Introduction to safety engineering

Michal Sojka

Czech Technical University in Prague,
FEE and CIIRC

January 3, 2024

Outline

Outline

What is safety?

What is safety?

Classical definition

Freedom from those conditions that can cause death, injury, occupational illness, damage to or loss of equipment or property, or damage to the environment.

What is safety?

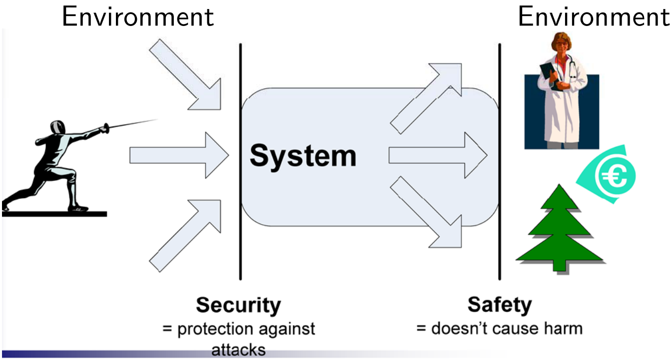
Classical definition

Freedom from those conditions that can cause death, injury, occupational illness, damage to or loss of equipment or property, or damage to the environment.

Alternative definition

Safety = Managing complexity without going crazy and ensuring completeness and consistency.

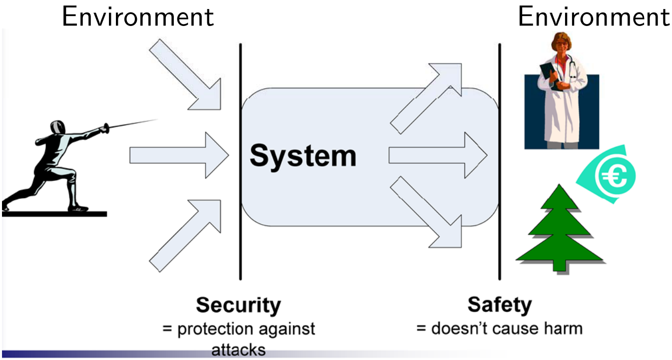
Safety vs. security



Source: TU Wien

- Safety = protection of environment from the system.
- Security = protection of the system from the environment.

Safety vs. security



Source: TU Wien

- Safety = protection of environment from the system.
- Security = protection of the system from the environment.
- But, environment is a system as well. So both safety and security represent a protection of one system from another...

What kind of system we have in mind?

- In general any system that can cause death, injury, ...
- In this course we deal mainly with software systems and also with electric/electronic systems.
- But safety is much broader term – you will see later.

How to start with safety?

So you want to develop safe systems?
How to start with that?

- Hard to say

How to start with safety?

So you want to develop safe systems?
How to start with that?

- Hard to say
- Safety is not a set of facts

How to start with safety?

So you want to develop safe systems?
How to start with that?

- Hard to say
- Safety is not a set of facts
- It is a wide range of knowledge that needs to be related

How to start with safety?

So you want to develop safe systems?
How to start with that?

- Hard to say
- Safety is not a set of facts
- It is a wide range of knowledge that needs to be related
- This relation happens at multiple (all) levels

How to start with safety?

So you want to develop safe systems?
How to start with that?

- Hard to say
- Safety is not a set of facts
- It is a wide range of knowledge that needs to be related
- This relation happens at multiple (all) levels
- Everybody starts with naive concepts of safety

How to start with safety?

So you want to develop safe systems?
How to start with that?

- Hard to say
- Safety is not a set of facts
- It is a wide range of knowledge that needs to be related
- This relation happens at multiple (all) levels
- Everybody starts with naive concepts of safety
- All basic facts seem obvious, but their correct implementation in engineering projects is difficult

Outline

System safety

System safety [DOD MIL-STD 882D Clause 3.2.13]

The application of engineering and management principles, criteria, and techniques to achieve acceptable mishap risk, within the constraints of operational effectiveness and suitability, time, and cost, throughout all phases of the system life cycle.

System safety

System safety [DOD MIL-STD 882D Clause 3.2.13]

The application of engineering and management principles, criteria, and techniques to achieve acceptable mishap risk, within the constraints of operational effectiveness and suitability, time, and cost, throughout all phases of the system life cycle.

Why management? Because experience has shown that many failures are not due to systems being built the wrong way but actually the wrong systems having been built. With other words management is there to make sure that engineering actually is doing the right thing (in all aspects).

System safety

System safety [DOD MIL-STD 882D Clause 3.2.13]

The application of engineering and management principles, criteria, and techniques to achieve acceptable mishap risk, within the constraints of operational effectiveness and suitability, time, and cost, throughout all phases of the system life cycle.

- Safety is not checklist
- It is necessary to interpret the principles

System safety

System safety [DOD MIL-STD 882D Clause 3.2.13]

The application of engineering and management principles, criteria, and techniques to achieve acceptable mishap risk, within the constraints of operational effectiveness and suitability, time, and cost, throughout all phases of the system life cycle.

- What is acceptable risk?
- 100% guarantee is never achieved!

System safety

System safety [DOD MIL-STD 882D Clause 3.2.13]

The application of engineering and management principles, criteria, and techniques to achieve acceptable mishap risk, within the constraints of operational effectiveness and suitability, time, and cost, throughout all phases of the system life cycle.

- Perfect technical solution is not always possible.

System safety

System safety [DOD MIL-STD 882D Clause 3.2.13]

The application of engineering and management principles, criteria, and techniques to achieve acceptable mishap risk, within the constraints of operational effectiveness and suitability, time, and cost, throughout all phases of the system life cycle.

- When we are done with system safety?
- When it went through all life-cycles stages:
  - Initial requirements
  - Design
  - Implementation
  - Service
  - Decommissioning
  - Disposal

System safety

System safety [DOD MIL-STD 882D Clause 3.2.13]

The application of engineering and management principles, criteria, and techniques to achieve acceptable mishap risk, within the constraints of operational effectiveness and suitability, time, and cost, throughout all phases of the system life cycle.

What is a system?

DOD MIL-STD 882D Clause 3.2.12:

An integrated composite of people, products, and processes that provide a capability to satisfy a stated need or objective.

ECSS-P-001A Rev A 1997 Clause 3.144:

System: Set of interdependent elements constituted to achieve a given objective by performing a specified function (IEC 50:1992).
NOTE: The system is considered to be separated from the environment and other external systems by an imaginary surface which cuts the links between them and the considered system. Through these links, the system is affected by the environment, is acted upon by external systems, or acts itself on the environment or the external systems.

What is a system?

IEC 61508-1 3.3.1:

Set of elements which interact according to a design, where an element of a system can be another system, called a subsystem, which may be a controlling system or a controlled system and may include hardware, software and human interaction

MOD 00-58 Clause 4.1.23:

A bounded physical entity that achieves in its environment a defined objective through interactions of its part.

NASA SP 6105 Rev1 2007:

A construct or collection of different elements that together produce results not obtainable by the elements alone.

What is a system?

RTCA DO 178C Annex B:

A collection of hardware and software components organized to accomplish a specific function or set of functions.

SAE ARP 4754a RevA 2010:

A combination of inter-related items arranged to perform a specific function(s)

Is terminology important?

- Safety is about communication at all levels.
  - engineers, managers, computer networks
- Goal: Establish common understanding of concepts.
- Implementation = transformation of concepts to actions
- If concepts differ but actions are coupled $\Rightarrow$ problems
- Terminology is not about finding the "true meaning"
- It teaches us to be sensitive to imprecision when communicating abstract concepts

Is terminology important?

- Safety is about communication at all levels.
  - engineers, managers, computer networks
- Goal: Establish common understanding of concepts.
- Implementation = transformation of concepts to actions
- If concepts differ but actions are coupled ⇒ problems
- Terminology is not about finding the "true meaning"
- It teaches us to be sensitive to imprecision when communicating abstract concepts
- Systems safety is about mitigation of problems arising from application of non-matching concepts

Is terminology important?

- Safety is about communication at all levels.
  - engineers, managers, computer networks
- Goal: Establish common understanding of concepts.
- Implementation = transformation of concepts to actions
- If concepts differ but actions are coupled ⇒ problems
- Terminology is not about finding the "true meaning"
- It teaches us to be sensitive to imprecision when communicating abstract concepts
- Systems safety is about mitigation of problems arising from application of non-matching concepts

If you don't understand anything in this lecture, just ask me!

Functional safety

Part of the overall safety [...] that depends on the correct functioning of the electrical and/or electronic and/or programmable electronic safety-related systems and other risk reduction measures.

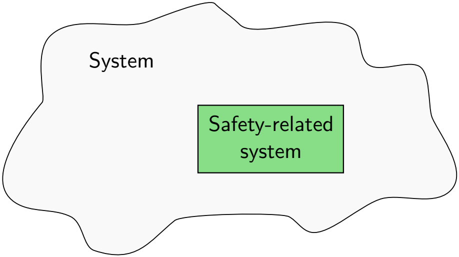
[IEC 61508-4/Ed.2, clause 3.1.12]

Functional safety

Part of the overall safety [...] that depends on the correct functioning of the electrical and/or electronic and/or programmable electronic safety-related systems and other risk reduction measures.

[IEC 61508-4/Ed.2, clause 3.1.12]

- Safety-related system (element): element which has the potential to contribute to the violation of or achievement of a safety goal.
  [ISO 26262-1:2010(E), 1.113]
- Example: fire alarm in a building, seat belt in a car



Functional safety

in other words

- Some elements in the system are more important (with respect to system safety) than others
- Those elements are called safety-related, or more informally safety-critical
- Functional safety standards define, how we should develop those elements (see the next section)

Outline

What are safety standards?

DIN 820 Part 1

Standardization is the collaborative unification of material and immaterial objects by interested parties for the good of the general public.

- Standards are created because there is a need for them.
  - By industry, governments or international bodies.
- The need arises because something went wrong without the standard (incompatibility etc.).
- Standards are here to help you.
- They contain useful knowledge that is hard (or painful) to gain. The knowledge was gained from failures in the past.
- Safety standards should not be followed without thinking.

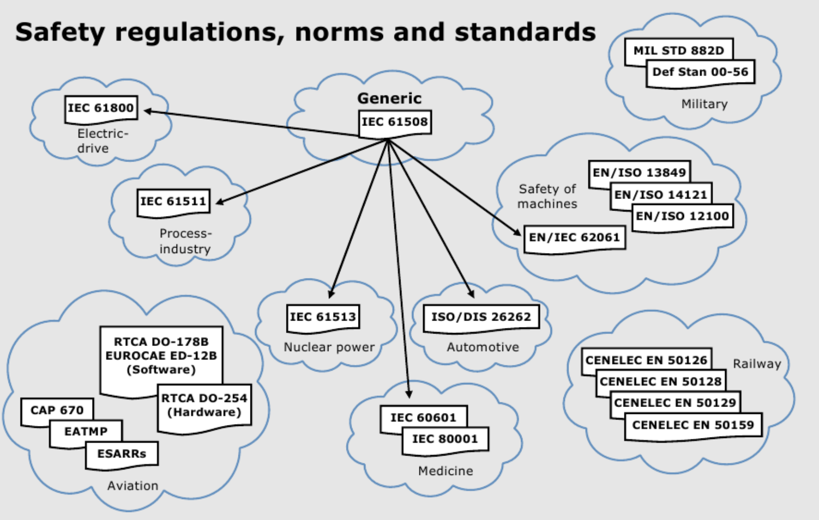
Liability

MOD Def Stan 00-74 Part 1 Preface (or any other Def Stan)

- Compliance with this Defence Standard shall not in itself relieve any person from any legal obligations imposed upon them.
- This standard has been devised solely for the use of the Ministry of Defence (MOD) and its contractors in the execution of contracts for the MOD. To the extent permitted by law, the MOD hereby excludes all liability whatsoever and howsoever arising (including, but without limitation, liability resulting from negligence) for any loss or damage however caused when the standard is used for any other purpose.

- European law contains similar things: When you produce something, you are liable (at least partially) for potential damage.
- $\Rightarrow$ IOW: Following the standards without thinking doesn't guarantee that you won't go to jail.

Hierarchy of safety standards



Source: FH Campus Wien

Hierarchy of safety standards

MIL-STD-882E – high-level standard

Section 4 gives good overview of system safety! We will see this later.

IEC 61508 – generic functional safety standard

This International Standard sets out a **generic approach** for all safety lifecycle activities for systems comprised of electrical and/or electronic and/or programmable electronic (E/E/PE) elements that are used to perform safety functions. This unified approach has been adopted in order that a rational and consistent technical policy be developed for all electrically-based safety-related systems. A major **objective is to facilitate the development of application sector standards**.
… enables application sector international standards, dealing with E/E/PE safety-related systems, to be developed; the development of application sector international standards, within the framework of this standard, **should lead to a high level of consistency** (for example, of underlying principles, terminology etc.) both within application sectors and across application sectors; this will have **both safety and economic benefits**;

Outline

Example: Ariane 5 Flight 501 Failure

4 June 1996

What happened?

- H0+36.7 seconds: Failure of the back-up Inertial Reference System followed immediately by failure of the active Inertial Reference System;
- swivelling into the extreme position of the nozzles of the two solid boosters and, slightly later, of the Vulcain engine, causing the launcher to veer abruptly;
- self-destruction of the launcher correctly triggered by rupture of the links between the solid boosters and the core stage.
- Loss: US $ 370 million

Source: `http://www.di.unito.it/~damiani/ariane5rep.html`

Failure event chain

in reversed time

- The launcher started to disintegrate at about H0 + 39 seconds because of high aerodynamic loads due to an angle of attack of more than 20 degrees that led to separation of the boosters from the main stage, in turn triggering the self-destruct system of the launcher.

Failure event chain

in reversed time

- The launcher started to disintegrate at about H0 + 39 seconds because of high aerodynamic loads due to an angle of attack of more than 20 degrees that led to separation of the boosters from the main stage, in turn triggering the self-destruct system of the launcher.
- This angle of attack was caused by full nozzle deflections of the solid boosters and the Vulcain main engine.

Failure event chain

in reversed time

- The launcher started to disintegrate at about H0 + 39 seconds because of high aerodynamic loads due to an angle of attack of more than 20 degrees that led to separation of the boosters from the main stage, in turn triggering the self-destruct system of the launcher.
- This angle of attack was caused by full nozzle deflections of the solid boosters and the Vulcain main engine.
- These nozzle deflections were commanded by the On-Board Computer (OBC) software on the basis of data transmitted by the active Inertial Reference System (SRI 2). Part of these data at that time did not contain proper flight data, but showed a diagnostic bit pattern of the computer of the SRI 2, which was interpreted as flight data.

Failure event chain

in reversed time

- The launcher started to disintegrate at about $H0 + 39$ seconds because of high aerodynamic loads due to an angle of attack of more than 20 degrees that led to separation of the boosters from the main stage, in turn triggering the self-destruct system of the launcher.
- This angle of attack was caused by full nozzle deflections of the solid boosters and the Vulcain main engine.
- These nozzle deflections were commanded by the On-Board Computer (OBC) software on the basis of data transmitted by the active Inertial Reference System (SRI 2). Part of these data at that time did not contain proper flight data, but showed a diagnostic bit pattern of the computer of the SRI 2, which was interpreted as flight data.
- The reason why the active SRI 2 did not send correct attitude data was that the unit had declared a failure due to a software exception.

Failure event chain

in reversed time

- The launcher started to disintegrate at about H0 + 39 seconds because of high aerodynamic loads due to an angle of attack of more than 20 degrees that led to separation of the boosters from the main stage, in turn triggering the self-destruct system of the launcher.
- This angle of attack was caused by full nozzle deflections of the solid boosters and the Vulcain main engine.
- These nozzle deflections were commanded by the On-Board Computer (OBC) software on the basis of data transmitted by the active Inertial Reference System (SRI 2). Part of these data at that time did not contain proper flight data, but showed a diagnostic bit pattern of the computer of the SRI 2, which was interpreted as flight data.
- The reason why the active SRI 2 did not send correct attitude data was that the unit had declared a failure due to a software exception.
- The OBC could not switch to the back-up SRI 1 because that unit had already ceased to function during the previous data cycle (72 milliseconds period) for the same reason as SRI 2.

Failure event chain (cont.)

- The internal SRI software exception was caused during execution of a data conversion from 64-bit floating point to 16-bit signed integer value. The floating point number which was converted had a value greater than what could be represented by a 16-bit signed integer. This resulted in an Operand Error. The data conversion instructions (in Ada code) were not protected from causing an Operand Error, although other conversions of comparable variables in the same place in the code were protected.

Failure event chain (cont.)

- The internal SRI software exception was caused during execution of a data conversion from 64-bit floating point to 16-bit signed integer value. The floating point number which was converted had a value greater than what could be represented by a 16-bit signed integer. This resulted in an Operand Error. The data conversion instructions (in Ada code) were not protected from causing an Operand Error, although other conversions of comparable variables in the same place in the code were protected.
- The error occurred in a part of the software that only performs alignment of the strap-down inertial platform. This software module computes meaningful results only before lift-off. As soon as the launcher lifts off, this function serves no purpose.

Failure event chain (cont.)

- The internal SRI software exception was caused during execution of a data conversion from 64-bit floating point to 16-bit signed integer value. The floating point number which was converted had a value greater than what could be represented by a 16-bit signed integer. This resulted in an Operand Error. The data conversion instructions (in Ada code) were not protected from causing an Operand Error, although other conversions of comparable variables in the same place in the code were protected.
- The error occurred in a part of the software that only performs alignment of the strap-down inertial platform. This software module computes meaningful results only before lift-off. As soon as the launcher lifts off, this function serves no purpose.
- The alignment function is operative for 50 seconds after starting of the Flight Mode of the SRIs which occurs at H0 – 3 seconds for Ariane 5. Consequently, when lift-off occurs, the function continues for approx. 40 seconds of flight. This time sequence is based on a requirement of Ariane 4 and is not required for Ariane 5.

Failure event chain (cont.)

- The internal SRI software exception was caused during execution of a data conversion from 64-bit floating point to 16-bit signed integer value. The floating point number which was converted had a value greater than what could be represented by a 16-bit signed integer. This resulted in an Operand Error. The data conversion instructions (in Ada code) were not protected from causing an Operand Error, although other conversions of comparable variables in the same place in the code were protected.
- The error occurred in a part of the software that only performs alignment of the strap-down inertial platform. This software module computes meaningful results only before lift-off. As soon as the launcher lifts off, this function serves no purpose.
- The alignment function is operative for 50 seconds after starting of the Flight Mode of the SRIs which occurs at H0 – 3 seconds for Ariane 5. Consequently, when lift-off occurs, the function continues for approx. 40 seconds of flight. This time sequence is based on a requirement of Ariane 4 and is not required for Ariane 5.
- The Operand Error occurred due to an unexpected high value of an internal alignment function result called BH, Horizontal Bias, related to the horizontal velocity sensed by the platform. This value is calculated as an indicator for alignment precision over time.

Failure event chain (cont.)

- The internal SRI software exception was caused during execution of a data conversion from 64-bit floating point to 16-bit signed integer value. The floating point number which was converted had a value greater than what could be represented by a 16-bit signed integer. This resulted in an Operand Error. The data conversion instructions (in Ada code) were not protected from causing an Operand Error, although other conversions of comparable variables in the same place in the code were protected.
- The error occurred in a part of the software that only performs alignment of the strap-down inertial platform. This software module computes meaningful results only before lift-off. As soon as the launcher lifts off, this function serves no purpose.
- The alignment function is operative for 50 seconds after starting of the Flight Mode of the SRIs which occurs at H0 – 3 seconds for Ariane 5. Consequently, when lift-off occurs, the function continues for approx. 40 seconds of flight. This time sequence is based on a requirement of Ariane 4 and is not required for Ariane 5.
- The Operand Error occurred due to an unexpected high value of an internal alignment function result called BH, Horizontal Bias, related to the horizontal velocity sensed by the platform. This value is calculated as an indicator for alignment precision over time.
- The value of BH was much higher than expected because the early part of the trajectory of Ariane 5 differs from that of Ariane 4 and results in considerably higher horizontal velocity values.

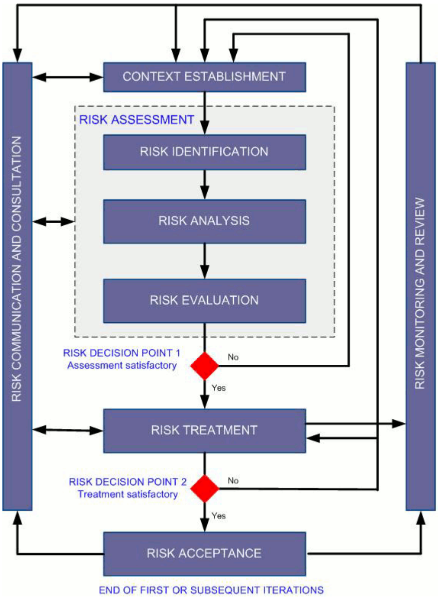
Outline

How to achieve safety?

By *"application of engineering and management principles, criteria, and techniques [...], throughout all phases of the system life cycle."*

[DOD MIL-STD 882D Clause 3.2.13]

Risk-based approach



Using standards

- Top level standards
  - Lower level standards/procedural standards
    - Project/management plans
      - Safety concept
        - Safety case (see later)

Generic safety process

Top level standard MIL-STD-882E contains a concise overview in Section 4: `https://system-safety.org/resource/resmgr/documents/MIL-STD-882E.pdf`
Look at the standard yourself! It is not complex and difficult to read.

Generic safety process

Top level standard MIL-STD-882E contains a concise overview in Section 4: `https://system-safety.org/resource/resmgr/documents/MIL-STD-882E.pdf`
Look at the standard yourself! It is not complex and difficult to read.

1. Document the system safety approach
2. Identify and document hazards
3. Assess and document risk
4. Identify and document risk mitigation measures
5. Reduce risk
6. Verify, validate and document risk reduction
7. Accept risk and document
8. Manage life-cycle risk

Risk mitigation measures

MIL-STD-882E, Clause 4.3.4

Eliminate the hazard if possible. When a hazard cannot be eliminated, the associated risk should be reduced to the lowest acceptable level within the constraints of cost, schedule, and performance by applying the system safety design order of precedence.

1. Eliminate hazards through design selection.
2. Reduce risk through design alteration.
3. Incorporate engineered features or devices.
4. Provide warning devices.
5. Incorporate signage, procedures, training, and personal protective equipment.

Lowest-level “standard”

- Top level standards
  - Lower level standards/procedural standards
    - Project/management plans
      - Safety concept
        - **Safety case**

Safety case

Definition

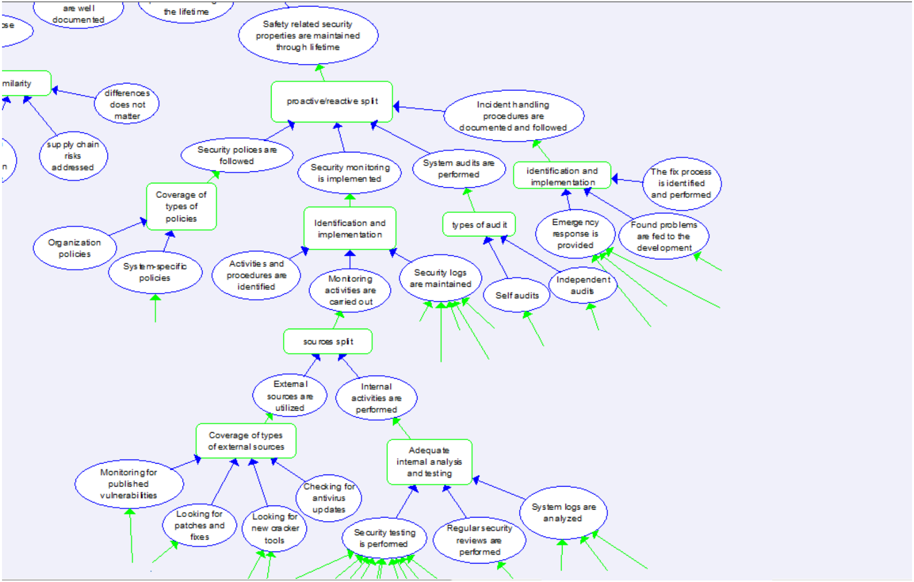
A safety case is an evidence-based explanation of why it is believed that a system is safe enough to be used in its intended application.

Specific for each application. This can be a Word document or a structured safety case.

Structured safety case

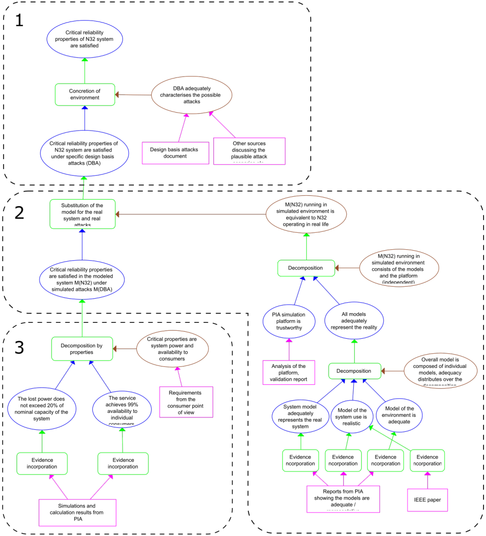
- Overall approach.
- Claim-Argument-Evidence (CAE) structure.
- Top-level claim supported by arguments and evidences that the arguments are correct.
- Split into sub-claims.
- Safety standards give guidance how to construct arguments.

CAE example


are well documented
the lifetime
Safety related security properties are maintained through lifetime
proactive/reactive split
differences does not matter
Incident handling procedures are documented and followed
supply chain risks addressed
Security polices are followed
Security monitoring is implemented
System audits are performed
identification and implementation
The fix process is identified and performed
Coverage of types of policies
Identification and implementation
types of audit
Emergency response is provided
Found problems are fed to the development
Organization policies
System-specific policies
Activities and procedures are identified
Monitoring activities are carried out
Security logs are maintained
Self audits
Independent audits
sources split
External sources are utilized
Internal activities are performed
Coverage of types of external sources
Adequate internal analysis and testing
Monitoring for published vulnerabilities
Checking for antivirus updates
Looking for patches and fixes
Looking for new cracker tools
Security testing is performed
Regular security reviews are performed
System logs are analyzed

CAE example II.

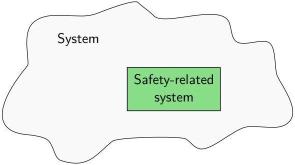
Security-informed safety case



Source: City University London, P. Popov

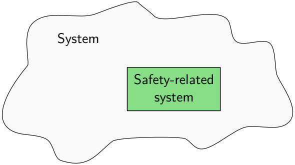
Safety integrity

Safety-related systems are used to reduce the identified risks to tolerable level. Therefore, safety of the whole system depends on proper function of these systems.



Safety integrity

Safety-related systems are used to reduce the identified risks to tolerable level. Therefore, safety of the whole system depends on proper function of these systems.



Definition (Safety integrity)

The probability of a safety-related system satisfactory performing the required safety functions under all the stated conditions within a stated period of time.

- Two components:
  - Random failure integrity (≈ hardware reliability)
  - Systematic failure integrity (≈ human mistakes, all SW failures)
- Question: How can I determine, that my safety-related system has sufficient safety integrity?

Safety integrity level

Standards (i.e. IEC 61508) gives the answer by defining “safety integrity level” and requirements for each level.

Definition (Safety Integrity Level)

Required level of protection against systematic failure in specification of the functions allocated to the safety-related systems.

Safety integrity level

Standards (i.e. IEC 61508) gives the answer by defining "safety integrity level" and requirements for each level.

Definition (Safety Integrity Level)

Required level of protection against systematic failure in specification of the functions allocated to the safety-related systems.

Safety integrity levels: target failure measures for a safety function operating in low demand mode of operation

Safety integrity level	Average probability of dangerous failure on demand of the safety function
4	$\geq 10^{-5}$ to $< 10^{-4}$
3	$\geq 10^{-4}$ to $< 10^{-3}$
2	$\geq 10^{-3}$ to $< 10^{-2}$
1	$\geq 10^{-2}$ to $< 10^{-1}$

How to achieve that?

Safety integrity levels: target failure measures for a safety function operating in high demand mode of operation or continuous mode of operation

Safety integrity level	Dangerous failure rate of the safety function [hr^{-1}]
4	$\geq 10^{-9}$ to $< 10^{-8}$
3	$\geq 10^{-8}$ to $< 10^{-7}$
2	$\geq 10^{-7}$ to $< 10^{-6}$
1	$\geq 10^{-6}$ to $< 10^{-5}$

Safety integrity level – cont.

Safety integrity levels: target failure measures for a safety function operating in high demand mode of operation or continuous mode of operation

Safety integrity level	Dangerous failure rate of the safety function [hr^{-1}]
4	$\geq 10^{-9}$ to $< 10^{-8}$
3	$\geq 10^{-8}$ to $< 10^{-7}$
2	$\geq 10^{-7}$ to $< 10^{-6}$
1	$\geq 10^{-6}$ to $< 10^{-5}$

In other words:

- SIL1 $\approx$ one dangerous failure per 10 years
- SIL4 $\approx$ one dangerous failure per 10000 years

How to achieve that?

- Testing is impossible – we don't have enough time.
- Instead we use indirect ways to achieve these goals (see next slides)

Achieving SIL requirements in general

- SIL1 demands basic sound engineering practices, such as adherence to a stan dard quality system, repeatable and systematically documented development processes, thorough verification and validation, documentation of all decisions, activities an d results, and independent assessment.
- Higher SILs, in turn, demand this foundation plus further rigour.

[Source: Felix Redmill, Understanding the Use, Misuse and Abuse of Safety Integrity Levels, 2000]

Some SIL requirements from IEC61508 Ed. 2

Table A.1 – Software safety requirements specification

(see 7.2)

	Technique/Measure*	Ref.	SIL1	SIL2	SIL3	SIL4
1a	Semi-formal methods	Table B.7	R	R	HR	HR
1b	Formal methods	B.2.4	---	R	R	HR
2	Forward traceability between the system safety requirements and the software safety requirements	TBA	HR	HR	HR	HR
3	Backward traceability between the safety requirements and the perceived safety needs	TBA	HR	HR	HR	HR
4	Computer-aided specification tools to support appropriate techniques/measures above	B.2.4	R	R	HR	HR

NOTE 1 – The software safety requirements specification will always require a description of the problem in natural language and any necessary mathematical notation that reflects the application.

NOTE 2 – The table reflects additional requirements for specifying the software safety requirements clearly and precisely.

NOTE 3 – See Annex C table C-A.1.

* Appropriate techniques/measures shall be selected according to the safety integrity level. Alternate or equivalent techniques/measures are indicated by a letter following the number. Only one of the alternate or equivalent techniques/measures has to be satisfied.

Legend: **NR** – not recommended, **R** – recommended, **HR** – highly recommended

Some SIL requirements from IEC61508 Ed. 2

**Table A.2 – Software design and development:
software architecture design**

(see 7.4.3)

NOTE - Some of these methods are about design concepts, others about how the design is represented.

Technique/Measure*		Ref	SIL1	SIL2	SIL3	SIL4
	Architecture and design feature					
1	Fault detection	C.3.1	---	R	HR	HR
2	Error detecting codes	C.3.2	R	R	R	HR
3a	Failure assertion programming	C.3.3	R	R	R	HR
3b	Safety bag techniques	C.3.4	---	R	R	R
3c	Diverse redundancy	C.3.5	---	---	---	HR
3d	Backward recovery	C.3.7	R	R	R	R
3e	Stateless design (or limited state design)	TBA	---	---	R	HR
3f	Re-try fault recovery mechanisms	C.3.9	R	R	R	HR
3g	Memorising executed cases (e.g. in transaction systems)	C.3.10	---	R	R	HR
4	Graceful degradation	C.3.11	R	R	HR	HR
5	Artificial intelligence - fault correction	C.3.12	---	NR	NR	NR
6	Dynamic reconfiguration	C.3.13	---	NR	NR	NR
7	Modular approach	Table B.9	HR	HR	HR	HR
8	Use of trusted/verified software modules and components (if available)	C.2.10 C.4.5	R	HR	HR	HR
9	Forward traceability between the software safety requirements specification and software architecture	TBA	HR	HR	HR	HR
10	Backward traceability between the software safety requirements specification and software architecture	TBA	HR	HR	HR	HR
	Design methods, notations and tools					

Some SIL requirements from IEC61508 Ed. 2

Table A.9 – Software verification

(see 7.9)

	Technique/Measure*	Ref	SIL1	SIL2	SIL3	SIL4
1	Formal proof	C.5.13	---	R	R	HR
2	Animation of specification and design	TBA	R	R	R	R
3	Static analysis	B.6.4 Table B.8	R	HR	HR	HR
4	Dynamic analysis and testing	B.6.5 Table B.2	R	HR	HR	HR
5	Forward traceability between the software design specification and the software verification (including data verification) plan.	TBA	HR	HR	HR	HR
6	Backward traceability between the software verification (including data verification) plan and the software design specification .	TBA	HR	HR	HR	HR

Some SIL requirements from IEC61508 Ed. 2

Table 5 – Minimum levels of independence of those carrying out functional safety assessment (overall safety lifecycle phase 9, including all phases of E/E/PE~~S~~ system and software safety lifecycles (see ~~f~~Figures 2, 3 and 4))

Minimum level of Independence	Safety integrity level			
	1	2	3	4
Independent person	HR	HR 1	NR	NR
Independent department	–	HR 2	HR 1	NR
Independent organization (see ~~n~~Note 2 of 8.2.12)	–	–	HR 2	HR
NOTE ~~–~~ See 8.2.12 (including notes), 8.2.13 and 8.2.14 for details on interpreting this table.				

Traceability

- Evidence that the low level requirements were translated to Source Code
- Evidence that no code was added that does not stem from a requirement
- Traceability means that you can answer the question "Why is this code here in this form" for every line I point at.

Process of determining safety integrity level

1. Hazard identification (e.g. HAZOP study – next lecture)

Process of determining safety integrity level

1. Hazard identification (e.g. HAZOP study – next lecture)
2. Assign a probability of occurrence to each of the identified hazards

Process of determining safety integrity level

1. Hazard identification (e.g. HAZOP study – next lecture)
2. Assign a probability of occurrence to each of the identified hazards
3. Identify mechanisms which protect against particular hazards

Process of determining safety integrity level

1. Hazard identification (e.g. HAZOP study – next lecture)
2. Assign a probability of occurrence to each of the identified hazards
3. Identify mechanisms which protect against particular hazards
4. Identify effects

Process of determining safety integrity level

1. Hazard identification (e.g. HAZOP study – next lecture)
2. Assign a probability of occurrence to each of the identified hazards
3. Identify mechanisms which protect against particular hazards
4. Identify effects
5. Identify severity of effects

Process of determining safety integrity level

1. Hazard identification (e.g. HAZOP study – next lecture)
2. Assign a probability of occurrence to each of the identified hazards
3. Identify mechanisms which protect against particular hazards
4. Identify effects
5. Identify severity of effects
6. Calculate the risk class (e.g. according to the next slide)
   - I Intolerable risk
   - II Undesirable risk
   - III Tolerable risk
   - IV Negligible risk

Process of determining safety integrity level

1. Hazard identification (e.g. HAZOP study – next lecture)
2. Assign a probability of occurrence to each of the identified hazards
3. Identify mechanisms which protect against particular hazards
4. Identify effects
5. Identify severity of effects
6. Calculate the risk class (e.g. according to the next slide)
   - I Intolerable risk
   - II Undesirable risk
   - III Tolerable risk
   - IV Negligible risk
7. Identify safety integrity level
   - We want to reduce risk to class IV (negligible)
   - Systems with SIL1 capability can be used to reduce risk by one level
   - Systems with SIL4 capability can be used to reduce risk by three levels

Risk classes

MIL-STD-882E, Table III (not IEC61508 Ed. 2!)

RISK ASSESSMENT MATRIX				
SEVERITY / PROBABILITY	Catastrophic (1)	Critical (2)	Marginal (3)	Negligible (4)
Frequent (A)	High	High	Serious	Medium
Probable (B)	High	High	Serious	Medium
Occasional (C)	High	Serious	Medium	Low
Remote (D)	Serious	Medium	Medium	Low
Improbable (E)	Medium	Medium	Medium	Low
Eliminated (F)	Eliminated			

Safety culture

- We learn from mistakes → Traceability, consistency and completeness
- Safety culture is about ensuring that at every step you ask "what could happen".

References

- Nicholas McGuire, Safety Manuals
- Dr. Andreas Gerstinger; Safety Engineering Course Slides, TU Wien.
- Gregor Pokorny; Safety Related Systems Slides, FH Campus Wien.
- F. Redmill, M. Chudleigh, J. Catmur; System Safety: HAZOP and Software HAZOP, Wiley, 1999.
- F. Redmill, Understanding the Use, Misuse and Abuse of Safety Integrity Levels, 2000
- SESAMO project