The Swedish Defence Materiel Administration´s Handbook on Software in Safety-Critical Applications

H ProgSäk E 2018

FMV

Öppen/Unclassified **BESLUT**

Datum	Diarienummer	Ärendetyp
2019-11-14	19FMV5884-1:1	Beslut
	Dokumentnummer	Sida
	ange	1(1)
Giltig t.o.m.	Upphäver	
ange	ange	

Beslutande
Gustaf Fahl

Föredragande
Svante Wåhlin

Approval of Handbook on Software in Safety-Critical Applications 2020 (H ProgSäk E 2020).

The Handbook on Software in Safety-Critical Applications, H ProgSäk E, 2020 (English edition), M 7762-001181, is hereby approved for use.

This English edition is a translation of the Swedish edition (Handbok för programvara i säkerhetskritiska tillämpningar, H ProgSäk, M7762-001041), approved for use in decision 15FMV2367-28:1. In the same decision, the former English edition (H ProgSäk E 2005), M7762-000621, was withdrawn.

In case of difficulties with regard to interpretation, the Swedish version applies.

This decision has been prepared by Svante Wåhlin, LOG / Requirements and Methodology.

FÖRSVARETS MATERIELVERK

Gustaf Fahl
Chief Technical Director
Swedish Defence Materiel Administration

Distribution: Swedish Defence Materiel Administration
Print: Elanders Sverige AB, november 2019

Preface

Scope

In this manual, FMV has compiled software standards, manuals and experiences of safe designs for technical systems that contain software in safety critical applications. This is to avoid, as far as possible, ill-health, accidents, damage to the environment and economic damage in the use of military technical systems and to increase the confidence in such technical systems during training, exercises and combat.

This manual is based on the Swedish Armed Force's view on system safety activities and is based on the methodology of the Armed Forces System Safety Handbook (H SystSäk). To achieve the required tolerable level of risk, there are both general design-oriented requirements and operational requirements in H SystSäk, and the more specific requirements for the design and manufacturing process for software for safety-critical applications are contained in this manual.

This manual is primarily intended for stakeholders who specify, procure, develop, modify or rent technical systems that contain software in safety critical applications and thus apply to other actors within the Swedish state.

Application of the Manual

Handbook Software in Safety Critical Applications (H ProgSäk) has no legal status but its use is governed by agreements or regulations. Its application for the procurement, development, and modification and leasing of technical systems is governed by the agency's own design organization. The manual provides information on appropriate requirements for procurement where software will be included in safety-critical applications. This will provide safer design solutions as well as providing background information, references and recommendations based on military operational requirements. The requirements may be formulated to better harmonize with the current technical system.

Standards usually contain different examples of documented knowledge. Following a standard is voluntary and a reference to a standard should be seen as a recommendation to comply with regulations or EU directives. However, in some regulations there is a direct reference to a specific standard to be followed. If FMV has ordered the contracted industry to comply with a certain standard, it will also be mandatory.

Experience shows that there may be conflicting requirements between standards from different technology areas, and this has to be managed on a case-by-case basis.

Advice to Readers

The reader who is unfamiliar with system safety activities should read the Armed Forces Manual System Safety (H SystSäk). For weapons and ammunition, refer to FMV Manual Weapon and Ammunition Safety (H VAS), which also contains a special section on software in ignition systems.

H ProgSäk is a complement to H SystSäk and can mainly be read and applied independently, but references to the text in H SystSäk are included in some sections. For some functions or subsystems, both H ProgSäk and H VAS need to be applied in parallel with these manuals.

Appendix 3 and *4* are important for the overall understanding of the work method. There are examples of the workflow from the Swedish Armed Forces' Requirements Document to the contracted industry's Development work.

Chapter 3 provides a simplified description of the division of work between the Armed Forces and FMV. The Coordination Agreement (SamO), which has been established between the Armed Forces and FMV, should be read for additional information about the work method.

Requirements Numbering

In *chapter 8*, the manual contains requirements for procurement of technical systems and products. The different sections begins with facts and explanatory text to the requirements. If applicable, the requirements are also commented.

The requirements in this manual are numbered according to the following principle: 2.801.03-A there:

2	prefix for requirements in H ProgSäk
801	Chapter 8, Section 1 (= 8.1)
03	sequence number
A	Administrative requirement (usually included in Business Obligation Specification, OUR)
T	Technical requirement (usually included in the Technical Specification, TS)

Improvement Suggestions

Proposals for improvements to H ProgSäk are sent to:

FMV

System Safety

SE-115 88 Stockholm

Sweden

E-mail: systemsakerhet.fmv@fmv.se

Table of Content

1 Aim of the Manual

1.1 Background

The Handbook Software in Safety Critical Applications 2018 (H ProgSäk 2018) is a further development of earlier editions (2001 in Swedish and 2005 in English). The previous issue was a manual jointly issued by the Armed Forces and FMV. This edition is an FMV publication.

1.2 Purpose

The purpose of this new edition of H ProgSäk is to be able to set the right requirements for software development to obtain safe technical systems. The manual contains basic requirements corresponding to the lowest accepted criticality level (GKPS) and guidance for the work. For all higher criticality levels, established software standards should be followed in addition to the basic requirements of the manual. This manual provides examples of commonly used established software standards

Safety-critical applications can be found in products and systems from, for example, ammunition, with a very limited software size, to very large and complex systems with many connected computers and different software components. For example, a management system can include hundreds of computers or exist as an integrated computer system on a ship. It should be possible to use the manual regardless of the size or complexity of the technical system.

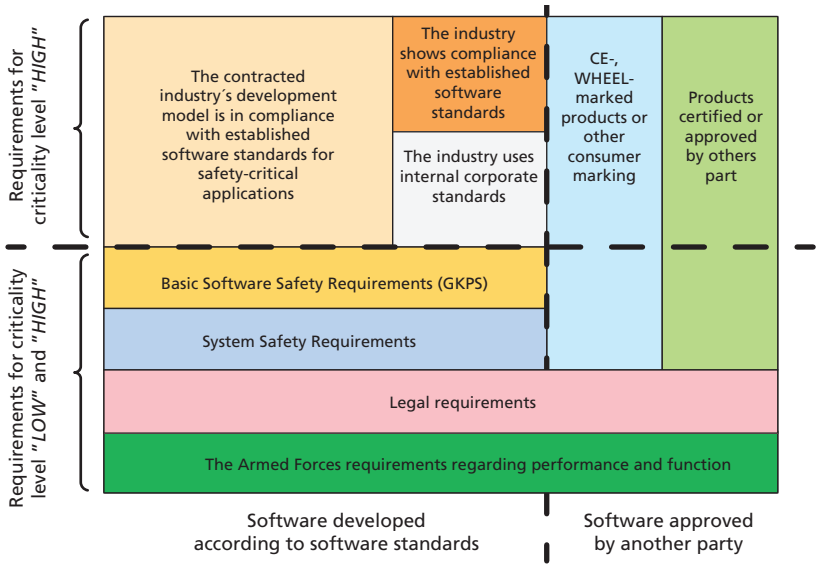


Figure 1:1 Illustration of requirements for technical systems with safety-critical software

The manual describes a methodology for defining, at an early stage in the development, a safety architecture so that the most critical errors that can be caused by the safety-critical computer system can be reduced, among other things, by means of added safety features.

The criticality of the safety-critical computer system depends partly on the consequence of an presumed accident, partly on the probability of a hazard, and on how the safety architecture of the system is designed. The goal of designing a safety architecture is to reduce the criticality of the safety-critical computer system as far as this is practically possible in the technical system. If the criticality level of the computer system can be lowered, the criticality of the software can usually be lowered. A methodology for initially determining the criticality of the safety-critical computer system and thus the included software is described in this manual.

The manual uses a number of basic concepts as follows.

Safety function

"An added function whose purpose is to reduce the likelihood of a hazard occurring in the event of a failure in the safety-critical function"

Source: H ProgSäk

Safety Critical Function

"QA function that controls or monitors energies and which in case of a fault can lead to a hazard and eventually accidents."

Source: H ProgSäk

Safety-critical computer system

"A computer systems that directly or indirectly controls or monitors energies and which, in the event of failure, can cause a hazard and, eventually, accidents."

Source: H ProgSäk

Safety-critical software

"Software that controls or monitors energies and which in case of error can cause a hazard and eventually accidents."

Source: H SystSäk

Software

"Contains computer programs, procedures, rules, associated documentation and data."

Source: AQAP 2210

Data

"Refers to information, often stored as files or databases, which the software uses when it provides functions or generates other information."

Source: H ProgSäk

Criticality

"A relative measure of the impact of a technical system on system safety."

Source: H ProgSäk

When designing technical systems that contain software in safety-critical applications, it is important to have a well-considered safety architecture, a structured approach and balanced technical requirements. The basic requirements in this manual aim to achieve this even if the specific military requirements imposed on military equipment used for training, practice and operations can lead to that the technical conditions for achieving optimal combat effectiveness can be difficult to achieve. When combat effectiveness and safety conflict with each other, these factors are weighed together and proportional efforts are made to create an optimal balance between safety and combat effectiveness. Such balancing takes place in consultation with the Armed Forces and FMV

1.3 Contents

The manual shows a method for both national and international procurement of safe software, including non-critical applications, as part of a technical system, system of systems, or for a separate product.

The manual describes the actions required by the various stakeholders involved in obtaining a secure technical system or product against the specified requirements. It is important that all stakeholders such as the Armed Forces, FMV and the contracted industry contribute to the safety of the technical system or product. The manual shall therefore be used for technical systems and products in all technical fields.

H ProgSäk is not a standard for, nor does it replace any standard for software development. The manual does not describe how programming is done or how software is to be developed.

The manual is advisory and the content is recommendations on methods for achieving the stated requirements for technical systems and products and can be made compulsory by entering into contracts with industry.

The basic requirements of the manual (GKPS) apply to the production of all software and for all technical systems where software is included. For software in non safety-critical applications, the reason is to ensure the possibility of future system updates. In developing software for initial criticality classification HIGH according to this manual, an established software standard shall be applied, in addition to the basic requirements (GKPS). The manual presents a selection of recommended standards in the field of safety-critical software.

A company standard cannot replace an established standard. If a company standard is applied, cross-references shall be established. Use of a company-standard shall, if applicable, be agreed with FMV in connection with the contract review.

Since there are many different software standards in the field and some concepts have different definitions, H ProgSäk contains attachments with explanations of the different concepts. Furthermore, it is important to know that there are different definitions in different software standards. This should be addressed in a Systems Safety Management Plan (SSMP).

The manual requires that, prior to the development work, the contracted industry shall describe which software standard(s) they intend to follow. Requirements verification and possible adaptation of the standard shall be agreed with FMV.

The manual states requirements regarding content in the documentation to be delivered to FMV and the Armed Forces.

The manual contains a general description of possible methods regarding dealing with complex electronic issues, such as programmable electronic circuits.

It is important that criticality rating of software is done in a correct and traceable manner. The manual highlights the benefits, but also the problems surrounding this. Methods for breaking down overarching system safety requirements are also described in the manual. The manual contains the recommended workflows that should be included in the planning of software development and its documentation.

The manual lists several adjacent areas, and these are summarized in *chapter 12*. For example, the manual does not cover the information security area, but it should be remembered that you cannot have a safe technical system unless you have taken into account both safety and security. There may be conflicting requirements between system safety and information security and these requirements must be dealt with in parallel to avoid poor solutions or the cost-driving requirements.

Also, the manual does not include the requirements for software development based on the need to deliver the desired functionality to counter hostile (or own) fire (countermeasure systems, IFF systems) or how important the function is to perform a particular assignment (mission critical). There is experience from foreign military development projects using criticality classification methodology, which means that the criticality classification of software (and thus software development requirements), according to current software standards, could also be applied to these aspects.

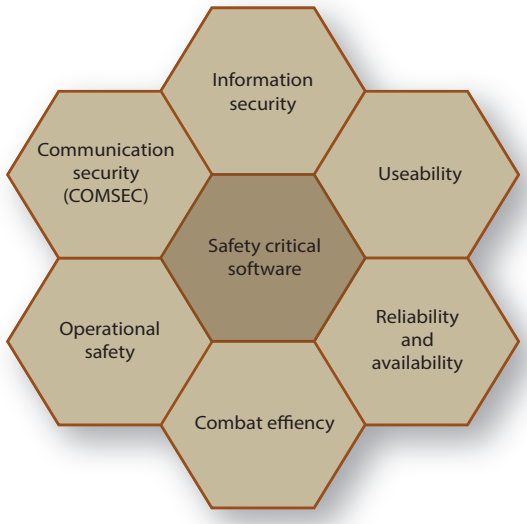


Figure 1:2 Scope and adjacent areas of software in safety critical applications.

1.4 Application

Legislation may grant exceptions to military technical systems and products and to military operations. Technical systems and products intended for the Armed Forces, which are normally used only in the event of war or preparedness, and during field exercises may require additional design-oriented requirements to achieve tolerable risk levels. Civil legislation may be governing in exercises where military activities are conducted in parallel with civilian activities, such as in civilian airspace or in actions that cannot be expected in war or during preparedness. Exemptions from the legislation do not of course mean a deviation from the general principles of occupational health and safety that all employees must be protected against ill health and accidents. The protection against ill health and accidents shall be applied to the Armed Forces personnel in the same way as for other employees in society. Further information can be found in H SystSäk.

This manual deals with software in safety critical applications, which can affect system safety on top system levels and are therefore classified as safety critical. The manual addresses the requirements for software with an initial criticality classification called LOW corresponding person, economic and environmental injury class IV in H SystSäk. It is recommended that software that does not have system safety impact but which is intended to be managed during its life-cycle is procured with the same requirements and procedures. In this manual, these requirements are referred to a "Basic Requirements Software Safety" (GKPS) and can be found in *chapter 8*.

The purpose of these basic requirements (GKPS) is that the software shall be of sufficient quality and that documentation for audit and configuration management is available to enable future system updates.

For procurement of software with a criticality classification ***HIGH***, this manual recommends that the contracted industry uses an established software standard that includes requirements for the processes used for developing safety-critical software.

Products that may contain software for performing certain activities, and that are already on the market (COTS products), where system software updates are not planned, may be procured as a product. Software in products that are CE marked for standalone use and which may not be integrated into military systems or which will not be modified, such as medical devices and measuring instruments, are handled in accordance with *chapter 10*.

Handling of Previously Developed Software (PDS), can be found in *chapter 11*.

This manual only complies with the design of safety critical software, data management and integration of PDS software. For specific activities such as medical technical equipment, specific requirements for design of the software are added through various laws, regulations and statutory collections. Design of software and maintenance equipment to support various types of maintenance work is handled in this manual.

1.5 Stakeholders

The stakeholders in this manual are the Armed Forces, FMV and the contracted industry. The roles can normally be described as follows:

Armed Forces	User
FMV	Design Authority
Contracted industry	Developer / Integrator

In cases where the Armed Forces acts as the Design Authority, it is recommended that the requirements in *chapter 7* be followed. If the Armed Forces or FMV assumes the role of developer / integrator, the requirements in *chapter 8* must be followed. This manual may also be used if the Swedish Armed Forces or FMV themselves acquire software for their own use.

1.6 Application Outside Sweden

When producing this manual, account has been taken of the EU regulations, EU directives and harmonized standards used internationally, so the manual is deemed to be applicable in its entirety, also in international procurement. When development assignments are submitted to a foreign supplier, system safety activities shall be carried out in accordance with the same procedures as with Swedish suppliers.

When purchasing already developed systems abroad, always ensure that information and documentation is obtained so that system safety evaluation can be performed.

1.7 Other Customers and Authorities

The manual is primarily aimed at stakeholders who acquire, develop or update software in technical systems. This also applies to rented or leased technical system.

FMV may use this manual during procurement or in cooperation with other authorities such as the Fortifications Agency (FORTV), the Defence Research Institute (FOI) and the Defence Radio Department (FRA).

2 Laws, Standards and Manuals

The purpose of this chapter is to provide an introduction and background based on laws, the most commonly used software standards and some manuals in the area.

2.1 Legal Requirements for use of Software in Products

Laws and ordinances are often written in a comprehensive manner and provide no details for the use of programmable systems. It is unusual to find support for how to develop and use software in safety-critical applications. In the past, programmable systems could be explicitly banned from being used in safety critical applications. Before the programmable technology was considered mature, relay-based logic was often provided for safety-related functions. However, in military technology systems there may still be grounds for applying conservative requirements and technologies in safety-critical applications.

Formulations in laws and regulations are often written technologically neutral, which means that the requirements are expressed in such a way that it does not matter what technology systems are being built with. Safety features can be realized with various technologies such as pneumatics, hydraulics, pyrotechnics, electrical circuits, electronics or software. The important thing is that industry has carried out a system safety analysis and then used techniques and methods to avoid design failures and to handle malfunctions during operation that could lead to accidents. A well-considered safety architecture should therefore be prioritized in the design work.

2.2 European Regulations

Within the European Union, efforts are being made to harmonize legislation in several areas. Therefore, European directives are expected to be incorporated into the laws and regulations of the different Member States. A directive is binding on the result to be achieved, but leaves the national authorities responsible for determining the implementation approach.

Several directives, such as the Low Voltage Directive, the EMC Directive, the Radio Directive, the Machinery Directive and the Medical Equipment Directive deal with product safety. When a manufacturer certifies that the product meets the basic health and safety requirements according to all applicable directives, the product may be CE marked as a sign of this. Particularly hazardous products shall be subject to a third party review by a notified body which, in turn, issues a certificate of verification which forms the basis of the CE marking.

Declaration of Conformity, DoC shall apply to all applicable directives for the product. A machine in the engineering industry is often covered by both the Machinery Directive, the Low Voltage Directive and the EMC Directive. Agricultural machinery, machinery in the engineering industry, packaging machines, printing machines and automatic doors are examples of machine types covered by the Machinery Directive. Among the machine types that the machine directive does not cover are weapons, motor vehicles and ships.

An advantage of common rules for product safety is that a product can be marketed in several Member States in the same configuration, without a repeated approval procedure in each individual member country. The basic health and safety regulations are the same in all EU Member States. The requirements of the directives are mandatory and must be met in order for the product to be put on the market.

For certain technology areas, the EU issues regulations, which in turn point directly to the regulatory framework to be followed. For the vehicle sector, EU regulations point directly to ECE regulations.

2.3 Standardization

The European directives state basic health and safety requirements without going into what this means. Detailed technical design questions are referred to standards. This approach means that the directives become stable and do not need to change at the same rate as today's technology level changes. An example of this principle is the Machinery Directive, which states, for example, that "*A control system must be designed and manufactured so that hazardous situations cannot occur. …*". To seek guidance on what the wording of the directive means in practice, read the European standards that deal with machine control systems.

Standards are updated regularly. It is common for a standard to be released in a new issue every five years.

In order to comply with the requirements of the directives, it is possible to follow the so-called harmonized European standards, issued by CEN, CENELEC or ETSI. That the standards are harmonized means that they are reviewed and comply with the requirements of the corresponding directive. CEN is the general European standardization organization for machines etc., CENELEC mainly covers standardization in the field of electrical systems and electronics. ETSI develops standards in telecommunications. In Sweden, standardization is managed through SIS and SEK.

There are also other standardization organizations that work to develop standards, but who are almost always linked to a particular industry field. Examples of such organizations are the *Society of Automotive Engineers* (SAE) and *Radio Technical Commission for Aeronautics* (RTCA). Military standards are issued by the *NATO Standardisation Agency* (NSA). Military standards include Def Stan, MIL-STD and *Standard NATO Agreement* (STANAG), including the *Allied Ordnance Publication* (AOP).

A standard is voluntary to follow. However, laws and regulations are mandatory and they impose requirements on product safety on, for example, machine safety, electrical environment (EMC), explosion protection (ATEX) and electrical safety (LVD). A manufacturer of equipment may choose a different way from that described in standards to achieve a product with high level of

safety. It is often requires a lot of work to produce own safety evidence, but it may be necessary in cases where the product is not in line, but more advanced, compared with the level of technology assumed by the corresponding standard.

The customer who purchases equipment, of course always has the possibility to demand that certain standards be met, even if the laws of the country do not require it. An example of this is large companies, which often have company-internal rules for the design of equipment to be used in their facilities, such as electrical installations.

2.4 Standards and Manuals for Software in Safety Critical Applications

Sections 2.5 – 2.19 show examples of the most common software standards, as well as a few manuals in the area that gained the widest knowledge of life cycle management, criticality classification, and techniques and methods to apply to safety-critical software in civil and military technical systems. The standard IEC 61508 is considered to be the model of several of the sector-specific standards.

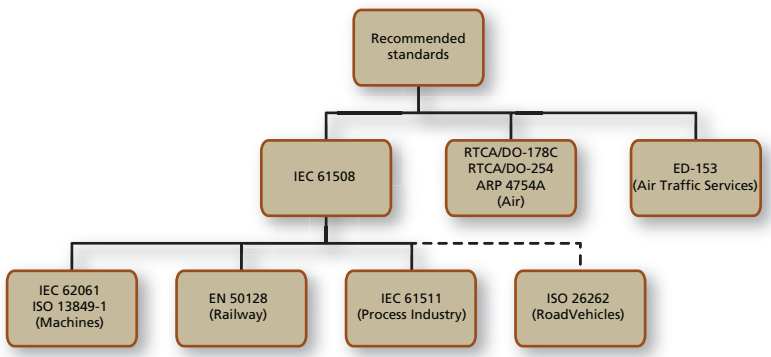


Figure 2:1 Relationships between recommended standards

The purpose of the sections below is to briefly describe different software-related standards and manuals with regard to content, scope and applicability. Furthermore, a number of concrete methodology-related questions are answered. For full understanding and to be able to use the standard, the user should have the specific standard available.

> Users of certain standards should always obtain it from the publisher. Firstly, to have access to the latest release, and also for possible copyright reasons.

In this manual's comparison between the standards, a number of aspects regarding similarities and differences between them are highlighted. The standards are different to their layouts and content, which means that comparisons must include many different aspects and representations. In *appendix 1*, this summary and comparison can be found in tabular form containing the following aspects:

- administrative aspects
- criticality classification
- technical scope
- techniques and methods
- methodology.

The tables in *appendix 1*relate to the main option, for example, if a standard is mainly for software, but if system aspects are mentioned to a small extent, the assessment will still be software. *table A1:2 Criticality classification* is a comparison of criticality levels between different standards.

2.5 ISO/IEC 61508 (Electrical / Electronic / Programmable Electronic Systems)

IEC 61508 *Functional safety of electrical/electronic/programmable electronic safety-related systems* is an internationally established software standard.

The standard has been developed within the *International Electrotechnical Commission* (IEC) but has also been adopted as a European Standard, known as EN 61508. In addition, it is a Swedish standard with the designation SS-EN 61508. The same information is available in IEC 61508, EN 61508 and SS-EN 61508.

2.5.1 Content and Scope

The standard covers the entire life cycle but is not harmonized with the Machinery Directive because the standard is generic. This means that it is used as a basis for sector-specific standards, such as IEC 62061 (Machines), IEC 61513 (Nuclear Power) and IEC 61511 (Process Industry).

The standard focuses on safety features. If the requirement for error rate is less stringent than 10^{-5} / hour, IEC 61508 is not intended to be applied. The standard states that if a safety-critical feature requires a lower error rate than 1×10^{-5} / hour, the entire function can be viewed as a safety feature and thus the standard will apply to the entire development process, which may be considered to be applicable to general software development.

The standard uses the so-called V model, which makes it applicable to processes that are built according to this model.

The first four parts of the standard are normative, while the other three are informative. See *table 2:1*.

Table 2:1 ISO / IEC 61508 different parts of the standard

Part	Title	Normative/Informative
1	General requirements	Normative
2	Requirements for electrical/electronic/ programmable electronic safety-related systems	Normative
3	Software requirements	Normative
4	Definitions and abbreviations	Normative
5	Examples of methods for the determination of safety integrity levels	Informative
6	Guidelines on the application of IEC 61508-2 and IEC 61508-3	Informative
7	Overview of techniques and measures	Informative

Note that software requirements are covered in Part 3, while hardware and systems are both part of Part 2. Included informative elements should be interpreted as being not mandatory, but they are highly recommended. Like all standards, there is room for interpretation. Some examples of inaccurate formulations are *should*, *consider*, *ensure*, *be detailed* and *appropriate*. Therefore, relevant assessments and interpretations of these terms need to be made.

The standard covers phases throughout the life cycle for one or more safety features in a technical system. The standard can be applied to all, or part of, a safety feature.

If a technical system integrates multiple safety features with other non-critical control elements, the entire system needs to be handled as safety critical. It is therefore desirable to distinguish safety features from "common control" in order not to get an overly expensive design and verification process, that is, trying to get some form of independence between them. If the standard is to apply to a part of the safety function, it is necessary to demonstrate independence to other parts.

The requirements for integrity (risk reduction) of safety features are assessed according to *Safety Integrity Level 1–4* (SIL 1–4)), where SIL 4 sets the highest requirements. A function takes in inputs, analyses them and sets outputs. This feature includes both hardware and software, but other ways of reducing accident risks, such as mechanical strengthening, are not included.

Appropriate SIL levels are determined by carrying out hazard analysis and the greater the risk of an accident, the higher the value of SIL is required. The standard divides the consequences into hazards to people, equipment, environment, information safety and financial damage. The standard couples the probability of hazardous errors to each SIL level. The higher the SIL level, the lower the probability of dangerous errors is tolerated. The requirement of lower probability of hazardous errors increases with increasing SIL levels.

Part 5 Annex E of the standard shows how SIL is calculated. Note, however, that part 5 is informative, that is, you don't have to follow it. To obtain the SIL level, a risk matrix based on the consequence of an unwanted event (wet event), as well as the likelihood of the event occurring is used. The value of the occurrence probability includes the frequency with which the risk of the event is predicted to be occurring as well as the likelihood that the safety functions will not succeed in avoiding the event when the risk occurs.

An example of safety function is a speed protection for a machine (referred to as EUC Equipment Under Control). The safety function is required to reduce the risk of injury to a person. Note that the standard does not say anything about the design of the EUC, but is completely focused on the safety function.

IEC 61508 distinguishes between probabilities for *Low demand mode*, *High demand mode* and *Continous operation*. For *Low demand mode* the probability of malfunction when the safety function needs to be used is specified. For *High demand mode* or *Continuous operation* the error rate per hour is specified. Safety features that are used less often than once a year are considered *Low demand mode*. For *High demand mode* and *Continuous mode*, the safety function's mean error rate (in the "per hour" unit) is specified

The entire Control System for *Equipment Under Control* (EUC) may in certain cases, when the error rate requirement is $<10^{-5}$/ hour, be considered as a safety feature in *Continuous operation*.

By considering the *Hardware Fault Tolerance*, (HFT) and *Safe Failure Fraction* (SFF), the standard describes the maximum SIL level achievable with a particular design architecture. The requirements for hardware error control increases with higher SIL levels.

For hardware, a number of values should be calculated to verify that the SIL level is met, while different methods are specified for software, that is, no calculations are made for the software. For hardware, see Part 2, Annex A and B *Techniques and Measures*, and the corresponding for software can be found in Part 3 Annex A and B. All SIL-dependent software requirements are collected

in *Techniques and Measures*. Annex B in Part 3 is informative, but should also be used for the software. *Techniques and Measures* software specifies methods and techniques with regard to SIL level.

Part 3 of the standard contains a number of sub-phases and requirements for these (*Software Safety Lifecycle Requirements*, SSLR). For the software design and development phase, sub-phases are also available, and requirements for these are defined.

As the standard contains many requirements, a *Functional Safety Assessment* (FSA) for the assessment is always required. Relevant documentation should be made available if the selection of SIL level indicates this.

2.5.2 Applicability

The standard IEC61508 is generic and independent and has no particular civil or military aspect. The standard applies specifically to safety features but many parts of the standard, but not all, can be used for the entire technical system. No special requirements exist for damage to property or the environment. No connection is made to areas such as land, sea or air. An area excluded in the standard is medical equipment which is instead covered by the standard IEC 60601 *Electrical Equipment for Medical Use*.

The standard does not intend to cover information safety aspects. However, it is considered to be a de facto standard for component suppliers of certified and standardized components for industry, such as sensors, actuators and logic elements with associated software as well as pure software components such as communication stacks and drivers.

2.6 ISO 26262 (Road Vehicles)

The standard ISO 26262 *Road Vehicles - Functional Safety* (Swedish title: *Vägfordon – Funktionssäkerhet i el- och elektroniksystem* is an international standard intended for the automotive industry and the first edition was issued in 2011. The standard has been developed by ISO Technical Committee ISO / TC 22 Road Vehicles, Subcommittee SC 3, Electrical and Electronic Equipment. It covers the entire life cycle from concept development, system design, hardware development, software development, evaluation, operation and maintenance.

2.6.1 Content and Scope

The standard consists of ten parts and software is dealt mainly with in Part 6. See *table 2:2*.

Table 2:2 ISO 26262 different parts of the standard

Part	Title
1	Vocabulary
2	Management of functional safety
3	Concept phase
4	Product development at the system level
5	Product development at the hardware level
6	Product development at the software level
7	Production and operation
8	Supporting processes
9	Automotive Safety Integrity Level (ASIL)-oriented and safety-oriented analyses
10	Guideline on ISO 26262

The standard ISO 26262 is based mainly on the standard IEC 61508, but is a sector-specific version for functional safety in the automotive industry. This is e.g. due, to the that automotive industry works with large-scale production. The term *Start of Production* (SOP) is central. Validation of the safety is performed before production start. The work is divided into concept phase, product development and activities after production start.

Part 7 of the Standard specifies requirements for production, which is not part of IEC 61508. It can be argued that a large number of features of a vehicle are safety related and always affect vehicle safety, do not have the character of prominent safety features such as overload protection.

The standard argues that system safety is achieved by actions with different technologies such as mechanical, hydraulic, pneumatic, electrical and programmable electronic systems. Although the standard covers functional safety in electrical and electronic systems, the standard provides a framework that can also be used for other technologies.

The standard describes the process from a lifecycle perspective. The initial conceptual phase includes the definition, risk analysis and concept of functional safety. The development work is described at the system level as well as for hardware and software. The life cycle also includes activities after production start.

ISO 26262 has a way of classifying hazards that are adapted to the automotive industry. Depending on severity, probability and verifiability, an ASIL class (A, B, C or D) is chosen, where D corresponds to the highest degree of risk reduction. Severity is classified from S1 to S3, where S3 is the highest severity. The probability is classified from E1 to E4, where E4 is the highest probability. Controllability is a measure of how well a driver can handle an emerging situation and is classified from C1 to C3, where C3 is a situation that cannot be handled by the driver. The greatest accident risks are assessed in a matrix in the standard where the situation is described with S3, E4 and C3. If any of the variables get a risk classified as S0, E0 or C0, then no values are used for the other variables. For these risks, no ASIL is assigned, but the standard estimates that normal quality management (*Quality Management*, QM) is sufficient.

For software development, the standard offers a reference model called V-model. Work on software development shall be adapted to the actual case, but be based on this reference model.

2.6.2 Scope

The automotive industry needs common guidelines on how to deal with safety critical embedded systems, and the standard has therefore come to wide use. It is important, not least because the number of “smart” features in vehicles increases rapidly. New support functions intended to increase safety are presented and the industry plans for functionality regarding autonomously driven passenger cars and trucks.

The standard is intended to be applied to safety-related systems that contain one or more electrical / electronic subsystems installed in a serialized passenger car with a total weight of up to 3 500 kg. The standard does not refer to electrical / electronic systems in special vehicles such as vehicles for the disabled. As the standard draws a limit of 3 500 kg, the standard does not apply to trucks and buses. However, the industry still refers to ISO 26262 because a corresponding standard is missing for heavier vehicles. In the future release 2 of ISO 26262 (2018), the limit of 3 500 kg is removed and a section delaing with motorcycle will also be included.

The standard manages risks due to malfunction of electrical and electronic systems, but does not handle risks such as electrical safety, fire protection, smoke, radiation, poisoning or corrosion, unless this is directly caused by malfunction of an electrical or electronic system. Nor does the standard address the performance of electrical or electronic systems, even though performance standards include brake systems, airbags, cruise control and automatic brakes are available.

The standard focuses exclusively on personal safety and omits damage to property and the environment. Nor is the standard intended to cover information safety issues.

2.7 EN ISO 13849-1 (Machine Controls)

Standard EN ISO 13849-1 *Safety of machinery - Safety-related parts of control systems - Part 1: General design principles* provide safety requirements and guidance on design principles and integration of safety features in machine control systems. The standard is based on EN 954-1 and ISO 13849-1.

The standard has been developed by ISO/TC 199, *Safety of Machinery*, a technical committee of the International Standardization Organization ISO, and has been adopted as EN ISO 13849-1: 2008 by the European Standardization CEN/TC 114, *Safety of Machinery*. A Swedish language version SS-EN ISO 13849-1: 2015 has been issued and has been adopted as EN ISO 13849-1. The first version of EN ISO 13849-1 issued in 2008 and an updated version was released in 2015.

2.7.1 Content and Scope

The parts of a machine's control system designed for the protection functions are called safety-related parts in control systems (Safety Related Parts / Control Systems, SRP / CS). These can consist of hardware and software. In addition to protection features, SRP / CS can also handle functions for operation of the machine such as two-hand devices or stops.

The work methodology required for machines is to avoid hazards by design, protect against residual hazards and, in cases where nothing else is possible, warn of remaining hazards. As part of the overall risk reduction strategy for a machine, the designer often tries to take measures to hazards by using technical protection with one or more protection features.

The need for risk reduction is assessed by combining the severity of injury with the exposure rate. The risk assessment is based on a situation before the intended protection function is applied and thus applies regardless of whether software is included in SRP / CS or if the logic is built with other technology.

Depending on the risk analysis, the protection functions are designed with different *Performance Levels* (PL). The standard specifies characteristics of the safety-related parts of the control system including the required performance level for performing the protection features. Performance levels are divided into five levels by the likelihood per hour of hazardous malfunction. This probabilistic approach differs from the qualitative approach, such as single error failure, which has previously been common for describing machine control performance. The method of expressing probability of malfunction can be compared to the standard IEC 61508, which also deals with error probabilities.

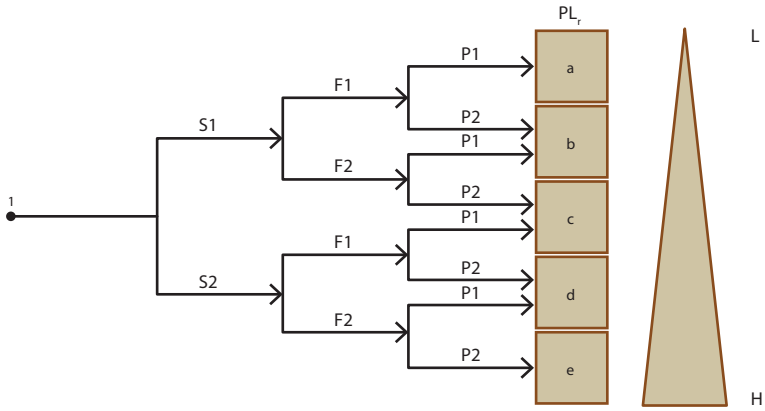
All machine controls are assumed to be in *High demand* or in *Continuous mode*. Therefore, probability values can be compared to SIL 1-3 according to IEC 61508.

Table 2:3 Performance Levels (PL) reproduced from EN ISO 13849-1

Performance Level	Average probability of hazardous malfunction per hour
a	$\geq 10^{-5} - < 10^{-4}$
b	$\geq 3\times10^{-6} - < 10^{-5}$
c	$\geq 10^{-6} - < 3\times10^{-6}$
d	$\geq 10^{-7} - < 10^{-6}$
e	$\geq 10^{-8} - < 10^{-7}$

Note: In addition to the average probability of hazardous malfunction per hour, other measures must also be taken to meet the performance level (PL).

The need for risk reduction can be difficult to assess. By combining the severity of the injury with the exposure rate, an assessment can be made. See example in *figure 2:2* below. The risk assessment is based on a situation before the intended protection function applies and applies regardless of whether software is included in SRP/CS or if the logic is built with other technologies.



Explanation

1 starting point for evaluating a protection function's contribution to risk reduction
L low contribution to risk reduction
H high contribution to risk reduction
PL$_r$ required level of performance

Risk Parameters

S the severity of the injury
S1 light (usually transient damage)
S2 severe (usually incurable injury or death)
F frequency and / or exposure time for the risk source
F1 rarely to less frequent and / or short exposure times
F2 often to continuous and / or long exposure times
P possibility to avoid the source of risk or limit the damage
P1 possible under certain circumstances
P2 hardly possible

Figure 2:2 Risk diagram to determine the required performance level according to EN ISO 13849-1

Performance level is determined by estimating the following aspects:

- Mean time for hazardous malfunction,
- Mean *Time to Hazardous Failure* ($MTTF_d$ value for individual components)
- Error Detection, *Diagnostic Coverage* (DC)
- *Common Cause Failures* (CCF)
- Structures
- The behaviour of the protection function at fault condition
- Safety-related software
- Systematic errors
- Ability to perform a protective function under expected environmental conditions.

In order to facilitate the assessment of the achieved performance level, the Standard applies a methodology based on categorization according to specific design criteria and specified behaviour in case of error. These categories are assigned one of five levels, called category B, 1, 2, 3 and 4. See *table 2:4*.

Table 2:4 Summary of requirements for different categories reproduced from EN ISO 13849-1

Category	Summary of Requirements	System behaviour	Principles for achieving safety
B	SRP / CS and / or their protective equipment, as well as their components, shall be designed, manufactured, selected, assembled and combined according to relevant standards to withstand expected impact. Basic safety principles shall be applied.	Error states that occur may result in loss of protection function.	Mainly through selection of components.
1	Requirements in B must be met. Well proven components and well-proven safety principles shall be used.	Error states that occur may result in loss of protection function, but the likelihood that they occur is lower than in category B.	Mainly through selection of components.
2	Requirements in B must be met and well-proven safety principles shall be used. Protective function shall be checked at appropriate intervals of the machine's control system.	Error states that occur may result in loss of protection between check-ups. Loss of protection function is detected by control.	Mainly through the structure of the system.
3	Requirements in B must be met and well-proven safety principles shall be used. Safety-related parts shall be designed to: • single fault states in any of these parts do not result in loss of protection function and • whenever practicable, the single state of error is detected.	When a single fault condition occurs, the protection function always remains. Some but not all error states are detected. Accumulating unidentified error states can lead to loss of protection function.	Mainly through the structure of the system.

Category	Summary of Requirements	System behaviour	Principles for achieving safety
4	Requirements in B must be met and well-proven safety principles shall be used. Safety-related parts shall be designed to: • a single fault condition in any of these parts does not result in loss of protection function and • The single fault state is detected when, or before, the protection function is called for the first time, but if this detection is not possible, an accumulation of undeclared error states shall not lead to loss of protection function.	When a single fault condition occurs, the protection function always remains. Detection of accumulated error states reduces the probability of loss of protection function (high DC). The error states are detected in time to prevent loss of protection function.	Mainly through the structure of the system.

Performance-level aspects can be grouped into quantifiable aspects (mean time between hazardous errors, $MTTF_d$ values for individual components, *Diagnostic Coverage* (DC), *Common Cause Failure* (CCF), Structure), and non-quantifiable qualitative aspects that affect the behaviour of SRP / CS protection function behaviour at fault, safety-related software, systematic error and environmental conditions). The standard shows the performance levels that are achievable with different category selection. In order to achieve the highest performance level, the control system must be designed according to category 4 and its $MTTF_d$ value shall be high.

The standard provides guidance for software development by providing overall requirements for control systems using programmable electronic systems. It often refers to the standard IEC 61508 for detailed techniques and methods. Software safety requirements include life-cycle activities for the software to be readable, understandable, and possible to test and update. The activities aim primarily at avoiding error states that occur during the program's life cycle.

Often, the software is built into the control system in such a way that it is not intended to be changed by the user. This is called the *Safety Related Embedded Software* (SRESW) standard. There is also software developed by the engineer for the special machine management, so-called *Safety Related Application Software* (SRASW,). Because SRESW and SRASW are handled differently, the requirements are also different.

The standard deals with both restricted program language *Limited Variability Language* (LVL) and non-restricted language capabilities *Full Variability Language* (FVL). LVL is often used for automated PLC systems where programming is strictly controlled by, for example, programming in function blocks. FVL means that the programmer can write his code freely, for example in common high level languages like C or Ada.

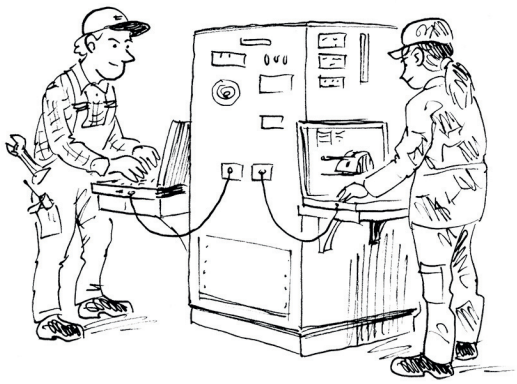


Figure 2:3 Industry develops and programs the computer system while the user sets parameters within approved limits

EN ISO13849-1 also deals with software-based parameterization of safety-related parameters. This is considered a safety-related aspect of the design and will be described in the specification of software safety requirements. Parameter setting shall be done with a custom software tool from the SRP / CS manufacturer and the safety of all data used for parameterization is maintained.

An informative part of the standard gives examples of typical activities to realize SRESW:

- Application of the V model for the software's safety life cycle (defined *Safety Life Cycle*, SLC,)
- Verification of software specification
- Programming rules at program structure level
- Programming rules when using variables
- Programming rules at the function block level.

The standard EN ISO 13849-2 describes validation. This section also includes validation of safety-related software, but only describes validation activities in a comprehensive manner.

2.7.2 Scope

Basic health and safety requirements for machinery within the EU are provided by the European Machinery Directive. For detailed information on safety aspects, refer to standards. The standard EN ISO 13849-1 is intended to provide guidance to those who work with the design and assessment of machine control systems. The standard provides no specific guidance for compliance with other EU directives.

Components such as limit switches and programmable control systems (PLCs), which can be certified by the manufacturer, can be used in safety features with a certain performance level. The standard does not specify the protection features or performance levels to be used in a single case. To find out what is required of a certain protection function for a particular machine, a risk analysis must be carried out

The standard focuses exclusively on personal safety and omits damage to property and the environment. Nor does the standard cover information safety issues.

Both EN ISO 13849-1 and EN 62061 cover machine controls and it is considered less appropriate for the same scope to have two different EU standards. A synergy between the standards is discussed.

2.8 EN 62061 (Machine Controls)

The Standard EN 62061 *Safety of machinery - Functional safety of electrical, electronic and programmable electronic safety-critical control systems* provides safety requirements and guidance on design principles and integration of safety features for machines. The standard is based on IEC 61508 and is a sector-specific application for the machinery industry. See *section 2.7*.

The standard has been developed by IEC TC 44, the, *Safety of Machinery - Electrotechnical Aspects* a technical committee of the International Electrotechnical Standardization Organization (IEC). The standard has been adopted as EN IEC 62061 by the European Standardization Organization CENELEC and as a Swedish Standard by the *SEK Svensk Elstandard* (SEK).

2.8.1 Content and Scope

The standard describes the functional safety of electrical, electronic and programmable electronic safety-critical control systems (*Safety-Related Electrical Control Systems*, SRECS). The risk assessment results in a risk reduction strategy that identifies *Safety-Related Control Functions* (SRCF). The features are documented in a functional requirement specification and a requirement specification for safety integrity.

Methodology and requirements are given to assign the required *Safety Intergrity Level* (SIL) to each safety-related control function conducted by Safety-Related Electrical Control Systems (SRECS). The standard provides support in specification, design and validation. Integration of protection functions developed according to EN ISO 13849-1 is also supported by the standard EN 62061.

Depending on the outcome of the risk analysis, *Safety-Related Control Functions* (SRCF) are designed according to different SIL levels, SIL 1-3, where SIL-3 sets the highest requirements. In the standard IEC 61508, SIL 4 is also defined, but this high degree of risk reduction is not considered necessary for machines. All

machine controls are assumed to be in *High Demand* or *Continuous Mode*. The probability values for SIL 1-3 are the same as in standard IEC 61508.

An informative part of the standard shows an approach to assessing the SIL level. Severity is judged on a scale from 1 to 4 where 4 corresponds to the most serious consequences. Similarly, exposure, likelihood and ability to avoid danger are added to a where a high value corresponds to high probability. At high severity and high probability, SIL 3 is chosen, while for lower risk of accidents, normal quality assurance measures (*Other Measures*, OM) may be sufficient. *Safety-Related Control Functions* (SRCF) with negligible accident risks are not assigned a SIL level.

By considering the *Hardware Fault Tolerance*, HFT and *Safe Failure Fraction* (SFF), the standard describes the maximum SIL level that can be achieved with a particular design architecture. The hardware failure requirement increases with a higher SIL level, while the standard allows single-channel designs to be used up to SIL 3, provided that the potential for error detection is high enough.

A *Software Safety Requirements Specification* (SSRS) specification shall be provided for each subsystem.

For design and development of embedded software, refer to IEC 61508-3. However, the standard sets up parametrization and application software development as these can be expected to be activities performed by many machinery manufacturers. Often, machine control is based on a modular control system (*Programmable Logic Controller*, PLC) where the designer inputs parameters and software to control the machine.

Parameterization provides for the integrity of data to be maintained, e.g. by checking that data is within valid range and data is not corrupted. Requirements are also stated for the tool used for parameterization and on the approach to change the safety-critical parameters.

When developing application software, IEC 61508-3 is to be followed when using *Full Variability Language* (FVL). Examples of FVL are programming in high level languages such as C or Ada.

If the machine control is programmed in a *Low Variability Language* (LVL), the EN 62061 states a number of requirements for the development process, configuration management, software architecture, tools, development methodology and testing. Examples of LVL are programming in function blocks for PLC.

The cases where the application software controls both safety-related and non-safety related features, the entire application software shall be considered safety-related, provided that insufficient independence between the program components cannot be shown.

2.8.2 Scope

Basic health and safety requirements for machinery in the EU are given by the European Machinery Directive. For detailed information on safety aspects, refer to standards. The standard EN 62061 is intended to provide guidance to those who work with the design and assessment of machine control systems. It provides no specific guidance for compliance with other EU directives.

Components such as limit switches and programmable control systems (PLCs), which can be certified by the manufacturer, can be used in safety features with a certain SIL level. The standard does not specify the protection features or SIL levels to be used in a specific case. To establish what is required of a certain protection function on a particular machine, a risk analysis must be carried out.

The standard is focused solely on personal safety and omits damage to property and the environment. Nor does the standard cover information safety issues.

A discussion is under way on how to coordinate between EN 62061 and EN ISO 13849-1. See *section 2.7.2*.

2.9 RTCA DO-178C/EUROCAE ED-12C (Air)

RTCA DO-178C *Software Considerations in Airborne Systems and Equipment Certification* is an internationally established standard that focuses on airborne application software. The current version of the standard is C and it was released in 2011. The standard covers the entire life cycle of the software and relates in part to hardware but only in relation to software. For example, there is no process description for hardware.

The standard has been developed in cooperation between the RTCA (*Radio Technical Commission for Aeronautics*) Special Committee 205 (SC-205) and EUROCAE Working Group 71 (*European Organization for Civil Aviation Equipment WG-71*).

2.9.1 Content and Scope

One aspect that permeates the standard is civil certification. The standard is consistently elaborated and can be considered complete with regard to the extent of the standard. The standard also contains a lot of guidance, approaches, examples, definitions and explanations. The standard also contains two ANNEXES and two APPENDICES.

One particular aspect is that there are *shall* or *musts* in the standard but just should. The reason is that there is no legal requirement to comply with the standard. The document is based on consensus in the aviation industry, but acknowledges that there may be alternative methods. This is why the words *shall* and *must* are avoided in the text. On the other hand, the word *may* is used throughout. How to apply the standard is determined by a PSAC. A PSAC is agreed between industry and FMV or between industry and the certifying authority if the industry is to deliver a certified product in accordance with applicable aviation regulations.

The standard is focused on what to do but not how to do it. This means that the review, for example in the case of certification, must take accurate account of if the correct scope has been included and if the content is in accordance with the parts of the standard used.

By virtue of the fact that in principle all parts of the text (headings, purposes, lists, definitions, examples, etc.) have identifiers, references can be made, for example, in ANNEX A, related to references and documentation to Software Level.

The standard is structured in different sections. Note that *Integral Processes* are performed in parallel with *Software Planning Process* and *Software Development Processes* throughout the life cycle. The standard describes the software process' relationship to system processes and there is a large information exchange between these different processes.

The documents that can be included are listed in *table 2:5*. The standard section 11: *Software Life Cycle Data* contains a description of the content.

Table 2:5 Examples of documentation specified in the RTCA DO-178C

Document	Created/Updated in Process
Design Description	Software Design Process
Executable Object Code	Integration Process Software Development Process
Parameter Data Item File	Integration Process Software Development Process
Plan For Software Aspects Of Certification	Software Planning Process Certification Liason Process
Problem Reports	Software Configuration Management Process
Software Accomplishment Summary	Certification Liaison Process
Software Code Standards	Software Planning Process
Software Configuration Index (SCI)	Software Configuration Management Process Certification Liaison Process
Software Configuration Management Plan	Software Planning Process
Software Configuration Management Records	Software Configuration Management Process
Software Design Standards	Software Planning Process
Software Development Plan	Software Planning Process

Document	Created/Updated in Process
Software Life Cycle Environment Configuration Index (SECI)	Software Configuration Management Process
Software Quality Assurance Plan	Software Planning Process
Software Quality Assurance Records	Software Quality Assurance Process
Software Requirements Data	Software Requirement Process
Software Requirements Standards	Software Planning Process
Software Verification Cases And Procedures	Software Verification Process
Software Verification Plan	Software Planning Process
Software Verification Results	Software Verification Process
Source Code	Software Coding Process Software Development Process
Trace Data	Software Development Processes Software Verification Process

There are five defined levels of Software Level (SL) in terms of software severity. See the standard RTCA DO-178C for complete definitions:

- Level A: "…resulting in a catastrophic failure…"
- Level B: "…resulting in a hazardous failure…"
- Level C: "…resulting in a major failure…"
- Level D: "…resulting in a minor failure…"
- Level E: "…no effect on aircraft operational capability or pilot workload…".

The standard defines criticality levels based on consequence. The included example of a system safety process indicates a collision with many dead (A), few people (passengers) injured or deceased (B), significant reduction in safety margins or functionality (C), some reduction in safety margins or functionality (D), no effect terms of seriousness (E).

Economic damage and damage to the environment are not included. There is no support for how to choose the level without referring to system processes.

For software that is part of a technical system, the criticality level is determined based on a risk analysis conducted as an example according to the SAE-ARP 4754A Standard, see section *section 2.11*.

ANNEX A specifies what activities and what results are required in relation to Software Level (SL).

Requirement breakdown is done as follows:

1. Safety-related requirements, including *Software Level* (SL) for software, are obtained at system level (*System Safety Assessment Process* results)
2. Requirements are broken down to software high level requirements depending on *Software Level* (SL)
3. Requirements are broken down to successively lower levels (if applicable) for software.
4. Low-level requirements (minimum level) are defined for software.
5. Derived requirements should be identified at both high and low levels.
6. Requirements tracking depending on *Software Level* (SL).

The PDS (referred to as COTS in the standard) sets the same requirements as on proprietary software, see DO-278A.

2.9.2 Scope

The standard is aimed at civil aviation but can also be used for military aviation. The standard applies to software products in airborne systems, as shown by the definition of *Software Level* (SL). There is no connection to sea or land applications in the standard.

The connection to system processes is weak in the standard and only shows those directly related to software processes. Hardware processes are not covered at all. Therefore, additional management of system processes and hardware processes is required, but there is no guidance for choosing these in the standard.

Despite the specific references to airborne systems, the standard can in principle be used for other types of technical systems, but then needs to be interpreted and adapted. However, one must clearly state why choosing such a standard, which parts are included and interpretation of the content. In order to have complete management, system processes and hardware processes must also be included.

The standard does not intend to cover information safety aspects.

In support of DO-178C and DO-278 application, DO-258C can be used.

There are four additional documents that address specific aspects of software development. See the corresponding references below:

- The RTCA DO-330 *Software Tool Qualification Considerations* was developed to provide guidance on how tools used in development can be qualified.
- RTCA DO-331 *Model-Based Development and Verification Supplement to DO-178C and DO-278* describes model-based development and verification.
- RTCA DO-332 *Object Oriented Technology and Related Techniques Supplement to DO-178C and DO-278A* lists object-oriented software and the conditions under which it can be used.
- The RTCA DO-333 *Formal Methods Supplement to DO-178C and DO-278A* describes formal methods and how they can supplement testing.

For aircraft certification, there are also RTCA DO-297 *Integrated Modular Avionics* (IMA) Design Guidance and Certification Considerations.

Advice for application of DO-178C and DO-278A is given in DO-248C.

A Reference Guide / Adjustment of the DO-178C for Air Traffic Control Systems can be found in the RTCA DO-278A *Guidelines for Communication, Navigation, Surveillance, and Air Traffic and Air Traffic Management (CNS / ATM) Systems Software Integrity Assurance*.

2.10 RTCA DO-254 (Programmable Logic, Air)

The standard RTCA DO-254 *Design Assurance Guidance for Airborne Electronic Hardware* specifies guidelines for the design of electronic hardware for the aerospace industry.

The standard has been developed in collaboration between RTCA (*Radio Technical Commission for Aeronautics*) Special Committee 180 (SC-180) and EUROCAE Working Group 46 (*European Organization for Civil Aviation Equipment WG-46*).

2.10.1 Content and Scope

The purpose of the standard is to ensure that the intended functions are performed in a safe way. Although the standard refers to hardware, it should be considered when discussing software, as it covers programmable circuits. The difference between logic in terms of software for a computer and logic in terms of the contents of a programmable circuit may seem small.

The standard defines five different levels, *Design Assurance Level* (DAL), to ensure proper development of the system and the hardware, DAL A-E, respectively. These levels are based on the consequences of errors that can cause accidents. In order to develop electronic hardware corresponding to DAL A, much more extensive verification and validation is needed than for DAL E.

Design work begins at the system level by distributing various hardware features and allocating the DAL level to the hardware to ensure proper development. A feature in the system can be distributed to a hardware device, software component or to a combination of both hardware and software.

The standard points to three system safety evaluation processes:

- Functional Hazard Assessment (FHA)
- Preliminary System Safety Assessment (PSSA)
- System Safety Assessment (SSA).

A lifecycle for development work is described with five main processes that can be followed for complete hardware devices, circuit boards or *Application Specific Integrated Circuits/Programmable*

Logic Devices (ASIC/PLDs). The main processes are requirements, conceptual design, detail design, implementation and transfer to production.

To design electronic hardware, the design engineer is dependent on tools. This is particularly true for ASIC and PLD. An error in the design can very well be introduced by improperly functioning design tools. Similarly, an incorrectly functioning test tool can amiss finding errors in the design.

Therefore, the standard prescribes that tools should be evaluated before use and that the result of tool qualification is documented and saved.

If the result of the tool qualification is to be subject to an independent evaluation, the tool itself is not required to be evaluated. For the lower levels, it is also estimated that evaluation of the tool is not required. On the other hand, DAL A, B and C design tools, and DAL A and B test tools must be evaluated. The only exception is if it can be shown that there is a relevant history from previous use of the tool.

Formal methods are described as a technique that can provide further evidence in the design process (see *RTCA/DO-254, appendix B, section 3.3.3*). In order to use formal methods, the requirement specification must be formally written. The degree of detail in the formal description of a component depends on the objectives of the selected formal analysis methods.

Another aspect is that there are no *shall* and basically no *musts* in the standard just should. The reason is that there are no legal requirements to comply with the standard. The document is based on consensus in the aviation industry, but at the same time allows alternatives methods. In addition, the word *may* is commonly used in the standard.

2.10.2 Application

The standard RTCA / DO-254 treats electronic hardware for use in the aviation industry. The standard is applicable, but not limited to:

- Replaceable modules (*Line Replaceable Unit*, LRU)
- PCBs
- Microcoded components such as ASICs and PLDs
- Components with technology-integrated circuits, such as multichip modules.

The standard is not intended to cover information safety aspects.

2.11 ARP 4754A (Air)

The standard SAE ARP4754A *Aerospace Recommended Practice - Guidelines for Development of Civil Aircraft and Systems* applies to system aspects and refers to DO-178C / ED-12C for software development and DO-254 / ED-80 for hardware development.

The standard has been developed in cooperation between the *Systems Integration Requirements Task* (SIRT) and EUROCAE Working Group 46 (*European Civil Aviation Equipment WG-46*).

2.11.1 Content and Scope

The standard contains three APPENDICES (the fourth appendix has been withdrawn):

- APPENDIX A; Process objectives data
- APPENDIX B; Safety program
- APPENDIX C; FDAL/IDAL Assignment process example.

The processes in the standard are divided into three main groups:

- Aircraft and system development process
- Integral processes
- Modifications to aircraft or systems.

Because in principle all parts of the requirements text such as headings, purposes, lists, definitions, examples and checklists have identifiers references are simplified. For example, APPENDIX A relates references and documentation to *Development Assurance Level*.

One aspect that permeates the standard is certification. The standard is consistently implemented and can be considered complete with regard to the present extent. The standard also contains guidance, procedures, examples, checklists, definitions and explanations.

Another aspect is that there are no *shall* requirements and in principle no *musts* in the standard but just *should* requirements. The reason is that there is no legal requirement to comply with the standard. The document is based on consensus in the aviation industry, but at the same time allows alternative methods. In addition, the word *may* is commonly used in the standard.

The standard is strongly focused on *what to do*, but also in many cases in the form of examples, *how*. This means that the review, for example in the case of certification, must take accurate account if the correct scope has been included and if the content is in accordance with the parts of the standard used.

The standard is comprehensive and therefore terminology becomes extra important not least because new concepts are included and other concepts have a different definition compared to other standards.

Classification matches *Assurance Level* according to *Catastrophic - A, … No Safety Effect - E* (see Table 2). Note that the *fault*, *error* and *failure* differ from Laprie's definitions (commonly used in academia). Level A - E is also used for information exchange according to DO-178B / ED-12B and DO-254 / ED-80.

The standard applies great importance to requirements and suggests early validation of requirements, although validation may need to be carried out once design and implementation has been completed. Applying great importance to requirements formulation is justified as very cost-effective. Great importance is also given to defining functions, analysing *Common Cause Failure* (CCFs), and managing *Derived Requirements*, that is, lower-level requirements that cannot be directly traced to higher system level requirements.

There is a clear explanation for software and hardware development where information exchange is also described between system processes and software / hardware processes.

What is special about this standard is the assignment of FDAL to *functions* and IDAL to components (*items*). This means that the same procedure can be used, for example, comparing independence between functions and independence between components. The standard lists an example showing the method of allocating FDAL and IDAL. It is an application where no dependencies exist and which is structured in the form of a fault tree, where the basic event is a fault and the top event is a catastrophic error. In order to mitigate the impact of a top event, FDAL / IDAL assignment must come from both branches leading to the top event and list the incorrect combinations that are relevant. That is, the shortest paths to the top event, i.e. *Minimal Cut Set* (MCS). It is then necessary to determine the FDAL for the functions and IDAL for the components.

For both verification and validation of requirements, the methods and data are specified in the standard according to *Development Assurance Level A-E*, (DAL A-E). For both verification and validation, there is a possibility of tayloring of the level of certification (R-*Recommended for certification*, A - *As negotiated for certification*, N-*Not required for certification*). Therefore, there is also system-level quality assurance. However, it should be noted that both verification and validation are done at the system level.

2.11.2 Application

The standard has no specific civil or military aspect. The standard applies to the system components of airborne systems and has a clear link to these in the text, as reflected in the definition of *Software Level* (SL), certification and many other places where *aircraft* and *airborne* are specified. There is thus no connection to sea or land applications in the standard.

Software processes and hardware processes are not covered but can be found in associated standards. These are so strongly coupled to the *SAE ARP4754A* that they are not suitable to be replaced by other standards.

The standard is not intended to cover information safety aspects.

Despite the specific references to airborne systems, the standard can in principle be used for other types of applications. However, a clear explanation of the choice of this standard is required, as well as the specification of which parts are included and the interpretation of the content. An example is *Classification* that may need to completely redone.

The standard cannot be used separately but must be used in conjunction with associated software or hardware development standards. Since much information is available and required, users should be properly trained before applying the standard. A chapter requiring a deeper review because of its complexity and importance is *section 5.2 Development Assurance Level Assignment*.

2.12 EN 50128:2011 (Railway)

There are three international standards for railway installations that together form a entity:

- System aspects are specified in CENELEC, *Railway Applications - The Specification and Demonstration of Reliability, Availability, Maintainability and Safety* (RAMS), EN 50126.
- Software aspects are specified in CENELEC, *Railway Applications - Communication, Signaling and Processing Systems - Software for Railway Control and Protection Systems*, EN 50128.
- Hardware aspects are specified in CENELEC, *Railway Applications - Communication, Signaling and Processing Systems - Safety Related Electronic Systems for Signaling* EN 50129.

2.12.1 Content and Scope

The following is a summary of EN 50128 and the current version from June 2011. It is not entirely certain when a new official version is expected to be released. The three standards will be seen as a specialization of the generic standard IEC 61508 and created for the railway infrastructure sector. The application is railway installations but there is nothing that prevents applying the standard in other areas after some adaptation. EN 50128 thus addresses safety features in software and consists of the following parts:

Chapter 1 – Scope

Chapter 2 – Normative References

Chapter 3 – Terms, Definitions and Abbreviations

Chapter 4 – Objectives, Conformance and Software Safety Integrity Levels

Chapter 5 – Software Management and Organization

Chapter 6 – Software Assurance

Chapter 7 – Generic Software Development

Chapter 8 – Development of Application Data or Algorithms: Systems Configured by Application Data or Algorithms

Chapter 9 – Software Deployment and Maintenance

Annex A – Criteria for the Selection of Techniques and Measures

Annex B – Key Software Roles and Responsibilities

Annex C – Documents Control Summary

Annex D – Bibliography of Techniques

Two examples of lifecycle models for software development are given in this standard, one of which is the V model. Both are related documents and these are also listed in Annex C.

Terminology exists in all three parts. Note that safety is not information safety but is defined as: “Safety, as an element that characterizes the resilience of a railway system to vandalism and unreasonable human behaviour, …”. Security is only defined and is not otherwise covered in EN 50126, EN 50128 or EN 50129. Information safety is not addressed.

The risk graph method is application-dependent, but a valid example is shown below. This is stated in EN 50126. Here, “Risk Levels” can be replaced with the corresponding safety features requirement; for example SIL 1 - Negligible, SIL 2 - Tolerable, SIL 3 - Undesirable, SIL 4 - Intolerable.

Frequency	Risk levels			
Frequent	Undesirable	Non tolerable	Non tolerable	Non tolerable
Likely	Tolerable	Undesirable	Non tolerable	Non tolerable
Temporary	Tolerable	Undesirable	Undesirable	Non tolerable
Possible	Negligible	Tolerable	Undesirable	Undesirable
Unlikely	Negligible	Negligible	Tolerable	Tolerable
Incredible	Negligible	Negligible	Negligible	Negligible

Figure 2:4 Occurrence in case of danger, reproduced from EN 50126

Software development is specified in *Chapter 7, Generic Software Development*, which includes the following parts:

1. Lifecycle and Documentation for Generic Software.
2. Software Requirements
3. Architecture and Design
4. Component Design
5. Component Implementation and Testing
6. Integration
7. Overall Software Testing/Final Validation

A number of activities run in parallel (see chapter 6) with software development: *Software testing, Software verification, Software validation, Software assessment, Software quality assurance and Modification and change control*.

As for tools, the same classification is used as IEC 61508.

Chapter 8 addresses parameterization, called *application data*, and a special process is used for this. Software tool support is usually required. The idea is that Generic software is first developed and is largely application-independent, and then application-spe-

cific with *application data*. Some important things to consider are: development process for *application data*, SIL for application data and unauthorized combinations of *application data*.

Removal of deleted software is not addressed in EN 50128 but is done for the entire system according to EN 50126.

A major focus is on roles (see Annex B) and the following roles are defined: *Requirements Manager*, *Designer*, *Implementer*, *Tester*, *Verifier*, *Integrator*, *Validator*, *Assessor*, *Project Manager*, *Configuration Manager*. A person/organization may have more than one role. Requirements are set as to what the respective roles should do; the production of special documents and the implementation of activities. How the different roles interact is also stated.

There are a few *should* in the requirements section and some in NOTE. Otherwise, the word *shall* is used.

Note that RAMS is not really mentioned in EN 50128 and EN 50129 but is entirely covered by EN 50126.

Annex A: Criteria for the Selection of Techniques and Measures.
Dependency is made similar to that of IEC 61508, that is, only by means of tables, but these differ from IEC 61508 Part 3

Some comments in relation to IEC 61508:

- It is good that M (Mandatory) is introduced because there are things that cannot be excluded or negotiated.
- The B tables have been removed and the information has instead been transferred to the new A tables.
- SIL 1 and 2 and SIL 3 and 4 have been combined, which means increased requirements for SIL 1 and SIL 3 respectively.
- Backward Recovery and Forward Recovery are called NR because they are difficult to implement in practice,
- The suitability of different program languages differs.

Annex B: Key software roles and responsibilities.
The different roles and responsibilities are defined in Annex B. In addition, the skills of each role are specified.

Annex C: Documents Control Summary.
The document list is defined in Annex C and also lists relationships with roles.

Annex D: Bibliography of techniques.
Here are 71 different methods described in the same way as in IEC 61508 Part 7. You do not have to follow these but they are a good start and usually sufficient.

2.12.2 Application

The standard applies to software and has no particular civil or military aspect but is intended for railway installations. The standard comes from CENELEC and is independent of ground, sea and air application, but it has strong connection to IEC61508 and therefore applicable to safety features. The concept *Hazard* is used but no connection to the type of damage (human, equipment, economy, environment) is made. The concept Safety is also used in connection to humans. The standard is not intended to cover information safety aspects

2.13 ED-153 (Air Traffic Services)

EUROCAE, ED-153 *Guidelines for ANS Software Safety Assurance*, ED-153, is an international standard for *Air Navigation Service* (ANS) software (SW). The current version is from August 2009. The standard is generally used for applications, i.e. not especially for safety features (for example, IEC 61508).

2.13.1 Content and Scope

The standard also includes managing infrastructure and projects and consists of the following parts:

Chapter 1 – Introduction

Chapter 2 – Document Strategy

Chapter 3 – Software Safety Assurance System

Chapter 4 – Primary Lifecycle Processes

Chapter 5 – Supporting Lifecycle Processes

Chapter 6 – Organisational Lifecycle Processes

Chapter 7 – Additional ANS Software Lifecycle Objectives

Chapter 8 – Software Safety Folder

Annex A – Reference to Existing Software Standards

Annex B – Roles and Responsibilities Scenarios

Annex C – Traceability with ESARR6

The standard is applicable to air traffic management, but there is nothing that prevents it from being applied to other areas. In *Chapter 8, Software Safety Folder*, documents and evidence of SWAL (equivalent to *Safety Case* in other contexts) are specified. Two different approaches are described: *Project-based Structure* and *Compliance-Based Structure*.

In Annex A, a comparison is made between different standards: ISO / IEC 12207, ED-109 / DO-278, ED-12B / DO-178B, IEC 61508, CMMI. Appendix B shows different examples of roles and responsibilities. Annex C provides traceability between ED-153 and ESARR6 (*Software in ATM Functional Systems*). The following terminology is used:

Table 2:6 Terminology of the Standard ED-153

Term	Explanation
ANS	Air Navigation Service
COTS	Commercial Off The Shelf. COTS denotes purchased SW, previously developed SW using ED-153, etc. PDS (Previously Developed Software) is also used as a designation for COTS.
Independence	Need to be handled by: other person, different departments within companies, different organizations etc
SWAL	Software Assurance Level, 1 - 4, there is 1 with the highest requirements (most critical) and 4 with the minimum requirements.
ESARR	Eurocontrol Safety Regulatory Requirement

A relatively large number of processes are included in the standard and there is an initial description of the respective process. Each process contains a number of *objectives* that can be seen as requirements. Each requirement is referred to as "shall" with a number, that is, the identifier. There are a number of examples, *should* and *notes* as well, but these are not considered as requirements and therefore are not numbered. Each objective shows how applicable it is with regard to SWAL (1 - 4) and in addition if it requires independence, i.e. work by an independent party (according to any level). Output is also specified for each objective. Some processes are not focused on, or not applicable to software development, but instead on system aspects, including procurement, delivery, validation, operation and maintenance. *Organizational Lifecycle Processes* may also be relatively independent of software. The standard has a fairly detailed description and requirements for COTS (many "should" and "may"). However, COTS cannot be used for SWAL1 (see *chapter 7.2.0 NOTE*). Advice is also given on how to qualify tools.

For hazards, a *cause-effect* principle is used, for example, using FTA (Fault Tree Analysis) for "cause" and for instance ETA (Event Tree Analysis) for "effect" (which produces the effects from "hazard"). *Table 2:7* below shows how SWAL is chosen based on probability and severity.

Table 2:7 Selecting SWAL based on probability and severity

Likelihood of generating such an effect (Pe × Ph)	Effect Severity Class			
	1	2	3	4
Very Possible	SWAL 1	SWAL 2	SWAL 3	SWAL 4
Possible	SWAL 2	SWAL 3	SWAL 3	SWAL 4
Very Unlikely	SWAL 3	SWAL 3	SWAL 4	SWAL 4
Extremely Unlikely	SWAL 4	SWAL 4	SWAL 4	SWAL 4

Effect Severity Class 1 - 4 is not defined but is defined by the application. This means that it is difficult to compare risks with other standards. *Severity Class 1* is the most serious, then 2, 3 and 4. For probability (*Likelihood*), the following is defined: *Very Possible*, *Possible*, *Very Unlikely*, *Extremely Unlikely*. Risk is not specifically defined but should be seen as a combination of *Severity Class* and *Likelihood*.

2.13.2 Application

The standard applies to software development and has no particular civil or military aspect but is intended for air traffic management applications (ANS). However, the background is civilian, the standard comes from EUROCAE (*The European Organization for Civil Aviation Equipment*). There is no particular aspect with regard to human, economic, environmental and environmental damage, nor is it linked to land and sea applications. There are four *objectives* that address *safety*. Although not explicitly defined, *security* is assumed to apply to information safety.

2.14 IEC 61511 (Process Industry)

The Standard IEC 61511 *Functional Safety – Safety Instrumented Systems for the Process Industry Sector* is an internationally established standard consisting of three parts, first published in 2003. The current version was published in 2016. The standard covers the entire life cycle.

2.14.1 Content and Scope

The first part of the standard is normative while the other two are informative:

- Framework, definitions, system, hardware and application programming requirements (normative).
- Part 2: Guidelines for the application of IEC 61511-1 (informative).
- Part 3: Guidance for the determination of the required safety integrity levels (informative).

The concept *Safety Instrumented System* (SIS) is used to define safety-related control systems. An SIS can implement several safety features *Safety Instrumented Function* (SIF). The intention is that IEC 61511 should be useful to those who build an SIS by connecting multiple components. In the process industry, it is common to purchase off-the-shelf control systems (PLC), sensors and actuators to be responsible for functionality and off-the-shelf software. In this case, the standard IEC 61508 can be used by component manufacturers and IEC 61511 is used by the systems manufacturer.

The standard contains a number of technical requirements, all of which are found in Part 1. Section 1, Part 2 and Part 3, provide support for application of the Standard.

The life cycle is based on the lifecycle of the IEC 61508 standard.

The requirements for integrity (risk reduction) of the safety features are assessed according to SIL 1-4 (*Safety Integrity Level 1-4*) where SIL 4 sets the highest requirements. Both hardware and software are included.

The standard is not intended for use if the functional safety requirements do not match any level SIL 1-4.

The standard differentiates between application software and embedded software. Built-in software is provided by the manufacturer and is not available for user change. Application software is specific to the application and three different types can be identified:

- Fixed program language, FPL: Only possible to change using parameters.
- Limited variability language, LVL: programming language for industrial control systems with limitations in which features can be programmed.
- Full variability language, FVL: a general programming language with the ability to create desirable functions and applications. For applications written in FVL, reference is made to IEC 61508-3: 2010.

IEC 61511-1 contains requirements for application programming. Among other things, information can be found about:

- Application programming life cycle (section 6.3).
- Application Program Development (Section 12).

IEC 61511-2 contains guidance for the application of the standard. This includes support for application programming, including information about:

- Application programming life cycle (Section A.6).
- Application Program Development (Section A.12).
- Examples of development with function blocks (Appendix B).
- Application programming methods and tools (Appendix E).
- Examples of relay schedule programming (Appendix F).
- Application Programming Methods (Appendix G).

2.14.2 Application

Standard IEC 61511 is a sector-specific application of IEC 61508 primarily intended for the process industry.

Within the process industry, safety often considered in terms of protection designed in several layers. The process that can cause a hazardous event should be protected in several ways. The control system is just one of these layers. Mechanical protection, safety-critical control systems, warning and evacuation are other measures that can be used to achieve a tolerable risk level.

2.15 MIL-STD 882E SYSTEM SAFETY

The Swedish Armed Forces and FMV System Safety Methodology is based on the US Department of Defense (DoD) Military Standard *MIL-STD 882E SYSTEM SAFETY*. System Safety Methodology is described in the Armed Forces Manual System Safety (H SystSäk). Section 4.4 and Appendix B of the MIL-STD 882E are mainly replaced by this manual (H ProgSäk).

MIL-STD 882E describes how criticality rating of software should be part of overall system safety work. Based on system-level risk classification, a methodology describes the Software Criticality Indices (SWCI) based on the Software Control Category, SCC and Severity Category, SC. SWCI then becomes a system safety requirement for software development.

Software Control Categories		
Level	Name	Description
1	Autonomous (AT)	• Software functionality that exercises autonomous control authority over potentially safety-significant hardware systems, subsystems, or components without the possibility of predetermined safe detection and intervention by a control entity to preclude the occurrence of a mishap or hazard. *(This definition includes complex system/software functionality with multiple subsystems, interacting parallel processors, multiple interfaces, and safety-critical functions that are time critical.)*
2	Semi-Autonomous (SAT)	• Software functionality that exercises control authority over potentially safety-significant hardware systems, subsystems, or components, allowing time for predetermined safe detection and intervention by independent safety mechanisms to mitigate or control the mishap or hazard. *(This definition includes the control of moderately complex system/software functionality, no parallel processing, or few interfaces, but other safety systems/mechanisms can partially mitigate. System and software fault detection and annunciation notifies the control entity of the need for required safety actions.)* • Software item that displays safety-significant information requiring immediate operator entity to execute a predetermined action for mitigation or control over a mishap or hazard. Software exception, failure, fault, or delay will allow, or fail to prevent, mishap occurrence. *(This definition assumes that the safety-critical display information may be time-critical, but the time available does not exceed the time required for adequate control entity response and hazard control.)*
3	Redundant Fault Tolerant (RFT)	• Software functionality that issues commands over safety-significant hardware systems, subsystems, or components, requiring a control entity to complete the command function. The system detection and functional reaction includes redundant, independent fault tolerant mechanisms for each defined hazardous condition. *(This definition assumes that there is adequate fault detection, annunciation, tolerance, and system recovery to prevent the hazard occurrence if software fails, malfunctions, or degrades. There are redundant sources of safety-significant information, and mitigating functionality can respond within any time-critical period.)* • Software that generates information of a safety-critical nature used to make critical decisions. The system includes several redundant, independent fault tolerant mechanisms for each hazardous condition, detection and display.
4	Influential)	• Software generates information of a safety-related nature used to make decisions by the operator, but does not require operator action to avoid a mishap.
5	No Safety Impacts (NSI)	• Software functionality that does not possess command or control authority over safety-significant hardware systems, subsystems, or components and does not provide safety-significant Information. Software does not provide safety-significant or time sensitive data or information that requires control entity interaction. Software does not transport or resolve communication of safety-significant or time sensitive data.

Figure 2:5 Software Control Categories

Software Safety Criticality Matrix				
	Severity Category			
Software Control Category	Catastrophic (1)	Critical (2)	Marginal (3)	Negligable (4)
1	SwCI 1	SwCI 1	SwCI 3	SwCI 4
2	SwCI 1	SwCI 2	SwCI 3	SwCI 4
3	SwCI 2	SwCI 3	SwCI 4	SwCI 4
4	SwCI 3	SwCI 4	SwCI 4	SwCI 4
5	SwCI 5	SwCI 5	SwCI 5	SwCI 5

Figure 2:6 Software Safety Critical Matrix

The assessed criticality class (SwCI) is then used to determine what activities to be performed in the development of the software. Which activities (*Level of Rigor*, LOR) to be implemented depend on the nature of the technical system and should be agreed between industry and FMV.

SwCI	Level of Rigor Tasks
SwCI 1	Program shall perform analysis of requirements, architecture, design, and code; and conduct in-depth safety-specific testing
SwCI 2	Program shall perform analysis of requirements, architecture, design; and conduct in-depth safety-specific testing
SwCI 3	Program shall perform analysis of requirements, architecture and conduct in-depth safety-specific testing
SwCI 4	Program shall conduct in-depth safety-specific testing
SwCI 5	Once assessed by safety engineering as Not Safety, then no safety specific analysis or verification i required

Figure 2:7 Level of choice software related activities

For the selection of activities (LOR) and how the activities are to be performed, refer to AOP-52 (Ammunition) and *Joint Software Systems Safety Engineering Handbook*. See *section 2.16* and *2.17*, respectively.

2.16 AOP-52 (Ammunition)

The standard AOP-52 edition 1, *Guidance on Software Safety Design and Assessment of Munition-related Computing Systems* describes the entire system safety process and is referred to by STANAG 4452 *Safety Assessment Requirements for Munition Related Computing Systems* and MIL-STD 882E, Appendix B.

The standard, however, is not a requirements fulfilment document but should be seen as a guide and recommendation. The standard is an *Allied Ordnance Publication* developed by the NATO Standardization Agency (NSA). AOP-52 is not ANSI *approved* or DoD *adopted*, which means that the Department of Defense does not use this as a requirements document. FMV therefore does not require the AOP-52 standard to be followed by industry. For special application, see also the Weapons and Ammunition Safety Guide (H VAS).

The AOP-52 standard is limited to ammunition-related software and should be seen as a complement to other standards such as MIL-STD 882 and DEF-STAN 00-56. The standard describes the total activities to be performed in software development. Additionally, it specifies the link between the system safety process and the unique activities to be performed for software development. The standard refers to AOP-15 regarding how to define the acceptable accident risk for the ammunition.

Central to the standard is the information on how to define the criticality index *Software Safety Criticality Index* (SSCI). The Criticality Index is obtained by combining the accident definition according to AOP-15 Edition 3, *Guidance on The Assessment of the Safety and Suitability for Service of Non-nuclear Munitions for NATO Armed Forces* and *Software Control Categories*. This indicates how malfunctioning software can contribute to accidents. This then lays down the activities and requirements governing the development of software for the development, coding, testing and integration of software in the technical system.

2.17 Joint Software Systems Safety Engineering Handbook

The Manual *Joint Software Systems Safety Engineering Handbook* has been developed in collaboration between the Department of Defense (DoD), NASA, U.S. Army, U.S. Coast Guard, U.S. Navy, U.S. Marine Corps, U.S. Air Force, Missile Defense Agency and US Defense Industry.

MIL-STD-882E refers to the manual as a software development guide in safety-critical applications. The manual describes processes and software development activities and the manual's processes interact with system safety work for the technical system.

The manual describes that, during the planning phase, design requirements should be identified, process activities and test activities defined, which subsequently will be planned and implemented. The manual describes how the development process can be adapted based on different criticality classifications. For a certain selected criticality classification, a given set of activities is not described. Instead, the manual describes a large number of activ-

ities and requirements where a number of relevant activities are determined per project depending on the nature of the technical system.

The choice of activities (LOR) to be implemented depends on the nature of the technical system and is to be agreed between the development industry and FMV.

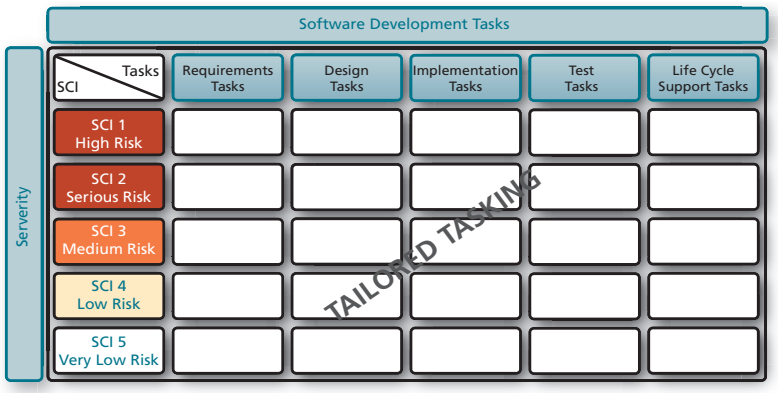


Figure 2:8 Principles of Selection of Development Techniques depending on criticality

Design Requirements	Process Tasks	Test Tasks
Fault Tolerant Design	Design Reviews	Safety-Significant Function Testing
Fault Detection	Safety Reviews	Functional Thread Testing
Fault Isolation	Design Walkthroughs	Limited Regression Testing
Fault Annunciation	Code Walkthroughs	100% Regression Testing
Fault Recovery	Independent Reviews	Failure Modes and Effects Testing
Warnings Cautions, and Advisories	Independent Walkthroughs	Safety-CriticalInterfaceTesting
Redundancy	Traceability of Safety-Significant Requirements to Design	COTS, Government Off-the-Shelf Input, Output Test, and Verification
Independence	Traceability of Safety-Significant Requirements to Code	Independent Testing of Prioritized Safety-Related Functions
Functional Partitioning	Traceability of Safety-Significant Requirements to Test	Functional Qualification Testing
Physical Partitioning	Safety Test Results Review	Verification and Validation
Design Safety Standards	Software Quality Assurance Inspections and Audits	Independent Verification and Validation
Design Safety Guidelines	Traceability of Safety-Significant Requirements to Hazards	Full Screening of All COTS Features
Design Safety Lessons Learned	Specific Software Language Requirements	
Full COTS Features Disclosure and Analysis		

Figure 2:9 Examples of some of the development requirements and activities described in the manual

2.18 NASA Software Safety Guidebook (NASA-STD-8719.13)

The *NASA Software Safety Guidebook* (NASA-STD-8719.13) was developed to provide specific information of and guidance on the process of creating and ensuring that software in safety-critical applications is sufficiently secure.

The manual addresses a broad target group such as system safety engineers, software developers, quality engineers, project managers and system engineers. In the introduction section the manual's provides guidance on which parts of the manual that are of particular interest to the different target groups.

The manual is intended to be more than just a collection of development methods and analyses. The goal is to open up new ways of thinking about software from a safety point of view. The manual points to things to look for (and watch out for) in the development of safety-critical software. The manual contains development methods, safety analyses and testing methods that lead to improved safety in the computer system. There is also a review of different programming languages.

The focus of the manual is the development of software in safety-critical applications. Much of the information and guidance is also applicable for the development of mission-critical software.

2.19 Def Stan 00-56

The United Kingdom military standard Def Stan 00-56 edition 4, *Safety Management Requirements for Defence Systems*, specifies special requirements for system safety operations. The standard will be applied primarily by developing industry in collaboration with the British military authorities. The standard defines system safety to include freedom from personal injury and property damage. It requires system safety operations to be carried out for the technical system throughout its lifetime.

The standard is divided into two parts. Part 1 specifies requirements for operations and Part 2 is a guide to Part 1. Part 2 also provides guidance for complex electronic safety systems.

Central to the standard is the concept of *Safety Case* that specifies the process to be performed to obtain safe technical systems. Through a defined *Safety Case*, detailed activities are detailed. The results of the completed activities are reported in a *Safety Case*. Notably, the term "As Low As Reasonably Practicable" (ALARP) is a statutory term in the United Kingdom.

The standard covers electronics and software as part of the technical system. Part 2 provides guidance on how to apply the requirements for activities under Part 1. There is a system safety section for systems that contain complex electronic devices that consists of both hardware and software. For this kind of electronics, potential hazardous events that it can cause or contribute to

shall be identified, mitigating and risk-reducing measures shall be taken, and evidence that errors and error probabilities are relevant be presented.

The criticality level of electronics and software shall be defined during development. Standards such as IEC 61508, RTCA DO-178C or Def Aust 5679 *The Procurement of Computer-Based Safety-Critical Systems* should be used. However, Def-Stan 00-56 contains no detailed description of how software development shall be performed.

3 Workflow Between the Armed Forces, FMV and Industry

This chapter describes the workflow from the Armed Forces' requirements, through development by industry, to FMV's handover of the technical systems to the Armed Forces.

3.1 Overall Process Chart, Different Perspectives

The Armed Forces technical systems usually contain large stored energy that is controlled and monitored by different computer systems. The road to safe technical systems begins with requirements from the Armed Forces to FMV which are then passed on to industry developing the system.

The workflow between the Armed Forces, FMV and industry is described in detail in *figure 3:1*. The main purpose of the image is to show important steps and deliveries between the different stakeholders, and the following sections explain the different steps. By stating requirements on performance, system safety, methodology and documentation at the right time, industry is given the opportunity to carry put its work in a structured and cost-effective way.

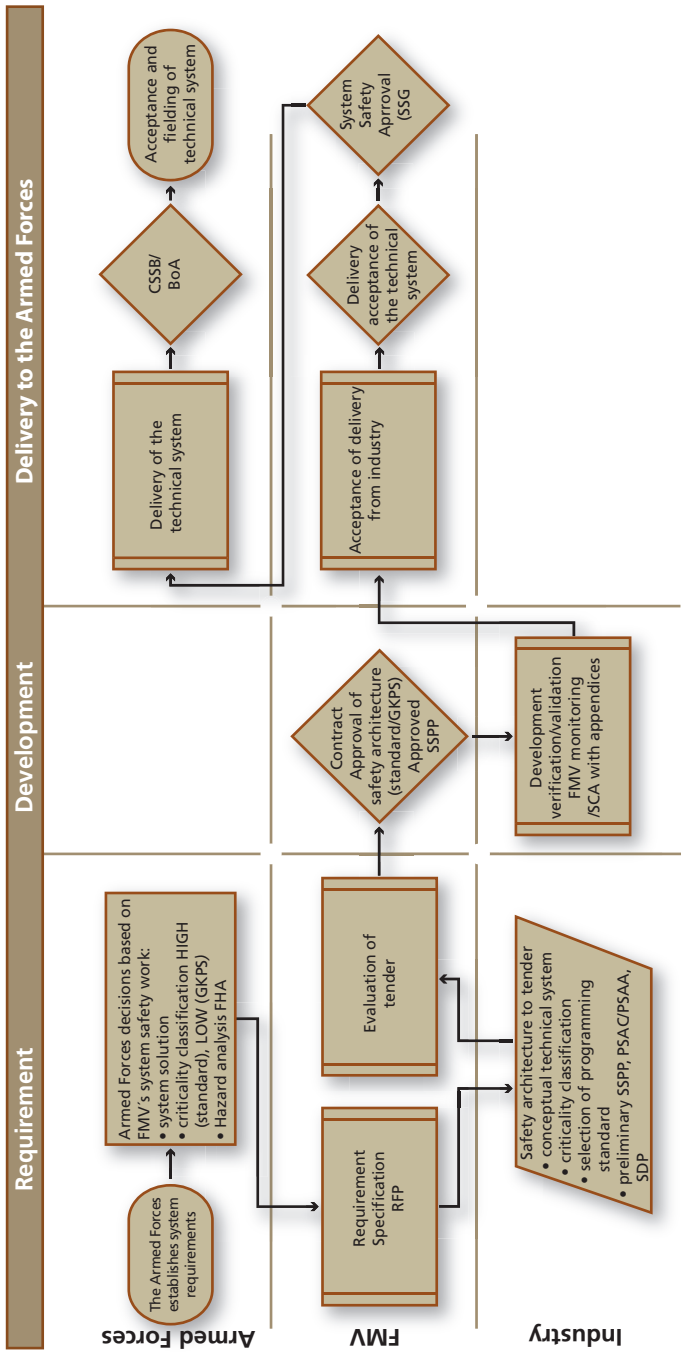


Figure 3:1 Simplified process chart showing the work flow of the Armed Forces, FMV and industry

The complexity when specifying requirements for technical systems containing computer systems can vary a lot. *Figure 3:2* below points to a number of different aspects that must be considered during the lifecycle of software for safe use in military technical systems.

Since there is no established software standard covering all of these aspects, the manual describes a way to handle them. This is intended, e,g, to ensure that the safety requirements are met.

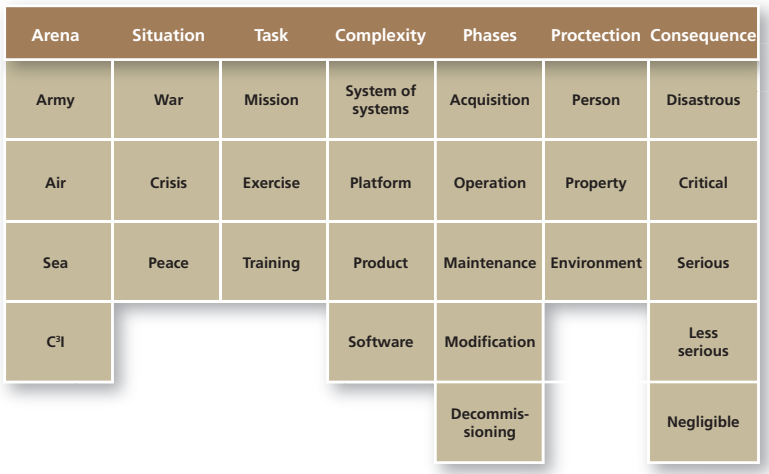


Figure 3:2 Different aspects to take into account the requirement for technical systems containing computer systems

The biggest impact on the requirement, and ultimately the total cost of software development, refers to the arena in which the technical system is to be used, its complexity and the consequences for the person, property and external environment in case of an accident. The above image can be supported in the dialogue between the Armed Forces and FMV in the early stages. It can also be used as support for a contractual review between FMV and the development industry.

3.2 The Armed Forces' objectives

The Armed Forces identifies the need for new ability, or retained capacity through material turnover, into one or more relationships, and whether the ability can also be used as support for civil society in peacetime. The Armed Forces establishes Bandwidth Objectives, which should include requirements for individual risk for personnel in current relationships. Corresponding requirements for property and external environment shall also be stated. Then the Armed Forces order a system work by FMV. The order also includes information on the conditions required for FMV to begin its system work. See *chapter 6*.

FMV performs system work, including from perspective function, technology, maintenance solution, commercial and legal. FMV hereby follows the requirements in *chapter 7* as part of the system work. System work comes out in one or more possible alternatives to concepts and maintenance solutions. Following the Armed Forces decision on what option to be realized, a development assignment is commissioned by FMV.

As part of FMV's Material Performance Preparation Work, FMV makes an initial system safety analysis called *Functional Hazard Analysis* (FHA) to identify the diminishing accident risks. The result is reported to the Armed Forces in order to ensure that the technical system will achieve the required skills.

The Armed Forces determine the Material Objective, which includes FMV's proposal for a tolerable risk level for persons, property and external environment for the technical system. The Material Objectives also include life and operating profile. Upon submission of the technical system, a return of claim fulfilment must be made to the Armed Forces.

3.3 FMV Initial System Safety Analysis

FMV, together with the Armed Forces, conducts a *Functional System Safety Analysis* (FHA) at system or subsystem level. As an initial risk analysis, the method is used only at the highest system level. FHA is mainly used to identify and classify system functions, as well as to assess the consequences of errors in these functions.

FHA is also used to identify external environmental and health related consequences due to functional-related errors. The result of the risk analysis is then used to determine which requirements are to be stated in the procurement documentation.

The following workflow can be applied:

1. Obtain a functional description of the system.
2. Identify hazards (top events) that may result from missing function, impaired function, malfunction or undesired activation of function.
3. Evaluate the most serious risks (top events) that may be associated with any identified malfunction of a function.
4. List requirements and suggestions for risk mitigation measures, such as when implemented, eliminate or reduce the risk (top event). Evaluate whether the identified risk mitigation measures can be implemented in hardware, software or similar, depending on the criticality of these features.

An FHA report should contain the following information:

- A description of the technical system and its main functions.
- Identified risks (top events) and their most serious consequences on personnel, property and the environment.
- The result of the risk analysis where the identified the accidents with their assessed consequences are listed in a Risk Log.
- Statement of requirements in the procurement documentation should be based on the use of an established software standard, applicable in the field of technology, or if the *Basic Requirements for Safety Critical Software* (GKPS) can be considered sufficient.
- A description of the risk assessment method used.

3.4 FMV Requirements in Specifications for Industry

Based on the results of the functional-based system safety analysis FHA, FMV uses the application matrix for an initial criticality classification of the technical system in *section 4.2*. Depending on whether the technical system can cause serious consequences (*HIGH*) or less serious consequences (*LOW*) for personnel, property or the environment, an initial criticality rating is made as shown in *figure 4:1*. The result of the criticality classification provides guidelines to FMV for which requirements to be specified for the development of the software. FMV shall also specify tolerable risk levels and indicate total operating time for the technical system. The detailed workflow is described in *appendix 3*.

3.5 Tender Submitted to FMV

FMV always specifies that the *Basic Software Safety Requirements* (GKPS) according to *chapter 8* shall be met and industry shall therefore always confirm this in the tender. If FMV has also specified that an established software standard be followed, industry must indicate in the tender which software standard that industry intends to follow and state reasons for this.

The industry's offer shall include a presentation of the safety architecture for the proposed technical system. Guidance and work flow developing a safety architecture are described in *section 4.3* and with examples in *appendix 4*.

The tender must always include a preliminary *System Safety Program Plan* (SSPP) and a *Development Plan Software* (SDP) shall also be attached. Where applicable, a *Software Certification Plan* (PSAC) or *Software Acceptance Plan* (PSAA) shall also be attached. Requested information in the specified documents is reported in the document list, *chapter 9*.

3.6 Contracts Between FMV and Industry

FMV places an order to industry based on the stipulated requirements in the specifications. The contract shall indicate, where applicable, which established software standard that the contracted company is committed to comply with.

During the FMV and industry contract review, the *System Safety Program Plan* (SSPP), The *Software Development Plan* (SDP) and, if applicable, e.g. the *Software Certification Plan* (PSAC) or *Software Acceptance Plan* (PSAA) are established.

The contracted industry is refining the safety architecture and, in the course of the contract, is able to justify the choice of criticality rating for the software based on the selected software standard. The contracted industry shall report whether any individual requirement in the GKPS is not applicable and/or otherwise fulfilled. The contracted industry and FMV will also agree on how the GKPS is to be verified. Agreements shall be documented in minutes between the parties.

If the contracted industry in its architecture work can show a system solution where the software has criticality rating ***LOW***, the parties can agree that *Basic Software Safety Requirements* (GKPS) is enough to follow. Such an agreement shall be documented in the contract between FMV and the contracted industry.

3.7 FMV's Monitoring of the Contracted Industry's Work

FMV should require monitoring of the contracted industry's development and manufacturing process. This can be done by contracting AQAP 2110/2210 standards. See *section 5.2*.

In connection with design reviews between FMV and the contracted industry, the system safety activities are monitored on the basis of agreed plans such as the SSPP and SDP, as well as, where applicable, the PSAC/PSAA certification documents.

During the development, the documentation shall be reviewed by FMV. Early validation of requirements and design should be planned with respect to system safety requirements. This is particularly important in the design of user interfaces where the operator's ability to identify a fault or hazardous situation is identified, and action can be taken before it can lead to hazardous events and accidents.

FMV shall also be given the opportunity to participate in verifying the technical system at the contracted industry.

3.8 FMV's Delivery Inspection of Technical Systems

Prior to delivery of technical systems to the Armed Forces, FMV reviews the contracted industry's System Safety Statement (SCA) and its appendices. As part of the review, it is included to verify that the contracted industry has fulfilled and verified the system safety requirements set by FMV, including the *Basic Software Safety Requirements* (GKPS). The documentation agreed in the SSPP, based on the document list in *chapter 9*, is reviewed and approved by FMV. FMV should participate in validation of the technical system of the industry prior to delivery.

3.9 FMV's Delivery of Technical Systems to the Armed Forces

Before FMV issues a System Safety Approval (SSG), a dialogue should be conducted with the Armed Forces on aspects of system updates of the technical system. These aspects are covered in *chapter 6*. When all out standing issues have been declared, FMV deliveries the technical system to the Armed Forces in accordance with usual delivery procedure.

3.10 The Armed Forces' Taken Delivery and Commissioning of Technical Systems

Based on FMV's documentation regarding system safety, the Armed Forces can issue *Central System Safety Decisions* (CSSB).

3.11 System updates During In-service

System updates can be initiated either by the developing industry based a product liability, or by FMV on behalf of the Armed Forces. This applies to corrections of software errors as well as function growth in the technical system or function adaptation to surrounding systems.

System update initiated by the developing industry to correct software errors is carried out by the developing industry in consultation with FMV. System updates in the form of software function growth, carried out by the developing industry under contract with FMV. Introduction into in-service systems are carried out in accordance with FMV Technical Orders (TO). Exceptions may exist if the Armed Forces is Design Authority for the system.

If the technical system contains previously developed software (PDS), FMV can choose to sign a maintenance agreement with the PDS provider. This is done to obtain information about updates, as well as to access these updates, including certain documentation.

Any change in the software of the technical system shall be considered a major change and shall be followed by new system safety decisions in accordance with H SystSäk.

Conversion of changeable parameters can be allowed if implemented system safety analyses have shown that this does not change the assessment of previously identified hazards. For example, a changeable parameter may change the danger area for a specific type of ammunition. In this case, it can be considered a minor change according to H SystSäk.

3.12 Software decommissioning in the technical system

FMV's letter of decommissioning shall also describe how the software, the development environment, software licenses and maintenance agreements are to be handled.

4 Safety architecture and methodology

The purpose of this chapter is to describe the importance of developing a well-thought-out safety architecture for computer systems based on the Armed Forces' need for technical systems. In support of this, a methodology for development and testing is presented. This involves all stakeholders regardless of life cycle phase and system level.

4.1 Application Matrix for Initial Criticality Classification of the Technical System

The Armed Forces' System Safety Handbook (H SystSäk) is a Swedish adaptation of MIL-STD 882E. Section 4.4 and Appendix B of the MIL-STD 882E are mainly replaced by this manual (H ProgSäk). The underlying standard AOP-52 does not apply, see *section 2.15*.

Below is the application matrix describing the link to the risk matrix used in H SystSäk to report remaining hazards for the system and possible consequences of hazardous related to the software.

Initial criticality classification is performed according to *figure 4:1*. By choosing an appropriate safety architecture, the criticality level of the safety critical computer system can be kept low, see *section 4.3*. Final criticality classification is carried out according to the architecture work according to *section 4.4*.

Software development is primarily carried out by applying general or sector-specific established software standards. A selection of software standards is described in *chapter 2*. The standards provide methods for reducing systematic errors during the development of the software.

If industry in its systems architecture work (with added safety function, diversity, redundancy, monitoring, etc.) can show that the system's risk of accident, which the computer system may affect, has low or negligible consequences for personal, financial

and / or environmental damage, the it is sufficient to meet is *Basic Requirements for Safety Critical Software* (GKPS) in *chapter 8*. The use of GKPS only must be agreed with FMV.

In cases where the product will be used independently and is CE marked, or will be CE marked, *section 10.1 – 10.3* of this manual can be applied This also applies to technical systems approved by other authorities, such as foreign forces or NATO agencies. See s*section 10.4*.

For initial criticality classification of the technical system implemented by FMV, *figure 4:1*. Application Matrix for FMV's initial criticality classification of technical systems shall be applied as described below.

For technical systems, the most serious hazards for persons, property and the environment are identified and analysed. For these hazards an estimate of their most serious consequences shall be made:

a. If the consequences are judged to be ***HIGH*** (high, serious or overage), FMV shall, in the invitation to tender, require industry to apply an established software standard in the development work in parallel with the *Basic Software Safety Requirements* (GKPS) as per *chapter 8*.
b. If the consequences are judged to be ***LOW*** (low or no consequence), FMV must, in the invitation to tender, require industry to always apply the *Basic Software Safety Requirements* (GKPS) according to Chapter 8. However, industry is always free to comply with an established software standard in parallel with the Basic Software Safety Requirements. (GKPS).

Application matrix in accordance with MIL-STD 882E for FMV's initial criticality classification of technical systems			
Conse-quence level	Description	Application	FMV´s initial criticality classification
High	Technical system containing safety-critical software where the consequence of an accident results is **catastrophic** for the person, the economy and/or the environment *(multiple or single deaths, total system loss and/or permanent environmental damage)*.	An agreed software safety standard is applied and the highest criticality require-ments are applied. FMV's documentation requirements are met	***HIGH*** FMV requires industry to comply with established soft-ware standards.
Critical	Technical system containing safety-critical software where the consequence of an acci-dent results is **critical** for the person, the economy and/or the environment *(serious and permanent personal injury, extensive economic and/or environmental damage)*.	An agreed software safety standard is applied and higher criticality require-ments are applied. FMV's documentation requirements are met.	
Serious	Technical system containing safety-critical software where the consequence of an accident results in **serious** consequences for the person, the economy and/or the environment *(serious but non-permanent personal injury, significant economic and/or environmental damage)*.	Agreed software security standards are applied and medium criticality require-ments are applied. FMV's documentation requirements are met.	
Marginal	Technical system containing safety-critical software where the consequence of an accident results in **marginal** consequences for person, economy and/or environment (less serious personal injury, less economic and/or environ-mental damage).	Basic requirements for software development for the lowest tolerable level of criticality are applied (GKPS).	***LOW*** FMV requires industry to use at least GKPS. *(However, the industry may choose to follow an established software standard)*
Neglible	Technical system containing software where the conse-quence of an accident results in negligible consequences for the person, the economy and/or the environment.	Basic requirements for software development for the lowest tolerable level of criticality are applied (GKPS).	

Figure 4:1 Application matrix linked to MIL-STD 882E for FMV's initial criticality classification of technical sys-tems

4.2 Computer System Characteristics

A computer system with its software has some unique features. Combinations of identical computer systems or different computer systems also affect system safety and availability.

4.2.1 Software Characteristics

Software has special features that differ from mechanical and electrical systems. Below are a number of system safety-related features listed.

Software:

- Contains only systematic errors and has no random errors, even though software's errors can be perceived as random e.g. due to random inputs.
- Errors are introduced when producing the requirements specification and/or when coding. These systematic errors are present in the design from the beginning but can cause errors much later in a changed usage profile or mode of use.
- Does not wear out over time.
- Different parts may require different criticalities, where the parent system set a the level of criticality of the software. The highest criticality level determines the overall criticality of the entire software, according to the method of respective software standards in *chapter 2*.
- Introduction of redundancy into the technical system by y does not reduce systematic errors, but allows for increased system-level availability (see explanation of software experience).
- Can be integrated into the technical system through diversity, which can reduce systematic errors (see explanation of software diversity and function monitoring).

4.2.2 Error Detection in Systems

For technical systems, it is important with functionality to detect random hardware failures. Errors in the system that are not detected in time can lead to errors in a safety-critical function. Fault detection can be accomplished in various ways and through different combinations of the following techniques.

- *Built-in test* (Build In Test, BIT) in the form of *Safety Check*, SK (Safety Check, SC / Power On Bit, PBIT) at boot-up.
- *Function monitoring*, Functional monitoring (FM/Continuous BIT, CBIT) in operation, see example in *figure 4:11* and *figure B4:8*.
- *Functional check*, FK (Functional Check, FC/Initiated BIT, IBIT) as a precautionary or malfunctioning pre-start or maintenance tool.
- *Integrity check*, checksum of software to ensure that it has not changed.
- *Error management*, Errors that arise during operation can be managed in such a way that the system continues to function with reduced functionality or performance.
- *Comparative*, in order to choose a redundant or diversified channel.
- *Watchdog*, a Watchdog in the computer system, along with structured and deterministic software, makes it possible to detect errors in software execution.
- *Voltage monitoring*, a computer system where voltage supply fails to meet defined requirements, may cause all or part of computer hardware to fail as intended. Voltage monitoring is handled by specially designed hardware circuits that can also provide conditions for the Watchdog feature.

4.2.3 Redundancy and Diversity in Computer Systems

By inserting two identical computer systems with the same software to solve the same function (replica), you can detect random hardware failures in the system. By comparing outputs from the computer systems, you can notice if the results differ too much from each other and thus determine if something is wrong. Thus, in a two-channel system, both channels must show compliance with full functionality in the system, see *figure 4:2*. Redundancy may exist for sensors, actuators, in computer systems with software as well as for outputs to operators.

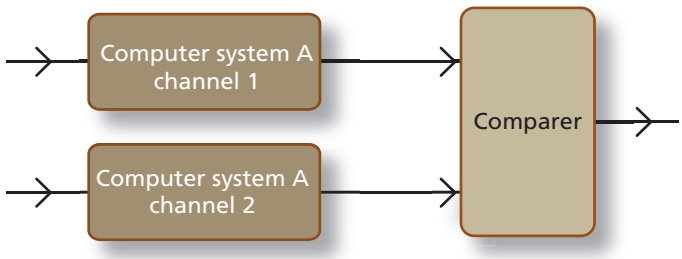


Figure 4:2 Redundant two-channel systems with identical computer systems and software with separate input channels

If there are three identical control systems, and one of these has a failure, it is often possible to determine which of these is wrong, that is, two computer systems show similar results, and the third is different from the others. Redundant voting systems can detect random hardware failures, thus increasing both availability and system safety in a technical system. See *figure 4:3*.

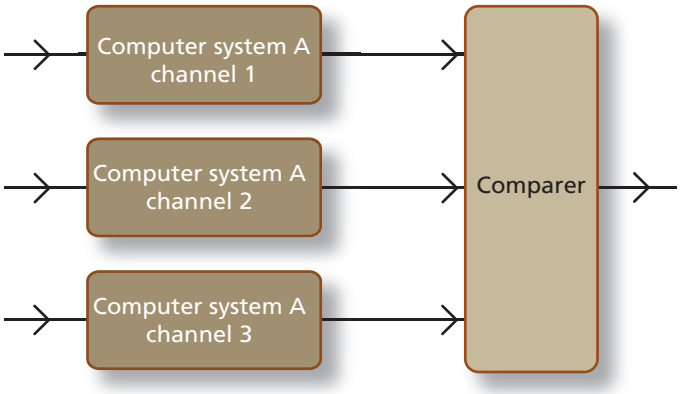


Figure 4:3 Redundant multi-channel system with three identical computer systems and software with separate input channels

A redundant voting system between three different computer systems with diversity, in addition to the random hardware failure, can also detect systematic errors in both in the software and hardware, see *figure 4:4*.

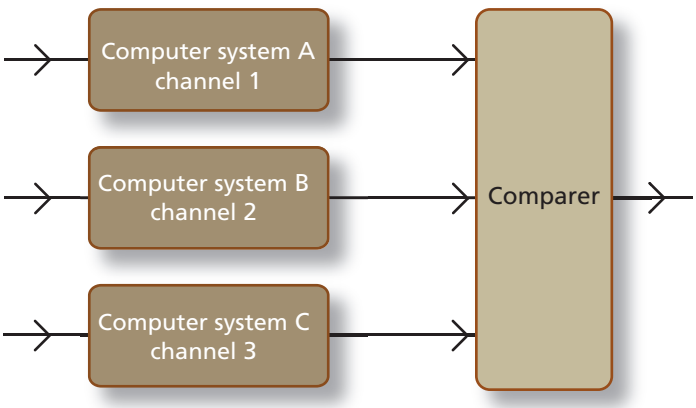


Figure 4:4 Redundant multi-channel system with three different computer systems and three different software and separate input channels

By introducing diversity with different software in computers for the same function, the possibility of detecting errors in the software also increases. Diversity can be included in the design in different ways during development, partly through functional diversity (i.e., not having a common requirement), and partly through

design diversity (no common errors in methods, tools etc.). The choice of which diversity approach appropriate for the current technical system should be carefully considered so it does not to lead to excessive complexity. It can also be difficult to show that diversity has also been achieved in order to mitigate systematic errors.

In multi-channel systems, the comparator becomes the most critical component.

4.2.4 Safe Mode for the Technical System

When errors are detected in a computer system, the fault condition can often be handled. One way is to put the function or part function in Safe Mode, known as *Safe State*. Safe mode often means that the system enters a state with reduced functionality or performance.

A safe mode varies from system to system and therefore cannot be defined in general. For each function, therefore, the fail-safe mode must be specified as detailed as possible, that is, the location where a hazardous event due to computer system failure can be prevented. A rotating system can assume safe mode when the mechanical brakes are activated and the control from the computer system is disabled. A firing system can assume safe mode when the power to the ignition circuit is disconnected. An airplane can assume safe mode on the runway when taking off is prevented due to a detected error in the computer system. If there are operating instances where a safe mode cannot be defined, this should be documented

Every conceivable error in the system needs to be analysed with respect to consistency and possible impact, as well as how the fault is to be found, how it affects the function and how it should be handled. A safe mode can be sufficient to reduce the consequence of a serious fault so that the hazardous event can be avoided. For critical functions, backup and / or emergency systems may be required. When rebooting, the system is based on a defined safe state

4.3 Safety Architecture, Methodology and Workflow

In order to determine the degree of software impact on the final technical system, the software needs to be classified as critical. This manual is based on the Swedish Armed Forces' Principle of Requirements for Relationships and Technical Systems and addresses where a safety architecture for the computer system is to be developed. Below is a model based on the same principles as in a fault tree analysis. Other models than the one described below can be found.

4.3.1 Accident Model

Accidents are often very complex events considering the causes and the indirect conditions that caused them. Every accident is also a unique event. An accident model can therefore never fully describe all possible accidents but only express a general picture. However, the accident model in *figure 4:5* can provide support to risk management.

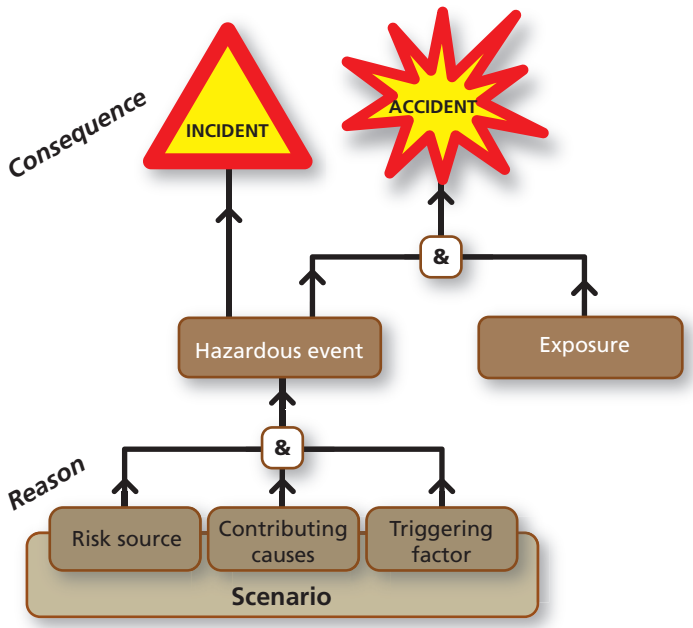


Figure 4:5 Simplified accident model according to H SystSäk

The following points explain the concepts of the accident model and put them in relation to each other:

- A *Risk source* is something that can harm a person, property or external environment through its dangerous properties
- A *Scenario* is a situation or system state where a hazard is present.
- A *Hazardous event* is an unwanted event that has occurred unplanned, by accident hazardous, that is without intent, and which can result in an accident if someone or something is exposed to the hazard.
- A *Hazardous event* always has one or more *contributing causes*. One or more of the direct causes that triggered the risk hazardous event are called *triggering factors*.
- An *Incident* is a hazardous event that does not result in any damage.
- An *Accident* is the result of a *Hazardous event* when someone/ something is exposed to the risk source and is then injured.
- The *Consequences* of an accident can be injury to a person, property or to the environment and are reported for each injury class in H SystSäk.

The term “Technical systems” stands for all types of platforms such as aircraft, ships and combat vehicles, as well as products such as medical devices and home appliances.

Technical systems usually include a control or monitoring function that is implemented using a computer system. By definition, the computer system is safety critical if it controls or monitors energies, which in an uncontrolled series of events can cause hazardous events and subsequently accidents. Computer systems in safety and emergency systems are also included in this category, even though they do not directly control hazardous sources.

In particular, the above definitions are risk sources the energies the computer system controls or monitors directly or indirectly. Deficiencies in the computer system's control or monitoring functions can be seen as contributing causes of hazardous events.

Performance requirements can be in contrast to the safety requirements because complex safety features can lead to reduced operational performance and availability. Therefore, in the design of a safety-critical computer system, the aim should always be to keep the system within a tolerable level of risk throughout its entire life cycle, without imposing restrictions on operational use.

In the following, measures are discussed to reduce the probability of hazardous events in a technical system. The requirement for a tolerable risk level for the technical system is broken down into the probability requirements for the respective hazardous event based on the requirement-based use profile. In system design, these requirements should be considered early so that a system structure can be obtained where there are reasonable conditions for detecting and verifying system safety requirements.

The Armed Forces require a tolerable risk level for individual accident risk based on a given operating profile and operating environment. The tolerable level of risk of an individual accident risk is linked to the probability of the hazardous event by defining the likelihood of exposure = 1. The contracted industry is then developing a technical system where the probability of hazardous events is so low that compliance with the tolerable risk level for accident risk is achieved.

4.3.2 Demand Breaking of Dimensioning Hazardous Events Requirements Breakdown

The purpose of developing a safety architecture is to reduce the criticality of the computer system in a technical system as far as this is practically possible, that is, a compilation of system safety and availability requirements linked to cost.

The choice of safety architecture should be done in such a way that it does not increase the complexity of the technical system design. A balance should always be aimed at to achieve key safety principles such as simplicity, independence and determinism. This facilitates understanding of the technical system structure, provides more favourable conditions for verification, and facilitates future system updates.

Software has special inherent features and is, in principle, impossible to write completely without errors for all modes of use and combinations of input data. By using function monitoring during the computer system's different operating modes, where comparison is made with expected results, and many random errors in the computer system can be identified early before leading to a dangerous error affecting the user environment. The use of diversity in function monitoring can also b predict and eliminate certain systematic errors.

Below is a model for how a breakdown of requirements can be carried out. Based on the entry requirement for the probability of a hazardous event (top event), a requirement breakdown is made in a general fault tree model to an assumed error probability for each base event.

The assumed broken down requirement for error probability in the base event may then be the entry requirements for selecting appropriate processes in the design work. For hardware there are calculation models that are able to predict error probabilities, but for software (systematic errors) this is not possible, but instead, the broken down requirement represents an input value for the selection of development methods with appropriate stringency.

The principle is to assume the most critical hazardous events that may occur in the technical system and to influence the design of the safety architecture so that the criticality of the computer system becomes as low as practicable. Early in the architecture work, a fault tree model can be developed for the technical system's most critical hazardous events.

The purpose of the breakdown of requirements is to identify the parts that will control the criticality level of the computer system, that is, both the error probability of random hardware failure and the stringency of software development methodology early in the work of the safety architecture. If independent safety features are introduced into the technical system, the level of criticality can also be lowered correspondingly to the safety-critical function, and thus also for the computer system.

A breakdown of requirements is performed for the most critical hazardous events (probability P2) so that the requirement for the probability of an accident (with probability P1) for the technical system can be based on a given operational profile and given operating conditions.

The breakdown of requirements can be presented in a general fault tree, see *figure 4:6*, and consists of at least one:

- Safety-critical function (with probability P4).
- Safety function (with probability P5).

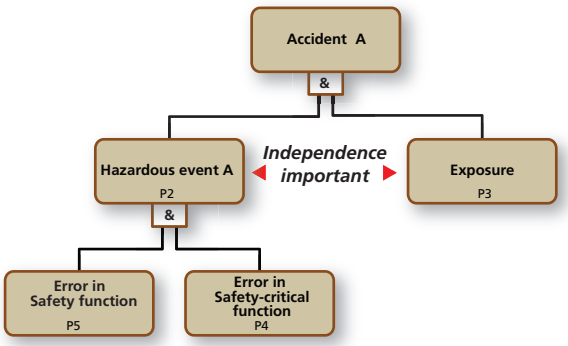


Figure 4:6 General fault tree to describe relationships in the accident model

The Safety-critical function (P4, computer system) refers to a system function which, in case of a dangerous error, can cause a hazardous event. It is in the safety critical function (P4) that the technical system's logic resides that controls or affects the risk source of and which gives the system its desired function.

The safety function (P5) means an added completely independent function whose sole purpose is to reduce the likelihood of a hazardous event (P2) occurring in case of a malfunction of the safety critical function (P4).

An accident A (P1, top event) occurs only if the hazardous event A (P2) occurs while a person, property or the environment is exposed to the hazard (P3) as, as the same time a failure in both the safety function (P5) and failures in safety critical function (P4)

occur. The exposure is affected by the operational profile, which is defined by the Armed Forces and may change over the lifetime of a technical system.

In the breakdown of requirements, therefore, initially a conservative probability of exposure (probability = 1) is applied, which gives the probability of accident = the probability of a hazardous event, that is, P1 = P2, see *figure 4:7*.

A simplified assumption of exposure of person, property and external environment:

It is initially assumed that the likelihood of exposure (P3) = 1. This assumption may sometimes be too conservative. If the assumption leads to unreasonable requirements regarding the likelihood of a hazardous event, an analysis should be conducted to define a realistic exposure level. The new assumed exposure level shall be agreed with FMV.

The requirement for a hazardous event A (P2) is broken down into safety function (P5) and safety critical function (P4), see *figure 4:7*.

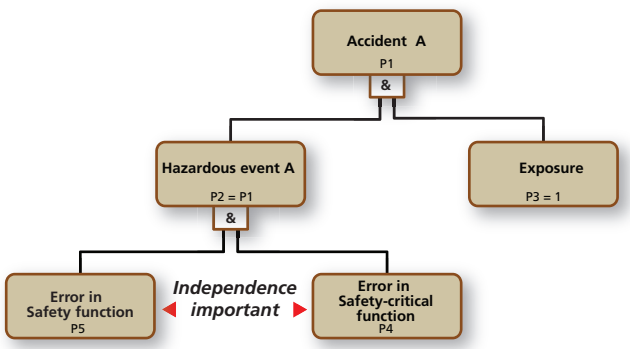


Figure 4:7 Breakdown of a hazardous event into safety function and safety critical function

The unit of probability must be defined by the Material Objective, i.e. per system per unit of time for death, property loss or serious environmental damage (only serious consequence, injury

class I in H SystSäk). Note that the error rate or error probability can be specified in different ways, such as per session, per hour, per year or per total lifetime.

In the design of the safety function, simplicity and known proven technologies should preferably be used. If the safety function can be realized with subsystems with extensive previous experience, and where error modes and error rates are known, this also simplifies verification of the requirements.

In the development of the safety-critical function, the goal will of course result in as low a likelihood of dangerous errors as is practically possible, but this can be difficult to verify if the safety-critical feature is realized in a computer system with many different software components that work together. From a system safety and verification perspective, it is usually a better strategy to allocate safety requirements to system safety features.

4.3.3 Requirements Break-down of the Hazardous Event

When breaking down the requirements, the safety-critical function can also be divided into a number of independent redundant diversified functions. See *figure 4:8* for a multi-channel system architecture, i.e. there must be a simultaneous hazard in both channels A1 and A2 in order to create a dangerous error in the safety-critical mode. In this way, the broken down requirement of the safety-critical function may in the ideal case be further broken down into independent sub-functions. A redistribution can then be done early in the system design if unreasonable or requirements difficult to verify have been identified.

Based on this breakdown of the safety function requirements, safety critical function, redundancy and diversity, the choice of safety architecture is then made to ensure that the requirement for hazardous event can be accommodated.

A complete independence is practically difficult to realize, the important thing is that the possible dependencies are identified.

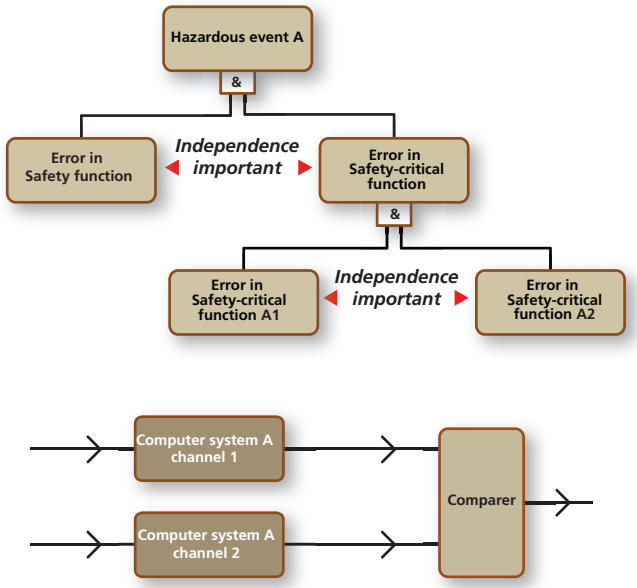


Figure 4:8 Safety-critical system, multi-channel redundant system (replica)

If a single-channel safety architecture is used, the breakdown of requirements should be distributed so that as much as possible of requirements are attributed to the safety function. This is because the safety feature is easier to verify than the safety-critical feature, see *figure 4:9* below.

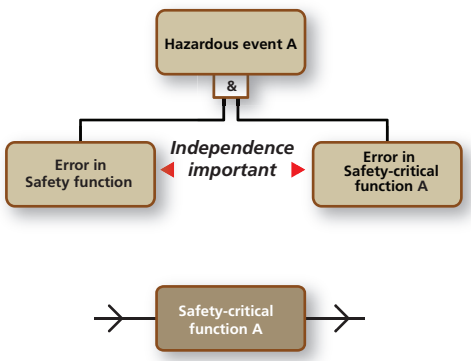


Figure 4:9 Safety-critical system, single-channel

4.3.4 Generic Fault Tree for Requirement Break-down of a Hazardous Event

In the following description, the example of a one-channel safety critical system is used. This is also applicable in each branch under a safety critical function in the multi-channel example of *figure 4:8*.

In the subsequent beak-down, the safety-critical function of three branches is specified in the fault tree. These are actuators, sensors and computer systems. All branches can individually, directly or indirectly, cause a dangerous error in the safety-critical function, hence the "OR" gate, see *figure 4:10*.

The **Actuator** in the fault tree symbolizes the computer system's connection to the risk source. It is via the actuator that the computer system controls or affects its connected energies. A dangerous error in the safety-critical function can directly be caused by a dangerous error in the actuator, that is, the computer system controls the actuator as intended, but the fault in the actuator results in a dangerous error in the safety-critical function.

Sensors in the fault tree symbolize the computer system's feedback on how the source of risk is controlled. An error in the sensor causes the computer system to receive an incorrect feedback of previously executed controls via the actuator. A fault in the sensor may result in the computer system controls the actuator in an incorrect way so that a dangerous error occurs in the safety-critical function.

The **Computer system** in the fault tree symbolizes both computer hardware and software. An error in the computer system may result in the actuator being controlled in an uncontrolled manner, which may result in a safety-critical failure in the safety-critical function.

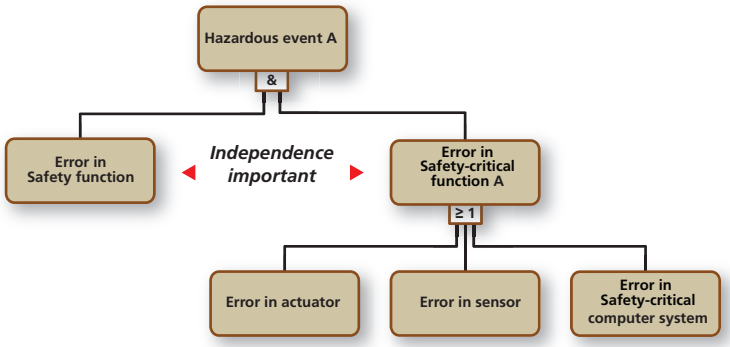


Figure 4:10 Reduced generic fault tree, single channel safety critical system

In order to further reduce the probability of a hazardous event, the next step will be monitoring/diagnosing the safety function, actuators and sensors. The purpose of the monitoring is to be able to detect errors in the respective monitored part before the error causes a dangerous error in order to further reduce the criticality level of the safety-critical computer system. The fault tree according to *figure 4:10* is then expanded as shown in *figure 4:11* below.

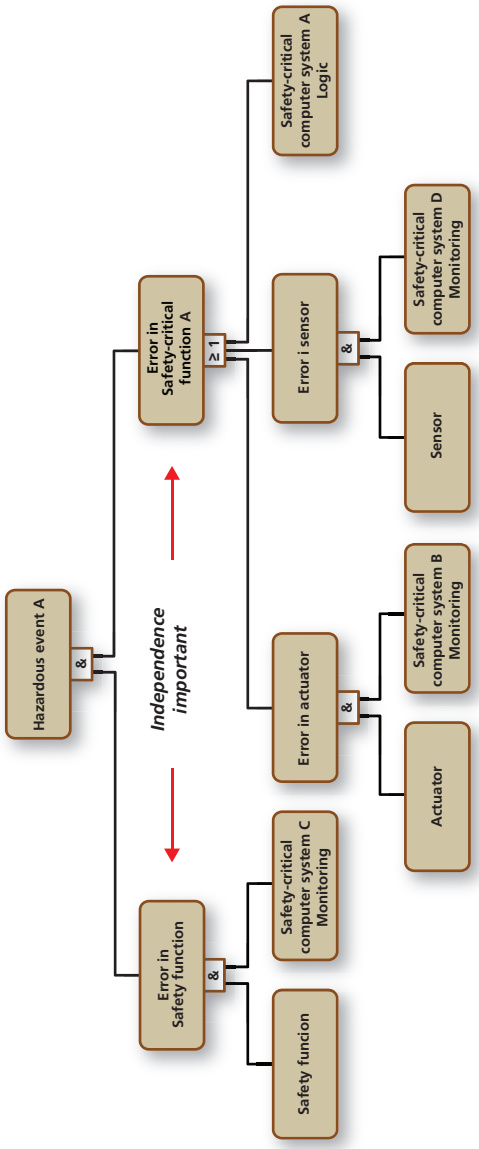


Figure 4:11 Reduced generic fault tree for a single-channel safety critical system with independent monitoring

The event *Safety Function Error* is defined here consisting of two basic events, a *Safety Function*, and an independent monitoring function (*Safety Critical Computer System C Monitoring*). Monitoring is provided for the purpose of detecting randomly dangerous errors in the *Safety Function*.

The event *Safety function failure* can only occur if there is a dangerous error in both the *Safety function* and that the independent monitoring (computer system C) cannot detect the hazardous error. *Actuators* and *sensors* are handled in the same way where the purpose of the independent monitoring is to detect random hazardous faults before they lead to that the *Safety-critical computer system A* will create a dangerous situation due to errors in the *actuators* or *sensors*.

A prerequisite is that the added monitoring in *Computer System C* can be considered independent of errors in the safety-critical *Computer Systems A*, *B* and *D*.

In the event of *Error in Safety Critical Function A*, the *safety-critical Computer System A* will actually contribute to the majority of the broken-down sub-requirement. It is in this branch that the system's logic is implemented and where the most complex functionality exists in both hardware and software. Therefore, from a system safety perspective, the goal is to keep the broken down requirement in this part as reasonable as possible to facilitate the verification of the requirement for hazardous event A.

If monitoring is added to detect random errors in actuators and sensors, the contribution of these two branches to the overall error probability of hazardous errors in the safety critical function can be further reduced.

When all hazardous events for injury class I (catastrophic consequence for person, property and/or the environment) have been broken down in each respective fault tree, the dimensioning fault tree can be identified. Possible safety features that can be realized have been identified in all fault trees, and a reasonable effort has been made at a verifiable level on the probability of a random dangerous error in the safety function. What remains with regard to the broken-down requirement for the safety-critical feature of the strictest requirement then becomes dimensioning in the development of the *safety-critical Computer System A*.

As *Computer System A* can be included in several fault trees for various hazardous events, this requirement will ultimately govern the development of *Computer System A*.

As both actuators and sensors can be found below the OR-gate, their contribution to the error probability must also be kept lower than the requirement for *Failure in Safety critical function A*. Depending on the broken-requirement for the error probability for actuators and sensors, requirements for independent monitoring in *Computer Systems B* and *D* are also provided. Similarly, the requirement for monitoring the safety function in *Computer System C* is also handled.

Note that the requirements of GKPS include both hardware requirements to reduce the likelihood of random errors, as well as requirements for stringent software development methods in order to limit the introduction of systematic errors. The probability of a deconstructed hazardous event requirement is only valid for the random hardware faults. The GKPS Critical Classification ***LOW*** defines the minimum subset of the requirements to limit the introduction of systematic errors.

Examples are given in *appendix 4*.

4.4 Criticality Classification of the Technical System

Prior to procurement, FMV has carried out an initial criticality classification according to section 4.1 for one representation of the technical system. If FMV's *Functional Hazard Analyzis* (FHA) Initial Critical Rating specifies ***LOW***, it is sufficient to set requirements according to the Basic Requirements for Safety Critical Software (GKPS). If FHA's initial criticality rating indicates ***HIGH***, requirements should also be set to apply any established software standard applicable.

The contracted industry will then develop a concept for a safety architecture and development process in order to balance and reduce the requirement for the probability of hazardous events. However, the break-down of requirements is independent of the selected software standard.

In order for the Armed Forces requirements for tolerable risk levels to be captured, FMV and the contracted industry will agree on the criticality level necessary for the computer system.

If the requirements or the chosen system solution is such that the broken-down requirement for the hazard event for a hazardous failure in a safety critical function can be assumed to have a probability of at least 10^{-1}/system/year, then only the basic requirements (GKPS) are sufficient. If the broken down requirement is lower than 10^{-1}/system/year, then an optional established software standard, applicable in the area of technology, shall be applied.

The basic requirements GKPS contain requirements for the design of the computer system to counter random hardware failures as well as requirements for the development methodology used in order to reduce the introduction of systematic errors in the hardware and software of the computer system for criticality classification ***LOW***. The estimated level of GKPS for error probability 10^{-1} / system / year is chosen to be below SIL 1, according to IEC 61508. See *table 4:1* below.

By introducing several independent safety features, the requirements for the safety-critical function can be lowered.

> To introduce several redundant independent safety features for the sole purpose of applying GKPS is not permitted.

For a technical system in continuous-mode operation, an estimated error rate of 10^{-1}/system/year (or 10^{-5}/system/hour), which corresponds to a total operating time of the computer system of 10,000 hours, i.e. about 1 year is assumed. If another total operating time is used, the error probability requirement will also be recalculated according to *table 4:1*. Below is a conversion table for the lowest estimated error probability (10^{-5}/system/hour) based on total operating time.

Table 4:1 Conversion table, application of GKPS for continuous operation

System in continuous operation Total operating time during service life	Minimum permitted probability of error in a safety critical function for criticality level ***LOW***
≤ 100 h	1×10^{-3} (p)

System in continuous operation Total operating time during service life	Minimum permitted probability of error in a safety critical function for criticality level ***LOW***
< 500 h	5×10^{-3} (p)
< 1 000 h	1×10^{-2} (p)
< 5 000 h	5×10^{-2} (p)
< 10 000 h	1×10^{-1} (p) (1 year continuous operation = 8 760 h) (1 year ≈ 10 000 h)
< 50 000 h	5×10^{-1} (p)
≥ 100 000 h	= 1

If the breakdown of requirements results in a lower error probability than 10^{-1}/system/year (10^{-5}/system/h) as described above, the basic requirements GKPS must be supplemented with requirements according to the agreed established software standard.

For a function in a demand mode technical system, such as emergency systems, rescue systems or systems with short operating hours, the assumed probability can be at least 10^{-1} / system / year (see IEC 61508, Part 1 Table 2). In this case, GKPS is enough. For this type of system, *section 4:1* shall not apply.

Figure 4:12 contains criticality levels for different software standards. A direct comparison between the criticalities of the different standards cannot be done. In a technical system, there may be different levels of criticality for included systems, as well as different standards used during the development work. If this is the case, FMV and industry must agree on how to apply a criticality matrix to the current project, agreeing on how different standards used in the development process relate to each other.

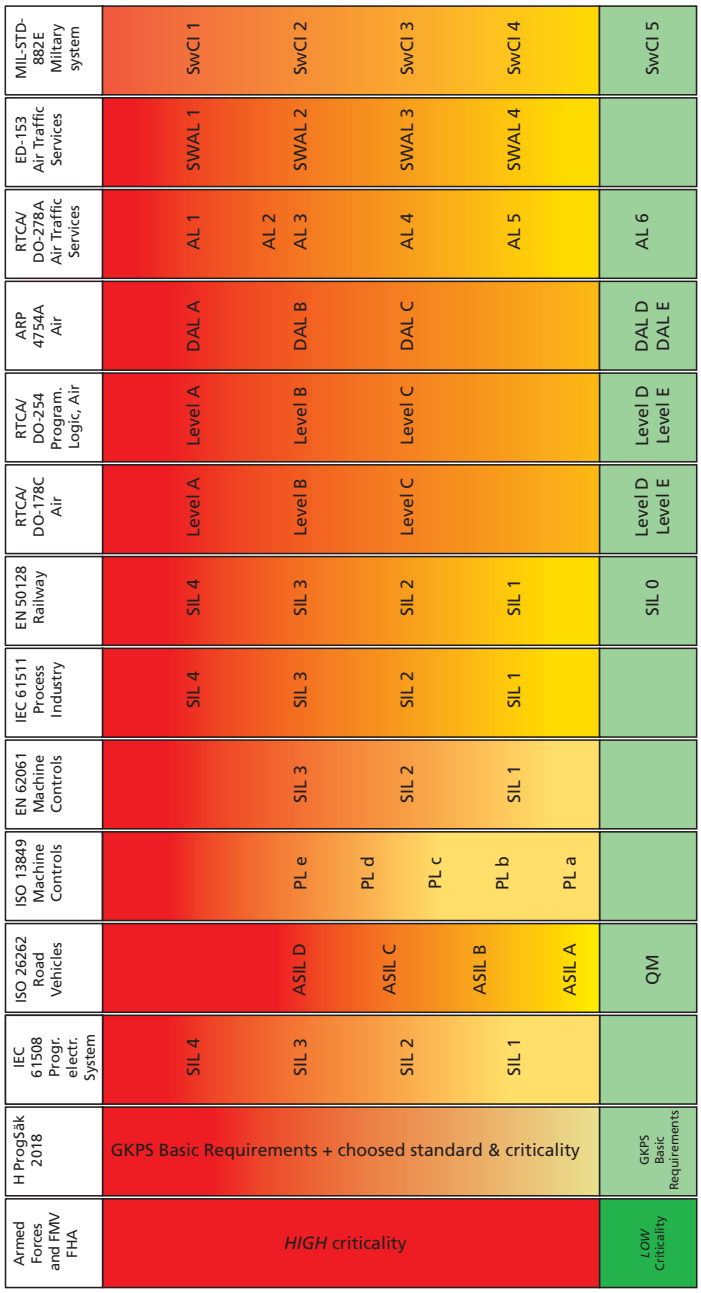


Figure 4:12 Criticality Levels for Different Software Standards

A direct comparison between the criticalities of the different standards cannot be done.

4.5 Data

The term Data refers to information, often stored as files or databases, which the software uses when it provides function or generates other information.

The following types of data can be identified:

- External information: For example terrain data (maps and more), road information (usage, properties), airspace information and airports.
- Calibration data: For example, engine values for a motor control systems to provide the correct operation of an, accelerometer calibration information and orientation for a navigation system.
- Configuration data: For example, the configuration status of devices included in a system, and thus what features a software should control and in what way.
- Parameters: Values of parameters that control software functionality, such as how data from sensors is interpreted and managed, which may vary for different system installations.

Data can be input to the functions of the computer system or software, and can also control what features the computer system (software) should have.

The above reasoning means that data has an impact on the computer system's function and hence its safety if the computer system influences system safety. You need to make sure that data that can affect the safety level is sufficient and of the right quality.

Within the air segment, there are requirements for quality assurance of *aeronautical data*, airspace data, etc., as that type of data is generated and handled by several different actors. How data can be quality assured is described in the standard RTCA DO-200B. The standard states requirements on the processes (including tools, etc.) that are used to generate and manage data.

The starting point is that data requirements are formulated based on their criticality. Data requirements are formulated with respect to the following characteristics:

- Accuracy
- Resolution
- Assurance level
- Traceability
- Timeliness
- Completeness
- Format.

Assurance level refers to the required level of work processes used to create and manage data. If the data has high impact on safety it higher requirement levels are placed on what activities to be performed and how they are documented and quality assured. Requirements are also made for the technical systems that handle data. Methods for formulating data requirements can also be helpful in other technology areas.

4.6 Maintenance Equipment

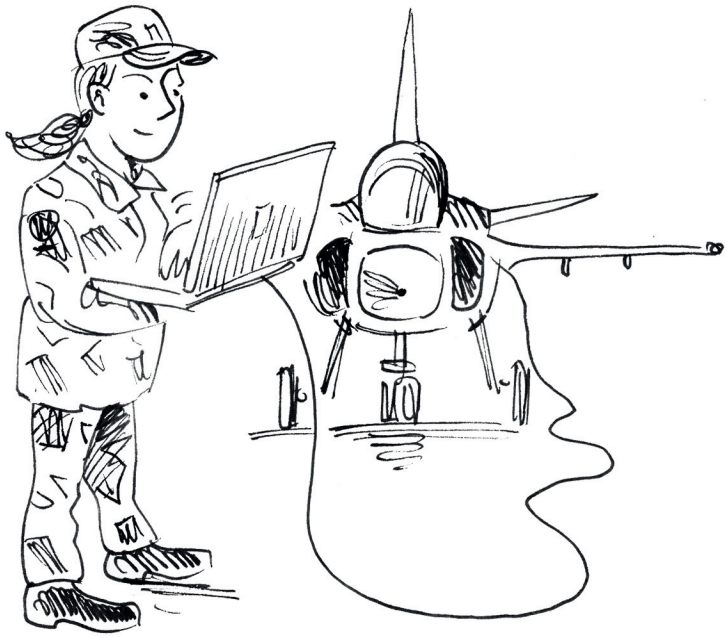
The technical system includes maintenance equipment and these are to be analysed as part of the system safety work. A well-designed maintenance concept facilitates both the development and maintenance of the technical system.

This manual provides specific guidance for externally connected equipment that can be used for software update, management of changeable parameters, loading of program code and data, logging, and troubleshooting.

Handling of classified parameters (using encryption devices etc.) is not part of this manual.

Maintenance equipment shall be developed at the same time as the development of the technical system, since adaptations of interfaces may need to be designed and adapted. Maintenance equipment, through its interfaces to the technical system, can both provide stimuli to and monitor test points in system func-

tions. Hazardous events that occur when using maintenance equipment together with the technical system shall be included in the initial system safety work. Hazardous events may also occur in case of failure of or improper use of maintenance equipment.



The maintenance equipment can also be used more extensively when verifying the technical system when the equipment can interact with system functions when the system is in operation. For example, the equipment may input errors to verify that the system's normal safety functions detect the error and activate possible protection features.

The maintenance equipment should also be able to read all system logs and save them in a database for later analysis of the occurrence of fault types and error states. However, in order to be able to analyse logs at a later time, there must be a defined system time that can be referred to a known time base, such as *Coordinated Universal Time* (UTC).

If the computer system is required with a so-called repetition function, for example, recording of an operator process, registered errors in the maintenance function should be directly linked to the repetition function.

In order for the maintenance equipment to be connected to, and thereby gaining access to, and exchanging information with the technical system it requires a system safety analysis to be performed and a system safety approval to be issued for this purpose.

Changes made to the technical system must also be logged by both the maintenance equipment and the technical system. This must also be considered from an information security perspective.

5 Lifecycle and Quality Management

Software, including program code, data, documentation and the development environment must be handled during the different phases of the technical system's lifecycle, including development, system updates and decommissioning. It is important to retain the development environment and competence of the project for future software updates. During the decommissioning phase, software licenses will be reviewed and any development environment will be phased out.

5.1 Operations Management System

All stakeholders must have a management system in order to conduct quality assurance activities that can be based on one or more different standards. Below is a selection of operations managements standards that specifically highlight software development. Together, these three standards constitute what may be referred to as a generally accepted practice for how software development can be described.

The standard ISO / IEC 15288 describes system lifecycle processes. For software development, standard ISO / IEC 12207 describes software lifecycle processes. Evaluation of processes in information technology are described by the ISO / IEC 15504 standard. The issues covered in these standards are applicable in many sectors of industry.

The three standards ISO / IEC 15288, ISO / IEC 12207 and ISO/ IEC 15504 relate to each other and are described in general below. There is the possibility of independent certification for the different operations management systems.

5.1.1 ISO/IEC 15288 Systems and Software Engineering - System Life Cycle Processes

The standard describes life cycle processes in general for different types of technical systems and constitutes a framework. Annex B provides connections to ISO / IEC 15504 Part 2. The purpose of the standard is to facilitate assessment of the life cycle process with support from ISO / IEC 15504th Annex E gives a comparison between the processes of ISO / IEC 15288 and ISO / IEC 12207. ISO / IEC 15288 refers to:

- ISO/IEC 15504 Part 2
- ISO/IEC 12207.

5.1.2 ISO/IEC 12207 Systems and Software Quality

The purpose of ISO / IEC 12207 is to be a software specialization of the general life cycle processes covered in ISO / IEC 15,288. The two standards are harmonized with each other so that they can be used concurrently. The level of ISO / IEC 12207 is relatively general and details such as specific methods and procedures are not included. Annex B provides connections to ISO / IEC 15504 Part 2. The purpose with this is to manage evaluation (process assessment) with support from ISO / IEC 15,504. Table B.2 in the standard lists all processes.

Annex D gives a comparison between ISO / IEC 15288 and ISO / IEC 12207 processes. ISO / IEC 12207 refers to:

- ISO/IEC 15288
- ISO/IEC 15504 Part 2.

5.1.3 ISO/IEC 15504, Information Technology

The purpose of the standard is to facilitate evaluation of the life cycle process. The standard consists of five parts:

- Part 1: Concepts and vocabulary
- Part 2: Performing an assessment
- Part 3: Guidance on performing an assessment
- Part 4: Guidance on use for process improvement and process capability determination
- Part 5: An exemplar Process Assessment Model.

Part 1 contains definitions. Part 2 contains requirements for evaluation, implementation, management and classification (including Level 1-5). Other parts are available as support. Part 5, section 4.2.1. lists all processes. ISO / IEC 15504 refers to:

- ISO/IEC 15288 from Part 1-4
- ISO/IEC 12207 from Part 1-5.

5.2 Quality Management of Defence Equipment

ISO 9001 is the most commonly used standard of quality management. The Allied Quality Assurance Publications (AQAP) are standards for quality management systems. The standards have been developed by NATO for quality assurance of defence equipment and can be used by all NATO countries and their partners. Requirements that selected AQAP standards are to be followed can thus be contracted with industry. The AQAP system is described in detail in STANAG 4107. There are currently two main types of AQAP standards; contractual, written as part of a technical specification, and for guidance only.

5.2.1 AQAP 2110, NATO Quality Assurance Requirements for Design, Development and Production

The Defence Standard AQAP 2110 is specifically aimed at suppliers of military technical systems, products and services. AQAP 2110 contains NATO's additional requirements in addition to ISO 9001 requirements for quality management in design, development and manufacturing. The requirement that AQAP 2110 shall be followed is applicable if the contracted industry already complies with the requirements of ISO 9001. Requirements to comply with AQAP 2110 provide, i, FMV e.g. with the right of monitoring industry's work during the implementation of the project.

5.2.2 AQAP 2210, NATO Supplementary Software Quality Assurance Requirements to AQAP 2110

The Defence Standard AQAP 2210 is intended to be used as a complement to AQAP 2110 in projects that also include software development. AQAP 2210 contains specific requirements for the supplier's quality management system and associated configuration management requirements. If FMV has contracted the supplier to comply with AQAP 2110/2210, these requirements will be compelling.

AQAP 2210 contains project-oriented requirements for managing the quality of the software development process. Both administrative and technical processes must be addressed to:

- Establish the visibility of the software development process
- Identify software problems as early as possible in the software lifecycle
- Provide data to quality control to quickly implement effective mitigation
- Confirm that quality is maintained in the software development process
- Provide assurance that the software produced complies with contractual requirements
- Ensure that appropriate software support is provided for activities at the system level and, as required by the contract, as well as to address safety requirements and the terms and conditions of the project.

In addition to the above, the use of an International Quality Agreement (GQA) may also be used. See also AQAP 2070

5.3 Configuration Management (ISO 10007:2003, IDT)

Configuration Management, CM is a methodology that applies technical and administrative control to configuration objects (Configurable Item CI) and their configuration information during the whole lifecycle of the system. Configuration management can be applied to fulfil requirements with regard to identification and traceability specified in ISO 10007:2003.

The methodology is used to establish, document and maintain a technical system's physical and functional requirements, performance, function, and physical components with its requirements, design and operational information. The choice of configuration objects and their interrelationship are based on the agreed system definition. Agreed criteria should be used when configuration objects are identified and the criteria should be selected so that

their functional and physical properties can be handled separately in order to achieve the overall performance of the configuration objects in end-use.

Configuration management in software development shall provide support to the operations and ensure that:

- The status and history of the software for a technical system are documented throughout its life cycle.
- There is an approved and frozen structure for the software where only approved changes are allowed.
- There is traceability for all events and decisions regarding everything that is part of a software system, such as deviation management, problem reports and any change requests.

Configuration information shall be relevant, traceable and updated.

5.4 Software Development Environments

During the implementation of a software development project, both a qualified development environment and competence to handle it are required. Tools include the equipment required for software verification, such as rigs, simulators, including system simulators. Data supply and configuration management equipment may also be required. It is important that there is an agreement with the supplier of the development environment, so that errors detected are reported and corrections can be obtained. Changes in the development environment may require renewed qualification when the software is updated. This is governed by the applicable development standard.

FMV's technical design responsibility also includes creating conditions for emerging needs and planned future system updates of the technical system. FMV may need to contract industry to maintain the development environment and skills during the technical system's life cycle at the appropriate level. It may even require premises for the equipment to be installed.

6 Requirements from the Armed Forces

This chapter describes the conditions FMV needs from the Armed Forces and is required to achieve adequate system safety in technical systems. The Armed Forces need to indicate the use environment, operating conditions, tolerable risk level and requirements for in-service management of the technical system. Certain conditions should be given by the Armed Forces before a procurement assignment is given to FMV. Responses to management requirements must be provided before the technical system is handed over to the Armed Forces prior to use as this may affect the contents of FMV's system safety approval.

6.1 Conditions and Requirements for the Development of Technical Systems

The System Safety Handbook (H SystSäk) describes the system safety activities carried out by the different stakeholders during a technical system's life cycle. It is the Armed Forces that decides on a tolerable level of risk for the technical system. Initial value for system safety requirements may be previous experience of the system's use and environment, as well as the operating conditions that apply during training, exercise and deployment in the field. Through appropriate architecture and motivated criticality classification of computer systems, requirements for tolerable risk levels for the new technical system can be met.

The Armed Forces shall define the abilities of the unit that is going to use the system. Based on these needs, FMV is developing a materiel objectives for the technical system, which is then established by the Armed Forces. This requires good co-operation between all parties involved, including the end users. This is essential in order for the right technical system to be procured and that the design, verification and validation as well as entering into operation can be carried out in a cost-effective manner.

In order for over-arching needs at the highest system level to be met, a comprehensive picture, the use environment, as well as operating conditions where the technical system is to be used and what tasks it will perform, is needed.

If the technical system is intended to be used in both military use environments and as support to society during peacetime, this should be apparent from the materiel objective. The following simplified examples can be used as a model to describe functional performance requirements of the technical system.

Simplified example

The Armed Forces need a new air defence system. The system shall be used primarily in combat, but it should also be able to support society during peacetime, for example, during major events if terrorist threats a represent. The Armed Forces need to answer if the air defence system is to fire on all aircraft or if it should only be possible to act against aircraft which are classified as hostile. Criticality rating of the computer system and the safety architecture will have a considerable impact on the development of the software for the system

The system safety requirements for the computer system are formulated with the aim that the final technical system will meet the Armed Force's requirements for tolerable risk levels. Specified functionality should be balanced against identified overall hazards. Before the Armed Forces order development of a system from FMV, correct and balanced system safety requirements shall be provided. Based on the requirements set by the Armed Forces with regard to tolerable level of risk, FMV can then break down the requirements for the computer system according to the model described in *chapter 4*.

FMV shall request operational experience from previous equivalent technical systems from the Armed Forces. FMV can also participate in user meetings, or to make direct contact with users to get a comprehensive picture of possible hazards.

FMV should request by the Armed Forces if there are directives regarding how software licenses and maintenance agreements are handled to match other purchased software. If the Armed Forces already has multi-user licenses on a variety of software, it's good if this is known when purchasing new software. Perhaps the Armed Forces themselves want to buy licenses and rights.

Another option is that licenses should be included in the procurement. Since there are several ways to go with licenses, it is important that FMV clarifies how this issue should be solved, so that the license issue does not become unnecessarily complicated or cost-effective.

The following conditions apply throughout the entire life cycle of the equipment from need to settlement. FMV will ask the Armed Forces by the Armed Forces on the following directions.

2.601.01-A FMV shall request that the Armed Forces specify the context, use and external environment and operating conditions of the technical system.

Comment: This applies to both military use and, where appropriate, support to society during peacetime.

2.601.02-A	FMV shall request that the Armed Forces define overall functional performance requirements for the technical system.
2.601.03-A	FMV shall request the Armed Forces to define the tolerable level of risk for the technical system throughout its life.
2.601.04-A	FMV shall request that the Armed Forces make operational experience available from previous similar technical systems.

6.2 Prerequisites for the Development of Technical Systems

During the development of the technical system, certain hazards can be identified which are difficult to mitigate to a tolerable risk level. With a dialogue between FMV and the Armed Forces, including the participation of the Armed Forces designated end-users, these problems can be addressed at an early stage.

In good time before handing over the system to the Armed Forces, FMV needs to be informed of which stakeholder that will be technically responsible.

2.602.01-A	A FMV shall request from the Armed Forces which actor is chosen to be the technical design authority for the system. *Comment:* If another stakeholder than FMV is the technical design authority, this must be stated in FMV's System Safety Approval (SSG).

6.3 Prerequisites for Handover and use

Prior to handing over the system, the Armed Forces shall notify FMV of how reporting and follow-up of operational experiences and deviations shall be carried out unless the regular reporting systems are to be used. In addition, FMV needs to know how the Armed Forces intend to introduce system updates during in-service. This is especially important to clarify before for possible deployment, and whether there will be other restrictions that need to be addressed in FMV's system safety approval.

System updates can be done directly by the contracted industry. This can also be done by FMV issuing a Technical Order (TO) that stipulates who is responsible and how the system update shall be carried out, including an instruction how to verify that the update was performed correctly. For all system updates, new safety decisions shall be taken.

2.603.01-A	FMV shall request that the Armed Forces have a deviation reporting system for technical systems where deviations can be reported. *Comment:* If other deviation reporting systems than the Armed Forces are to be used, FMV needs to beware of this.
2.603.02-A	FMV shall request that the Armed Forces comply with the instructions submitted regarding the operation, in-service use maintenance, and procedures for performing system updates on handed over materiel. *Comment:* If a stakeholder other than FMV is to be Technical Design Authority, the Armed Forces need to inform FMV of this.
2.603.03-A	FMV shall, on the basis of the Armed Forces' requirements, specify what restrictions and requirements that apply to personnel who handle, use / maintain or perform system updates on handed over materiel. *Comment:* This is especially valid during tactical deployments where system updates may need to be carried out by the Armed Forces own personnel.

6.4 Prerequisites for Maintenance

When using and maintaining technical systems, deviation reports may be issued. Follow-up of these and proposed measures can be handled in the System Safety Working Group (SSWG). If required, FMV may request the participation of the Armed Forces in SSWG in order to jointly find proposed solutions.

2.604.01-A	FMV shall request from the Armed Forces Deviation Reports for the Technical System. *Comment:* The information may be submitted to the System Safety Working Group (SSWG).
2.604.02-A	FMV shall request that the Armed Forces participate in the System Safety Working Group (SSWG).

6.5 Prerequisites for Decommissioning

The Armed Forces decide on decommissioning of a system. The decision shall also include technical systems (or parts thereof) with computer software. The decision should also include the resources used for support of the development and maintenance of systems such as development environments and other support systems. The following are included:

- Software development environment such as development tools, rigs, simulators, premises, user licenses and software updates.
- Personal resources agreements to maintain tools, rigs, simulators, configuration management tools, etc.
- Resources and data supply agreements.
- Secret information such as documentation, hard disks, and computers.

Please note that equipment and documentation can be found both at the Armed Forces, FMV and at the contracted industry.

7 Operational Requirements for FMV

This chapter contains requirements and guidelines for work within FMV with software architecture, drafting of documentation for procurement, follow-up of industry's work and maintenance. This chapter therefore states requirements for FMV's workmethods. Co-operation with the Armed Forces is described in ***chapter 6***.

The development of technical systems including large amount of software requires a well-structured approach to safety-promoting activities and techniques to avoid systematic errors such as incorrect requirements with can lead to a large costs to rectify. The cost of redesigning technical systems with safety-critical software tends to be high due to high costs for necessary testing and documentation.

7.1 FMV's Work During the Life Cycle

FMV's activities shall be planned so that proper system safety work, including software safety, is implemented in all life cycle stages of the respective technical systems during the concept, development, production, maintenance and decommissioning phases. FMV's work shall be in accordance with the current system safety management plan (SSMP) for the system, or the system safety plan for the specific project (SSPP), if applicable. The SSMP and SSPP shall include software safety activities. FMV is responsible for the SSWG, which will treat software in safety-critical applications. FMV requests the participation of the Armed Forces in the SSWG in order to jointly find proposed solutions and to allow the Armed Forces to make the necessary decisions.

The work at FMV with the system and in the individual project shall be subject to quality assurance requirements in accordance with FMV's internal working methods. See further H SystSäk.

2.701.01-A	FMV System Safety Management Plan (SSMP) shall address software safety requirements. *Comment:* FMV's SSMP shall address handle the Armed Forces' requirements for tolerable risk levels for all system levels of the technical system. In cases where FMV issues an internal SSPP for a project, it should also include software safety.
2.701.02-A	Software safety issues shall be handled by the System Safety Working Group (SSWG).

7.2 Concept Phase Before the Armed Forces Development Commission to FMV

FMV and the Armed Forces identify together solutions that exist in the form of technical systems and services based on requirements on operational performance.

Based on this work, FMV carries out software architecture work in which technical systems are designed that deliver the right functionality and meet applicable non-functional requirements and meet requirements for tolerable risk levels. During the software architecture work, FMV identifies and specifies what products (technical systems or products included in the technical systems) to be procured.

7.3 Development, Production and Acquisition

The contracted industry shall work in accordance with the mandatory System Safety Program Plan (SSPP), agreed with FMV during the contract review. Industry's development work shall also be subject to quality assurance in accordance with standards AQAP 2110/2210 unless otherwise agreed. See *section 5.2*.

At the contractual review between FMV and the contracted industry, minutes of meeting shall be issued. These minutes shall specify which software standard and what criticality level the

contracted industry will follow in the development of the computer system. The minutes shall also state that the contracted industry will comply with the GKPS (requirements in *chapter 8*).

The System Safety Program Plan (SSPP) produced by the contracted industry shall be in accordance with H SystSäk and it shall also indicate how the requirements in *chapter 8* will be met. In addition, a *Software Development Plan* (SDP) shall be attached. In cases where a software standard is required, additional activities will be added. The plans shall include activities throughout the software's lifecycle phases, such as requirements management, configuration management, coding practices, reuse, testing and documentation. The plans shall also cover how the various activities are monitored, reported and delivered.

FMV shall ensure that industry during the development uses a deviation reporting system where deviations in work processes and deviations in the expected functionality of the software are recorded and monitored continuously (bug reports, *problem reports*). The Deviation Reporting System shall enable analysis of individual deviations as well as statistical analysis of the total number of deviations. The system shall also enable identification of the software configuration status associated with the respective error report. This can be done using a configuration management system. The contracted industry can use the support systems, tools etc. that are normally use in their operations if they meet the requirements set by FMV.

Industry shall show that any errors generated by the development environment can be detected in subsequent testing.

FMV shall issue a System Safety Approval (SSG) for the complete technical system handed over to the Armed Forces. This assumes that FMV ensures that the contractors of present system safety analyses and risk mitigation measures that include software for the computer systems that are included. It also means that FMV shall ensure that the provider of software that has a criticality level ***LOW*** according to *section 4.1* shows compliance with the basic requirements of this manual.

For software that has an impact on system safety with a criticality level ***HIGH***, the industry, in addition to the basic requirements (GKPS), shall also show compliance with an established and with agreed FMV standard.

An independent review of the technical system shall be carried out by the contracted industry in accordance with the software standard applied for the development work and in accordance with H SystSäk for activities contained therein. Independent review is defined differently depending on which software standard is applied, what task to be performed and that criticality level to show compliance with. The most common interpretations that document reviews or other activities are conducted by someone who has not participated in the development of the software including its documentation.

When placing a contract, FMV specifies the extent of the contracted industry's work during in service and operation. FMV arranges for the industry to access the data from use in the Armed Forces and data from any other stakeholders needed for the analyses to be performed.

The system safety activities shall ensure that the system for software updates still meets the required level of risk specified by the Armed Forces. Examples of hazards connected to work on the system can be that protective devices or other components are removed when testing the system, which may expose personnel to hazards.

When a technical system is modified, renewed system safety work must be carried out. FMV identifies industries involved which are tasked to carry out system safety work for each sub-area to provide a basis for a renewed system safety approval (SSG) for the entire system, based on renewed system safety statements (SCA) from industry.

2.703.01-A FMV shall ensure that the SSPP includes software-related system safety activities prior to signing a contract.

Comment: The SSPP shall be written in accordance with H SystSäk and contain all necessary activities and methods for implementing the software safety work and, where appropriate, in accordance with the agreed software standard.

2.703.02-A FMV shall, for software with an initial criticality classification, HIGH agree with industry which established software standard, including level of criticality, applicable in the field of technology with which industry shall demonstrate conformity.

Comment: For criticality level ***LOW***, the basic requirements (GKPS) are sufficient.

2.703.03-A FMV shall ensure that the minutes from the contract review show any possible deviation from GKPS as agreed.

Comment: In the minutes it shall be stated that the contracted industry will meet other requirements according to GKPS. Several contract reviews can be completed during the implementation of the project.

2.703.04-A	FMV shall ensure that the contracted industry reports deviations that are significant for system safety identified during development and operation as well as the total number of deviations. *Comment:* At the time of delivery, the report shall include at least any deviations that are open or closed from verification testing. FMV shall include any open remarks in the system safety work, at least by taking the view that they do not cause system safety measures.
2.703.05-A	FMV shall specify that industry show compliance with the basic requirements (GKPS) in this manual for all software regardless of criticality level.
2.703.06-A	FMV shall ensure that industry can provide support for analysis and actions for emerging system safety issues during total system life. *Comment:* FMV shall contract support from the manufacturer according to the extent and time FMV and the Armed Forces need. Consideration shall be given to the characteristics and expected life of the technical system.
2.703.07-A	FMV and industry's system safety work including software safety must be completed and a system safety approval (SSG) to be issued prior to handing over the system to the Armed Forces.

7.4 Usage and System Updates

During use and maintenance, the user reports deviations in the function of the technical system. The reports are analysed by FMV and by the respective industry to identify the need for actions due to malfunctions or system safety issues.

In order to be able to analyse any occurred deviations, FMV needs to require the contracted industry to have access to development environments for testing.

In cases where the contracted industry intends to use previously developed software (PDS) from a subcontractor, FMV should require the contracted industry to ensure access to future system updates through specific maintenance agreements.

Implemented software may need to be changed. Changes shall be treated in the same way as software development with the same criticality level of the modified software. System safety work will be carried out and a new System Safety Statement (SSA) based on System Safety Statement (SCA) shall be issued.

2.704.01-A	FMV shall ensure that the contracted industry has access to the software development environment throughout the product's entire life cycle to the extent necessary. *Comment:* The Scope is specified in the contract given by FMV.
2.704.02-A	System Safety Approval (SSG) update shall always be made when changing or modifying a technical system. *Comment:* See H SystSäk regarding System Safety Approval (SSG).

7.5 Decommissioning of a Technical System

When decommissioning of the technical system, software related activities and equipment shall be disposed of. For example, development environments, any software licenses or software-based support systems shall be identified and discontinued. In order to enable decommissioning, supplies such as software and computers, including those in the development and test environments, shall be registered in relevant management support systems already at delivery from industry.

2.705.01-A	Supplies such as software and computers must be registered in relevant support systems in connection with delivery from industry. *Comment:* This also applies to supplies transferred into the governments possession, but is still in use by the contracted industry.
2.705.02-A	Decommissioning of a technical system (or part of a technical system) shall include the resources used for support for the development and maintenance of the systems. *Comment:* This includes software development environments, agreements for activities including personnel and supply of data etc.

8 Basic Requirements (GKPS) for the Contracted Industry

This chapter contains the basic requirements for software safety (GKPS), which must be met by the contracted industry in developing and updating software in computer systems regardless of criticality.

8.1 Requirements for the Development of Technical Systems

Software development in a safety critical computer system requires a structured approach and techniques to create robustness and to avoid systematic errors in the design.

The purpose of formulating system safety requirements for the software is that the final technical system must meet a tolerable risk level. The contracted industry will report to FMV the results of the system safety analyses and risk mitigation measures taken for the technical system. The governing document is the agreed Systems Safety Program Plan (SSPP).

For an initial criticality rating assessed as ***LOW*** (as shown in *figure 4:1*), all of the following requirements must be met by the contracted industry. For an initial criticality classification ***HIGH***, general or sector-specific established software standards are also applied. However, the contracted industry must always report in writing to FMV how the requirements of this chapter will be met. This can be done in the System Safety Program Plan (SSPP) or in the *Plan for Software Aspects of Certification* (PSAC) / *Plan of Software Aspects of Approval* (PSAA).

If the contracted industry in its preliminary criticality classification can show that the hazards that may be affected by the technical system software have low or negligible consequences for personnel, property and / or the environment, the basic requirements (GKPS) in this chapter are sufficient to meet

The basic requirements (GKPS) provide the conditions for achieving the required tolerable risk level. This leads to that industry identifies and corrects systematic errors during the development of the technical system. As a result, development and maintenance costs can be reduced throughout the life cycle while maintaining the required tolerable risk level over the life of the system.

8.1.1 Staff Competence Requirements

Personnel developing computer systems and software shall have a good knowledge of established development technology, safety architecture, methods, tools and programming languages, as well as having knowledge and experience of applicable software standards in the field of similar technical systems. For initial criticality classification ***HIGH***, additional competence is required according to the selected software standard.

2.801.01-A Roles including the required level of competence shall be agreed with FMV.

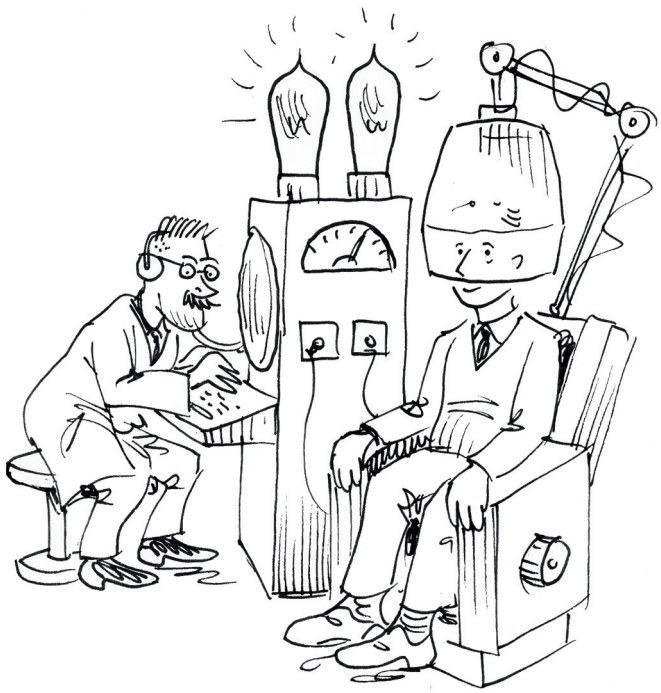
Comment: The competence profiles for personnel involved in the development of the technical system, such as project managers, technical leads responsible for system architecture, verification managers and independent auditors are documented.

2.801.02-A At least two persons should be familiar with the chosen system architecture.

Comment: The choice of system architecture based on the implementation of analysis of dimensioning hazards shall be known by at least two persons.

2.801.03-A The contracted industry shall designate a software safety point of contact.

Comment: This person ensures that the agreed work practices and methodologies for system safety work are followed and are responsible for the verification of the basic requirements (GKPS) and reports that these requirements are met.



8.1.2 Requirements for Operational and System Safety Management

The contracted industry's business management system aims at conducting quality assurance activities. The contracted industry shall apply AQAP 2110/2210 to give FMV special right of insight in the development of the technical system. In the contract between FMV and industry there shall be an agreement of which the parts of the standards to be applied.

Industry's system safety work during development is governed by the agreed System Safety Program Plan (SSPP), see *chapter 9*. This plan shall be approved by FMV before project start. Where applicable, software development may be described in a S*oftware Development Plan* (SDP) or in a PSAC / PSAA.

The scope of the *System Safety Program Plan* (SSPP) or *Software Development Plan* (SDP) depends on the complexity of the technical system and may need to be reviewed during the course of the project due to changing conditions.

In the case where the safety architecture requires initial criticality classification ***HIGH***, the contracted industry shall report an adjustment to the agreed established software standard. This can be reflected in the *Software Development Plan* (SDP), see *chapter 9*.

2.801.04-A	The contracted industry must comply with AQAP 2110. *Comment:* This applies primarily to the right of insight.
2.801.05-A	The contracted industry must comply with AQAP 2210.
2.801.06-A	The contracted industry shall issue a System Safety Program Plan (SSPP). *Comment:* The System Safety Program Plan (SSPP) shall also include required activities such as requirements documents, test plans and software development test procedures, as well as a description of agreed development tools.
2.801.07-A	The contracted industry shall state in the System Safety Plan (SSPP) how GKPS will be met.
2.801.08-A	The system safety analysis shall cover the computer system's impact on the entire system of the technical system during its life. *Comment:* The analysis shall be performed in an iterative way during the development phase, from requirements analysis to completed verification.

8.1.3 Requirements for Safety Architecture Design

When designing safety-critical computer systems, it is important to assume the technical system's most critical hazardous events for criticality class I, according to H SystSäk, and let these identified hazards influence the design of the safety architecture. If hazards belonging to criticality class I are not identified, class II hazards shall be used.

The aim is to be able to establish a safety architecture early in the development of the system so it can provide the conditions for achieving as low criticality level for the computer system and software as possible. See methodology in sections *section 4.1–4.4*.

Identification of possible hazards in the technical system is a work that begins early and is ongoing throughout the development phase. Before choosing a safety architecture, this shall be weighed against the required system safety and performance requirements. As a safety principle, simplicity in the design must be sought. The design principles should identify which error detection, fault tolerance and failure policies to apply. The report shall also include verification methods and acceptance criteria.

During the development work, new hazards can be identified and a review of the chosen safety architecture may therefore be necessary. Therefore, in the case of long development and operational periods, it is important to document the design decisions so that reassessment does not become person-dependent.

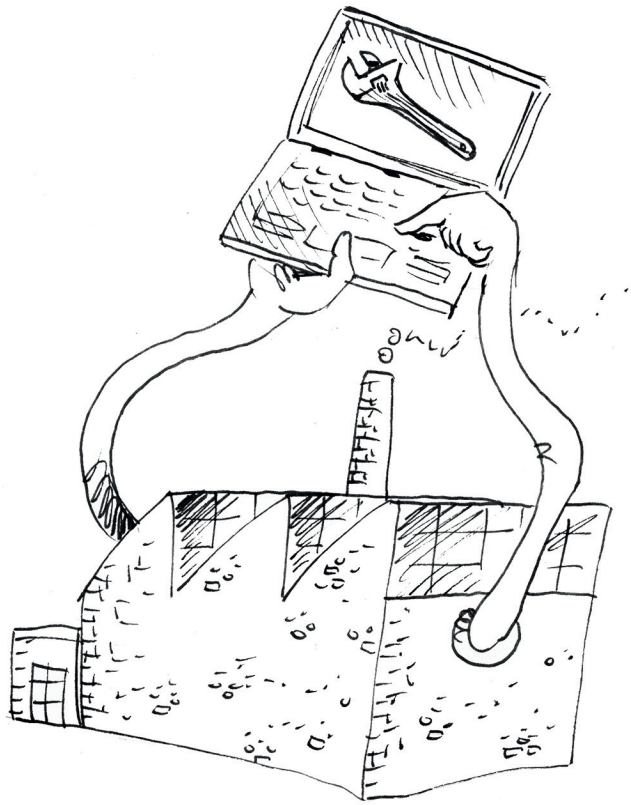
2.801.09-A	For the computer system, safety architecture and design principles shall be documented and reported. *Comment:* The contracted industry shall present a safety architecture according to *section 4.3*, which is reported in the System Specification / System, Subsystem Specification (SSS).
2.801.10-A	The design principles shall determine which strategies for error detection, fault tolerance and error safety that are applied. *Comment:* The statement shall state and justify the chosen design principles.
2.801.11-A	Design decisions regarding the selected safety architecture shall be documented and include the assumptions and justification for the selected design options.

8.1.4 Development Tools Requirements

Selection of development tools must be agreed with FMV before the System Safety Program Plan (SSPP) is established. The agreed tools for requirements tracking, configuration management, deviation reporting and test data tools shall work in a tool chain and be designed in such a way that information can be exchanged between FMV and the contracted industry. Details shall be stated in the agreement between FMV and the contracted industry.

During the development phase insight and transparency are important for both FMV and the contracted industry. This leads to increased participation and provides the conditions for making correct and joint priorities for actions during development and maintenance of the technical system.

2.801.12-A	Requirements Tracking Tools shall be used and agreed with FMV. *Comment:* The tool shall meet the requirements tracking process requirements of IEC.
2.801.13-A	Configuration Management Tools shall be used and agreed with FMV. *Comment:* The tool shall meet the process requirements for configuration management in accordance with IEC 12207.
2.801.14-A	Deviation Reporting Tools shall be used and agreed with FMV. *Comment:* The tool shall meet the process requirements for error reporting according to IEC 12207.
2.801.15-A	FMV shall be provided with information regarding requirements tracking, configuration management, deviation reporting and test data. *Comment:* FMV needs to ensure that there are conditions present for managing and reading the information.



8.1.5 Documentation Requirements

During the development phase, the contracted industry produces a large amount of documentation. The documents to be included in the delivery of a complete technical system shall be agreed with FMV. This should be documented in the System Safety Program Plan (SSPP), which is a mandatory document. The documentation may also consist of automated reports from the development tools, but the content should include required information. A proposed list of documents can be found in *chapter 9*.

A digital platform for the delivery of documentation needs to be agreed between FMV and the contracted industry. The platform should also provide support for configuration management of related documentation during the lifecycle of the technical system.

Please note that the time for development and maintenance can be so long that the access to information must be independent of the chosen platform. Consideration should also be given to the technical system's security management plan and information safety classification when choosing a platform.

2.801.16-A A document list shall be agreed with FMV.
Comment: This shall be defined on the basis of the document list in *chapter 9*. A delivery plan for the documentation shall be provided.

8.2 Operational Requirements for the Development of Technical Systems

During the development phase, important design decisions must be agreed with FMV. Earlier conducted safety analyses may need to be reassessed as the development progresses. It is therefore important that the contracted industry uses a system where deviations are recorded on a regular basis and that FMV has access to such information in order to be able to follow the development work. This will be agreed in the contract between FMV and the contracted industry.

In cases where deviation reports indicate that a reorganization or change of work methods is necessary, this shall be agreed and recorded with FMV at formal review meetings where all relevant parties are represented.

8.2.1 System Safety Analysis Requirements

System safety analyses begin at overall system level and are successively broken down for each subsystem according to the required system safety methodology.

During the initial system safety work, a safety architecture shall be developed to, if possible, reduce the criticality level of the subsystems.

The in-depth analysis of the design will provide a basis for improvements in the safety architecture so that the criticality level can be maintained at the assumed level or at lower levels.

A greater level of detail leads to increased insight as to what parts, and to what extent, these parts affect the identified hazards. Different methods of system safety analysis can be applied to investigate where and how the software can be involved and which, in the long run, can lead to accidents. For example, fault tree analysis (FTA) can be used.

An independent review shall be carried out of identified dimensioning hazards, as well as how these have also rule the choice of safety architecture. The person performing this independent review shall not have participated in the development work, see also H SystSäk.

Data handled by the safety-critical computer system shall also be analysed and have the criticality classification required in the current technical system. Data shall be classified on the same basis as for other software, that is, based on the effect it may have on the faults. See *section 4.5*.

2.802.01-A	Traceability shall exist between computer systems and its impact on the identified hazards belonging to the technical system. *Comment:* Requirements tracking in both directions can be reported in the safety architecture work.
2.802.02-A	The system safety analysis shall report the criticality level of included software in the technical system. *Comment:* Refers to analysis of the safety architecture.
2.802.03-A	The choice of safety architecture shall be motivated based on analysis of the dimensioning hazards.

2.802.04-A	Independent audits and reviews shall be performed in accordance with the agreed System Safety Program Plan (SSPP). *Comment:* An independent reviewer refers to a person who has not participated in the development work.
2.802.05-A	Data shall have the criticality classification required by the current technical system. *Comment:* Data refers to both static and real-time generated information.

8.2.2 Design Requirements

When developing software in a safety-critical computer system, there are many requirements and aspects to take into account. Below is a minimum amount of detailed requirements to be handled by the contracted industry for technical systems with an initial criticality classification ***LOW***.

Selection of safety features and function monitoring shall be done in such a way that this does not unnecessarily complicate the software system. Balance should always be done to achieve key safety principles such as simplicity, independence and determinism. This facilitates understanding of the software system's structure and verification for future updates.

There should be independence between critical and non-critical parts. However, complete independence may be difficult to achieve, and the identified dependencies shall be documented with an assessment of their impact.

There should be independence between critical and non-critical parts. However, complete independence may be difficult to achieve, and the identified dependencies shall be documented with an assessment of their impact.

Built-in test (BIT) coverage indicates how much of the hardware features or the hardware's possible malfunctions in the computer system the built-in test BIT is able to detect. BIT is a software that runs in parallel with other operating software in the computer system.

An important feature is logging of the system's internal states and events in order to detect malfunction of any possible events in the software system. Logging should be possible with different degree of detail in order to identify fault cases.

In the SDP, the methods that can be used to prove *Proven in use* shall be defined. The justification and criteria for using *Proven in use* for the cases used shall be documented.

2.802.06-T	Selecting safety features and function monitoring shall be done in such a way that this does not unnecessarily complicate the software system. *Comment:* A balancing should always be done to achieve key safety principles such as simplicity, independence and determinism.
2.802.07-T	Established programming languages shall be used in developing safety-critical software. *Comment:* Selected programming languages shall be reported to FMV together with design principles and safety architecture.
2.802.08-T	For each operational state, the technical system must be able to enter a safe state. *Comment:* For initial criticality classification, ***LOW*** safe states refer to states where control of executing parts of the system have been controlled / terminated in a safe way or where a rescue system has assumed control.

2.802.09-T All error states that may affect system function shall be logged in a format that is possible to evaluate.

Comment: There shall be traceability between the triggering fault situation / fault criterion and the state of the technical system has entered so that faults can be detected in the computer system. Logging can be done internally in the computer system or logged in to an external system.

2.802.10-T The technical system shall be in a safe state during boot-up.

Comment: This also includes rebooting of the computer system.

2.802.11-T At the start-up of the technical system, the software shall check that the defined safe state has been entered before critical parts of system are activated.

Comment: The level of safety can be checked by re-reading critical control or sensor signals.

2.802.12-T Unreasonable inputs, which, according to the system safety analysis, may affect the functioning of the system, shall be detected and disposed of so that a hazard does not occur.

Comment: Unreasonable data means all data outside the defined value range or data at the wrong time.

2.802.13-T Operator actions and presented information relating to safety-critical functions shall be recorded.

Comment: The ways to record this may vary based on system configuration, complexity and the situation.

2.802.14-T Built-in Test (BIT) shall contain Safety Control (SK / PBIT) during start up, Function Monitoring (FÖ / CBIT) during operation and Manual Initiated Test / Function Control (FK / IBIT) during maintenance.

2.802.15-T	BIT features for boot-up and maintenance shall not be inadvertently activated during operation of the system. *Comment:* Safety functions, such as blocking, shall exist so that handling errors can be avoided.
2.802.16-T	The independent watchdog function must be activated before the computer system can perform critical controls. *Comment:* The independent watchdog function is preferably implemented in hardware.
2.802.17-T	Watchdog (WD) must have a defined time window (that is, min / max time for WD triggers). *Comment:* Recovery of watchdog is performed by the software.
2.802.18-T	The Watchdog (WD) shall be subject to Safety Check (SK / PBIT) at boot-up and approved results shall be a criterion for activating the watchdog function.
2.802.19-T	Voltage monitoring shall be performed continuously on the power supply voltage of the computer system. *Comment:* A control signal from the voltage monitoring can be one of the criteria in the watchdog function.
2.802.20-T	Resource utilization at the first serial delivery shall be defined. *Comment:* The requirement relates to CPU, memory and communication links and should be no more than 50%.

8.2.3 Requirements for Software Development Environment

For long-term development projects, development tools will be used to update and change the computer software. A functional configuration management throughout the technical system's life cycle is therefore important.

When updating the software development environment, re-verification of the developed software shall be carried out so that no unintended changes have occurred in the function. If there is a regression testing environment, this may be used as partial verification that the updated development environment has not caused an undesired impact on the software.

Testing tools that introduce changes to the source code and are necessary to perform verification in the target environment should have such properties that the changes are retained in the source code after completion of the verification process. Analysis of any impact from the test tool shall be performed as evidence to prove that the verification is still valid.

2.802.21-A	The choice of software development environment shall be justified and documented for the technical system. *Comment:* Industry standards and past experiences shall be considered based on the chosen criticality level.
2.802.22-A	Audit history shall be reported for the use of the development environment. *Comment:* The development environment shall be under configuration management during the entire lifecycle of the software.
2.802.23-A	When updating the development environment during the development of the software, re-verification shall be carried out both of the development environment and the developed software. *Comment:* Approaches and criteria are described in the SSPP or in any other agreed document.

2.802.24-A Testing tools that introduce changes to the software shall not be used for verifying a specified software version.

Comment: If modifications are necessary for the test tool to be used, these changes shall be seen as part of the software version.

8.2.4 Verification Requirements

System safety testing shall be performed when verifying software included in a safety critical computer system. This activity is part of the work of fully verifying and validating the technical system. The intention is to verify that safety features and monitoring are properly implemented in the target environment and that they can detect errors before they can cause a fault that can lead to a hazard. System safety testing shall also include improper operation of the technical system and include all use phases such as training and maintenance.

Test code verification coverage is to verify the amount of requirements and code that are implemented and passed through test sequences.

The test cases used shall also be subject to an independent review by a person who has not participated in the development of the technical system.

System safety testing shall be performed with the established system version. Frozen system version refers to the version of the technical system to be delivered to FMV. This also means that the intended target environment shall have final status. In the event of uncertainties surrounding the target environment, FMV may need to specify this. Alternatively, industry can document what assumptions are made regarding the target environment.

The result of the system safety test should shall that access to functions intended for certain operating mode / system state cannot be entered during another operating mode / system state. All intended operating modes such as normal use, maintenance and training shall be tested. The documentation shall indicate which features are available, or are locked, in all different operating modes.

Adaptations and special test devices (test boxes) may be required to enable malfunction / error simulation in the target system's regular interfaces. The development of a test device should be coordinated with the development of maintenance functions for the technical system, c.f. *section 4.6*.

In connection with the system safety test, maximum utilization of the computer system should also be verified and documented. The purpose of this is partly to ensure that both CPU, memory and links have sufficient capacity in normal operation mode and partly to enable future functional growth. A rule of thumb may be that the maximum resource utilization margin is around 50% at first system delivery with full functionality.

Safety-critical functions should be deterministic, that is, they are executed in a predetermined order. This feature should be verified during system safety testing.

For safety-critical software, it is most likely that several updates will take place during the life of the technical system. If this is planned for during the development phase and also how to reverify, both the fulfilment of system the safety requirements is ensured and the costs of reverification can be calculated. If test sequences and evaluation are automated (regression testing) then the time for reverification can be reduced.

If previously developed software PDS is to be used in the technical system and when verifying this functionality will be based on previous experiences (*Proven in use*), the criteria for this must be documented in the SDP (*Software Development Plan*).

2.802.25-A System safety testing shall be planned, performed and reviewed and detected errors shall be resolved and approved.

Comment: The results are presented and any identified measures agreed with FMV.

2.802.26-A Test cases for system safety testing shall be subject to an independent review of a person not involved in the development.

2.802.27-A System safety testing shall be performed on a frozen system version of the technical system.

Comment: Frozen system version refers to the version of the technical system to be delivered, that is, even the target environment must have established status.

2.802.28-A System safety testing shall include error injection in all interfaces of the safety-critical signals identified in the system safety analyses.

Comment: The system safety test is intended to show that function monitoring can detect critical errors.

2.802.29-A System safety testing shall demonstrate that functions intended for a specific operating mode / system state cannot be entered under other operating mode / system state.

Comment: Also observe incorrect handling and operating conditions such as during training and maintenance.

2.802.30-A Maximum resource utilization of the computer system shall be verified and documented.

Comment: The requirement relates to CPU, memory and communication links.

2.802.31-A	Verification must be performed by correct order of program execution and at the right time for time-critical functions. *Comment:* Verification of execution order can also be performed using the development environment.
2.802.32-A	Test coverage (BIT) of safety features in the technical system shall be verified.
2.802.33-A	Use criteria for *Proven in use* must be agreed with FMV. *Comment:* The criteria are documented in the SDP (Software Development Plan).

8.3 Requirements for Delivery to FMV

Prior to each delivery of a new technical system, or a new updated configuration of the technical system, the agreed delivery activity according to the system safety plan (SSPP) shall be followed. This involves the contracted industry to issue a Safety Compliance Assessment (SCA) with the required attachments according to H SystSäk. For technical systems that contain computer systems, these attachments may include, among other things, the document list in *chapter 9*.

As the contracted industry shall always comply with and confirm that the *Basic Software Safety Requirements* (GKPS) according to the contract are met, this should be stated in the *Safety Compliance Assessment* (SCA) or in an attachment. If there are agreed exceptions from GKPS, this shall also be stated with reference to minutes from the contract review. If the contracted industry has chosen to follow an established software standard, this requirement is stated by referring to the different elements of the software standard where there is consistency between GKPS and requirements in the standard.

The contracted industry shall always ensure that all previous known errors in the software are taken care of and corrected. However, if there are still known errors in the software, the con-

tracted industry shall report this in the *Software Version Description* (SVD). In addition, the contracted industry shall, in the *Safety Compliance Assessment* (SCA), state that the required tolerable level of risk is met despite remaining known errors. FMV can then, based on this, approve or reject the delivery.

If the contracted industry has used dedicated software solely for the purpose of testing of verification of requirements compliance, and this software is not required for operational use, this must be included in the delivery.

2.803.01-A	A list of remaining known errors shall be issued for the delivered version of the technical system. *Comment:* As stated in the Software Version Description (SVD) according to the document list.
2.803.02-A	A In spite of the remaining known errors, the contracted industry must show that the technical system nevertheless fulfils the Armed Force's requirements for tolerable risk level.

8.4 System Update Requirements

Software updates can be initiated either by the contracted industry based on product liability or by FMV on behalf of the Armed Forces. FMV shall first consult with the Armed Forces regarding the introduction of the updated software into the system. All system updates to a fielded system are initiated by FMV by issuing a Technical Order (TO). Contracted industry issues a new Safety Compliance Assessment (SCA) and advises FMV on how system updates are to be implemented. If the change relates only to changeable parameters, no new Safety Compliance Assessment is required. The impact of these parameters on the technical system shall be analysed, verified and specified in the Safety Compliance Assessment for the version of the system

2.804.01-A	In the case of a new version of the technical system, re-verification shall be carried out. *Comment:* The need for reverification is determined after analysis of which parts are affected by the change.
2.804.02-A	In connection with a system update, a new system Safety Compliance Assessment shall be issued. *Comment:* Is not applicable to changeable parameters.

8.5 Requirements for Decommissioning of Resources at the Contracted Industry

The contracted industry has product liability under the Product Liability Act for Products and Technical Systems for 10 years after the individual product was placed on the market. Industry is responsible for maintaining the required technical competence of the products and for storing sufficient documentation about the product in case of a safety issues that need to be investigated. Longer product liability can be agreed between FMV and industry.

As long as the Armed Forces and FMV use the technical systems, software-related activities and equipment should remain with industry. This includes operating logs, development environments, test environments, and software-based support systems. FMV and the contracted industry can agree on this by agreements in accordance with *chapter 7*.

9 Description of Documentation

The contracted industry follows the document list specified in the chosen standard with regard to the agreed criticality level. If the basic requirements (GKPS) in ***chapter 8*** are deemed sufficient to meet for the technical system, the document list below is used. Information to be presented according to the document list must always be submitted to FMV independently of the selected software standard. Software documentation is a necessity in order to be able maintain a technical system

9.1 Document List for Basic Requirements (GKPS)

In cases where the contracted industry develops a technical system with criticality classification ***LOW*** and where the basic requirements are considered to be sufficient, the following document list can be followed. If the corresponding documents are produced within the framework of the system safety work, the information may be presented together. However, if the technical system consists essentially of software, information about software safety in some of the documents listed below may benefit from being issued separately.

The document list below represents a minimum subset that is needed to provide documentation for the analysis and implementation of system safety activities regarding the safety critical computer system (SSHA Computer System).

The document list should be seen as support in selecting a document structure and distribution of content between documents. The quality of the information to be presented regarding software safety is more important than that of all the different documents are issued The contracted industry can therefore propose merging certain documents and agree this with FMV in the System Safety Program Plan (SSPP).

Corresponding names of different documents contained in the document list are available in many different standards. The documents in the different standards have similar purposes and contain corresponding headings, although may vary. However, headings may be changed and agreed with FMV.

Table 9:1 Sample Document List for Basic Requirements (GKPS) in Chronological Order

Example of Document List for Basic Requirements (GKPS)		
Acronym	Name	Description
SSPP	System Safety Program Plan	Special activities for the development of computer systems are included in the project's SSPP, see H SystSäk.
PSAC	Plan for Software Aspects of Certification	If the system is to be certified by an authority, a PSAC shall be issued and reported to the certification authority.
PSAA	Plan for Software Aspects of Approval	A PSAA should be developed to clarify the acceptance and delivery criteria of the system before project start.
SDP	Software Development Plan	Describes how software development work shall be performed.
SCMP	Software Configuration Management Plan	Describes how the configuration management of the software is to be carried out and to what level of detail.
SVP	Software Verification Plan	Describes the test strategy and how verification will be carried out.
SVR	Software Verification Report	Summary of the results of completed verification according to the SVP.
SQA Plan	Plan Software Quality Assurance Plan	Quality organization and objectives.
SQA Record	Records Software Quality Assurance Records	Quality objectives Report.
SSS	System, Subsystem Specification	Specification of system safety requirements for the technical system.

Example of Document List for Basic Requirements (GKPS)		
Acronym	Name	Description
SRS	Software Requirement Specification	Specification of system requirements to be implemented in the software.
IRS	Interface Requirement Specification	Specification of the electrical and software interfaces.
SDD	Software Design Document	Specification of the software component with connection to overarching requirements.
STD	Software Test Description	Specifies at detailed level how the respective tests are to be carried out and in what test environment.
STR	Software Test Report	Test report from STD.
SVD	Software Version Description Document	Describes the current system release status regarding function and configuration (both for SW and FW / HW and deviations).
SSTD	System Safety Test Description	System Safety Test Program.
SSTR	System Safety Test Record	Test report from System testing
SSHA CS	Sub System Hazard Analysis Computer System	SHA for the computer system in the technical system. Included as underlying documents in the SHA for the entire technical system, see H SystSäk.

9.2 Description of Specific Documents

Information about what should be reported and how it can be documented in different documents can be found in different standards (a relatively comprehensive compilation of different documents is found in ISO 15289). The following describes the overall purpose of the respective documents in the document list and requirements for their basic content. The documents constitute only minimum content requirements from a system safety perspective and can be coordinated with the more common document structure for development.

9.2.1 System Safety Program Plan, SSPP

The purpose of the *System Safety Program Plan* (SSPP) is to describe the planned system safety activities. Issuing of the SSPP is project-related regardless of the system level. For technical systems, where the contracted industry is responsible for the system., the SSPP shall be approved by FMV before the activities covered by the SSPP are conducted by industry. In cases where FMV is responsible for the system, issuing of the SSP applies to FMV.

At the contractual review between FMV and the contracted industry, minutes shall be issued. These shall state indicate which software standard and level of criticality the contracted industry will follow in the development of the computer system. The minutes also state that the contracted industry will comply with GKPS (requirements in *chapter 8*).

The System Safety Program Plan (SSPP) issued by the contracted industry shall be according to H SystSäk and it shall also state how the requirements of *chapter 8* will be met. The SSPP shall include activities throughout the lifecycle of the software. The plan shall also cover how the various activities are monitored, reported and delivered.

The SSPP is used to evaluate the potential contracted industry's understanding and prioritization of the system safety activities required in the development of a technical system. In this case, this applies to a computer system for safety-critical applications. is meant. More information about the contents of an SSPP can be found in H SystSäk. Software-related system safety activities shall be addressed in the in SSPP.

In the SSPP there should be an agreement on important principles, such as the agreed software standard with regard to handling of redundancy and diversity.

9.2.2 Plan for Software Aspects of Certification

If the technical system is to be certified by an authority, a *Plan for Software Aspects of Certification* (PSAC) shall be issued and approved by the certification authority before start of project. The PSAC document shall show that the intended life cycle of the software in the computer system complies with the regulatory framework required for the criticality level that the software is intended to meet. The PSAC is a document defined in DO-178. There are other similar documents, such as *Plan for Software Aspects of Approval* (PSAA), defined according to DO-278.

The PSAC document should contain a description of the technical system in general and a description of the software with its respective functions, which features are implemented in hardware or software. It shall describe the intended allocation of the software and how the work to ensure that the safety requirements will be met is carried out.

The document should also describe how the work is going to be carried out during the different life cycle phases of the software. The data needed for the software should be described how it is developed and maintained.

In addition to this, the document shall describe how the work is going to be carried out in order for the certifying authority to get the transparency required for the technical system to be certified. Also various aspects that may affect certification shall be stated, how customization (tailoring), against selected processes is done, how the development environment is qualified, how to manage *Previously Developed Software* (PDS) and deactivated (dormant) code, and how software and data loading in the target system is to be done.

9.2.3 Software Development Plan, SDP

The purpose of the *Software Development Plan* (SDP) is to describe how the development of the software is to be implemented in the current project and the plan shall agreed with FMV. If the chosen safety architecture requires criticality level ***HIGH***, the SDP shall also state adaptation to the agreed software safety standard and the chosen criticality level.

The SDP shall contain the following:

- Identification and system overview
- Project organization and resources
- Document overview and connection to other documents
- References
- Used Software Development Processes and Methods

  Here are stated the standards and coding rules that the development shall follow, as well as the tools and software products and standard libraries to be used.
- Configuration Management (CM)

  Overview of the CM plan, how the software baseline is defined, reusable components (PDS), and how deviation management and corrective actions in the software are to be handled.

  The decision-making process for change management is also specified. Details of the CM plan are reported in the Software Configuration Management Plan (SCMP).
- Requirements Management

  Describes how traceability is handled regarding requirements and verification of requirements. A Summary identification of how requirement tags are defined and how system safety requirements are identified.Also describes methods and requirements management tools.

- Development and testing environment

  Describing the development environment and how the test environment is structured and in what steps the software tests are conducted and how feedback to the requirements management is done. Describe how the development and testing environment has been approved for use.

  If regression testing is used, the principle of how the approval criteria are specified in the regression test and how these approval criteria can be tested.

- Reviews

  Principles of how the software, test and approval criteria are reviewed and what audit steps are reported to FMV.

- System integration

  Describes how integration and testing are conducted in the actual target environment and which requirements to be verified in the target environment.

- Tools used

  Specification of used requirements tracking tools and how requirement tags are designed.

- Safety architecture

  Description of safety architecture.

- Used software standard

  Reference to which software standard is being used and adaptations to this.

- Software delivery process

  Description of the process prior to delivery of a new software lease to FMV.

- Software maintenance

  Describes the process for maintenance of delivered software and change management.

9.2.4 Software Configuration Management Plan, SCMP

The purpose of the *Software Configuration Management Plan* (SCMP) is to describe how the configuration management of the software is performed and to what level of detail it is done. This in order to be able to identify the different parts of the software at any time during its life cycle and, if necessary, restore a specific version of the software. The *Software Configuration Management Plan* (SCMP) shall be agreed with FMV.

The SCMP shall include the following:

- Identification and system overview
- Configuration management organization and resources
- A document overview and connection to other documents
- References
- Tools used

  Specification of tools used in the development environment, as well as version and issue management, and the formats in which information assets can be exported.

  Describes how the configuration management of the tools will be implemented.
- Configuration structure

  Specification of the configuration management objects.
- Configuration Status

  Definition of which metrics of program development, and when, to be reported to FMV.

  Examples of metrics can be:

  – The number of problem reports resolved
  – Problem Type (Specification error or coding error)
  – When problems were resolved and time spent on resolving.
  – The number of remaining problem reports.
- Configuration audit

  Definition of how to perform configuration management audits.

- Problem reports

  How error reports are handled, classified, and how the cause is identified.
- Change Management

  Process for how actions, based on identified causes or changed requirements, will be introduced into future system versions based on a defined *baseline* version.
- Software system version

  State the steps in the process for releasing a system version and how it is created from a defined *base-line*:
  - Scope of Test Readiness Review (TRR) and in which process steps this is to be performed.
  - Identity and document structure for current system version
  - Used tools and standard components for the system version.
- Delivery Process

  Specify the format of delivery to FMV and how the installation process and activities such as *Factory Acceptance Test* (FAT) / *Site Acceptance Test* (SAT) are to be implemented and assured on the target object.

9.2.5 Software Verification Plan, SVP

The purpose of the *Software Verification Plan* (SVP) is to describe the test strategy and how verification of the software is going to be performed.

The SVP shall include the following:

- Test organization
- Requirements for assumed independence between development and testing
- Identification of the test environment and associated versions
- Test methods intended to be used
- Test methods for safety-critical functions
- Test methods for safety functions

- How input and output data from test cases shall be recorded and made traceable
- Approval criteria for test cases
- Traceability of requirements.

The verification plan shall also state the parts in which the test equipment may affect the test results and an assessment of the consequences. Details regarding test sequences and approval criteria can be reported in *Software Test Description* (STD).

9.2.6 Software Verification Report, SVR

The purpose of the *Software Verification Report*, SVR is to summarize the result of the verification carried out in accordance with the *Software Verification Plan* (SVP).

In the SVR, all requirements from the SRS shall be collected and the results shall be traceable down to the tests performed. Details of tests performed must be reported in the *Software Test Record* (STR).

Deficiencies with associated problem reports shall also be reported in SVR.

9.2.7 Software Quality Assurance Plan (SQA)

The purpose of the *Software Quality Assurance Plan* (SQA) is to describe the organization and goals for the quality work. The SQA shall also describe how the software quality is to be ensured, the metrics to be used and how the monitoring and verification of these are to be reported, see AQAP 2110/2210.

9.2.8 Software Quality Assurance Records (SQAR)

The purpose of the *Software Quality Assurance Records* (SQAR) is to report results from completed quality activities according to the established *Software Quality Assurance Plan* (SQA).

9.2.9 System, Subsystem Specification (SSS)

The purpose of the *System, Subsystem Specification* (SSS) is to specify all the requirements for the technical system including system safety requirements.

9.2.10 Software Requirement Specification (SRS)

The purpose of the *Software Requirement Specification* (SRS) is to specify the system requirements to be realized in software and that are traceable towards the system requirements. Derived requirements shall also be specified.

9.2.11 Interface Requirement Specification (IRS)

The purpose of the *Interface Requirement Specification* (IRS) is to specify the interfaces of the technical system. Particular focus is that safety-critical signals are identified and which test and monitoring points are available.

9.2.12 Software Design Document (SDD)

The purpose of the *Software Design Document* (SDD) is to specify, for the respective software component, the implemented functions in relation to the SRS requirements. The *Software Design Document* (SDD) is the minimum level of detail for software specification. Scope and depth shall be agreed with FMV.

The SDD shall include the following:

- Safety-critical features and safety features shall be marked in a special way with associated error handling and be traceable to system requirements.
- All internal and external interfaces shall be specified.
- Connection to Recycled Components (PDS) and standard libraries shall be specified.

- Safety-critical data should be described.
- Overarching architecture, and principles for execution and data exchange as well as other design rules should be specified.

9.2.13 Software Test Description (STD)

The purpose of the *Software Test Description* (STD) is to specify, at detailed level, how the respective test cases are to be performed. The STD can be included in the SVP for smaller systems, but it is recommended to keep these documents separated.

All system requirements and derived system requirements shall be traceable to test cases with defined approval criteria.

Testing of safety-critical features and safety features of the software shall be clearly described.

Requirements covered by automatic and / or manual tests shall be specified.

9.2.14 Software Test Report (STR)

The purpose of the *Software Test Report* (STR) is to report completed tests according to the *Software Test Description* (STD) above.

The STR shall include the following:

- Summary

  A short summary of the results of the completed test activities. If deviations have been identified during the test, these should be stated in the summary with the associated reference to the problem report.
- Identification and system overview
- Document overview and connection to other documents
- References

- Test objects and system version

  A brief description of the test object configuration and status with associated references and possible changes to established test programs.
- Testing Resources

  Describes where and when the test was carried out and with which personnel the test was carried out.
- Test equipment

  Specifies the test equipment used. All test equipment must be registered with the accompanying design documentation for both hardware and software.
- Summary of test results

  A detailed summary of the number of approved or non-approved tests.

  The unapproved tests shall be identified with the requirement designation and the associated problem report in order able to analyse the deviation.
- Test scope

  Stating test coverage regarding system requirements including derived system requirements and requirements for automated and / or manual tests.
- Requirements Tracking

  A compilation of the requirements tracking matrix with requirements and links to any partial test verifying fulfilment of a requirements.
- Log data and test results

  Traceability to log data and test results shall be provided. All test results shall be saved in such a format that a review of an individual test, including results, shall be possible to re-implement with reasonable effort.

9.2.15 Software Version Description Document, SVD

The purpose of the *Software Version Description Document* (SVD) is to describe the current software version's status regarding functionality against the system requirements and configuration. Furthermore, it is stated what new features are added and any deviations in the software that have been corrected since the previous software version.

Remaining known deviations shall be stated with reference to the associated problem report and any restrictions due to these.

The SVD shall include the following:

- Identification
- Document overview and connection to other documents
- References
- Software Version

  Description in tabular form of software components with version number for the current software version. The initial PDS shall be specified with the current version and checksum.
- Introduced changes

  Description of new and changed functionality as well as corrected errors since the previously delivered software version. All changes shall be tracked against requirements and / or problem reports along with documentation showing completed analysis and test of the change.
- Remaining known errors

  The remaining known errors shall be traced to the specified problem report and reported in at least three main groups:
  - Error with system safety impact and possible additional restrictions in use or maintenance.
  - Errors affecting functions and need for additional instructions for use and maintenance.
  - Other known faults or malfunctions.

9.2.16 System Safety Test Description (SSTD)

The purpose of the *System Safety Test Description* (SSTD) is to describe how verification of safety features in the technical system shall be performed and be traceable to the system requirements. It shall be possible to detect System errors all before they lead to a situation that cannot be controlled by the software system.

System safety testing shall be performed on a complete target system after the system version is frozen. Should any change be made to the system version, ie. hardware and / or software after the system safety tests have been carried out, re-testing shall be carried out. The re-test is preferably carried out together with FMV.

The SSTD shall include the following:

- Summary
- Identification
- Document overview and connection to other documents
- References
- Personnel

  Requirements for independence from development and testing teams.
- Test Objects and Software Version

  A short description of the test object configuration and status with associated references. The most important thing is to clearly identify permissible discrepancies on the test item used and a statement on the significance of these. The current software version shall be documented in accordance with the SVD.
- Test equipment

  Description of test equipment required for the system safety testing in order to inject errors in the system's normal interfaces. It is important that the test equipment does not affect the system function in any way other than the intended malfunction. The development of the test equipment should be coordinated with the development of the technical system maintenance functions.

- Test implementation

  The test implementation must be agreed with FMV and specified together with the conditions for the test and the expected results.

  The test consists of two parts, a regular test and an additional test. The normal test scope shall always be performed with each new system version. This test includes all testable safety features and function monitoring of these safety features (ie, both hardware and software features). The purpose is to ensure that no safety features in the new system version have been unintentionally affected.

  The additional test is specifically designed for added functionality, or corrected discrepancies since the previous system version which has been considered to affect system safety.

9.2.17 System Safety Test Report, SSTR

The purpose of the *System Safety Test Report* (SSTR) is to report results from the completed system safety testing. The test report shall be approved by FMV.

SSTR should include the following:

- Summary

  A brief summary of the results of the performed system safety test shall be provided. If any deviations have been identified during the test, they shall be stated with the associated reference to the problem report.
- Identification
- Document overview and connection to other documents
- Referenced documents
- Test Objects and Software Version

  A brief description of the test object configuration and status with associated references and possible changes to established test programs.

- Test resources

  State where and when the test was carried out, which personnel from FMV and industry that who participated and a statement on whether requirements for independence were met or not.

- Test equipment

  A list of used test equipment with serial numbers and any deviations or adjustments that have been made before or during the test.

- Results, regular tests

  An account of the results of the regular test. Any deviations and criteria for approval shall be reported together with the associated problem report. Measurement results and any log files shall be traceable to each completed test.

- Results, additional tests

  An account of the results from additional test points. Any deviations and criteria for approval shall be reported together with the associated problem report. Measurement results and any log files shall be traceable to the respective test.

- Conclusion

  A comprehensive summary of the results of the system safety test with recommendations for any additional restrictions on use or maintenance.

  All system safety requirements shall be presented in tabular form with reference to the corresponding test point.

9.2.18 Sub System Hazard Analysis Computer System (SSHA CS)

The purpose of the *Sub System Hazard Analysis Computer System* (SSHA CS) is to identify possible additional hazards after the initial risk identification and to verify compliance with the system safety requirements for the technical subsystems, in this case the computer system. Hazards that can be associated with errors in the computer system and operation of the computer system are analysed. Furthermore, risk mitigation measures are identified.

An SSHA CS can be documented according to DI-SAFT-80101B, System Safety Hazard Analysis Report. Examples of analysis methods for implementing an SSHA can be found in H SystSäk.

SSHA for computer systems shall include the following:

- Summary
- Introduction
- External and internal requirements
  (Legal requirements, manuals such as H SystSäk, H ProgSäk, H VAS, chosen standard)
- System safety activities
  Description of how the system safety work has been carried out and what methods that have been used.
- System description
  A brief functional description and overview of the computer system with all interfaces and defined names, references to design documentation, which PDS used with version identification, which operating system used and any hardware. The description shall be directly traceable to the respective design documentation.
- Product Identity
  Product name with article and document numbers, current software version with reference to the SVD for analysed release of the system version.
- History
  A list of corrected errors from the previous system version with System Safety Impact Rating shall be included. Reference to completed verification and validation, remaining known errors with classification as well as any additional restrictions regarding use and maintenance.
- Methods
  An analysis of the methodology used in the analysis and classification of hazards that has been performed.

- Hazards

  An account of identified hazards that can be initiated or affected by the computer system shall be included. A breakdown of requirements and identification of safety functions shall also be included.

- Safety architecture

  Justification of chosen safety architecture, including rationale, and selection of standards shall be provided.

- Verification of system safety requirements

  An account of pre and post mitigation hazards, all risk mitigation measures that lead to risk reduction in more than one step must be separately analysed and agreed with FMV. Listing of implemented system safety tests.

- Analyses results

  An extended summary of the results of the implemented system safety analyses with recommendations for any additional restrictions on use or maintenance. All system safety requirements shall be listed in tabular form with reference to the corresponding verification reports.

- Acronyms, abbreviations and definitions
- References

  References to all audit documents for the analyses shall be traceable with date and version number. It is not enough to refer only to an overhead document structure for the technical system. All documents must be available in digital format, such PDF.

- Attachments

  E.g. Hazard Log Computer System and FTA Computer System.

10 CE marked Products and Products Approved by Other Party

This chapter covers with products and technical systems approved by another trusted operator or by a foreign authority, whether or not the product is labelled with a consumer marking. FMV must always check that the product or technical system meets the Armed Force's requirements regarding the use environment and operating conditions, as well as ensuring that the product complies with laws and regulations.

10.1 General Information About CE marked Products

CE marking is mandatory in EU legislation for certain specified product categories. However, products specially developed for military purposes aimed at causing harm to an enemy cannot be CE marked. Machines manufactured for military purposes are thus not covered by the Machinery Directive. For example, a weapon cannot be CE marked because its main function is to harm a third party, but on the other hand, an Ammunition Clearing Machine and its associated weapons can be CE marked.

Through the CE marking, the manufacturer / distributor declares that the product complies with statutory requirements, including safety, health and the environment. For certain products, it is sufficient for the manufacturer / distributor to ensure that the product meets all requirements. For other products considered to be particularly hazardous, the manufacturer / distributor must allow an independent third party body to check the product.

As part of the CE marking, the manufacturer / distributor must establish technical documentation for the product and issue an EU declaration of conformity. The CE-marked product must be accompanied by a user manual containing all essential information in order for the product to be used safely for the intended purpose. For delivery to the end user, the product must be accompanied by a user manual in the language of the recipient country.

When acquiring commercial products, there are often difficulties in obtaining sufficient information about previously performed system safety analyses. Depending on how these products are intended to be used, they may become safety-critical.

This chapter is intended to define a reasonable scope of system safety activities for the procurement of CE-labelled products containing software intended to be used stand-alone from military systems. Should the CE-labelled product be integrated into the military technical system, this integration will be covered by a system safety analysis. Standalone use means that the product is powered, exchanges information or is integrated with other technical products in accordance with the manufacturer / distributor assembly instructions. This category of products may, in turn, be divided into subgroups for which the need for system safety activities varies in terms of contents and scope.

The breakdown of independent CE-marked products is as follows and the need for system safety activities is described for each according to the sub-sections:

- Products already on the market that contain safety-critical software, but it is not planned to perform own updates or other changes (*section 10.2.*)
- Newly developed products or technical systems not available on the market but are CE marked before delivery to FMV. Updates may be relevant (*section 10.3*).
- Products or technical systems approved by other trusted parties such as foreign power or offered through NATO (*section 10.4*).

10.2 CE Marked Products Already on the Market

This section describes CE-marked products that comply with all of the following statements:

- The product has been CE marked when it was placed on the market.
- The product is meant to be used alone and not integrated into a technical system.
- Any updates of the software are performed exclusively by the supplier.

For simple CE-labelled products that contain non-safety-critical software, usually no system safety activities are required in addition to just making the assessment and rating that the current product is of this simple nature. Examples of such products may be personal computers, monitors, home appliances (for example ovens, washing machines), tools for craft work (such as drilling machines, laser meters). For such products, there are often harmonized standards that should be met, which strengthens the seller's basis for CE marking and simplifies the customer's acceptance inspection. Upon delivery, the CE declaration (*Declaration of Conformity*, DoC) as well as the handling and maintenance documentation must be attached. The product may only be used / handled separately from other equipment in accordance with the manufacturer's / distributor's instructions.

If the CE marking shows that the harmonized standard is met, the product can be considered as tolerably safe provided that the customer does not perform its own software updates. Updates may, however, be carried out by the manufacturer / distributor as required under the CE marking. Examples of such products may, for example, be medical equipment. Upon delivery, the CE declaration (*Declaration of Conformity*, DoC) and the handling and maintenance documentation must be attached. The product may only be used / handled separately from other equipment in accordance with the manufacturer's / distributor's instructions.

10.3 CE Marked Products not on the Market

This category includes CE-marked products as well as products of a technical nature, such as boats and functional containers, which are not already on the market. In such procurements, FMV may impose special requirements on the use environment, operating conditions, system safety work and its documentation for the manufacturer's / distributor's CE marking.

Just like for products in *section 10.2* above, it is important to determine whether the software in the current product is safety critical in any sense and in the event that it is deemed safety critical, ensure that applicable system safety work on these software-controlled features is carried out. Once this is done and as long as the product is used within the framework of the CE marking and no changes are implemented in the operating instructions or in the design, the system can be considered as tolerably safe.

The difference between the types of products described in *section 10.2* is that the systems in *section 10.3* usually have a long service life. The long planned life expectancy may more likely result in the need for customer updates. If FMV intends to implement its own software updates in a CE-marked product, FMV needs access to the development environment, documentation and source code of the product. This is not always possible from the manufacturer / distributor and may cause the update not to be completed. If it has been more than 10 years since the last product was delivered, the manufacturer / distributor is no longer required to provide technical documentation for the product, which may complicate the documentation of its the update. If it is deemed possible to make customer updates of the software after this deadline is passed, this should be clarified before the Armed Forces issues a procurement assignment to FMV.

Updates of software that are not performed by the manufacturer / distributor or on the advice of the manufacturer / distributor may cause the manufacturer / distributor CE declaration to become out of date or not applicable. In cases where the original CE marking cannot be applied or when the intended use of the product is not covered by the declaration, a new CE marking may be carried out by a certification body on behalf of FMV, alternatively a CE-like process is carried out where EU requirements

non-compliance directives must be documented. For these unfulfilled requirements, a system safety analysis is carried out according to H SystSäk. FMV must, prior to the call for tenders, enter into, and after a dialogue with the Armed Forces, determine the extent of documentation to be ordered from the contracted industry to ensure the possibility of future own updates of the software.

When newly developed products not on the market are procured for independent use within the Armed Forces, the product may, with reference to the above be CE marked. Certain exceptions exist for products designed for specific military use and which are not subject to the CE marking requirements.

It is important determine early on whether, during the use phase, it may be appropriate for FMV to carry out own updates of the software in the product to be CE marked. This requires that FMV has full control over, among other things, the software development environment. This is not always possible to get from the manufacturer / distributor and may cause the update not to be implemented without a reduction in safety.

FMV procuring a newly developed product that is CE marked should in particular ensure that the manufacturer / distributor follows and reports which EU directives and harmonized standards the manufacturer / distributor refers its declaration of compliance to. Therefore, FMV needs to define the product's intended use profile, application area and operating conditions.

FMV states this in the requirements to the manufacturer / distributor so that the manufacturer / distributor is aware of the use. The manufacturer / distributor shall, in a system safety plan (SSPP), respond to which EU directives and harmonized standards the product will be CE-marked and how verification of compliance with these standards and requirements will be implemented.

In connection with delivery, the manufacturer / distributor must provide a *Declaration of Conformity*, DoC, and the technical documentation relevant to the product.

10.4 Products Certified or Approved by Another Party

In line with the EU's security and defence policy, efforts are being made to progressively establish a European defence equipment market and to meet the needs for military capabilities. In the work of strengthening a European Defence Industrial and Military Technology Base, Directive 2009/81 / EC provides guidance to contracting authorities to standardize technical specifications and take into account tenders based on equivalent solutions based on performance and functional requirements, as well as referring international, European or national standards.

The European Commission stated on 20 December 2013: *The European Defence Agency (EDA) and the Commission will prepare a roadmap for the development of defence industrial standards by mid-2014, without duplicating existing standards, in particular NATO standards*.

EDA is a service body for its Member States with the task of supporting, streamlining and coordinating the development and procurement of defence equipment. EDA develops and provides tools for contracting authorities, such as *Collaborative Database* (CODABA), *Third Party Logistic Support* (TPLS) *Platform and Procurement Experts Network* (PEN). EDA works to harmonize requirements for defence equipment and for multi-member cooperation contracts. One goal is to consolidate and standardize requirements for cost-effective defence equipment. One solution is so called *User Clubs*, with several Member States where requirements, development methods, standards, approval processes and more can be effectively coordinated.

Within NATO, the *NATO Support and Procurement Agency* (NSPA) is tasked with assisting Member States with procurement of defence equipment, primarily in the purchase of equipment "off the shelf". Among other things, NSPA has the task of developing and adapting technical documentation in connection with the sale of defence equipment.

For air traffic management systems, which also cover many of the military systems, there are governing directives. For example, EC 552/2004 *Operational Capacity of the European Air Traffic*

Management Network requires interoperability between national systems. Requirements are made for the submission of completed operating approval processes, including system safety approvals as well as software safety approvals, to be submitted to the requesting authority within the EU. Requirements are set at system level, for example on a radio communication system, and at component level, as for a flight radio. Suppliers of such systems and products should compile documentation for operating approval in view of the fact that the data can be disseminated to authorities across the EU.

For ammunition, which increasingly contains complex software, within EDA *The European Network of National Safety Authorities on Ammunition* (ENNSA) is available. ENNSA aims at *Better communication among national safety authorities on ammunition and to improve harmonization of national practices on ammunition safety standardization and test procedures where feasible*.

Within NATO, the *Munition Safety Information Analysis Centre* (MSIAC) is intended to support member states in ammunition safety issues. This organization has a broader scope than ENNSA. Technical questions regarding the ammunition's design and experiences during use can be put to MSIAC through Sweden's contact person, which is located at FMV.

Figure 10:1 provides an example of how a NATO standard with requirements for ammunition can be met using an international civilian standard for software in safety critical applications.

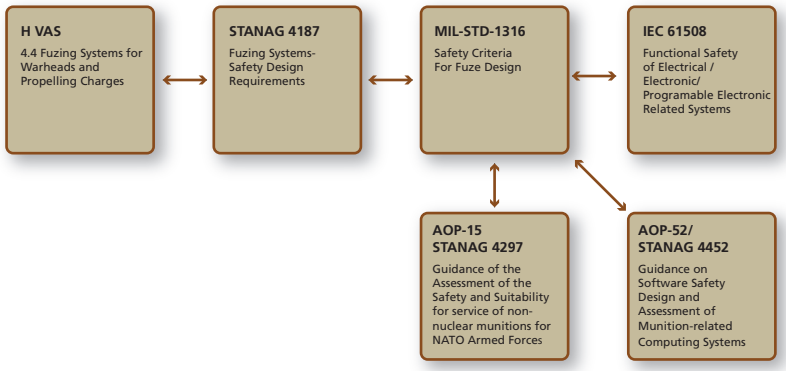


Figure 10:1 Connection between NATO Standards/requirements for Ammunition and civilian Standards

Equipment regulated by the Marine Equipment Directive shall be "steering wheel" marked and not CE marked. The steering wheel mark shows that the product meets the requirements of this directive. Type approval (steering wheel marking) of equipment is regulated by an EU directive. The Directive states down common rules aimed at eliminating differences in the implementation of international standards by having a clearly identified set of requirements and uniform certification procedures.

The new EU Parliament/Council Marine Equipment Directive 2014/90/EU about marine equipment is put in place by law (2016:768) on marine equipment and by a regulation (2016:770) on marine equipment together with the Swedish Transport Agency's Regulations (TSFS 2016:81) on Marine equipment. A product type-approved by a notified body in a Member State within the EU may be placed on an EU vessel, regardless of which flag it carries, which promotes the free movement of marine equipment on the EU's internal market

In addition to the above, there are additional EU directives with exceptions to military equipment, such as RoHS (electrical and electronic equipment) and Reach (chemical substances).

11 Handling of Previously Developed Software (PDS)

Previously Developed Software (PDS), is software, already fully developed when the new technical system is being developed. PDS can either be developed internally at the contracted industry or procured from another party. Variants of previously developed software can also be COTS (***Commercial Off The Shelf***), GOTS (***Government Off The Shelf***), MOTS (***Military Off The Shelf***), NOTS (***NATO Off The Shelf***), or ***Open Source Software*** (OSS).

When developing a new technical system containing software, the choice has to be made to develop completely new software, further develop existing software, or use previously developed software.

11.1 To Take Into Account When Using PDS

Acquiring PDS can initially be perceived to be more economically advantageous than developing new software for a particular function. However, PDS must meet the same requirements as for other newly developed software regarding various aspects such as system safety, IT security, testing, documentation, quality, configuration management.

Below are some aspects that can be expected to arise when PDS is used in new applications.

- PDS has most often been developed for wide use and to suit many users, or for use in previous versions of the current technical system. This can mean that the specific requirements for the new technical system are not fully met.
- If the specific requirements are not met, modification of the PDS may be required which may lead to extensive development work and testing, especially if there is no access to the full documentation. Sometimes extensive work has to be carried out on testing in order to qualify the software to the required level of criticality.

- If documentation is missing this may complicate future system updates and testing. If PDS is to be used in a context where a higher level of criticality is required than it has been tested and documented to, this may make it impossible to use.
- Lack of documentation can lead to design managers not understanding PDS functionality. This may lead to improper use of the software and that it cannot be sufficiently tested.
- If the PDS is intended to be general and fit for many different applications, there may be features that are not required for the current technical system. Thus, PDS can contain dead or deactivated code that can cause faults.
- To classify PDS to a higher criticality level than it is certified for can lead to high costs or, in practice, is impossible.
- For maintenance of long-life technical systems incorporating PDS, it is important to check that any licenses are still valid and therefore can be maintained and updated through maintenance agreements with the suppliers. If the need for new functionality in the software is needed, to give the technical system better performance or new features, then the PDS development history must be known and traceable.
- It must be ensured that the Armed Forces have the right of use of the software through licenses in cases where the contracted industry acquires PDS from another party for use in a technical system.
- If the supplier makes a special version of the PDS for the new technical system, it should be handled in the same way as for other newly developed software. The new PDS shall then be classified with regard to criticality and comply with the requirements of the selected software standard.

11.2 Prerequisites for Using PDS

Reusing PDS may be beneficial in cases where the software functionality matches what has been requested and has been used in similar applications. Examples of when PDS can be an option are:

- In large technical systems, such as a ship, where there can be a relatively low level of criticality on certain independent and uncritical functions.
- When the PDS provider has a certified software component or functional component according to certain software standards. For example, IEC 61508 certified components with a specified SIL level.
- For drivers in interfaces with standard hardware components.
- When the PDS provider can provide complete documentation for the required level of criticality and documented operational experience.

11.3 Evaluation of Supplier for PDS

When assessing different PDS solutions as an alternative to new development, the following aspects need to be addressed:

- Has the supplier been on the market for a long time and delivered similar products?
- Assess the suppliers interest and possibilities to implement future system updates and provide support for current PDS, for example through specific agreements.
- Assess possibilities for access to development environments and documentation such as source code, specifications, test documents, user manuals, design documents, descriptions, and implemented error corrections.
- Will the supplier submit quality documents that applies to the current PDS as proof that the documented development process has been followed, such as audit reports, test reports and quality audit reports.
- Has the supplier used the current PDS in other similar technical systems and provided evidence that PDS works correctly (*Proven in use*), referring to a *Service History* report system.
- Has the supplier defined methods that can be used to demonstrate *Proven in use* for the current PDS and are these methods documented?
- Can the supplier provide certificates from an independent third party review?

12 Related Methodology and Technology Areas

This chapter discusses a number of adjacent methodology and technology areas that are not discussed further in this manual. The purpose is to provide guidance on how neighbouring areas can be treated and thus indirectly also describe the applicability of the methodology described in this manual.

12.1 System Safety Operations

System safety operations are the total work carried out during the entire lifecycle of a technical system with the aim of identifying, analysing, assessing and managing accident risks. See H SystSäk.

The manual describes requirements that shall be specified for software and software development all based on the impact it has on system safety. System safety is defined below.

System Safety

"The property of a technical system not to inadvertently cause damage to person, property or the environment".

Source: H SystSäk

There are also other requirements for the technical system and the software that is included as well as how the software system is developed. Some selected areas are described below.

12.2 Operational Safety

Operational safety aims at the Armed Forces' ability to handle risks in all operations, not just the risks associated with technical systems. System safety activities are the total work undertaken during the entire lifecycle of the technical system in order to identify, analyse, evaluate and address hazards. See H SystSäk.

12.3 Information Security

Information security is aimed at protecting information assets. In connection with software, it is usually aimed at protecting the information contained in IT systems, or their function, against external attacks. This is handled both through technical and administrative measures in accordance with the *Swedish Armed Forces Security Service, Information Security* (H Säk Infosäk 2013), especially sections on IT security, communication and accreditation, as well as the latest version of *Security Requirements* (KSF).

All activities are dependent on assets to be protected in the form of personnel, information, equipment and facilities. Based on the nature of the business, some of these assets may be more in need of protection than others. The purpose of the information security area is to protect these assets against unwanted events.

From an information security perspective, the information shall be:

- Available to anyone who needs it
- Correct
- Protected against unauthorized access as well as
- Traceable.

If the protection of the information fails ie. so makes it unavailable, corrupt or undesirably spread to others, it has consequences for the use, equipment, personnel and facilities.

12.3.1 Information Security Declaration (ISD)

The Information Security Declaration (ISD) process is part of the Armed Forces accreditation process within the framework of the Armed Forces Materiel Supply. The main purpose of the ISD process is to create trust in a system's design and function through a certain structure and methodology from the perspective of information security. The Armed Forces can then operate the materiel with a tolerable risk with regard to information security. The ISD

creates consistency, clarity, traceability and efficiency in working with features related to information security in operations and with technical systems.

The ISD complies with ISO / IEC 15288 standards for Systems Engineering and ISO / IEC 27000 for *Information Security Management Systems*. Information about ISD is available on FMV's website.

The ISD is both a support process for information security work at FMV and a declaration on information security.

The purpose of the ISD process and its methodology support is to create uniformity, clarity, traceability and efficiency in the process of developing accreditation data and ISD declarations. The process starts with the requirement definition and develops during breakdown of requirements, production and handover. The subsequent declaration then states that FMV:

- Takes a design responsibility for the IT security solution
- Complies with the Armed Forces' requirements regarding information security
- Has designed the documentation according to the standard applicable to FMV
- Has dimensioned the IT safety solution based on a defined tolerable risk level set by the Armed Forces
- Has followed the established ISD plan regarding IT security work.

If there are special conditions for this declaration, these aspects shall also be included in the declaration.

How FMV is to conduct information security work can be found in the following ISD manuals:

- ISD IT Security Management
- ISD IT Security Independent Review
- ISD IT Security Use cases and architecture.

12.3.2 Communication Security (COMSEC)

When sensitive information is to be sent between different units within the Armed Forces, or between different authorities, it is encrypted. The special case of software with the main task of encrypting information is not covered in this manual.

Where signal COMSEC is included, information from the *COMSEC Handbook H TST Fundamentals* (M7746-734002) can be obtained.

12.4 Functional Characteristics

The functionality of a software can also affect the ability to carry out a mission, for example, whether a weapon can be aimed at an identified target or not. There may also be a lack of functionality, which can lead to exposing own troops to danger by enemy action or friendly fire because countermeasure or IFF systems do not work as intended.

There is experience from foreign military development projects where criticality classification methodology has been applied which means that the criticality classification of software (and thus the software development requirements) according to current software standards is also applied.

12.5 Usability

The vast majority of technical and non-technical systems will, in one way or another, communicate with operators and maintenance staff. If people can increase or decrease their ability in this interaction with technical systems, it is important for systems to be perceived safe, effective and useful.

Insight into how psychological, physiological, organizational and technical aspects interact in complex, stressful environments, create the conditions for providing safe, efficient and useful systems. FMV is responsible to the Armed Forces for this complex situation and this insight is necessary in order to cope with the Armed Forces' needs with regard to the technical systems.

The foundation for a technical system or product to become useful and beneficial to the users and operations is laid early in the development work. To compensate for a technical system's or a product's deficiencies with training, or requiring specialist knowledge from the users, is cost-driving and restricts the use. A system with deficiencies in safety, efficiency and usability can expose the users to life threatening hazards and can also cause damage to the environment or lead to financial losses.

When designing user interfaces, knowledge about human conditions and abilities is required, but also knowledge about human limitations as a user and as part of a system. Knowledge is also needed about how man, through his or her sensory organs, receives information and interprets the environment as well as about the human memory, thinking and decision making processes, etc.

In support of user interface design, there exist several general design principles, rules of thumb or guidelines, often based on experience and research. Examples of these principles are *Jakob Nielsen's rules of thumb* or *Ben Shneiderman's eight golden design rules*. In addition, there are platform-specific design principles for, for example, Microsoft Windows and international standards

For technical systems with a long life, further development of the user interface often takes place. This particularly concerns military systems where operators have been trained and practiced on a particular version of the user interfaces. Consequences of changed user interfaces can both result in new hazards and reduced efficiency as the operator's handling of the system or visual feedback changes.

For computer monitor work, AFS 1998: 05 *Arbete vid Bildskärm* (in Swedish) is used. In addition to regulations and standards, there is an FMV Handbook (H HFI) and MIL-STD 1472G.

12.6 Programmable Logic

Logic functions can be implemented, in software and also, with programmed and, in some cases, programmable circuits such as *Application Specific Integrated Circuits* (ASIC), *Field Programmable Gate Array* (FPGA), and *Programmable Logic Devices* (PLD). This raises, besides pure hardware issues, the same issues with systematic errors that exist for software. Therefore, for these components, and the functions they support, it may be necessary to apply a methodology similar to that applied to software.

Section 2.10 describes the standard RTCA DO-254 that applies to programmable logic within the aviation domain. Within the aviation domain, there are also comprehensive implementing rules for the US (FAA) and European Aviation Authorities (EASA) standards.

The description contained in the *Weapons and Ammunition Safety Handbook* (H VAS) for application of programmable logic in ignition systems is also useful for other types of control functions in technical systems. See also STANAG 4187. This area is not otherwise discussed in this manual, but the requirements for this in H VAS can be applied.

12.7 Methods for Rapid System Development

There are a number of methods to speed up system development and programming work in order to secure results that can be achieved faster than with traditional system development. In a more traditional way of working, it is likely that it takes a long time to handle customer needs, as it may take a long time to handle many requirements in relatively large development steps, instead of taking more but smaller ones, which each can provide a result that can be delivered or put on the market. Examples of methods include SCRUM and Agil system development.

Sometimes descriptions of these methods can be interpreted as not having full traceability of requirements, managed processes, complete documentation, or full verification of features. However, the same requirements apply to this type of work method as with the use of more conventional methods.

13 Compilation of requirements

Chapter6 Requirements from the Armed Forces

Section 6.1 Conditions and Requirements for the Development of Technical Systems

Requirement no	Content
2.601.01-A	FMV shall request that the Armed Forces specify the context, use and external environment and operating conditions of the technical system. *Comment:* This applies to both military use and, where appropriate, support to society during peacetime.
2.601.02-A	FMV shall request that the Armed Forces define overall functional performance requirements for the technical system.
2.601.03-A	FMV shall request the Armed Forces to define the tolerable level of risk for the technical system throughout its life.
2.601.04-A	FMV shall request that the Armed Forces make operational experience available from previous similar technical systems.

Section 6.2 Prerequisites for the Development of Technical Systems

Requirement no	Content
2.602.01-A	A FMV shall request from the Armed Forces which actor is chosen to be the technical design authority for the system. *Comment:* If another stakeholder than FMV is the technical design authority, this must be stated in FMV's System Safety Approval (SSG).

Section 6.3 Prerequisites for Handover and use

Requirement no	Content
2.603.01-A	FMV shall request that the Armed Forces have a deviation reporting system for technical systems where deviations can be reported. *Comment:* If other deviation reporting systems than the Armed Forces are to be used, FMV needs to beware of this.
2.603.02-A	FMV shall request that the Armed Forces comply with the instructions submitted regarding the operation, in-service use maintenance, and procedures for performing system updates on handed over materiel. *Comment:* If a stakeholder other than FMV is to be Technical Design Authority, the Armed Forces need to inform FMV of this.

Requirement no	Content
2.603.03-A	FMV shall, on the basis of the Armed Forces' requirements, specify what restrictions and requirements that apply to personnel who handle, use / maintain or perform system updates on handed over materiel. *Comment*: This is especially valid during tactical deployments where system updates may need to be carried out by the Armed Forces own personnel.

Section 6.4 Prerequisites for Maintenance

Requirement no	Content
2.604.01-A	FMV shall request from the Armed Forces Deviation Reports for the Technical System. *Comment:* The information may be submitted to the System Safety Working Group (SSWG).
2.604.02-A	FMV shall request that the Armed Forces participate in the System Safety Working Group (SSWG).

CHAPTER7 OPERATIONAL REQUIREMENTS FOR FMV

Section 7.1 FMV's Work During the Life Cycle

Requirement no	Content
2.701.01-A	FMV System Safety Management Plan (SSMP) shall address software safety requirements. *Comment*: FMV's SSMP shall address handle the Armed Forces' requirements for tolerable risk levels for all system levels of the technical system. In cases where FMV issues an internal SSPP for a project, it should also include software safety.
2.701.02-A	Software safety issues shall be handled by the System Safety Working Group (SSWG).

Section 7.3 Development, Production and Acquisition

Requirement no	Content
2.703.01-A	FMV shall ensure that the SSPP includes software-related system safety activities prior to signing a contract. *Comment:* The SSPP shall be written in accordance with H SystSäk and contain all necessary activities and methods for implementing the software safety work and, where appropriate, in accordance with the agreed software standard.
2.703.02-A	FMV shall, for software with an initial criticality classification, HIGH agree with industry which established software standard, including level of criticality, applicable in the field of technology with which industry shall demonstrate conformity. *Comment:* For criticality level ***LOW***, the basic requirements (GKPS) are sufficient.

Requirement no	Content
2.703.03-A	FMV shall ensure that the minutes from the contract review show any possible deviation from GKPS as agreed. *Comment:* In the minutes it shall be stated that the contracted industry will meet other requirements according to GKPS. Several contract reviews can be completed during the implementation of the project.
2.703.04-A	FMV shall ensure that the contracted industry reports deviations that are significant for system safety identified during development and operation as well as the total number of deviations. *Comment:* At the time of delivery, the report shall include at least any deviations that are open or closed from verification testing. FMV shall include any open remarks in the system safety work, at least by taking the view that they do not cause system safety measures.
2.703.05-A	FMV shall specify that industry show compliance with the basic requirements (GKPS) in this manual for all software regardless of criticality level.
2.703.06-A	FMV shall ensure that industry can provide support for analysis and actions for emerging system safety issues during total system life. *Comment:* FMV shall contract support from the manufacturer according to the extent and time FMV and the Armed Forces need. Consideration shall be given to the characteristics and expected life of the technical system.
2.703.07-A	FMV and industry's system safety work including software safety must be completed and a system safety approval (SSG) to be issued prior to handing over the system to the Armed Forces.

Section 7.4 Usage and System Updates

Requirement no	Content
2.704.01-A	FMV shall ensure that the contracted industry has access to the software development environment throughout the product's entire life cycle to the extent necessary. *Comment:* The Scope is specified in the contract given by FMV.
2.704.02-A	System Safety Approval (SSG) update shall always be made when changing or modifying a technical system. *Comment:* See H SystSäk regarding System Safety Approval (SSG).

Section 7.5 Decommissioning of a Technical System

Requirement no	Content
2.705.01-A	Supplies such as software and computers must be registered in relevant support systems in connection with delivery from industry. *Comment:* This also applies to supplies transferred into the governments possession, but is still in use by the contracted industry.
2.705.02-A	Decommissioning of a technical system (or part of a technical system) shall include the resources used for support for the development and maintenance of the systems. *Comment:* This includes software development environments, agreements for activities including personnel and supply of data etc.

CHAPTER8 BASIC REQUIREMENTS (GKPS) FOR THE CONTRACTED INDUSTRY

Section 8.1.1 Requirements for the Development of Technical Systems

Requirement no	Content
2.801.01-A	Roles including the required level of competence shall be agreed with FMV. *Comment:* The competence profiles for personnel involved in the development of the technical system, such as project managers, technical leads responsible for system architecture, verification managers and independent auditors are documented.
2.801.02-A	At least two persons should be familiar with the chosen system architecture. *Comment:* The choice of system architecture based on the implementation of analysis of dimensioning hazards shall be known by at least two persons.
2.801.03-A	The contracted industry shall designate a software safety point of contact. *Comment:* This person ensures that the agreed work practices and methodologies for system safety work are followed and are responsible for the verification of the basic requirements (GKPS) and reports that these requirements are met.

Section 8.1.2 Requirements for Operational and System Safety Management

Requirement no	Content
2.801.04-A	The contracted industry must comply with AQAP 2110. *Comment:* This applies primarily to the right of insight.
2.801.05-A	The contracted industry must comply with AQAP 2210.
2.801.06-A	The contracted industry shall issue a System Safety Program Plan (SSPP). *Comment:* The System Safety Program Plan (SSPP) shall also include required activities such as requirements documents, test plans and software development test procedures, as well as a description of agreed development tools.
2.801.07-A	The contracted industry shall state in the System Safety Plan (SSPP) how GKPS will be met.
2.801.08-A	The system safety analysis shall cover the computer system's impact on the entire system of the technical system during its life. *Comment:* The analysis shall be performed in an iterative way during the development phase, from requirements analysis to completed verification.

Section 8.1.3 Requirements for Safety Architecture Design

Requirement no	Content
2.801.09-A	For the computer system, safety architecture and design principles shall be documented and reported. *Comment*: The contracted industry shall present a safety architecture according to *section 4.3*, which is reported in the System Specification / System, Subsystem Specification (SSS).
2.801.10-A	The design principles shall determine which strategies for error detection, fault tolerance and error safety that are applied. *Comment:* The statement shall state and justify the chosen design principles.
2.801.11-A	Design decisions regarding the selected safety architecture shall be documented and include the assumptions and justification for the selected design options.

Section 8.1.4 Development Tools Requirements

Requirement no	Content
2.801.12-A	Requirements Tracking Tools shall be used and agreed with FMV. *Comment:* The tool shall meet the requirements tracking process requirements of IEC.
2.801.13-A	Configuration Management Tools shall be used and agreed with FMV. *Comment:* The tool shall meet the process requirements for configuration management in accordance with IEC 12207.

Requirement no	Content
2.801.14-A	Deviation Reporting Tools shall be used and agreed with FMV. *Comment:* The tool shall meet the process requirements for error reporting according to IEC 12207.
2.801.15-A	FMV shall be provided with information regarding requirements tracking, configuration management, deviation reporting and test data. *Comment:* FMV needs to ensure that there are conditions present for managing and reading the information.

Section 8.1.5 Documentation Requirements

Requirement no	Content
2.801.16-A	A document list shall be agreed with FMV. *Comment*: This shall be defined on the basis of the document list in *chapter 9*. A delivery plan for the documentation shall be provided.

Section 8.2.1 System Safety Analysis Requirements

Requirement no	Content
2.802.01-A	Traceability shall exist between computer systems and its impact on the identified hazards belonging to the technical system. *Comment:* Requirements tracking in both directions can be reported in the safety architecture work.

Requirement no	Content
2.802.02-A	The system safety analysis shall report the criticality level of included software in the technical system. *Comment:* Refers to analysis of the safety architecture.
2.802.03-A	The choice of safety architecture shall be motivated based on analysis of the dimensioning hazards.
2.802.04-A	Independent audits and reviews shall be performed in accordance with the agreed System Safety Program Plan (SSPP). *Comment:* An independent reviewer refers to a person who has not participated in the development work.
2.802.05-A	Data shall have the criticality classification required by the current technical system. *Comment:* Data refers to both static and real-time generated information.

Section 8.2.2 Design Requirements

Requirement no	Content
2.802.06-T	Selecting safety features and function monitoring shall be done in such a way that this does not unnecessarily complicate the software system. *Comment:* A balancing should always be done to achieve key safety principles such as simplicity, independence and determinism.
2.802.07-T	Established programming languages shall be used in developing safety-critical software. *Comment:* Selected programming languages shall be reported to FMV together with design principles and safety architecture.

Requirement no	Content
2.802.08-T	For each operational state, the technical system must be able to enter a safe state. *Comment:* For initial criticality classification, ***LOW*** safe states refer to states where control of executing parts of the system have been controlled / terminated in a safe way or where a rescue system has assumed control.
2.802.09-T	All error states that may affect system function shall be logged in a format that is possible to evaluate. *Comment:* There shall be traceability between the triggering fault situation / fault criterion and the state of the technical system has entered so that faults can be detected in the computer system. Logging can be done internally in the computer system or logged in to an external system.
2.802.10-T	The technical system shall be in a safe state during boot-up. *Comment:* This also includes rebooting of the computer system.
2.802.11-T	At the start-up of the technical system, the software shall check that the defined safe state has been entered before critical parts of system are activated. *Comment:* The level of safety can be checked by re-reading critical control or sensor signals.
2.802.12-T	Unreasonable inputs, which, according to the system safety analysis, may affect the functioning of the system, shall be detected and disposed of so that a hazard does not occur. *Comment:* Unreasonable data means all data outside the defined value range or data at the wrong time.

Requirement no	Content
2.802.13-T	Operator actions and presented information relating to safety-critical functions shall be recorded. *Comment:* The ways to record this may vary based on system configuration, complexity and the situation.
2.802.14-T	Built-in Test (BIT) shall contain Safety Control (SK / PBIT) during start up, Function Monitoring (FÖ / CBIT) during operation and Manual Initiated Test / Function Control (FK / IBIT) during maintenance.
2.802.15-T	BIT features for boot-up and maintenance shall not be inadvertently activated during operation of the system. *Comment:* Safety functions, such as blocking, shall exist so that handling errors can be avoided.
2.802.16-T	The independent watchdog function must be activated before the computer system can perform critical controls. *Comment:* The independent watchdog function is preferably implemented in hardware.
2.802.17-T	Watchdog (WD) must have a defined time window (that is, min / max time for WD triggers). *Comment:* Recovery of watchdog is performed by the software.
2.802.18-T	The Watchdog (WD) shall be subject to Safety Check (SK / PBIT) at boot-up and approved results shall be a criterion for activating the watchdog function.

Requirement no	Content
2.802.19-T	Voltage monitoring shall be performed continuously on the power supply voltage of the computer system. *Comment:* A control signal from the voltage monitoring can be one of the criteria in the watchdog function.
2.802.20-T	Resource utilization at the first serial delivery shall be defined. *Comment:* The requirement relates to CPU, memory and communication links and should be no more than 50%.

Section 8.2.3 Requirements for Software Development Environment

Requirement no	Content
2.802.21-A	The choice of software development environment shall be justified and documented for the technical system. *Comment:* Industry standards and past experiences shall be considered based on the chosen criticality level.
2.802.22-A	Audit history shall be reported for the use of the development environment. *Comment:* The development environment shall be under configuration management during the entire lifecycle of the software.
2.802.23-A	When updating the development environment during the development of the software, re-verification shall be carried out both of the development environment and the developed software. *Comment:* Approaches and criteria are described in the SSPP or in any other agreed document.

Requirement no	Content
2.802.24-A	Testing tools that introduce changes to the software shall not be used for verifying a specified software version. *Comment:* If modifications are necessary for the test tool to be used, these changes shall be seen as part of the software version.

Section 8.2.4 Verification Requirements

Requirement no	Content
2.802.25-A	System safety testing shall be planned, performed and reviewed and detected errors shall be resolved and approved. *Comment:* The results are presented and any identified measures agreed with FMV.
2.802.26-A	Test cases for system safety testing shall be subject to an independent review of a person not involved in the development.
2.802.27-A	System safety testing shall be performed on a frozen system version of the technical system. *Comment:* Frozen system version refers to the version of the technical system to be delivered, that is, even the target environment must have established status.
2.802.28-A	System safety testing shall include error injection in all interfaces of the safety-critical signals identified in the system safety analyses. *Comment:* The system safety test is intended to show that function monitoring can detect critical errors.

Requirement no	Content
2.802.29-A	System safety testing shall demonstrate that functions intended for a specific operating mode / system state cannot be entered under other operating mode / system state. *Comment:* Also observe incorrect handling and operating conditions such as during training and maintenance.
2.802.30-A	Maximum resource utilization of the computer system shall be verified and documented. *Comment:* The requirement relates to CPU, memory and communication links.
2.802.31-A	Verification must be performed by correct order of program execution and at the right time for time-critical functions. *Comment:* Verification of execution order can also be performed using the development environment.
2.802.32-A	Test coverage (BIT) of safety features in the technical system shall be verified.
2.802.33-A	Use criteria for *Proven in use* must be agreed with FMV. *Comment:* The criteria are documented in the SDP (Software Development Plan).

Section 8.3 Requirements for Delivery to FMV

Requirement no	Content
2.803.01-A	A list of remaining known errors shall be issued for the delivered version of the technical system. *Comment:* As stated in the Software Version Description (SVD) according to the document list.

Requirement no	Content
2.803.02-A	A In spite of the remaining known errors, the contracted industry must show that the technical system nevertheless fulfils the Armed Force's requirements for tolerable risk level.

Section 8.4 System Update Requirements

Requirement no	Content
2.804.01-A	In the case of a new version of the technical system, re-verification shall be carried out. *Comment:* The need for reverification is determined after analysis of which parts are affected by the change.
2.804.02-A	In connection with a system update, a new system Safety Compliance Assessment shall be issued. *Comment:* Is not applicable to changeable parameters.

Definitions and Explanations

The following definitions and explanations are used in the manual. A number of definitions are the manual's own and these are specifically marked with “H ProgSäk 2018” in the column “Reference” (the manual's own definition).

For other terms, reference to, e.g., Swedish Terminology Centre TNC and ISO / IEC / IEEE 24765: 2010 *Systems and software engineering - Vocabulary*.

Term	Reference	Definition / Explanation
Accident	H SystSäk 2011	Occurs when someone / something is exposed to a hazard or dangerous condition and is thereby injured (injury to person, property or the environment). An accident is always unplanned, not the result of, for example, hostile action.
Cause	H SystSäk 2011	Condition that led to the occurrence of errors.
CE-marking	Wikipedia	Product labelling in the EES. The letters CE are an abbreviation for the Conformité Européenne. A CE marking product may be sold in the EES without further requirements.
Computer Systems	H ProgSäk 2018	Contains hardware, software and data.
Continuous mode	IEC 61508	Where the safety function retains the EUC in a safe state as part of normal operation.
Damage class	H SystSäk 2011	For personal injury: Death, serious injury, minor injury and negligible damage. For financial damage: Comparable with total system loss, significant loss, limited loss, small loss Details are given in *H SystSäk part 1, section 4.2.3*.
Dangerous condition	H SystSäk 2011	A physical situation that can lead to an accident.

Term	Reference	Definition / Explanation
Data	H ProgSäk 2018	Refers to information, often stored as files or data bases, which the software uses when executing or generating other information.
Development environment	H ProgSäk 2018	Equipment required for development (such as compilers and linkers), software verification, rigs, simulators, data provision equipment and configuration management.
Deviation handling	H ProgSäk 2018	The process of how errors are reported by the customer, classified and resulting in one or more problem reports that describe how the deviation is handled
ECE Regulation	Swedish Transport Agency	ECE regulations are annexes to the 1958 agreement to adopt uniform technical regulations for wheeled vehicles or for equipment and parts that can be mounted or used on such vehicles.
Emergency system	H ProgSäk 2018	System that ensures that the main features are maintained in case of a malfunction in the technical system.
Emergency System	H ProgSäk 2018	System that ensures that safe state of the protection can be taken in case of a malfunction of the technical system.
Established coding directive	H ProgSäk 2018	Established collection of directives that have a wide use, and mainly within its industry sector. A Coding directive comprises: • rules for permitted and prohibited software structures • rules for labelling, commenting and naming of critical parts • instructions for minimizing complexity • restrictions due to problems in compiler or target systems and more • detailed rules for safe design in used low-level language.

Term	Reference	Definition / Explanation
Established software standard	H ProgSäk 2018	Internationally recognized software standard that has a wide use in its industry sector and is updated as needed.
Established standard	H ProgSäk 2018	Internationally accepted standard that has a wide use in its industry sector and is updated as needed.
EUC	IEC 61508	Equipment under control, EUC, equipment, machinery, apparatus or plant used for manufacturing, process, transportation, medical or other activities.
Function	H ProgSäk 2018	Contains hardware such as computer systems, switches, power supplies, operating parts and sensors.
Functional safety	SS 441 05 05	The ability of a device to perform a required function under given conditions over a given time interval.
Functionality	Joint Group for Swedish Computer Terminology	Ability of a product to perform the functions for which it is designed.
Hazardous event	H SystSäk 2011	Event occurring by misadventure, that is, without intent, unplanned and which may result in an accident or incident if someone or something is exposed to the hazard.
High demand mode	IEC 61508	See IEC 61508.
Incidents	H SystSäk 2011	Hazard that does not lead to an accident because nothing was exposed when the hazard accrued.
Independent investigation	H ProgSäk 2018	Review conducted by a person who did not participate in the development. Several levels of independence exist. Independence may require a person from another organization.
Low demand mode	IEC 61508	See IEC 61508.

Term	Reference	Definition / Explanation
Operational Safety	H SystSäk 2011	The Armed Forces' operational safety defines the Armed Forces' ability to handle risks during all activities so that legal requirements for health and safety for the Armed Forces personnel as well as the requirements for third party safety, the environment and property are met.
Problem Report	H ProgSäk 2018	Contains Internal information for handling registered deviations.
Protection Function	H ProgSäk 2018	Function to achieve or maintain a safe state in the controlled equipment.
Proven in use	H ProgSäk 2018	Software that has been used in similar technical systems with proof of proper operation and has traceable usage history.
Regression Testing Environment	H ProgSäk 2018	Testing environment for software that allows automatic testing and evaluation of program features. Regression testing is used to test all or part of the technical system after changes have been made. This is to ensure that the system works as before and that no new problems have occurred as a result of introduced changes
Reliability	H ProgSäk 2018	That a technical system provides a certain function with a certain probability over time or when the function is requested.
Review	H SystSäk 2011	The purpose is to examine in a quality assured and traceable manner mainly technical documentation.
Risk	H SystSäk 2011	Refers to risk of damage to person, property and / or the environment. Expressed as a function of the likelihood of an accident and its consequence (the consequence usually divided into four criticality classes for injury to person and harm to property.

Term	Reference	Definition / Explanation
Risk Source	H SystSäk 2011	Something that can lead to damage to person, property or the environment.
Safe state	H ProgSäk 2018	A state where control of operating functions have been commanded / terminated safely or where an emergency or rescue system has taken control.
Safety Architecture	H ProgSäk 2018	Method to reduce the criticality of computer systems in a safety critical technical system.
Safety Function	H ProgSäk 2018	Added function whose purpose is to reduce the likelihood of a hazard occurring as a result of a fault safety critical function.
Safety-critical computer system	H ProgSäk 2018	Computer systems that directly or indirectly control or monitor energies and an in case of a fault error can cause a hazard leading to accidents.
Safety-critical function	H ProgSäk 2018	A function that controls or monitors energies and in case of a fault can lead to a hazard leading to accidents. Comment: A function may contain both hardware and software.
SCRUM	Wikipedia	SCRUM is a methodology for system development created by Jeff Sutherland and Ken Schwaber. The word “SCRUM” comes from rugby and is a moment when the ball is put into play.
Software	H ProgSäk 2018	Contains program instructions, data and documentation for computer systems.
Software Development Environment	H ProgSäk 2018	An environment for software development that consists of a computer program for developing, producing, modifying, analysing and testing another program and its documentation. This may also include configuration management and requirements tracking software and associated documentation.

Term	Reference	Definition / Explanation
Steering wheel-marking	Swedish Transport Agency	Equipment regulated by the Marine Equipment Directive shall be Steering wheel marked and not CE marked. The steering wheel mark shows that the product meets the requirements of the directive. Type approval (steering wheel marking) of equipment is regulated by an EU directive. The Directive lays down common rules aimed at eliminating differences in the implementation of international standards by having a clearly identified set of requirements and uniform certification procedures.
System		See *Technical system*.
System Safety	H SystSäk 2011	The property of a technical system not to inadvertently cause damage to person, property or the environment.
System Safety Activities	H SystSäk 2011	Total work carried out during the entire lifecycle of a technical system in order to identify, analyse, evaluate and address hazards.
System Update	H ProgSäk 2018	Installing an updated system version according to the contracted industries' instructions, including verification that the installation has been correctly installed.
Systematic errors	H SystSäk 2011	An error that always occurs when using a system and leads to the same erroneous result every time. For example, the reason may be a logical error in the software that gives the same error during software execution.
Tailorisering	H ProgSäk 2018	Selection and adaptation of activities and / or documentation.
Technical system	H ProgSäk 2018	Comprises components, consumables and software, as well as instructions and other product information, organized to achieve one or more stated purposes in a given environment

Term	Reference	Definition / Explanation
Test coverage for built-in test	H ProgSäk 2018	Specifies how much of the hardware functions or possible faults in the in the computer system's hardware the built-in test BIT is able to detect. BIT is a piece of software that runs concurrently with other operating software in the computer system.
Test coverage for program code	H ProgSäk 2018	Indicates how well the software code has been tested, that all requirements have been verified and that all parts of the code have been tested. Measured after the code has been developed.
Test Tools	H ProgSäk 2018	Part of the development tools used in functional testing / verification of software.
Time-critical data	H ProgSäk 2018	Data where the timing of information must be known as it is of vital importance.
Tolerable risk level	H ProgSäk 2018	An Armed Forces / FMV requirement that at least meets the law's requirements for acceptable safety under on given conditions.
Total operating time	H ProgSäk 2018	Number of hours by a technical system has been used.

Acronyms/Abbreviations

The following acronyms and abbreviations are used in the manual.

Acronym/Abbreviation	Explanation
AFS	Swedish Work Environment Authority's Statute Book (Arbetsmiljöverkets författningssamling)
ANS	Air Navigation Services
AOP	Allied Ordnance Publication
AQAP	The Allied Quality Assurance Publications
ASA	Aircraft Safety Assessment
ASIC/PLD	Application Specific Integrated Circuits/Programmable Logic Devices
ASIL	Automotive Safety Integrity Level
ATEX	ATEX is an abbreviation of the French name of one of the directives, Appareils destinés à être utilisés an ATmosphères EXplosibles (Explosion Protection)
BIT	Built-In-Test
BOA	Decision on Use
CBIT	Continuous-Built-In-Test
CCF	Common Cause Failures
CE	Conformité Européenne
CEN	European Committee for Standardization
CENELEC	Committee for Electrotechnical Standardization
CM	Configuration Management
CODABA	Collaborative Database
COTS	Commercial off the Shelf
CSSB	Central System Security Decision
DAL	Design Assurance Level
DC	Diagnostic Coverage
DoC	Declaration of Conformity
EASA	European Aviation Safety Agency (
ECE	Economic Commission for Europe
EDA	European Defence Agency

Acronym/Abbreviation	Explanation
EMC	Electromagnetic compatibility
ENNSA	European Network of National Safety Authorities on Ammunition
ETSI	European Telecommunications Standards Institute
EUC	Equipment Under Control
EUROCAE	European Organization for Civil Aviation Equipment
FAA	Federal Aviation Administration
FAT	Factory Acceptance Test
FC	Functional Check
FDAL	Function Development Assurance Level
FHA	Functional Hazard Analysis
FHA	Functional Hazard Assessment
FM	Functional Monitoring
FORTV	Swedish Fortifications Administration (Fortifikationsverket)
FPGA	Field-Programmable Gate Array
FRA	Swedish National Defence Radio Establishment (Försvarets radioanstalt)
FSA	Functional Safety Assessment
FTA	Fault Tree analysis
FVL	Full Variability Language
GKPS	Basic software safety requirements
GOTS	Government Off The Shelf
H SystSäk	Armed Forces' Handbook on System Safety (Handbok Systemsäkerhet)
H VAS	Weapons and Ammunition Safety Manual (Handbok Vapen- och ammunitionssäkerhet)
HFI	Human Factors Integration
HFT	Hardware Fault Tolerance
HKV	Armed Forces Headquarters (Försvarsmakten Högkvarteret)
HW	Hardware)
IBIT	Initiated BIT

Acronym/Abbreviation	Explanation
IDAL	Item Development Assurance Level
IEC	International Electrotechnical Commission
IMA	Integrated Modular Avionics
IRS	Interface Requirement Specification
ISD	Information Security Declaration
ISO	International Organization for Standardization
ISO/TC	ISO Technical Committee
LOR	Level Of Rigor
LRU	Line Replaceable Units
LVD	Low Voltage Directive
M	Mandatory
MIL-STD	Military Standard
MOTS	Military Off The Shelf
MoU	Memorandum of Understanding
MCS	Minimal Cut Set
MSIAC	Munitions Safety Information Analysis Center
$MTTF_d$	Mean Time To dangerous Failure
NOTS	Nato Off The Shelf
NR	Not Recommended
NSPA	NATO Support and Procurement Agency
OM	Other Measures
OSS	Open Source Software
PASA	Preliminary Aircraft Safety Assessment
PBIT	Power on BIT
PDS	Previously Developed Software
PEN	Platform och Procurement Experts Network
PL	Performance Level
PLC	Programmable Logic Controller
PSAC	Plan for Software Aspects of Certification
PSSA	Preliminary System Safety Assessment
QM	Quality Management

Acronym/Abbreviation	Explanation
R	Recommended
RAMS	Reliability, Availability, Maintainability and Safety
RTCA	Radio Technical Commission for Aeronautics
SAE	Society of Automotive Engineers
SC	Severity Category
SC	Safety Check
SCA	Safety Compliance Assessment
SCC	Software Control Category
SCMP	Software Configuration Management Plan
SDD	Software Design Document
SDP	Software Development Plan
SEK	SEK Svensk Elstandard
SFF	Safe Failure Fraction
SFS	Swedish Code of Statues (Svensk författningssamling)
SHA	Safety Hazard Analysis
SIL	Safety Integrity Level
SIRT	Systems Integration Requirements Task
SIS	Swedish Standards Institute
SL	Software Level
SOP	Start Of Production
SQA Plan	SQA Plan Software Quality Assurance Plan
SQA Record	SQA Records Software Quality Assurance Records
SRASW	Safety-Related Application Software
SRCF	Safety-Related Control Functions
SRECS	Safety-Related Electrical Control Systems
SRESW	Safety-Related Embedded Software
SRP/CS	Safety-Related Parts/Control Systems
SRS	Software Requirement Specification
SSA	System Safety Assessment
SSG	System Safety Approval (Safety Statement)
SSHA CS	Sub System Hazard Analysis Computer System

Acronym/Abbreviation	Explanation
SSLR	Software Safety Lifecycle Requirements
SSMP	System Safety Management Plan
SSPP	System Safety Program Plan
SSRS	Software Safety Requirements Specification
SSS	System, Subsystem Specification
SSTD	System Safety Test Description
SSTR	System Safety Test Record
SSWG	System Safety Working Group
STANAG	Standard NATO Agreement
STD	Software Test Description
STR	Software Test Record
SVD	Software Version Description Document
SVP	Software Verification Plan
SVR	Software Verification Record
SW	Software
SWAL	Software Assurance Level
SwCI	Software Criticality Index
TNC	Swedish Terminology Center
TPLS	Third Party Logistic Support
TRR	Test Readiness Review
TS	Technical Specification
UTC	Universal Time Coordinated
VÅS	Statement of Work, SoW

References

The following documents are source documents for the manual. The specified document designations are those that were relevant at the manual's completion. In cases where a particular reference is required, it is recommended that the presence of a later issue be checked.

Title, document
AFS 1998:05, Arbete vid bildskärm, *Computer Monitor work*
AOP-15 edition 3 (2009), Guidance On The Assessment Of The Safety And Suitability For Service Of Non-nuclear Munitions For Nato Armed Forces
AOP-52 edition 1 (2009), Guidance On Software Safety Design And Assessment Of Munition-related Computing Systems
Def Aust 5679 (2006), The Procurement of Computer-Based Safety-Critical Systems
Def Stan 00-56 edition 4, Safety Management Requirements for Defence Systems
DoD, Joint Software Systems Safety Engineering Handbook Version 1.0 Published August 27, 2010
ED-153, Guidelines for ANS Software Safety Assurance
EC Marine Equipment Directive 2014/90 / EU (2016: 768) Marine Equipment and Regulation (2016: 770) Marine Equipment, together with the Swedish Transport Agency's Regulations (TSFS 2016: 81) on marine equipment
EN 50126, Railway applications - The specification and demonstration of Reliability, Availability, Maintainability and Safety (RAMS)
EN 50128, Railway applications - Communication, signaling and processing systems - Software for railway control and protection systems
EN 50129, Railway applications – Communication, signaling and processing systems – Safety related electronic systems for signaling
EN 62061, EN 62061, Safety of machinery - Functional safety of electrical, electronic and programmable electronic safety-critical control systems
EN ISO 13849-1 Safety of machinery - Safety-related parts of control systems - Part 1: General design principles
Armed Force's Handbook on System Safety (H SystSäk E 2011, Part 1), M7739-352031
Armed Force's Handbook on System Safety (H SystSäk E 2011, Part 2), M7739-352032
FMV Handbok i användbarhet (H HFI), *Handbook on Human Factors Integration*

Title, document
FMV Handbok ISD IT-säkerhet Användningsfall och arkitektur, *Handbook on ISD IT Security Use cases and architecture*
FMV Handbok ISD IT-säkerhet Management, *Handbook on ISD IT Security Management*
FMV Handbok ISD IT-säkerhet Oberoende granskning, *Handbook on ISD IT Security Independent Review*
FMV Weapons and Ammunition Safety Manual 2012 (H VAS E), M7762-000881
Handbok säkerhetstjänst informationssäkerhet 2013 (H Säk Infosäk 2013 M7739-352056), *The Armed Forces' handbook on Security Service, Information Security*
IEC 60601, Electrical Equipment for Medical Use
IEC 61508, Functional Safety of Electrical/Electronic/Programmable Electronic Safety-related Systems
IEC 61511, Functional safety – Safety Instrumented Systems for the Process Industry Sector
ISO 15289, Systems and Software Engineering – Content of Life-cycle Information Items
ISO 26262, Road vehicles - Functional safety
ISO 9001:2015 – Quality management system
ISO/IEC 12207, System and software quality – Life cycle processes for software
ISO/IEC 15288, System and software quality - System life cycle processes
ISO/IEC 15504, Information Technology – Process Assessment
ISO/IEC 27000 Management System in Information Security
ISO/IEC/IEEE 24765:2010, Systems and software engineering – Vocabulary
MIL-STD 1472G (2012), Department Of Defense Design Criteria Standard: Human Engineering
MIL-STD 882E, System Safety
NASA Software Safety Guidebook (NASA-STD-8719.13)
RTCA DO-178C, Software Considerations in Airborne Systems and Equipment Certification
RTCA DO-254, Design Assurance Guidance for Airborne Electronic Hardware
SAE ARP4754A, Aerospace Recommended Practice - Guidelines for Development of Civil Aircraft and Systems

Title, document
STANAG 4187 Fuzing Systems - Safety Design Requirements
STANAG 4452, Safety assessment requirements for munition related computing systems

Appendix 1 Comparison Between Software Standards

Comparison tables for selected software standards regarding certain aspects.

Table A1:1 Administrative aspects

	IEC 61508	ISO 26262	EN ISO 13849-1	EN 62061	RTCA/ DO 178C	RTCA/DO-254	ARP 4754A	ED-153	EN 50128	IEC 61511
Sector / scope	Programmable electrical systems	Road vehicles	Machine Control	Machine Control	Airborne Systems (SW)	Programmable Logic (HW)	Airborne Systems (system)	Airborne Systems	Railway	Process Control
Current Edition (Release Year)	2010	2011	2015	2015	2011	2000	2010	2009	2011	2016
Number of parts in the standard	7	10	2	1	1	1	1	1	1	3

Table A1:2 Criticality classification

	IEC 61508	ISO 26262	EN ISO 13849-1	EN 62061	RTCA/ DO 178C	RTCA/DO-254	ARP 4754A	ED-153	EN 50128	IEC 61511
Reason for classification	Severity, exposure, controllability	Severity, exposure, controllability	Severity, frequency, opportunity to avoid	Severity probability	Severity	Severity	Severity	Severity probability (as shown)	Severity, frequency	Severity, probability

	IEC 61508	ISO 26262	EN ISO 13849-1	EN 62061	RTCA/ DO 178C	RTCA/DO-254	ARP 4754A	ED-153	EN 50128	IEC 61511
Methodology for classification	Risk Graph	Risk Graph	Risk Graph	Table	Assessment severity	Assessment severity	Assessment severity	Risk Graph	Risk Graph	Several methods in IEC 61511-3
Levels of classification	SIL 1–4	ASIL A–D	PL a–e	SIL 1–3	Level A–E	DAL A–E	Level A–E	SWAL 1–4	SIL 0–4	SIL 1–4
Highest level	SIL 4	ASIL D	PL e	SIL 3	Level A	DAL A	Level A	SWAL 1	SIL 4	SIL 4

Table A1:3 Technical scope

	IEC 61508	ISO 26262	EN ISO 13849-1	EN 62061	RTCA/ DO 178C	RTCA/DO-254	ARP 4754A	ED-153	EN 50128	IEC 61511
Purpose	Safety function	Product	Safety function	Safety function	Product	Product	Product	Product	Product	Product
Addresses system	Yes	Yes	Yes	Yes	No	No	Yes	To a small extent	No	Yes
Addresses SW	Yes	Yes	Yes	Yes	Yes	No	No	Yes	Yes	Yes
Addresses HW	Yes	Yes	Yes	Yes	No	Yes	No	No	No	Yes
Addresses information security	No	No	No	No	No	No	No	Yes, to some extent	No	Yes, to some extent
The standard sufficient in itself	Yes	Yes	Yes	Yes	In principle, only SW is included	In principle, only SW is included	No	Yes	No	Yes

	IEC 61508	ISO 26262	EN ISO 13849-1	EN 62061	RTCA/ DO 178C	RTCA/DO-254	ARP 4754A	ED-153	EN 50128	IEC 61511
Addresses subcontractor	No	Yes	No	No	Yes	No	No	Yes	Yes	Yes, IEC 61511-1, Section 5.2.5
Harmonized against EU directives	No	No	Yes	Yes	No	No	No	No	No	No
Certification	Yes	Yes	Yes	Yes	Yes, if DO178C, DO-254, ARP4754A is included	Yes, if DO178C, DO-254, ARP4754A is included	Yes, if DO178C, DO-254, ARP4754A is included	Yes	No	No

Table A1:4 Techniques and methods

	IEC 61508	ISO 26262	EN ISO 13849-1	EN 62061	RTCA/ DO 178C	RTCA/DO-254	ARP 4754A	ED-153	EN 50128	IEC 61511
Method of risk analysis	Yes	Yes	Yes	Yes	No	No	No	Yes	Yes, as example	Yes, IEC 61511-1, Chapter 8. Several methods in IEC 61511-3

	IEC 61508	ISO 26262	EN ISO 13849-1	EN 62061	RTCA/ DO 178C	RTCA/DO-254	ARP 4754A	ED-153	EN 50128	IEC 61511
Method of determining severity	Yes	Yes	Yes	Yes	No	No	Yes	No	No	Several methods in IEC 61511-3
Requirements: *shall* or *should*	*Shall*	*Shall*	*Shall*	*Shall*	*Should*	*Should*	*Should*	*Shall*	*Shall* and some *should*	*Shall*
Terminology to some extent adapted to standard	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
V model for development	Yes	Yes	Yes	Yes	No, possible but not specified	Yes	Yes	No, possible but not specified	Yes, as example	Yes
Requirements for persons and / or organization	Yes	Yes	Yes	No	No	No	No	Yes	Yes	Yes, IEC61511-1, Chapter 5
Full life cycle defined (SW, and HW)	Yes	Yes	Yes	Yes	No, just SW	No, just HW	Yes	Support exists for SW but not defined	No, found in EN 50126	Yes

Table A1:5 Methodology

	IEC 61508	ISO 26262	EN ISO 13849-1
Does the standard set special requirements for company competence?	No, just requires independence from those who perform *Functional Safety Assessment*. See Part 1, Table 4-5.	Yes, see Section 2, Section 5.4.2, which deals with *Safety culture*.	No
Does the standard state specific requirements for the competence of the individual?	Yes, Part 1, sections 6.2.13-15, 8.2.9 and 8.2.14	Yes, see Part 2, Section 5.4.3.	No
Is there a routine for independent software review?	No, but referred to as a requirement to be implemented, see part 3, section 7.9.2.12	Yes, see Part 2, Section 6.4.7. The requirements for independence increase with increasing ASIL.	No, but validation must be performed by persons who are independent of the design, see Part 2, Section 4.1.
Is there a specific methodology for architectural work?	Yes, see section 3 7.4.3 and also some information in Part 7, Annex F.	Yes, the standard prescribes a system design that with system architecture requirements. See Part 4, Section 7.4. Software architecture is described in Part 6, Chapter 7.	The standard works with system architecture and defines five different categories, but does not include software architecture requirements.
Does the standard describe principles for software development or provides suggestions for detailed design solutions?	Sets requirements, but does not provide any design solutions.	Yes, techniques and methods are provided depending on ASIL.	Yes, describes the structure and requirements for software in section 4.6.

	IEC 61508	ISO 26262	EN ISO 13849-1
Is there a methodology for how to integrate PDS software?	No, but trusted / verified software elements are included as part of part 3, tables A.2 and A.4.	Yes, qualification of software components is described in Part 8, Chapter 12.	No
Are there any requirements for how to carry out configuration management and quality assurance of software?	Yes, see Part 1, Section 6.2 and Part 3, Section 6.2.	Yes, see Part 8, Chapter 7.	The standard has no requirements for this.

	EN 62061	RTCA/DO 178C	RTCA/DO-254
Does the standard set special requirements for skills at the company?	No	No	No
Does the standard state specific requirements for the competence of the individual?	No	No, just one aspect regarding test preparation in section 6.2e	No
Is there a routine for independent software review?	No	Yes	Appendix A provides guidance on independence in the verification process. All verification on Level A and B should be independent. Level C or lower levels do not require independent authentication.
Is there a specific methodology for architectural work?	The standard addresses system architecture in section 6.6.2.1 and software architecture in section 6.11.3.3	No, just general principles.	Yes, Section 2.3 *Hardware Safety Assessment* addresses aspects that are important for architectural choices.

	EN 62061	RTCA/DO 178C	RTCA/DO-254
Does the standard describe principles for software development or provides suggestions for detailed design solutions?	Describes the structure and requirements of software in section 6.11.3.	No	The standard describes activities that should be performed to achieve the goals, rather than detailing how the design itself should look.
Is there a methodology for how to integrate PDS software?	No	Yes, Section 12.1 describes *Use of Previously Developed Software*.	Yes, see section 11.2.
Are there any requirements for how to carry out configuration management and quality assurance of software?	Yes, see section 6.11.3.2.	Configuration management is provided in Chapter 7 *Software Configuration Management process*. Verification of source code is generally listed in Chapter 6 *Software Verification Process*.	Yes, see chapter 7.

	ARP 4754A	ED-153	EN 50128
Does the standard set special requirements for competence at the company?	No	Yes, see table A.9.	Yes, to a certain extent in the *Training Process* and in the *Improvement Process*.
Does the standard set specific requirements for the competence of the individual?	No	Yes, based on roles, see section 5.2 and Annex B.	Yes, see *Training Process*

	ARP 4754A	ED-153	EN 50128
Is there a routine for independent software review?	No	No, but requirements are contained in section 7.5 *Component implementation and testing*.	No, only mentioned in the *Verification Process*.
Is there a specific methodology for architectural work?	No	No	No
Does the standard describe principles for software development or provides suggestions for detailed design solutions	No	No	No
Is there a methodology for how to integrate PDS software?	No	No, but there are requirements (referred to as pre-existing SW, COTS or open source SW).	Yes, both requirements and advice can be found in section 7.2.

	ARP 4754A	ED-153	EN 50128
Are there any requirements for how to configure configuration management and quality assurance of software?	Configuration management is described in many places and specifically in Section 5.6. Quality assurance for the entire system using authentication and validation, as described in Chapter 5.	A *Software Configuration Management Plan* shall be issued. Configuration requirements for what is to be addressed are found in many places. There is also a role defined, *Configuration Manager Role Specification*, with specified responsibilities and skills, see Table B.10. Section 6.5 *Software Quality Assurance* applies to quality assurance and document *Software Quality Assurance Plan* and *Software Quality Assurance Verification Report* shall be issued.	See *Configuration Management Process* for configuration management. The *Quality Assurance Process* provides the overall quality assurance requirements

	IEC 61511
Does the standard set special requirements for competence at the company?	Yes, see IEC 61511-1, chapter 5.
Does the standard set specific requirements for the competence of the individual?	Yes, see IEC 61511-1, Chapter 5.
Is there a routine for independent software review?	There is no routine, but independence is required both for verification (IEC 61511-1, Chapter 7) and *SIS safety validation* (IEC 61511-1, Chapter 15).
Is there a specific methodology for architectural work?	No
Does the standard describe principles for software development or provides suggestions for detailed design solutions	No
Is there a methodology for how to integrate PDS software?	No, but component selection requirements are contained in Section 11.5 of IEC 61511-1
Are there any requirements for how to configure configuration management and quality assurance of software?	Yes, software configuration management is included in SIS configuration management (IEC 61511-1, Section 5.2.7)

Appendix 2 Template for FMV's Functional Hazard Analysis (FHA)

The purpose of implementing an FHA is to identify system safety related design oriented requirements in the Technical Specification (TS). The project manager at FMV issues carries out the FHA and reports the results of the Armed Forces. This is a customized FHA for this purpose.

Accident/Top Event	*Use Phase*	**Caused by:** • **Suspended Function** • **Reduced Function** • **Incorrect Function** • **Unintentional function activated**	**Consequence given that accident risk occurs**	**Suggested Risk Reduction Action**	**Design Requirements to Prevent Accident Risk**
Activated Impact Given the Risk of Accident Person, property and environmental damage due to accidental shooting of own aircraft	Battle	Caused by missing function of aircraft identification equipment (IFF)	Death, property loss and limited environmental damage caused by the loss of own air craft	Double sensor system of the air defence system to ensure the identity of the aircraft	Aircraft must be possible to be identified in two independent ways. The anti-aircraft system must have at least 50% safe identification of the aircraft before fire can be delivered.
…					

Appendix 3 Examples of FMV's initial criticality classification and requirements

General

The Armed Forces require system safety, supplemented with tolerable risk levels for the technical system, in the current Materiel Objective. Tolerable risk level requirements are not used as a basis for *FMV's initial criticality classification*, but are used in the system safety work to later report the fulfilment of requirements when the Technical System is handed over to the Armed Forces. Regarding the Armed Forces' requirements for system safety, including requirements for tolerable risk levels for the technical system, often expressed in a risk matrix, please refer to the methodology of H SystSäk.

In the Material Objective, the intended operating profile shall also be stated. The operating profile is used to calculate the total operating time throughout the life of the technical system. The operating time is of importance to the contracted industry's requirement for the absence of dangerous failures in safety-critical functions, see *appendix 4*.

FMV needs to identify the worst possible consequences for personnel, property and the environment and also the total operating time of the technical system. Below a fictional example and the intended workflow from the receipt of the Material Objectives to FMV's tender documents for the contracted industry are presented.

Basic requirements

The Armed Forces intend to procure a new technical system and therefore issues a Material Objective to FMV. FMV identifies that the technical system will contain safety-critical computer systems. In this example, the computer system will control and monitor large amounts of energy.

The Armed Forces require that the tolerable risk level for single deaths (injury class I, according to H SystSäk) for a particular accident, may not exceed 1×10^{-6}/system during its life. See *figure 3:1*

Person		A	B	C	D	E	F
0	Most deaths	NT	NT	NT	NT	NT	T
I	Occasional deaths	NT	NT	NT	NT	T	T
II	Serious injury	NT	NT	NT	T	T	T
III	Less serious injury	NT	T	T	T	T	T
IV	Insignificant injury	T	T	T	T	T	T

Figure A3:1 Example of the Armed Force's requirements for tolerable risk levels for injury to personnel

Corresponding requirements for tolerable risk levels are specified for property and the environment. However, note that tolerable risk levels also can be expressed in other ways than above, for example, that CE marking is sufficient.

FMV's initial criticality classification of the technical system

FMV performs a simplified *Functional Hazard Analysis* (FHA) to identify dimensioning accidents, so-called top events, see *appendix 2*. The top events of the technical system are also the hazards that the contracted industry will address in its safety architecture. FMV reports the results of the FHA to the Armed Forces. This provides the basis for the Armed Forces' decision on which option is to be chosen and ordered as a development assignment to FMV. The results of the FHA and the Armed Force's requirements documents provide the basis for the governing design-oriented system safety requirements, which can be included in the FMV Technical Specification (TS).

Following the Armed Forces decision on what option to be realized, a development assignment is commissioned to FMV.

If the top events of the technical system can lead to high, serious or medium impacts on personnel, property and / or the environments, according to the matrix in *figure A3:2*, the result of *FMV's initial criticality classification* is "***HIGH***". This means that FMV's specifications shall require that industry must meet both the requirements of an established software standard for system safety critical software development, and *Basic Software Security Requirements* (GKPS) according to *chapter 8*. of this manual

If the worst consequences of the technical system's top events only result in low or no consequences, the result of FMV's initial criticism classification will be "***LOW***". This means This means that FMV's tender documentation must state that the industry only must comply with the *Basic Requirements for software safety* (GKPS) according to *chapter 8*.

Application matrix in accordance with MIL-STD 882E
for FMV's initial criticality classification of technical systems

Conse-quence level	Description	Application	FMV´s initial criticality classification
High	Technical system containing safety-critical software where the consequence of an accident results is **catastrophic** for the person, the economy and/or the environment *(multiple or single deaths, total system loss and/or permanent environmental damage).*	An agreed software safety standard is applied and the highest criticality require-ments are applied. FMV's documentation requirements are met	***HIGH*** FMV requires industry to comply with established soft-ware standards.
Critical	Technical system containing safety-critical software where the consequence of an acci-dent results is **critical** for the person, the economy and/or the environment *(serious and permanent personal injury, extensive economic and/or environmental damage).*	An agreed software safety standard is applied and higher criticality require-ments are applied. FMV's documentation requirements are met.	
Serious	Technical system containing safety-critical software where the consequence of an accident results in **serious** consequences for the person, the economy and/or the environment *(serious but non-permanent personal injury, significant economic and/or environmental damage).*	Agreed software security standards are applied and medium criticality require-ments are applied. FMV's documentation requirements are met.	
Marginal	Technical system containing safety-critical software where the consequence of an accident results in **marginal** consequences for person, economy and/or environment (less serious personal injury, less economic and/or environ-mental damage).	Basic requirements for software development for the lowest tolerable level of criticality are applied (GKPS).	***LOW*** FMV requires industry to use at least GKPS. *(However, the industry may choose to follow an established software standard)*
Neglible	Technical system containing software where the conse-quence of an accident results in negligible consequences for the person, the economy and/or the environment.	Basic requirements for software development for the lowest tolerable level of criticality are applied (GKPS).	

Figure A3:2 Application matrix for FMV's initial criticality classification for software

In this example the consequences for injury to personnel in case of an accident are considered high. In the tender documentation, FMV will require that the *Basic Requirements for Software Safety* (GKPS) are met and that the contracted industry indicates which established software standard it intends to follow in the development of the technical system.

FMV's initial criticality rating is documented in FMV's System Safety Program Plan (SSPP).

Calculation of total operating time

The Material Objective states the intended operating profile of the technical system. The operating profile can be expressed in different ways, and it can also be conditional on, for example, requirements for international deployments. Below is an example of how the operating profile can be expressed in a Material Objective.

> "The technical system shall have an operational life of 20 years (should 30 years). During one year of operation, the operational profile is estimated to be 50% operations, 20% exercise, 10% training, 15% in storage and 5% maintenance. The technical system must be kept in storage for 2 years (should 4 years).
>
> During peacetime use, one year of training is expected to correspond to approximately 9 months of operation. Driving distance approximately 4000 km / year. Firing of appr. 600 rounds / year. During international deployment, the mileage is estimated at approximately 8000 km / year.Firing appr. 1000 rounds/year".

Based on the above example of an operating profile, FMV estimates the total operating time of the technical system to a maximum of 10,000 hours over 30 years of service life. FMV will document the these calculations in a document, to be used in reporting to the Armed Forces showing compliance with tolerable risk levels.

Requirement statement in the tender documents

Based on the above example, FMV states the following in the tender documents:

- Requirements for tolerable risk levels for single deaths (injury class I, H SystSäk) can be a maximum of 1×10^{-6} per system over a lifetime of 30 years.
- *Basic Software Safety Requirements* (GKPS) in H ProgSäk 2018 must be met.
- Optional established software standards, applicable in the field of technology, shall be specified and fulfilled (FMV shall not specify the level of criticality in the specifications).
- The operating profile corresponds to a total operating time of at least 10,000 hours for the technical system.

Appendix 4 Examples of industry's workflow prior to contract

Industry Receipt of Request for Tender Documents

FMV's request for tender documents, as shown in *appendix 3*, states that:

- Requirements for tolerable risk levels for single deaths (injury class I, H SystSäk) may be a maximum of 1×10^{-6} per system over the life of 30 years.
- *Basic Software Safety Requirements* (GKPS) in H ProgSäk 2018 must be met.
- An established software standard, applicable in the field of technology, shall be specified and fulfilled.
- The operating profile corresponds to a total operating time of at least 10,000 hours for the technical system.

Industry's Tender to FMV

Since FMV is obliged to make a formal evaluation under the principle of equal treatment between different tenders, the industry must respond that all of the above-mentioned requirements are met.

In this example, the contracted industry selects the software standard IEC 61508. The motive for the selection is based on the fact that the contracted industry is already working according to IEC 61508 and has good experience in applying the standard. The industry's offer to FMV states that IEC 61508 will be applied and that the requirements for *Basic Software Safety Requirements* (GKPS) will be met. In addition, the requirement for tolerable risk levels and total operating time will be met. If FMV has requested further details in the specifications, this is attached.

Contract Agreement Between FMV and the Contracted Industry

FMV places an order on the industry based on stipulated requirements in the specifications.

Contract Review Between FMV and the Contracted Industry

Prior to the formal contractual review between FMV and the contracted industry, certain preparations are made.

The contracted industry issues a more detailed concept of the technical system and conducts an initial system safety analysis aimed at identifying hazards. In this work, the contracted industry starts with the most critical hazards for injury class I (catastrophic consequence for personnel, property and / or the environment) that may occur in the technical system. If injury class I cannot occur, class II shall be used instead. A more detailed design of the technical system's safety architecture can now be obtained.

In the breakdown of the requirements, based on the required tolerable risk level, those parts of the design that will control the criticality level of the computer system are identified based on the intended safety architecture.

The contracted industry then analyses the application of GKPS in the detailed concept. Industry also motivates choice of criticality according to methodology in the selected software standard.

The contracted industry considers that one of the requirements of GKPS is not applicable.

"2.802.09-T All error states that may affect system performance must be logged in an format that is possible to evaluate".

The reason is that because the safety critical computer system is planned as a built-in system based on a 32-bit microcontroller with limited internal memory, the logging capabilities are limited.

At the formal contractual review it was agreed that requirement 2.802.09-T is not possible to realize with the proposed technical solution but instead it was agreed on how an alternative method of logging using external logging equipment that can be connected when troubleshooting. The contracted industry also justifies the choice of SIL level according to the methodology in the standard IEC 61508.

For technical systems to be certified by a relevant authority, it is recommended that the contracted industry, prior to the contract review, issues the *Plan for Software Aspects of Certification* (PSAC) and *Acceptance Plan* (PSAA).

Clarifications and agreements at the contract review are documented in minutes signed by both parties. In cases where the agreements are deemed to affect the contract, a contract change will be implemented.

Below alternative system solutions for safety architectures that reduce the criticality level of the computer system are shown.

Workflow in the Definition of Safety Functions

In the design of safety functions, simplicity as well as known and proven technologies should preferably be used. If the safety function can be realized with subsystems with previous extensive experience, and where error modes and error rates are known, this also facilitates verification of stated requirements.

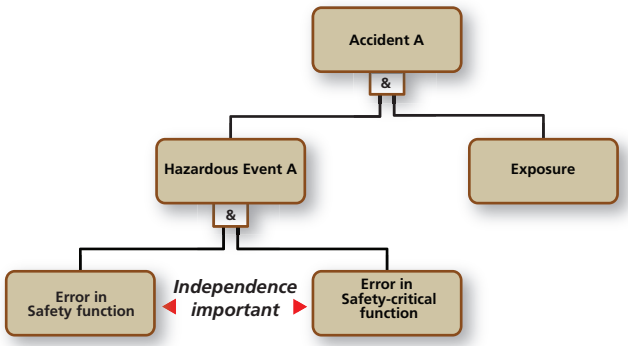


Figure A4:1 General fault tree for break-down of hazardous events

The aim of requirement break-down, of the probability of hazardous Event A, is to define an appropriate safety function so that as much of the requirement as possible is allocated to the safety function. In the case of highly specified requirements for the safety-critical function, this leads to a higher criticality rating for this part, affecting both the stringency of applied development methodology and verification of the stated requirement. From a verification point of view, in many cases, a better strategy is to allocate as much of the broken down safety function requirement.

Type Example (a)

Below is an example where a safety feature is introduced into the technical system so that the requirement for errors in the safety critical function A can be reduced to a level so that GKPS can be applied to this part of the technical system.

In the initial analyses, *Hazardous Event A* has been identified as the dangerous failure of the technical system that, when exposed, leads to *Accident A*.

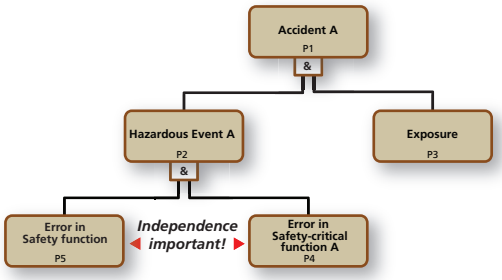


Figure A4:2 General fault tree to describe relationships in the accident model

The initial requirement for tolerable risk level (injury class I) may not exceed 1×10^{-6} per system over the life of 30 years.

The probability of exposure is set conservatively to p = 1 (P3), that is, the probability of *Accident A* (P1) = The probability of *Hazard A* (P2).

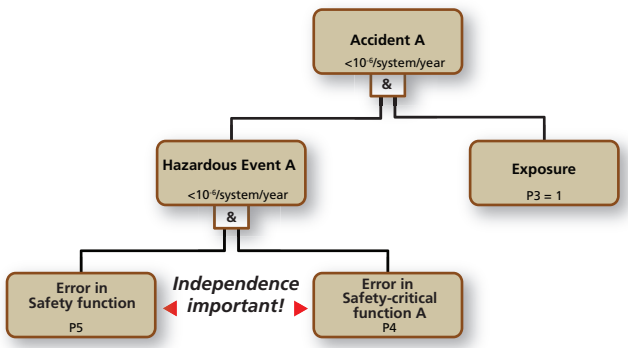


Figure A4:3 Examples of breakdown of requirement

Based on FMV's requirement for a 30-year operational life, the total operating time of the technical system is estimated to be up to 10,000 hours of operation (corresponding to approximately 1 year of continuous operation). This means that the tolerable risk

level for single deaths should be $<1 \times 10^{-6}$ per system / year or $<1 \times 10^{-10}$ per system/hour (10,000 hours equivalent to approximately 1 year).

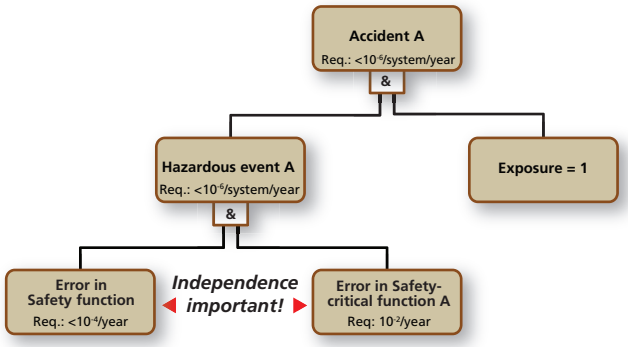


Figure A4:4 Examples of breakdown of requirement

If the safety function according to the break down of requirements in *figure A4:4* is assumed to have an error probability $<10^{-4}$ /system individual / year ($<10^{-8}$ / hour) then the safety critical function A must complete the remaining part, that is, an error probability $<10^{-2}$ /system / year ($<10^{-6}$ / hour).

Note that the independence between the two main branches of the fault tree must be considered so that the probabilities of both branches can be multiplied, that is, the probability of Hazard A is $<10^{-6} = (10^{-4} \times 10^{-2})$.

If the safety-critical function A is divided into two independent safety-critical functions (A1, A2) as shown in *figure A4:5*, a fault must occur at the same time in both channels A1 and A2 in order to create a dangerous fault in the safety-critical function A. In this ideal case, the broken down requirement of safety-critical function A can be broken down into two independent sub-functions A1 and A2.

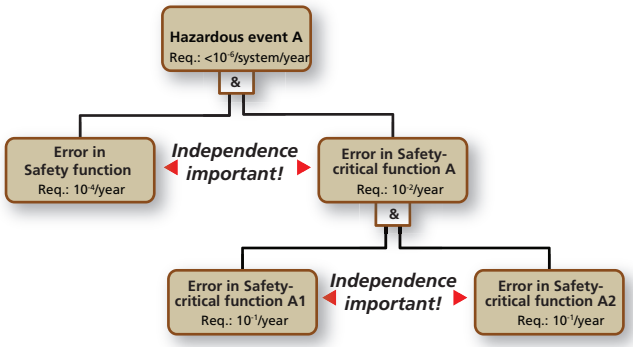


Figure A4:5 Requirement breakdown in safety-critical multi-channel diversified system

If, instead, a one-channel safety architecture is used, the breakdown of the requirements must be distributed so that the safety function takes a greater part of the broken down requirement.

Therefore, $<10^{-5}$/year for the safety function and remaining $<10^{-1}$/year for the safety-critical function, see *figure A4:6* below.

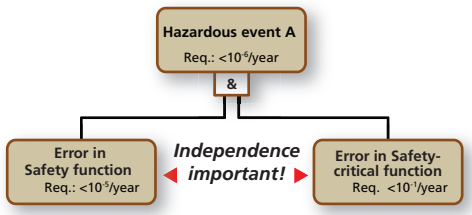


Figure A4:6 Example of requirements collision breakdown in a single-channel safety critical system

Note that the initial requirement for the technical system has *not changed* but only the part that could cause dangerous faults in safety-critical operation.

In both of these examples, the broken-down requirement of a randomly hazardous fault in the safety critical function is $<10^{-1}$ / year (or $<10^{-5}$ / hour). According to *table A4:1*, this requirement breakdown is implemented so that the initial criticality classification ***LOW*** and GKPS are considered sufficient to meet.

Application of GKPS for Continuous Operation

Table A4:1 Recalculation Table for Minimum applied error probability for random faults

System in Continuous Operation Total operating time during service life	Minimum Error Probability r in a Safety Critical Function with Initial Critical Classification ***LOW***
≤ 100 h	1×10^{-3} (p)
< 500 h	5×10^{-3} (p)
< 1 000 h	1×10^{-2} (p)
< 5 000 h	5×10^{-2} (p)
< 10 000 h	1×10^{-1} (p) (1 year continuous operation = 8 760 h) (1 year ≈ 10 000 h)
< 50 000 h	5×10^{-1} (p)
100 000 h	= 1

Type Example (b)

In this example, a one-channel safety critical function, ie. without redundancy but with basic events as described in *section 4.3.4* is used. This is also applicable to each branch under safety critical function in the multi-channel example shown in *figure A4:5*.

In *figure A4:7*, the continued requirement breakdown has been based on the previously assumed requirement for faults in a safety critical function A of $<10^{-1}$ /year (year ≈ 10,000h). The requirement has been allocated equally to the respective basic event (*fault in the actuator* or *malfunction of the sensor* or *failure in safety critical computer system*).

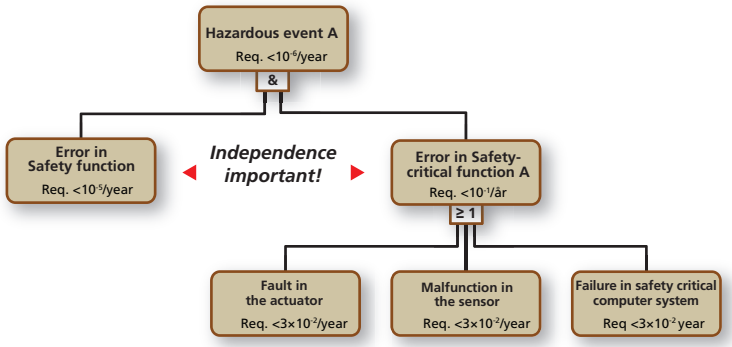


Figure A4:7 Reduced generic field tree, single channel safety critical system

The error probability for safety critical function A is approximated to 10^{-1} by summing the underlying base events.

In the next step, to further reduce the probability of *hazardous event A*, function monitoring / diagnostics of the *safety function* and *safety-critical function A's actuators* and *sensors* are also provided. The purpose of the monitoring is to detect random errors in order to further reduce the broken-down requirement of the safety-critical computer system A. The error rate as shown in *figure A4:7* is then increased as shown in *figure A4:8* below.

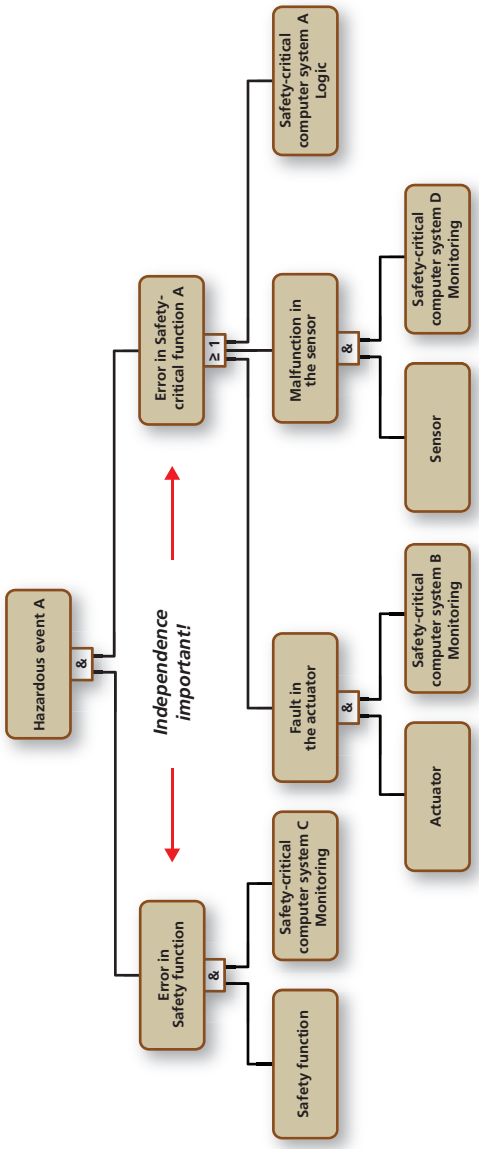


Figure A4:8 Reduced generic fault tree for a single-channel safety critical system with independent monitoring

A prerequisite is that the added monitoring in *Computer System C* can be considered independent of faults in the safety-critical *Computer systems A*, *B* and *D*. If the added safety monitoring

(*Safety Critical Computer System C, Monitoring*) safety is set to 10^{-1} / year, the probability allocated to the *Safety function* as shown in *figure A4:8* is reduced to $<10^{-4}$ / year.

When *Computer System C* is within the initial criticality rating ***LOW***, ie. 10^{-1} / year (year ≈ 10,000 hours), GKPS requirements are sufficient for the design of *Computer System C monitoring*.

If also defined in the same way, function monitoring of actuators and sensors via *Computer Systems B* and *D* monitoring also leads to an initial criticality rating ***LOW***, that is, with an error probability of 10^{-1} / year, the contribution from these two branches in the fault tree can be neglected in relation to a dangerous error in *Safety Critical Computer System A*, that is, the requirement for failure in *Safety Critical Function A* can be fully allocated to *Safety Critical Computer System A*, ie. 10^{-1} / year and thus initial a criticality rating ***LOW*** and GKPS are also applicable to this part.

Note that the requirements of GKPS include both hardware requirements to reduce the likelihood of random errors as well as requirements for stringency in the software development process to reduce systematic errors. The probability of hazardous event break-down is only valid for random errors. GKPS Initial Critical Classification ***LOW*** defines the minimum subset of the requirements that will mitigate the introduction of systematic errors for this level.

Appendix 5 Example of FMV's Requirements Fulfilment Template

The project manager at FMV shall ensure that the requirements of the contract with the contracted industry are met before delivery to the Armed Forces.

Req. no.	Requirement	Fulfilled	Not fulfilled	How is it fulfilled?	Reference / basis	Remarks/ other
2.801.03-A	Contracted industry shall designate a software safety contact.	×		Title/role, first name and last name.	Minutes xxx from contract review.	
2.802.07-T	Established programming languages shall be used in the development of safety-critical software.	×		Contracted industry uses C ++.	Minutes xxx from contract review.	
...						

The checklist is available in digital version on the FMV website. Whether a requirement is met or not, or if not applicable, can be specified in the file's requirement column (Yes / No / Not applicable).

Figures and Tables

Figures

Figure		Comment
4:4	*Redundant multi-channel system with three different computer systems and three different software and separate input channels*	
4:5	*Simplified accident model according to H SystSäk*	
4:6	*General fault tree to describe relationships in the accident model*	
4:7	*Breakdown of a hazardous event into safety function and safety critical function*	
4:8	*Safety-critical system, multi-channel redundant system (replica)*	
4:9	*Safety-critical system, single-channel*	
4:10	*Reduced generic fault tree, single channel safety critical system*	
4:11	*Reduced generic fault tree for a single-channel safety critical system with independent monitoring*	
4:12	*Criticality Levels for Different Software Standards*	
10:1	*Connection between NATO Standards/requirements for Ammunition and civilian Standards*	
A3:1	*Example of the Armed Force's requirements for tolerable risk levels for injury to personnel*	
A3:2	*Application matrix for FMV's initial criticality classification for software*	
A4:1	*Application matrix linked to MIL-STD 882E for FMV's initial criticality classification of technical systems*	
A4:2	*Redundant two-channel systems with identical computer systems and software with separate input channels*	
A4:3	*Examples of breakdown of requirement*	
A4:4	*Examples of breakdown of requirement*	
A4:5	*Requirement breakdown in safety-critical multi-channel diversified system*	
A4:6	*Example of requirements collision breakdown in a single-channel safety critical system*	
A4:7	*Reduced generic field tree, single channel safety critical system*	
A4:8	*Reduced generic fault tree for a single-channel safety critical system with independent monitoring*	

Drawings

Drawings can be found on the pages *22*, *66*, *83*, *86*, *113*, *117*, *122*, *130*, *137*, *141*, *185*, *193* and the cover.

Tables

Project Manager
Lars Lange, FMV

Subject Experts
Björn Koberstein, FMV
Mikael Lindbergh, FMV
Peter Djervbrant, Peter Djervbrant AB

Illustrations and cover
Stefan Gustafsson, UTBLICK MEDIA I HALLAND AB
Mats Lundgren, Combitech AB

Original
Mats Lundgren, Combitech AB

Digital edition
Mats Lundgren, Combitech AB