazardous events and the event sequences that relate them. Consideration must be given to the elimination of hazards, and human

[152] part 1 sections 7.2.2 - 7.2.3 pp19-21
[153] part 1 sections 4.4.2 - 4.4.3, pp7-8; section 7.2.4 pp21-22
[154] ARP4761, p16

factors must be taken into account. Probabilities of hazardous events are calculated, and may be quantitative or qualitative[155]. Hazard analysis is continued into the development lifecycle to ensure that emergent hazards are identified.

Safety functions of the overall system necessary to meet overall safety requirements and reduce hazard risk are then identified. Each overall safety requirement is allocated to the designated computer system, taking into account other technological systems and external risk reduction. The designated system safety requirements are then specified, taking into account all relevant modes of operation[156]. Software safety requirements are derived from these[157].

5.3 Risk and Integrity Assessment

Def(Aust) 5679 The Def(Aust) 5679 risk and integrity assessment is based on the concept of development integrity levels. Probabilistic interpretations of risk are explicitly excluded because of the scope for error or corruption in the quantitative analysis process, and because it is currently impossible to interpret or assess low targets of failure rates for software or complex designs[158].

For each potential accident identified by the Preliminary Hazard Analysis, a severity category (catastrophic, fatal, severe, and minor) is allocated, based on the level of injury incurred. Sequences of events that could lead to each accident are identified, and assigned a probability where estimation is possible[159].

One of seven Levels of Trust (LOTs) is allocated to each system safety requirement, depending on the severity category of the accidents which may result from the corresponding system hazard. The LOT may be reduced if each accident sequence can be shown to be sufficiently improbable. Each LOT defines the desired level of confidence that the corresponding system safety requirement will be met.

Next, one of seven Safety Integrity Levels (SILs) is assigned to each Component Safety Requirement (CSR), indicating the level of rigour required to meet the CSR. By default, the SIL level of the CSR is the same as the Level of Trust of the system safety requirement corresponding to the CSR. However, the default SIL may be reduced by up to two levels by implementing fault-tolerant measures in the design to reduce the likelihood of the corresponding hazard. As the standard prohibits allocation of probabilities to hazards, this is based on a qualitative argument[160].

MIL-STD-882C Risk assessment in MIL-STD-882C is based on determination of acceptable risk reduction.

An initial level of risk is determined during the Preliminary Hazard Analysis from the potential hazard consequence severity and the likelihood that the consequence will arise. This initial assessment assumes that no measures are taken to eliminate or reduce the system hazard.

The standard does not allocate levels of integrity or trust in the system, but where risk levels are determined as too high, measures are required to reduce the likelihood of hazard occurrence. During the Safety Requirements / Criteria Analysis, the risk is reassessed in light of the design decisions and an estimate of residual system risk is made[161]. No guidance is provided on whether the assessment of likelihood should be quantitative.

The determined risk level ranges from intolerable to acceptable. Different levels of intermediate risk may be accepted by the appropriate authority, with higher authority required to accept higher levels of residual risk[162].

Since the likelihood of software failure cannot be estimated, a different approach to risk assessment is suggested

[155] part 1 section 7.2.2 p26, section 7.3.3 p26; section 7.4 pp27-28
[156] part 1 pp30-31, part 2 pp14-15
[157] part 3 p17
[158] p49
[159] p48
[160] pp55-58
[161] p203-1
[162] pA-7

for software hazards, based on the notion of software control categories. In this case, the severity of the hazard is combined with the level of control exercise in the function. The resulting level of risk determines the resources to be applied in eliminating the software hazard[163]. However, no concrete guidance on acceptable measures to take for each risk level is provided.

STANAG STANAG 4452 requires the developer to identify a hazard category and a Software Control Category associated with each Safety Critical Computer System Function[164]. However, their use is not mentioned subsequently.

Def Stan 00-56 In Def Stan 00-56, accidents are classified as belonging to one of four severity categories and one of six probability categories. The correspondence between probability categories and actual probabilities must be stated and approved by the Independent Safety Auditor. Using these classifications, a risk class is assigned to each accident using a matrix approved by the Independent Safety Auditor before hazard analysis activities begin[165].

For systematic (as opposed to random) failures, the SIL (or actual data if available) determines the minimum failure rate that may be claimed of the function developed according to the SIL; such failure rates must be approved by the Independent Safety Auditor. Accidents in the highest risk class (A) are regarded as unacceptable, while probability targets are set for accidents in the next two risk classes (B and C). Accidents in the lowest risk class are regarded as tolerable. Accident probability targets are regarded as having a systematic and a random component. The consideration of accident probability targets and accident sequences determines the hazard probability targets with systematic and random components. These hazard probability targets must be approved by the Independent Safety Auditor[166].

The random element of each hazard probability target is apportioned to the lower level functions, providing hazard probability targets for these lower level functions. For the systematic element of each hazard probability target, the system function is implemented according to the SIL of the most severe resulting accident. Additional independent implementations are made according to a SIL depending on both the accident severity and the failure probability of the first function. (Presumably this failure probability is determined by the claim limit of the above paragraph; however, this is not stated.) Rules are given for implementing high-level functions of a certain SIL by combinations of independent sub-components of lower SILs[167].

A Safety Compliance Assessment is conducted using techniques such as Fault Tree Analysis. If the hazard probability target cannot be met for risk class C, then risk reduction techniques such as redesign, safety or warning features, or special operator procedures must be introduced. If risk reduction is impracticable, then risk class B may be used with the approval of the Project Safety Committee[168].

Def Stans 00-55 & 00-54 Since risk assessment is a system-level activity, Def Stans 00-55 and 00-54 require risk assessment to be conducted in accordance with Def Stan 00-56. Def Stan 00-55 explicitly mentions that software diversity may, if justified, reduce the required SIL of the application being developed. A discussion is given on the risks and benefits of software diversity[169].

ARP The avionics risk assessment framework is based on development assurance levels, which are similar to the Def(Aust) 5679 safety integrity levels.

Each functional failure condition identified under ARP4754 and ARP4761 is assigned a Development Assurance Level based on the severity of the effects of the failure condition identified in the Functional Hazard Assessment[170]. However, the severities correspond to levels of aircraft controllability rather than direct levels of

[163] pA-8
[164] STANAG 4452 section 6 p5, Appendix A; also mentioned in Analysis Tasks 1 and 2.
[165] part 1 sections 7.2.3.1-7.2.3.2 and 7.2.4.2-7.2.4.3 pp20-21, sections 7.3.1-7.3.2 pp23-25
[166] part 1 sections 7.2.3.1-7.2.3.2 and 7.2.4.2-7.2.4.3 pp20-21, section 7.4.3 p27, sections 7.4.6-7.4.7 pp28-29
[167] part 1 sections 7.2.3.1-7.2.3.2 and 7.2.4.2-7.2.4.3 pp20-21, section 7.4.3 p27, section 7.4.8 pp29-31
[168] part 1 section 7.5 pp31-33
[169] 00-55 part 1 section 31 p21; part 2 section 31 pp37-38
[170] ARP4754 p34

harm. As a result, the likelihood of accident sequences is not considered in the initial risk assessment.

The Development Assurance Level of an item in the design may be reduced if the system architecture provides multiple implementations of a function (redundancy); isolates potential faults in part of the system (partitioning); provides for active (automated) monitoring of the item; or provides for human recognition or mitigation of failure conditions. Detailed guidance is given on these issues[171]. Justification of the reduction is provided by the Preliminary System Safety Assessment.

Development assurance levels are provided with equivalent numerical failure rates[172] so that quantitative assessments of risk can be made. However, it is acknowledged that the effectiveness of particular design strategies cannot always be quantified and that qualitative judgments are often required[173]. In particular, no attempt is made to interpret the assurance levels of software in probabilistic terms. Like Def(Aust) 5679, the software assurance levels are used to determine the techniques and measures to be applied in the development processes.

When the development is sufficiently mature, actual failure rates of hardware components are estimated and combined by the System Safety Assessment (SSA) to provide an estimate of the functional failure rates. The assessment should determine if the corresponding development assurance level has been met. To achieve its objectives, the SSA suggests Failure Modes and Effects Analysis and Fault Tree Analysis, which are described in the appendices of ARP4761.

DO-178B Each failure condition in DO-178B is categorised according to the severity of its effect. Criteria include the capability of the aircraft and the ability of the crew to cope with an increased workload. The contribution of software to potential failure conditions determines its Software Level[174], which specifies the rigour to which the software should be developed. The software levels are similar to the SILs in Def(Aust) 5679. Guidance is given on architectural strategies that may limit, detect or react to errors in software, with the result that the software level may be reduced[175].

It is suggested that software developed to a higher level than necessary will ease the addition of system functionality later in the software development, as substantiating a higher software level later is likely to be more difficult.

IEC 61508 IEC 61508 risk assessment is based on a combination of risk reduction and integrity levels.

The necessary risk reduction is determined by the hazard and risk analysis for each identified hazardous event[176]. The required risk reduction is determined from the actual risk of equipment under control, assuming that no safety measures are taken, and the tolerable level of risk. No method for this is mandated but guidance is provided in Part 5. An overall Safety Integrity Level is assigned in order to satisfy the required risk reduction. Acceptable limits of numerical failure rates corresponding to integrity levels are provided[177].

Safety Integrity Levels are allocated to the safety functions in the designated computer system, taking into account other technological systems, external risk reduction facilities and the independence of these systems. Ideally, the allocation of risk should be justified by a quantitative argument[178] but it is recognised that calculation of probabilities is often not possible, and that in these cases qualitative judgments must be made instead[179].

The resultant Safety Integrity Levels are used to set numerical targets for failure rates of the hardware system and processes to be applied in software development.

---

[171] ARP4754 pp24-31
[172] ARP4761 p14
[173] ARP4754 p25
[174] pp8-11
[175] pp8-10, pp63-65, Annex A
[176] part 1 p29
[177] part 1 p33
[178] part1 p31
[179] part1 p33

5.4 Design Assurance

Def(Aust) 5679 Component design assurance in Def(Aust) 5679 provides assurance that each component design satisfies its Component Safety Requirements (CSRs) to the appropriate Safety Integrity Level (SIL).

Assurance is proof-based and the level of assurance determined by the rigour of proof. In particular, component design assurance is achieved if

- the specifications of CSRs;
- the model of the component design; and
- the verification that the component design meets all of its CSRs,

are all sufficiently formal according to each CSR's SIL.

In addition, once components are implemented, safety tests must be conducted against the CSRs[180].

MIL-STD-882C The MIL-STD-882C hazard analysis tasks 203 to 205 require production of evidence that the design specifications satisfy design safety criteria. The safety criteria are determined from the system hazards as well as generic design guidelines. Safety tests are prepared throughout the development in accordance with Task 302 and are conducted by Task 401, along with other verification techniques such as analysis, functional mockups and simulation.

STANAG STANAG 4404 requires that a system safety engineering team validate and verify that safety design requirements have been met. Verification of the restoration of safety interlocks removed during tests is required[181].

Analysis Task 2 of STANAG 4452 requires verification that the design implements the system level safety requirements.

Def Stan 00-56 Def Stan 00-56 requires a Safety Compliance Assessment to be conducted during design in order to show compliance with system safety requirements[182].

Def Stans 00-55 & 00-54 Def Stan 00-55 requires that requirements traceability be maintained throughout the development. The software requirements should be checked to be self-consistent and unambiguous. The use of structured design methods, and formal methods for specification and design, is required for the highest SILs. The specification must be validated using formal arguments or executable prototyping, with detailed requirements and guidance given. Factors such as size and complexity must be taken into account in the software design, and detailed guidance is given various factors such as fault-tolerance. For the highest SILs, the software design must be syntax and type-checked using an appropriate tool. For such SILs, the internal consistency of each development process output and refinement from the previous output must be verified formally, and performance modelling conducted. The proof obligations, formal arguments and performance modelling must be reviewed for correctness and completeness by the V&V Team. Any anomaly discovered must be corrected or justified, and the possibility of similar anomalies considered. Detailed requirements and guidance concerning development of the software design are given[183].

In Def Stan 00-54, the rigour of application of design assurance techniques is SIL-dependent. The hardware requirement is checked to be self-consistent, unambiguous and complete, with development proceeding only when issues in this respect have been resolved. A formal specification language must be used to specify the design, and justified in the Safety Programme Plan. Tools used must be selected according to the criteria for NDIs (see Section 5.8). The specification must be checked to be consistent and unambiguous using analytic means, and

[180] p63
[181] STANAG 4404 section 6.6 p6, section 7.5 p7
[182] part 1 section 4.4.5 p8, section 7.3.3 p26, section 7.5 pp31-33
[183] 00-55 part 1 section 26.2 p18; sections 32.2-32.5 pp22-25; section 33 p26; sections 34-36 pp26-30

must be shown to satisfy the safety requirements. Correspondence between the specification and design must be demonstrated by analytic means. Traceability of implementation to requirement must be maintained throughout the development, and a representative set of simulation results obtained at all stages of development. Simulation and physical test coverage should be as specified in the Safety Programme Plan. Any anomaly discovered must be corrected or justified[184].

ARP The avionics standards provide design assurance through a mixture of validation and verification processes performed throughout development. Validation is the process of assuring that the identified requirements are sufficiently correct and complete[185] while verification determines whether each level of implementation meets its specified requirements[186].

ARP4754 gives detailed guidance on the validation of requirements, with checklists given for correctness, completeness and various assumptions concerning the environment, interfaces, reliability of components, production, installation and maintenance. It is assumed that validation occurs throughout the development lifecycle[187]. Validation techniques include traceability, analysis, testing, and comparison with similar systems in service. The Development Assurance Level of the function determines the level of validation of a function[188].

Detailed guidance on the verification that each level of the implementation meets its requirements is also given. Verification methods include inspection, reviews, analysis, testing and comparison with similar systems in service. The Development Assurance Level of the system or item determines the level of verification activities.

DO-178B DO-178B gives general guidance on design assurance. For low Software Levels some activities need not be satisfied, whereas for high Software Levels some activities should be satisfied with independent review. High-level requirements should comply with system requirements. The requirements and software architecture should be traceable, verifiable and consistent. Derived requirements should be defined and analysed for consistency with high-level requirements. Control flow and data flow should be monitored. Some software verification activities are also relevant to design assurance. In particular, reviews of the integration process are conducted and tests for software/hardware integration are defined[189].

IEC 61508 IEC 61508 defines requirements for both validation and verification.

Validation activities are conducted in accordance with validation plans for the software, computer system and overall system to ensure that safety functions meet their requirements (see Section 4.3).

Verification of the computer hardware design is conducted according to Part 2 to ensure that derived computer system safety requirements satisfy the allocated system safety requirements. Consistency and completeness of the architecture design is verified, along with satisfaction of the safety requirements. Techniques for verification and validation are recommended, particularly testing, with stronger recommendations made for higher safety integrity levels[190].

While Part 3 defines requirements for software assurance, some design verification activities are performed, including programmable electronics integration and software system testing (validation)[191]. Testing techniques are recommended depending on the safety integrity level.

5.5 Software Assurance

Def(Aust) 5679 Once a software component has been designed and implemented, implementation assurance is conducted. It must be shown that:

[184] 00-54 part 1 section 12.2 p12; sections 13.2-13.6 pp14-15
[185] ARP4754 p38
[186] ARP4754 p51
[187] ARP4754 p38
[188] ARP4754 p49
[189] section 5.2.2 pp20-21; sections 6.3.1-6.3.3 pp27-28; sections 6.3.4 & 6.4 p29
[190] part 2 section 7.7 p34, section 7.9 pp35-37, table B.5 p55
[191] part 3 pp25-29, pp39-40, pp42-43

- the programming language is of a high level, preferably a safe subset;
- analysis of program control flow, information flow and data use has been conducted;
- the specification of the code is sufficiently formal;
- verification of the code against CSRs is sufficiently formal; and
- the code has been tested against expected behaviour according to suitable coverage criteria[192]

according to the SIL required of each CSR.

MIL-STD-882C MIL-STD-882C contains no specific software implementation assurance requirements, although Task 401 requires safety verification procedures to be conducted. Testing is mentioned as a possible method.

STANAG STANAG 4404 requires the use of static and dynamic analysis and debugging tools to provide software assurance[193]. Test coverage should be as complete as possible, and must consider input failure modes, input data rates, boundary testing, regression testing, operator interface testing and stress testing under peak loading conditions[194].

Analysis Task 3 of STANAG 4452 requires the analysis of program code and its interfaces to the system for faults that could contribute to hazards. Some suggestions as to appropriate subtasks are made, and appropriate techniques are mentioned in the appendix. Analysis Task 8 requires planning and implementation of safety testing of the software. The software must respond correctly to failures and overload conditions and performs no extraneous functions.

Def Stan 00-56 Def Stan 00-56 requires that a test program for safety features be implemented[195]. For particular Safety Integrity Levels, static and dynamic analysis techniques and independent testing are recommended[196]. Further software assurance is conducted in accordance with Def Stan 00-55.

Def Stans 00-55 & 00-54 Def Stan 00-55 requires the planning for verification and validation to include acceptance criteria for each item of software. Required verification methods include static analysis (such as control flow analysis, language subset analysis, complexity analysis, semantic analysis and information flow analysis) and dynamic testing. Formal verification of correctness is required for the highest SILs. The static analysis and formal verification must be conducted or reviewed by the V&V team. For the highest SILs, object code must be verified by static analysis, formal proof, the use of a formally verified compiler, or testing. Any anomaly discovered must be corrected or justified, and the possibility of similar anomalies considered[197].

Def Stan 00-55 gives detailed requirements and guidance on testing. Before testing, the test scope must be documented in the Software Verification and Validation Plan. The V&V Team must review the test specification. Tests must be conducted under configuration control. Discrepancies between expected and actual outputs must be justified. Test coverage criteria are given for the highest SILs. Integration and system tests should also be conducted. In particular, system tests should be designed from the software requirements and specification, and tests at the extremes of performance requirements should be conducted. The customer must perform acceptance testing prior to acceptance[198].

ARP Software assurance lies outside the scope of the ARP standards.

DO-178B DO-178B details guidelines for software verification. For low Software Levels some activities need

[192] pp64-65
[193] STANAG 4404 section 6.6 p6
[194] STANAG 4404 section 17 pp17-18
[195] part 1 section 4.5 p9, section 9 pp33-34
[196] part 2 table 3 pp36-37
[197] 00-55 part 1 section 24.4 p17; section 26.2 p18; sections 36.5-36.7 pp30-32
[198] 00-55 part 1 section 37 pp32-35; section 39 p36

not be satisfied, whereas for high Software Levels some activities should be satisfied with independent review. The standard recommends that high-level requirements should be shown to satisfy the system requirements, low-level requirements should be shown to satisfy the high-level requirements and the system architecture generate high-level software requirements. All requirements should be shown to be correct, complete and unambiguous. The software architecture should be shown to be compatible with high-level requirements and the target computer. The source code should comply with low-level requirements. The source code should be checked for correctness, with issues mentioned including stack usage, data corruption and exception handling. The integration process should be reviewed and analysed for completeness and correctness[199].

DO-178B places much emphasis on software testing, with detailed guidance given. Testing of requirements should include both normal and abnormal range test cases. Testing should cover hardware/software integration, software integration and low-level testing. Test coverage analysis should be performed to provide assurance that adequate testing has been conducted, with dead code removed. The use of formal methods is not mandated, but is suggested as a software verification method complementary to testing[200].

IEC 61508 Part 3 defines requirements for software assurance. Verification activities are planned during development, and consider whether or not the software architecture fulfils the software safety requirements; the software system design satisfies the software architecture; the module design fulfils the software system design; and the code conforms to module design[201]. Designs and requirements are checked for feasibility, testability, readability and safe modification, and are verified with respect to appropriate test specifications. Data are also verified. A number of generic verification techniques is nominated with recommendations for use dependent on the safety integrity level. Formal proof is highly recommended for the highest level of integrity.

Testing is performed at a number of levels, including module, software integration, programmable electronics integration and software system testing (validation)[202]. Testing techniques are recommended depending on the safety integrity level.

5.6 Hardware Assurance

Def(Aust) 5679 Non-custom hardware components need to be analysed for safety. Design of custom hardware components must be expressed using a well-known hardware description language and supported by the use of a reliable computer-aided design (CAD) tool[203].

Hardware design must be formally verified if the SIL requires. In addition, hardware must be tested and static timing analysis applied[204]. However, there is no physical reliability assessment for hardware.

MIL-STD-882C MIL-STD-882C contains no specific hardware implementation assurance requirements, although Task 401 requires safety verification procedures to be conducted. Testing is mentioned as a possible method.

STANAG Hardware assurance is outside the scope of STANAG 4404. Analysis Task 2 of STANAG 4452 requires hardware analyses to be conducted, but no details are given.

Def Stan 00-56 Def Stan 00-56 recommends the use of static analysis techniques, independent testing and computer simulation of hardware components for particular Safety Integrity Levels[205]. Further hardware assurance is conducted in accordance with Def Stan 00-54.

Def Stan 00-55 & 00-54 In Def Stan 00-54, design analysis, simulation and physical testing activities must be conducted, with the chosen method (including use of CAD tools) and extent of cover justified in the Safety

[199] pp27-29
[200] pp62-63
[201] part 3 pp31-35, p41
[202] part 3 pp25-29, pp39-40, pp42-43
[203] p29
[204] pp66-67
[205] part 2 table 3 pp36-37

Programme Plan. Components susceptible to random failures must be analysed using Failure Modes Effects Criticality Analysis and Fault Tree Analysis. Traceability of implementation to requirement must be maintained throughout the development, and a representative set of simulation results obtained at all stages of development. Simulation and physical test coverage should be as specified in the Safety Programme Plan. Any anomaly discovered must be corrected or justified[206].

ARP Hardware assurance lies outside the scope of the ARP standards.

DO-178B Hardware assurance lies outside the scope of DO-178B. However, a standard is being drafted by RTCA to recommend verification measures to show that the hardware implementation meets the required Development Assurance Levels[207]. Random hardware failures are currently considered in the system safety assessment process.

IEC 61508 Part 2 defines requirements for hardware assurance with attention focussing on random hardware failures. Probabilities of random hardware failure are calculated, taking into account the architecture of the system, including any common cause failures. These are used to determine if the target safety integrity levels have been met, with techniques suggested including fault tree analysis. Detailed guidance is given on diagnostic tests, including integration tests, that may be conducted for both random (Annex A of Part 2) and systematic (Annex B of Part 2) failures.

Detailed guidance is given on maximum safety integrity levels that may be claimed, depending on architecture, fault tolerance and diagnostic coverage. A quantitative analysis is always required for the highest level of integrity.

5.7 Human Factors

Def(Aust) 5679 Def(Aust) 5679 considers the human operators of the system to be part of the system, and stresses the importance of adequate training, the definition of standard and emergency procedures, and human factors analysis in the design and implementation of the system. In particular, some constraints relating to human-computer interaction are given. Safety Integrity Levels are defined for operator procedures in the same way as for other components. Assurance of compliance at the implementation level is provided by the level of operator skill and training[208].

MIL-STD-882C MIL-STD-882C advocates the use of procedures and training, with a formal proficiency certification process to be agreed between the Developer and the Customer, in cases where hazards cannot be adequately reduced through design selection or the use of safety and warning devices[209].

In the Subsystem Hazard Analysis (Task 204), humans are considered as components. Task 205 requires the consideration of human errors in System Hazard Analysis. In addition, Task 206 requires an Operating and Support Hazard Analysis, which includes hazard analysis for human operational procedures. This involves the identification of potential hazards and the implementation of actions or procedures to eliminate or reduce them, including use of safety and warning devices and operator training. This hazard analysis should be performed for system installation, commissioning and decommissioning as well as normal operation.

STANAG STANAG 4404 provides some guidelines on human factors. Identification of items used in simulations should be clear. Software control of critical functions should have feedback mechanisms which indicate the function's occurrence[210]. User interfaces must be designed so that the operator may abort execution of the software with a single action, and have the system revert to a safe state. Two or more unique operator actions are required to initiate a potentially hazardous sequence of functions. Safety-critical operator displays must be concise and unambiguous. Software must provide the operator with feedback on entries made, and provide status

[206] 00-54 part 1 section 2.2 p4; section 12.5 p13; sections 13.5-13.6 p15
[207] ARP4761 p21
[208] section 6.4.4 p30; section 17.7 pp67-68
[209] section 4.4.4 p11
[210] STANAG 4404 section 7.11, section 7.13 pp8-9

reports on action taken on entries. Signals are required to alert the operator to unsafe situations[211].

Analysis Task 2 of STANAG 4452 requires the preparation of user and operator manuals. Analysis Task 5 requires the developer to conduct a user-interface hazard analysis, relating the results of other hazard analyses to operator functions and displays. Potential operator errors must be analysed, operator manuals reviewed, and safety controls or warning devices implemented where appropriate. Detailed guidelines are given in the appendix.

Def Stan 00-56 Def Stan 00-56 advocates the use of training and operating procedures in risk reduction. Risk assessment requires the apportionment of failure rates to operators. These failure rates should be based on experience of similar tasks performed in similar situations[212].

Def Stans 00-55 & 00-54 Def Stan 00-55 requires the production of user manuals[213]. Def Stan 00-54 provides no guidance on human factors.

ARP Human factors receive limited attention in the ARP standards, although any assumptions generated about operational use are recommended for validation[214].

DO-178B Human factors are outside the scope of DO-178B.

IEC 61508 IEC 61508 suggests the specification of procedures for the training of operations staff; the training of staff in diagnosing and repairing faults and in systems testing; and the retraining of staff at periodic intervals[215]. Human factors should be considered in the hazard analysis and design[216].

5.8 Non Development Items

Def(Aust) 5679 Def(Aust) 5679 provides requirements and guidance for both Non Development Systems and Non-Development components[217].

Non development components require full design and implementation assurance to be assigned a SIL of S_3 or higher. Components developed in accordance with other safety standards may be assigned a level up to S_2 if the Auditor approves and evidence is provided that component specifications meet the derived component safety requirements. Otherwise, components may only be rated S_0.

Non-development systems must be developed to a safety standard. The processes of Def(Aust) 5679 must still be followed as far as is possible, including production of the Safety Case. However there is more room for discretion by the Auditor and Evaluator.

MIL-STD-882C MIL-STD-882C notes the difficulties presented by NDIs[218]. It recommends tailoring the safety program to incorporate management and assessment only for small NDIs through Tasks 101 and 301. For larger NDIs, a plan is recommended (Task 102), along with a safety working group (Task 105) and safety requirements/criteria analysis (Task 203). General consideration is given to the assessment of any documentation or operational evidence and to performing additional hazard analyses as necessary.

STANAG Analysis Task 8 of STANAG 4452 requires that commercial or government-furnished software be analysed and tested unless specifically excluded by the Managing Activity.

Def Stan 00-56 Def Stan 00-56 requires the production of a Safety Case for NDIs[219]. Detailed guidance on the

[211] STANAG 4404 section 13 pp12-13
[212] part 1 section 7.5.3 pp31-32
[213] 00-55 section 41 p37
[214] ARP4754 p44
[215] part 1 p17, part 2 pp32-33, p54
[216] part 1 section 7.4.2.3 p27, part 2 p30
[217] section 11
[218] section 60.5 pB-8
[219] part 1 section 4.7 p9

retrospective application of the standard is given in Annex D of part 2. In particular, a Safety Programme Plan, a Project Quality Plan, a Project Configuration Management Plan and a Hazard Log should be established. Existing safety analysis information, including service histories, should be examined for deficiencies, and augmented where necessary by hazard analyses, a System Criteria Definition, risk estimation, and a Safety Compliance Assessment.

Def Stans 00-55 & 00-54 In Def Stan 00-55, the use of previously developed software in a new or modified system must be shown not to adversely affect the safety of the new system. Reverse engineering, verification and validation activities are required for any software not produced according to the standard; this is likely to require access to source code. The extent of reverse engineering may be reduced if other assurance activities have been conducted or the in-service history of the software is sufficiently reliable, and detailed guidance is given on this matter. In particular, quantified error rates and failure probabilities for the software may be taken into account[220].

Def Stan 00-54 notes the widespread use of NDIs in hardware development. An NDI may only be used if evidence of its integrity can be gathered from the process used in its design and production, its source of supply, and its service history. The argument used is qualitative. The NDI must have been supplied with a comprehensive specification. Safety analysis is required to show that the item is not used outside the limits documented in the specification. Evidence is required of comprehensive testing that the item operates in accordance with its specification. Any faults attributed to the NDI must be recorded in the Safety Records Log, together with measures taken to prevent further occurrence of the fault. Any modifications to NDIs must be made in accordance with the standard. The configurations of all NDIs must be recorded[221].

ARP ARP4754 considers NDIs in relation to the modification of aircraft[222]. In particular, the problems of altering a legacy system and integrating a system with a different aircraft type are examined. In general, the certification data necessary to support the safety assessment are required. Credit may be sought for previous assurance activities if the system or aircraft is traceable to the certification data. Otherwise, the applicant should identify and substantiate the assumptions necessary to support the assessment. If it is unavailable, certification data may be generated by reverse engineering or from an analysis of the service history.

DO-178B DO-178B discusses use of existing software in new aircraft and software whose data does not satisfy the guidelines of the standard[223]. Certification data should be reviewed and upgraded to determine satisfaction of the safety assessment and verification activities. Reverse engineering may be employed if data are not available. The service history may be used provided the configuration can be identified and an analysis confirming relevance of the service history can be provided. Some estimate of the software reliability is also required based on length of service period and records of operational errors.

IEC 61508 IEC 61508 requires that, if standard or previously used components are to be used, they shall be clearly identified and the suitability justified[224]. Justification may be derived from operation in a similar application or subjection to the same verification and validation procedures. The constraints of the previous environment(s) should also be evaluated.

Acknowledgements

This research was funded by the Australian Defence Acquisition Organisation under Contract CA38809. We wish to thank Tony Cant and Peter Lindsay for their helpful comments.

[220] 00-55 part 1 section 30 pp20-21
[221] 00-54 part 1 section 11 pp10-11
[222] ARP4754 section 11
[223] sections 12.1.2, 12.1.4
[224] part 2 p31, part 3 p22