What is Safety?

- **Absolute definition:**
  - „Safety is freedom from accidents or losses.“
  - Nancy Leveson, „Safeware: System safety and computers“
- **But is there such a thing as absolute safety?**
- **Technical definition:**
  - „Sicherheit: Freiheit von unvertretbaren Risiken“
    - IEC 61508-4:2001, §3.1.8
- **Next week:** a safety-critical development process

Legal Grounds

- **The machinery directive:**
  - **Scope:**
    - Machineries (with a drive system and movable parts).
  - **Structure:**
    - Sequence of whereas clauses (explanatory)
      - followed by 29 articles (main body)
      - and 12 subsequent annexes (detailed information about particular fields, e.g., health & safety)
    - Some application areas have their own regulations:
      - Cars and motorcycles, railways, planes, nuclear plants ...
**What does that mean?**

- Relevant for all machinery (from tin-opener to AGV)
- Annex IV lists machinery where safety is a concern
- Standards encode current best practice.
  - Harmonised standard available?
- External certification or self-certification
  - Certification ensures and documents conformity to standard.
- Result:

> Note that the scope of the directive is market harmonisation, not safety – that is more or less a byproduct.

**The Norms and Standards Landscape**

- First-tier standards (A-Normen):
  - General, widely applicable, no specific area of application
  - Example: IEC 61508
- Second-tier standards (B-Normen):
  - Restriction to a particular area of application
  - Example: ISO 26262 (IEC 61508 for automotive)
- Third-tier standards (C-Normen):
  - Specific pieces of equipment
  - Example: IEC 61496-3 (“Berührungslos wirkende Schutzeinrichtungen”)
  - Always use most specific norm.

**Norms for the Working Programmer**

- IEC 61508:
  - Widely applicable, general, considered hard to understand
- ISO 26262:
  - Specialisation of 61508 to cars (automotive industry)
- DIN EN 50128
  - Specialisation of 61508 to software for railway industry
- RTCA DO 178-B:
  - “Software Considerations in Airborne Systems and Equipment Certification”
  - Airplanes, NASA/ESA
- ISO 15408:
  - “Common Criteria for Information Technology Security Evaluation”
  - Security, evolved from TCSEC (US), ITSEC (EU), CTCPEC (Canada)

**Introducing IEC 61508**

- Part 1: Functional safety management, competence, establishing SIL targets
- Part 2: Organising and managing the life cycle
- Part 3: Software requirements
- Part 4: Definitions and abbreviations
- Part 5: Examples of methods for the determination of safety-integrity levels
- Part 6: Guidelines for the application
- Part 7: Overview of techniques and measures

**How does this work?**

1. Risk analysis determines the safety integrity level (SIL)
2. A hazard analysis leads to safety requirement specification.
3. Safety requirements must be satisfied
   - Need to verify this is achieved.
   - SIL determines amount of testing/proving etc.
4. Life-cycle needs to be managed and organised
   - Planning: verification & validation plan
   - Note: personnel needs to be qualified.
5. All of this needs to be independently assessed.
   - SIL determines independence of assessment body.

**Safety Integrity Levels**

<table>
<thead>
<tr>
<th>SIL</th>
<th>High Demand (more than once a year)</th>
<th>Low Demand (once a year or less)</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>$10^{-6} &lt; P/hr &lt; 10^{-4}$</td>
<td>$10^{-4} &lt; P/hr &lt; 10^{-2}$</td>
</tr>
<tr>
<td>3</td>
<td>$10^{-7} &lt; P/hr &lt; 10^{-6}$</td>
<td>$10^{-5} &lt; P/hr &lt; 10^{-3}$</td>
</tr>
<tr>
<td>2</td>
<td>$10^{-10} &lt; P/hr &lt; 10^{-9}$</td>
<td>$10^{-6} &lt; P/hr &lt; 10^{-4}$</td>
</tr>
<tr>
<td>1</td>
<td>$10^{-15} &lt; P/hr &lt; 10^{-14}$</td>
<td>$10^{-8} &lt; P/hr &lt; 10^{-6}$</td>
</tr>
</tbody>
</table>

- P: Probability of dangerous failure (per hour/year)
- Examples:
  - High demand: car brakes
  - Low demand: airbag control
  - Which SIL to choose? → Risk analysis
  - Note: SIL only meaningful for specific safety functions.

**Establishing target SIL I**

- IEC 61508 does not describe standard procedure to establish a SIL target, it allows for alternatives:
  - Quantitative approach
    - Start with target risk level
    - Factor in fatality and frequency
  - Example:
    - Safety system for a chemical plant
    - Max. tolerable risk exposure $A=10^{-4}$
    - $B=10^2$ hazardous events lead to fatality
    - Unprotected process fails $C=1/5$ years
    - Then Failure on Demand $E = A(B^C) = 5\times 10^{-3}$, so SIL 2

**Establishing target SIL II**

- Risk graph approach
  - Example: safety braking system for an AGV
**What does the SIL mean for the development process?**

- In general:
  - „Competent” personnel
  - Independent assessment („four eyes“)
- SIL 1:
  - Basic quality assurance (e.g. ISO 9001)
- SIL 2:
  - Safety-directed quality assurance, more tests
- SIL 3:
  - Exhaustive testing, possibly formal methods
  - Assessment by separate department
- SIL 4:
  - State-of-the-art practices, formal methods
  - Assessment by separate organisation

**Increasing SIL by redundancy**

- One can achieve a higher SIL by combining independent systems with lower SIL („Mehrklangsysteme“).
- Given two systems A, B with failure probabilities $P_A$, $P_B$, the chance for failure of both is (with $P_{CC}$ probability of common-cause failures):
  
  $P_{AB} = P_A + P_B - P_{CC}$

- Hence, combining two SIL 3 systems may give you a SIL 4 system.
- However, be aware of systematic errors (and note that IEC 61508 considers all software errors to be systematic).
- Note also that for fail-operational systems you need three (not two) systems.

**The Safety Life Cycle**

**The Software Development Process**

- 61508 mandates a V-model software development process
  - More next lecture
- Appx A, B give normative guidance on measures to apply:
  - Error detection needs to be taken into account (e.g. runtime assertions, error detection codes, dynamic supervision of data/control flow)
  - Use of strongly typed programming languages (see table)
  - Discouraged use of certain features: recursion(!), dynamic memory, unrestricted pointers, unconditional jumps
  - Certified tools and compilers must be used.
  - Or „proven in use“

**Table A.2, Software Architecture**

<table>
<thead>
<tr>
<th>SIL</th>
<th>Zero Failure</th>
<th>One Failure</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>12 ops 12 yrs</td>
<td>24 ops 24 yrs</td>
</tr>
<tr>
<td>2</td>
<td>120 ops 120 yrs</td>
<td>240 ops 240 yrs</td>
</tr>
<tr>
<td>3</td>
<td>1200 ops 1200 yrs</td>
<td>2400 ops 2400 yrs</td>
</tr>
<tr>
<td>4</td>
<td>12000 ops 12000 yrs</td>
<td>24000 ops 24000 yrs</td>
</tr>
</tbody>
</table>

**Table A.4 – Software Design & Development**

<table>
<thead>
<tr>
<th>Variable/Method</th>
<th>Value</th>
<th>SIL 1</th>
<th>SIL 2</th>
<th>SIL 3</th>
<th>SIL 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Control Structure</td>
<td>90%</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Software structure</td>
<td>95%</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Software testability</td>
<td>85%</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Documentation</td>
<td>90%</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Inspection</td>
<td>90%</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Verification</td>
<td>90%</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Maintenance</td>
<td>90%</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Test Plans</td>
<td>90%</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

**Table A.9 – Software Verification**

<table>
<thead>
<tr>
<th>Verification Method</th>
<th>Value</th>
<th>SIL 1</th>
<th>SIL 2</th>
<th>SIL 3</th>
<th>SIL 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Software testing</td>
<td>90%</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Formal verification</td>
<td>90%</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Static analysis</td>
<td>90%</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Dynamic analysis</td>
<td>90%</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>
Table B.1 – Coding Guidelines

Table B.5 - Modelling

Certification

Security: The Common Criteria

Common Criteria (IEC 15408)

Common Criteria (CC)

Concept of Evaluation
Requirements Analysis

- The security environment includes all the laws, organizational security policies, customs, expertise and knowledge that are determined to be relevant.
  - It thus defines the context in which the TOE is intended to be used.
  - The security environment also includes the threats to security that are, or are held to be, present in the environment.

- A statement of applicable organizational security policies would identify relevant policies and rules.
  - For an IT system, such policies may be explicitly referenced, whereas for a general purpose IT product or product class, working assumptions about organizational security policy may need to be made.

The results of the analysis of the security environment could then be used to state the security objectives that counter the identified threats and address identified organizational security policies and assumptions.

- The security objectives should be consistent with the stated operational aim or product purpose of the TOE, and any knowledge about its physical environment.

Requirements Analysis

- The intent of determining security objectives is to address all of the security concerns and to declare which security aspects are either addressed directly by the TOE or by its environment.
  - This categorization is based on a process incorporating engineering judgment, security policy, economic factors and risk acceptance decisions.
  - Corresponds to (part of) requirements definition!

Requirements Analysis

- The IT security requirements are the refinement of the security objectives into a set of security requirements for the TOE and security requirements for the environment which, if met, will ensure that the TOE can meet its security objectives.
  - The CC presents security requirements under the distinct categories of functional requirements and assurance requirements.

- Functional requirements
  - Security behavior of IT-system
  - E.g. identification & authentication, cryptography,...

- Assurance Requirements
  - Establishing confidence in security functions
  - Correctness of implementation
  - E.g. Development, life cycle support, testing,...

Requirements Analysis

- A statement of assumptions which are to be met by the environment of the TOE in order for the TOE to be considered secure.
  - This statement can be accepted as axiomatic for the TOE evaluation.

- A statement of threats to security of the assets would identify all the threats perceived by the security analysis as relevant to the TOE.
  - The CC characterizes a threat in terms of a threat agent, a presumed attack method, any vulnerabilities that are the foundation for the attack, and identification of the asset under attack.

- An assessment of risks to security would qualify each threat with an assessment of the likelihood of such a threat developing into an actual attack, the likelihood of such an attack proving successful, and the consequences of any damage that may result.

Functional Requirement

- The functional requirements are levied on those functions of the TOE that are specifically in support of IT security, and define the desired security behavior.

  - Part 2 defines the CC functional requirements. Examples of functional requirements include requirements for identification, authentication, security audit and non-repudiation of origin.

Security Functional Components

- Class FAU: Security audit
- Class FCO: Communication
- Class FCS: Cryptographic support
- Class FDP: User data protection
- Class FIA: Identification and authentication
- Class FMT: Security management
- Class FPR: User data protection
- Class FPT: Protection of the TSF
- Class FRU: Resource utilisation
- Class FTP: TOE access
- Class FTP: Trusted path/channels

Security Functional Components

- Content and presentation of the functional requirements
Decomposition of FDP

- **FDP**: User Data Protection
  - FDP: Information Flow Control
  - FDP: Data Flow Control
  - FDP: Access Control
  - FDP: Authorization
  - FDP: Data Encryption

FDP – Information Flow Control

**FDP_IFC.1** Subset information flow control
Hierarchical to: No other components.

*Dependencies: FDP_IFC.1 Simple security attributes
  - FDP_IFC.1.1 The TSF shall ensure the (assignment: information flow control SFP) on (assignment: list of subjects, information, and operations that cause controlled information to flow to and from controlled subjects covered by the SFP).

**FDP_IFC.2** Complete information flow control
Hierarchical to: FDP_IFC.1 Subset information flow control

*Dependencies: FDP_IFC.1 Simple security attributes
  - FDP_IFC.2.1 The TSF shall ensure the (assignment: information flow control SFP) on (assignment: list of subjects, information, and all operations that cause that information to flow to and from subjects covered by the SFP).

FDP_IFC.2.2 The TSF shall ensure that all operations that cause any information in the TOE to flow to and from any subject in the TOE are covered by an information flow control SFP.

Assurance Requirements

**Assurance Approach**

“The CC philosophy is to provide assurance based upon an evaluation (active investigation) of the IT product that is to be trusted. Evaluation has been the traditional means of providing assurance and is the basis for prior evaluation criteria documents.”

Assurance Components

- **Class APE**: Protection Profile evaluation
- **Class ASE**: Security Target evaluation
- **Class ADV**: Development
- **Class AGD**: Guidance documents
- **Class ALC**: Life-cycle support
- **Class ATE**: Tests
- **Class AVA**: Vulnerability assessment
- **Class ACO**: Composition

Assurance Components: Example

**ADV_FSP.1** Basic functional specification

- EAL-1: ... The functional specification shall describe the purpose and method of use for each SFR supporting SFP.
- EAL-2: ... The functional specification shall completely represent the TSF.
- EAL-3: ... The functional specification shall summarize the SFR-supporting and SFR-non-interfering actions associated with each TSFI.
- EAL-4: ... The functional specification shall describe all direct error messages that may result from an invocation of each TSFI.
- EAL-5: ... The functional specification shall describe the TSFI using a semi-formal style.
- EAL-6: ... The developer shall provide a formal presentation of the functional specification of the TSF. The formal presentation of the functional specifications of the TSF shall be in a formal style, supported by informal, explanatory text where appropriate.


Evaluation Assurance Level

- EALs define levels of assurance (no guarantees)
  1. functionally tested
  2. structurally tested
  3. methodically tested and checked
  4. methodically designed, tested, and reviewed
  5. semiformally designed and tested
  6. semiformally verified design and tested
  7. formally verified design and tested

Assurance Requirements

- EAL5 – EAL7 require formal methods.
  - according to CC Glossary:
    - **Formal**: Expressed in a restricted syntax language with defined semantics based on well-established mathematical concepts.
Security Functions

- The **statement of TOE security functions** shall cover the IT security functions and shall specify how these functions satisfy the TOE security functional requirements. This statement shall include a bi-directional mapping between functions and requirements that clearly shows which functions satisfy which requirements and that all requirements are met.

- Starting point for **design process**.

Summary

- Norms and standards enforce the application of the state-of-the-art when developing software which is
  - safety-critical or security-critical.
  - Wanton disregard of these norms may lead to personal liability.
  - Norms typically place a lot of emphasis on process.
  - Key question are traceability of decisions and design, and verification and validation.
  - Different application fields have different norms:
    - IEC 61508 and its specialisations, DO-178B.