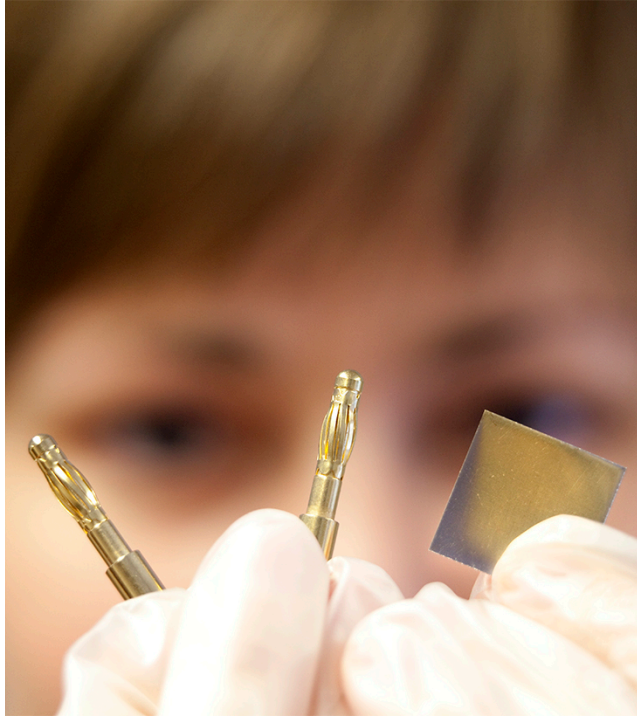
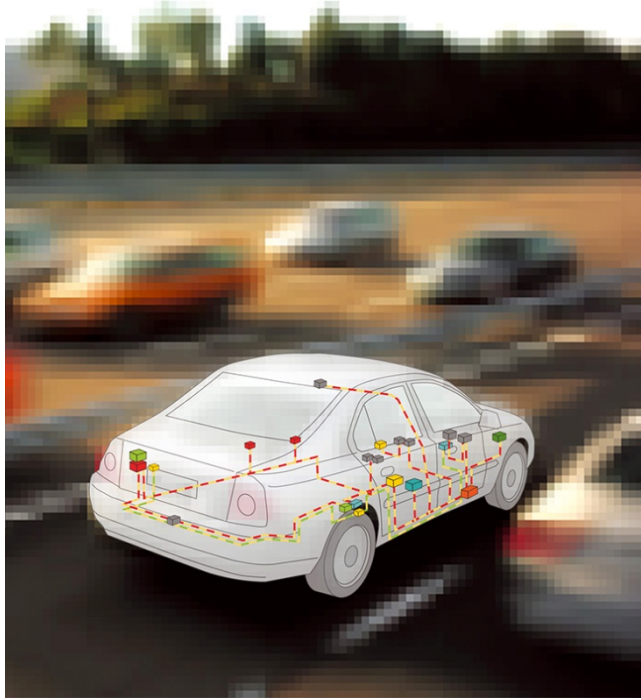
ISO/DIS 13849-1 SW + VALIDATION

Andreas Söderberg

Research Institutes of Sweden

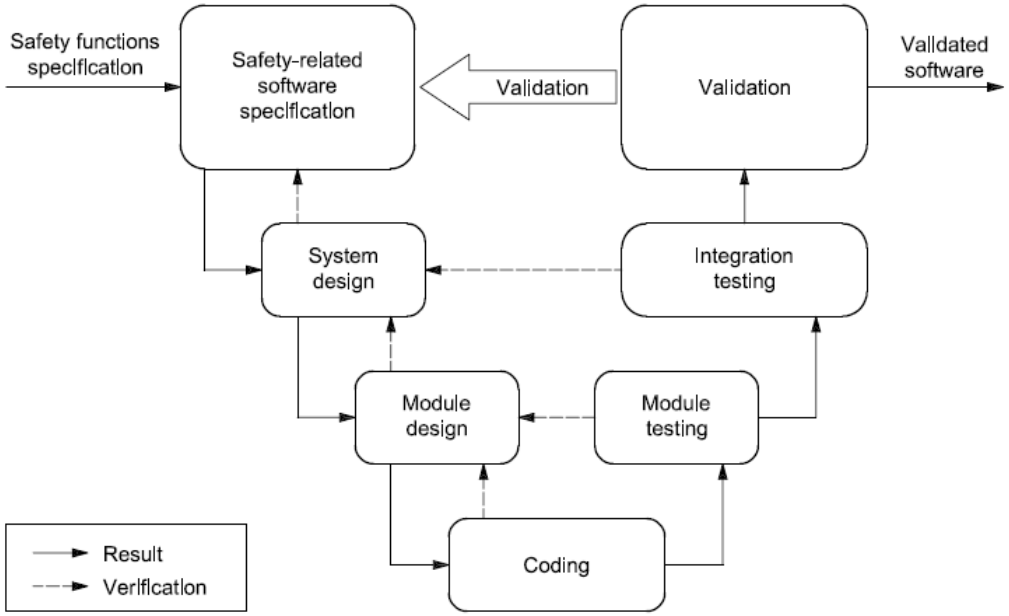
SÄKRA STYRSYSTEM
ELEKTRIFIERING & PÅLITLIGHET

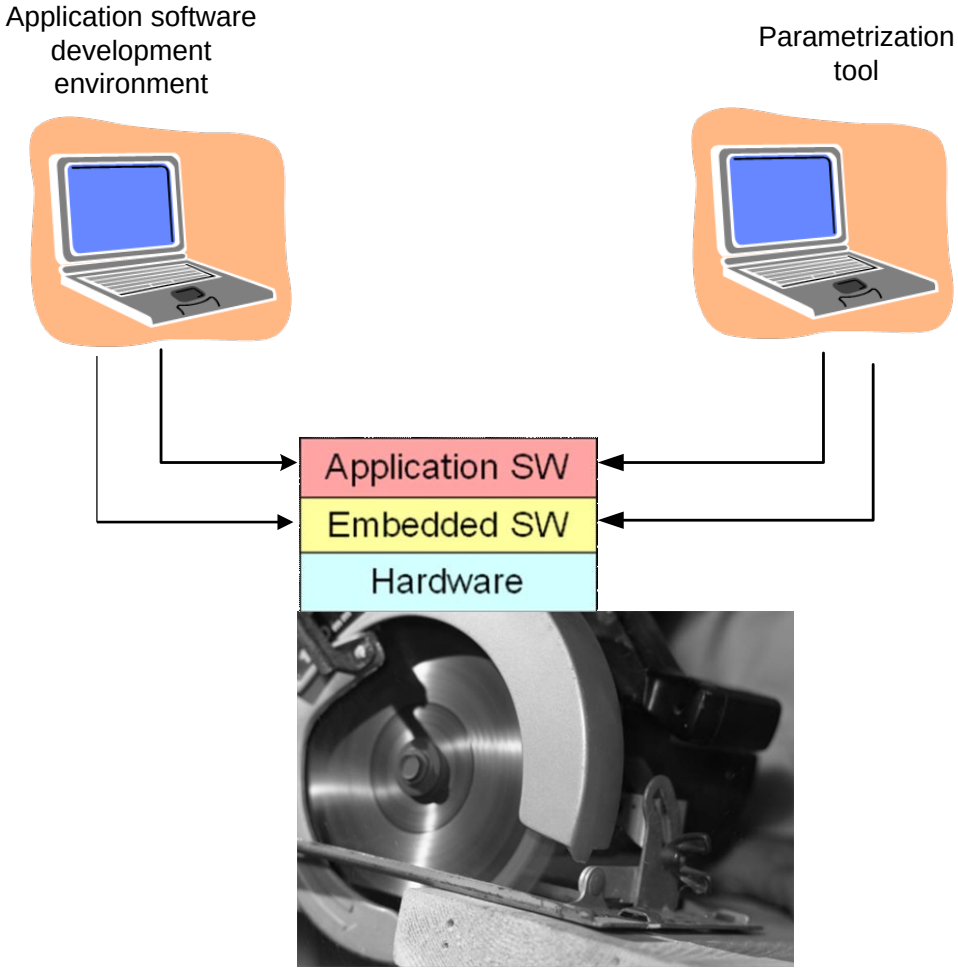




Programvara i styrsystem

Utveckling av programvara





Programvarubaserad parametrisering

ISO 13849 medger att parametrisering utförs med:

- Verktyg = SRP/CS (t ex robotpedant), eller
- Generellt verktyg (t ex persondator)

ISO/DIS 13849-1
Terminology / Software:

limited variability language (LVL)

type of language that provides the capability of combining predefined, application-specific library functions and basic logic to implement the safety requirements specifications

Clause 3.1.38 in the standard

- Typical examples of LVL are **ladder logic** and **function block diagrams** which are often used for programming **PLC**s.

ISO/DIS 13849-1
Terminology / Software:

full variability language (FVL)

type of language that provides the capability of implementing a wide variety of functions and applications

Clause 3.1.39 in the standard

- Typical examples of FVL are **C, C++** and **assembler**. In the field of machinery, FVL are typically used for programming embedded computer systems.

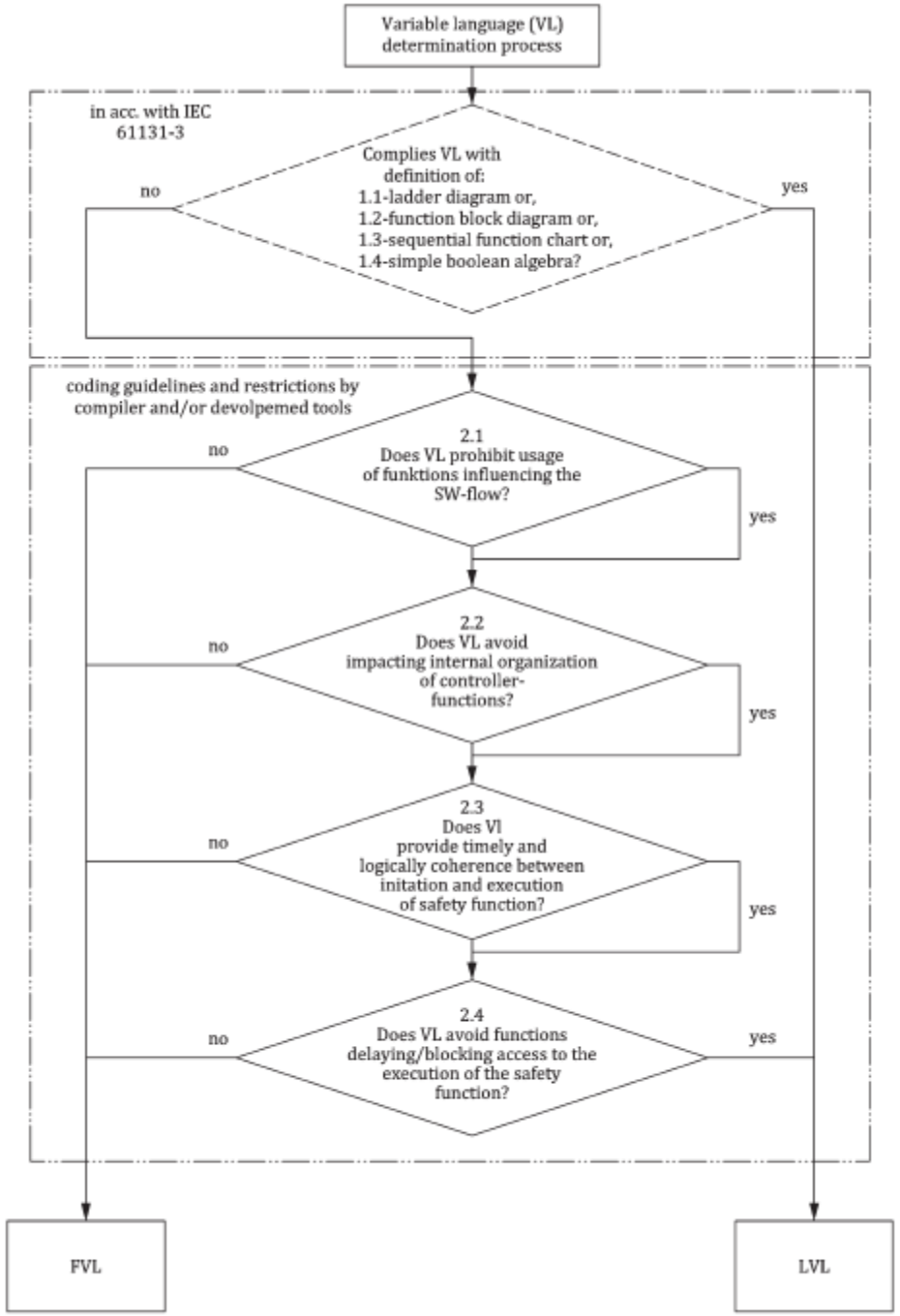


Figure 16 — decision guideline for FVL or LVL

ISO/DIS 13849-1
Terminology / Software:

embedded software / firmware / system software

software that is part of the system supplied by the manufacturer and is not accessible for modification by the user of the machinery

Clause 3.1.41 in the standard

- For the embedded software, the abbreviation **SRESW** (safety-related embedded software) is used.
- SRESW is usually written in **FVL**.

ISO/DIS 13849-1
Terminology / Software:

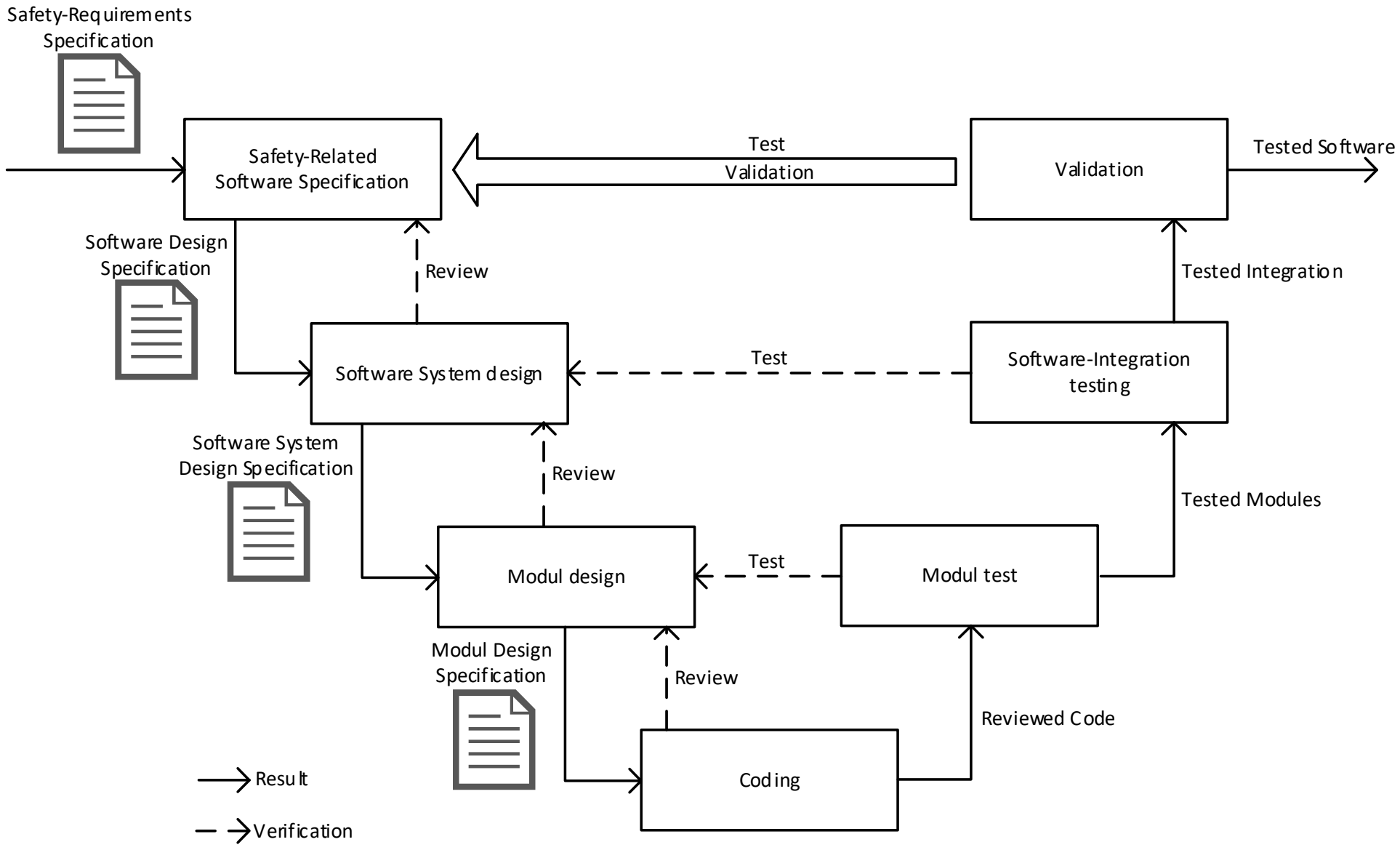
application software

software specific to the application and generally containing logic sequences, limits and expressions that control the appropriate inputs, outputs, calculations and decisions necessary to meet the SRP/CS requirements

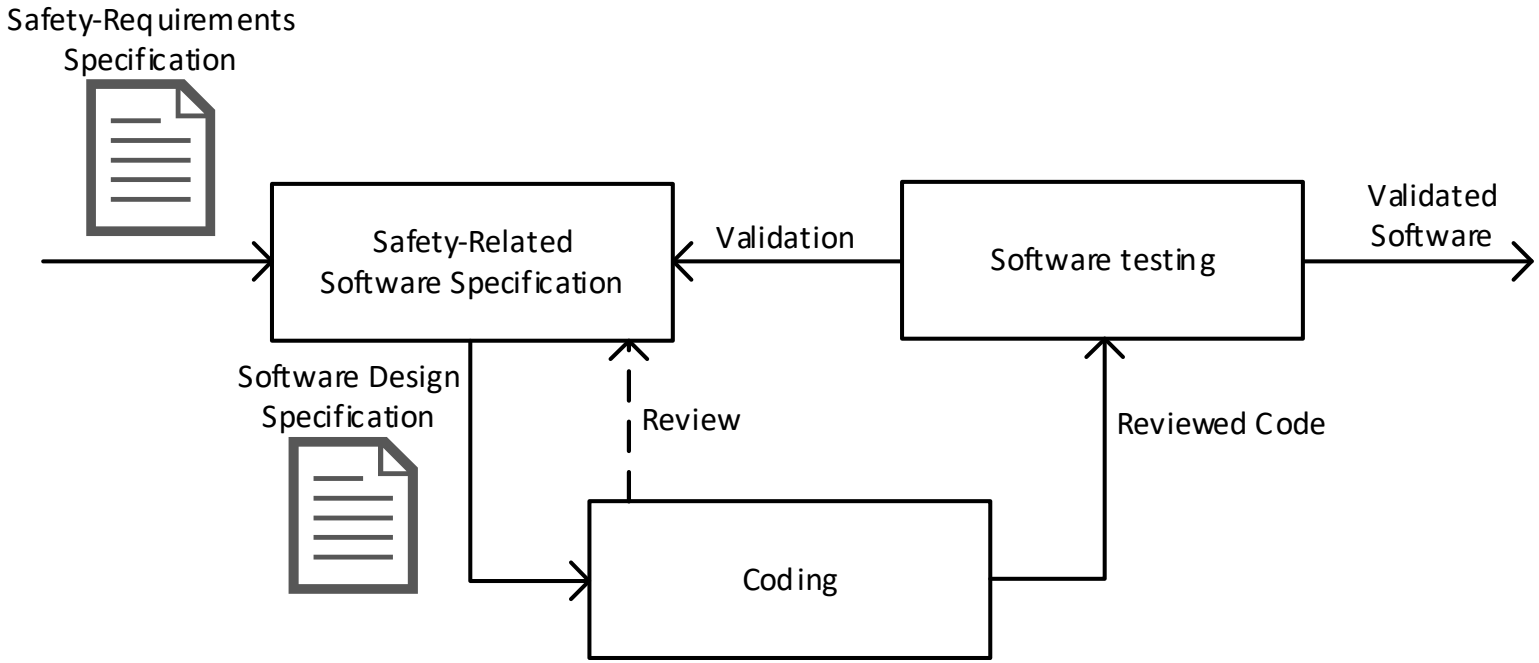
Clause 3.1.40 in the standard

- For the application software, the abbreviation **SRASW** (safety-related application software) is used.
- SRASW is usually written in **LVL** but may also be written in **FVL**.

Software – V-model (ISO/DIS 13849-1, Figure 14)



Software – V-model (ISO/DIS 13849-1, Figure 14-a)



- V-model for software if **pre-assessed** safety related hardware and software modules are used in combination with LVL
- E.g. Safety PLC programmed with Function block diagrams

SRESW

- Safety Related Embedded Software

For **SRESW** for components **with PLr a to d**, the following basic measures shall be adapted:

a) software safety lifecycle with verification and validation activities, see Figure 14;

b) documentation of specification and design, e.g. software design specification, software system design specification, module design specification, code listings including comments;

c) modular and structured design and coding, e.g. hierarchy and limitation of functionality, clear program structure, definition of interfaces, well-structured call-graph, avoidance of interrupts, use of coding guidelines;

d) control of systematic failures, e.g. program sequence monitoring, controlling errors in the data communication process (see G.2);

e) where using software-based measures for control of random hardware failures, verification of correct implementation, e.g. correct implementation of diagnostic measures, RAM/ROM/CPU tests, hardware tests, plausibility checks;

f) functional testing, e.g. black box testing, e.g. verification of correct output data based on input data (valid, invalid and border values), compatibility of interfaces, timing;

g) appropriate software safety lifecycle activities after modifications, e.g. based on an impact analysis.

For **SRESW** for components with **PLr c or d**, the following additional measures shall be applied:

h) project management and quality management system comparable to, e.g. IEC 61508, e.g. definition of workflow, responsibilities, configuration management, use of tools;

i) documentation of all relevant activities during software safety lifecycle, e.g. documentation of reviews, testing, validation and verification;

j) configuration management to identify all configuration items and documents related to a SRESW release, e.g. version control of code listings, modules, design documents, test plans, release control, archiving;

k) structured specification with safety requirements and design;

l) use of suitable programming languages and computer-based tools with confidence from use, e.g. programmers are trained to use the tools;

m) modular and structured programming, separation from non-safety-related software, limited module sizes with fully defined interfaces, use of design and coding standards;

n) coding verification by walk-through/review with control flow analysis (to check for faults, quality of comments, compliance with coding guidelines, clarity, readability, completeness);

o) extended functional testing, e.g. grey box testing, performance testing or simulation, e.g. using input unspecified data, extreme environmental conditions, full load, testing based on knowledge of internal coding;

p) impact analysis and appropriate software safety lifecycle activities after modifications.

SRESW for components with **PL_r = e** shall comply with IEC 61508-3:1998, Clause 7, appropriate for SIL 3.

When using diversity in specification, design and coding, for the two channels used in a subsystem with category 3 or 4, **PL_r = e** can be achieved with the above-mentioned measures for PL_r of c or d.

For components for which SRESW requirements are not fulfilled, e.g. PLCs without safety rating by the manufacturer, these components may be used under the following alternative conditions:

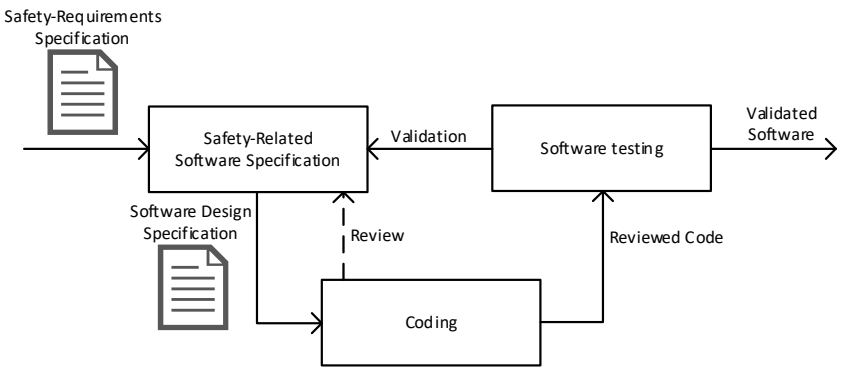
— the subsystem is limited to PL a or b and uses category B, 2 or 3;

— the subsystem is limited to PL c with category 2 or PL d with category 3 and it is necessary to fulfil the diversity requirements of the CCF, which means that both channels must have own non-safe PLC's, which both use diverse technologies compared to each other.

SRASW

- Safety Related Application Software

General requirements for SRASW, PL **a** to **e**



Requirements for PL **a** to **e**:

- SW safety lifecycle (Figure 14)
- If written in FVL, SRESW shall apply
- For components with different PL, the highest PL shall apply
- Documentation of lifecycle steps
- Modularity
- Functional testing
- Impact analysis

Additional requirements for SRASW, PL **c** to **e**

Review of specification shall be performed and made available for every person involved in the lifecycle and shall contain:

- Safety functions
- Performance criteria
- HW architecture
- Signal interfaces
- Detection and control of external failure.

Requirements for SRASW: Tools, libraries and languages

SRASW requirements for PL **c** to **e** :

- Tool confidence from use
- Tools that have features for systematic error detection
- Tools that enforce language subsets
- Tools that enforce coding guidelines
- Use subset of IEC 61131-3. Graphical languages are highly recommended (FBD, Ladder)

Additional SRASW requirements for **e**:

- **Either** use one tool that complies with a safety standard, (if one component)
- **Or** use two diverse tools with confidence from use, (if two components)
- Use validated function block (FB) libraries

Requirements for SRASW: SW design

SRASW design requirements for PL **c** to **e** :

- Use semi-formal methods
- Modularity
- Limited size of function blocks
- One entry, and one exit point of functions
- Architecture of model Input-Processing-Output
- Detection of external failure
- Defensive programming
- Safe state

Requirements for SRASW: Testing

If both safety-related and non safety related SW:

- Clearly separated in different function blocks
- No combination of safety-related and non-safety-related data

Code:

- Readable, understandable and testable
- Coding guidelines
- Data integrity and plausibility checks
- Tested by simulation

Additional for PL = **d** or **e**:

- Code shall be verified by control and data flow analysis

Requirements for SRASW: Testing

General requirements:

- Black-box testing as validation method of functinality
- I/O testing of safety related signals
- Test planning, test cases and completion criteria

Additional requirements for PL = **d** or **e**:

- Boudary value analysis

Requirements for SRASW: Documentation

General documentation:

- All lifecycle phases documented
- Complete, available, readable and understandable

Source code documentation:

- Module headers with legal entity
- Functional and I/O description
- Version and version of library functions
- Comments to lines inlcluding networks/statements and declarations

Requirements for SRASW: Modifications and CM

Configuration Management:

- Archive: documents, SW-modules, V&V results and tool configurations
- Data backup
- Establish procedures

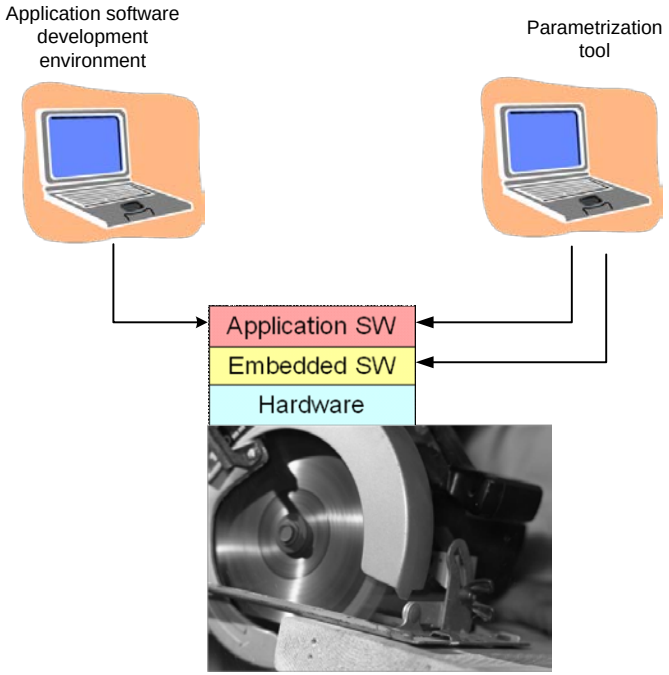
Modifications:

- Impact analysis
- Go back to appropriate lifecycle phase
- Persons involved. Decision procedures
- Document modification history

ISO/DIS 13849-1, 7.5 Software-based parameterization 1(4)

- Software-based parameterization of safety-related parameters shall be considered as a **safety-related aspect** of SRP/CS design to be described in the software safety requirements specification.

- Parameterization shall be carried out using a **dedicated software tool provided by the supplier** of the SRP/CS. This tool shall have its own identification (name, version, etc.) and shall prevent unauthorized modification, for example, by use of a password.



ISO/DIS 13849-1, 7.5 Software-based parameterization 2(4)

- The **integrity** of all data used for parameterization **shall be maintained**. This shall be achieved by applying measures to
  - control the **range** of valid inputs
  - control data **corruption** before transmission
  - control the effects of **errors** from the parameter **transmission** process
  - control the effects of **incomplete** parameter **transmission**
  - control the effects of **faults** and failures of hardware and software of the **tool** used for parameterization

ISO/DIS 13849-1, 7.5 Software-based parameterization 3(4)

- The parameterization tool shall **fulfil all requirements** for SRP/CS according to this part of ISO 13849.

- **Alternatively**, a special procedure shall be used for setting the safety-related parameters. This procedure shall include confirmation of input parameters to the SRP/CS by either
  - **retransmission** of the modified parameters to the parameterization tool
  - other suitable means of confirming the integrity of the parameters as well as subsequent confirmation, e.g. by a **suitably skilled person** and by means of an **automatic check** by a parameterization tool.

NOTE 1 This is of particular importance where parameterization is carried out using a device not specifically intended for the purpose (e.g. **personal computer** or equivalent).

ISO/DIS 13849-1, 7.5 Software-based parameterization 4(4)

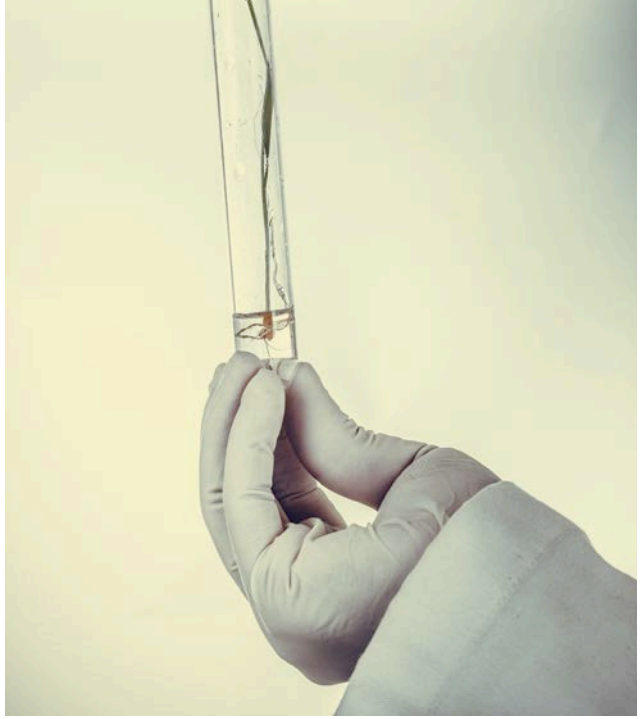
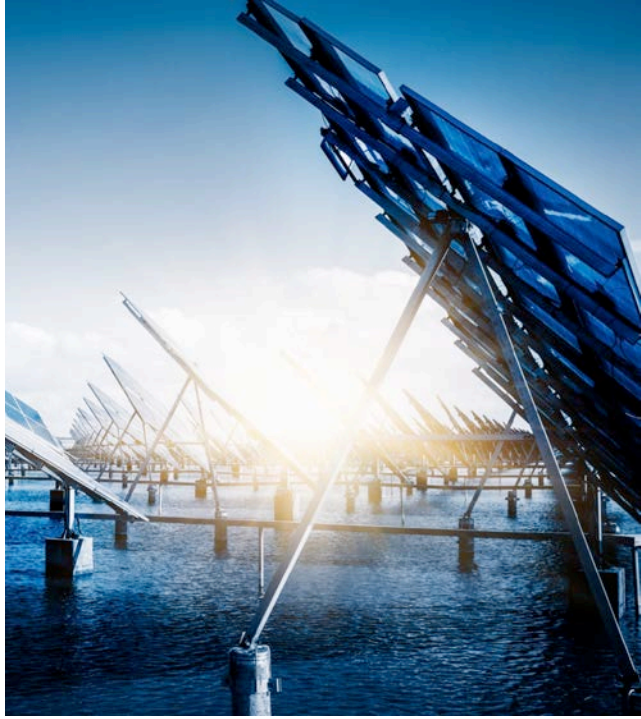
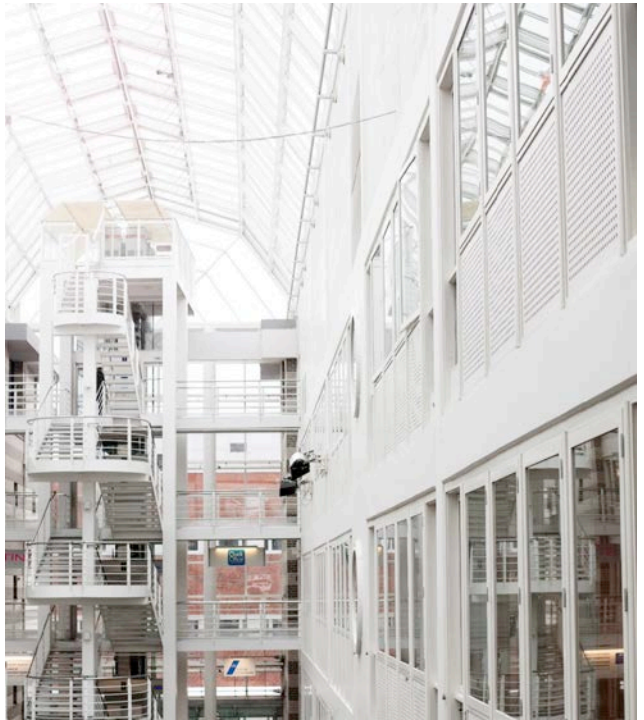
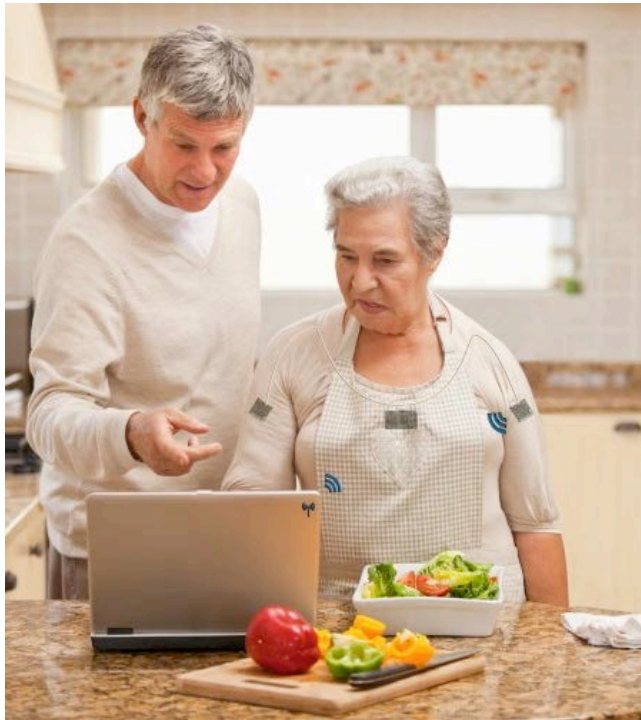
- The software modules used for encoding / decoding within the transmission / retransmission process and software modules used for visualization of the safety-related parameters to the user shall, as a minimum, use **diversity in function**(s) to avoid systematic failures.

- **Documentation** of software-based parameterization shall indicate **data used** and **information** necessary **to identify** the parameters associated with the SRP/CS, the **person**(s) carrying out the parameterization together with other relevant information such as **date** of parameterization.

- The following **verification activities** shall be applied for software based parameterization:
  - verification of the **correct setting** for each safety-related parameter (minimum, maximum and representative values)
  - verification that the safety-related parameters are **checked for plausibility**, for example by use of invalid values, etc.
  - verification that **unauthorized modification** of safety-related parameters is prevented
  - verification that the data/signals for parameterization are generated and processed in such a way that faults can not lead to a **loss of the safety function**

    NOTE 2 This is of particular importance where the parameterization is carried out using a device not specifically intended for this purpose (e.g. **personal computer** or equivalent).

ISO 13849-1 & 2
Introduktion till test & validering enligt standarder

Research Institutes of Sweden

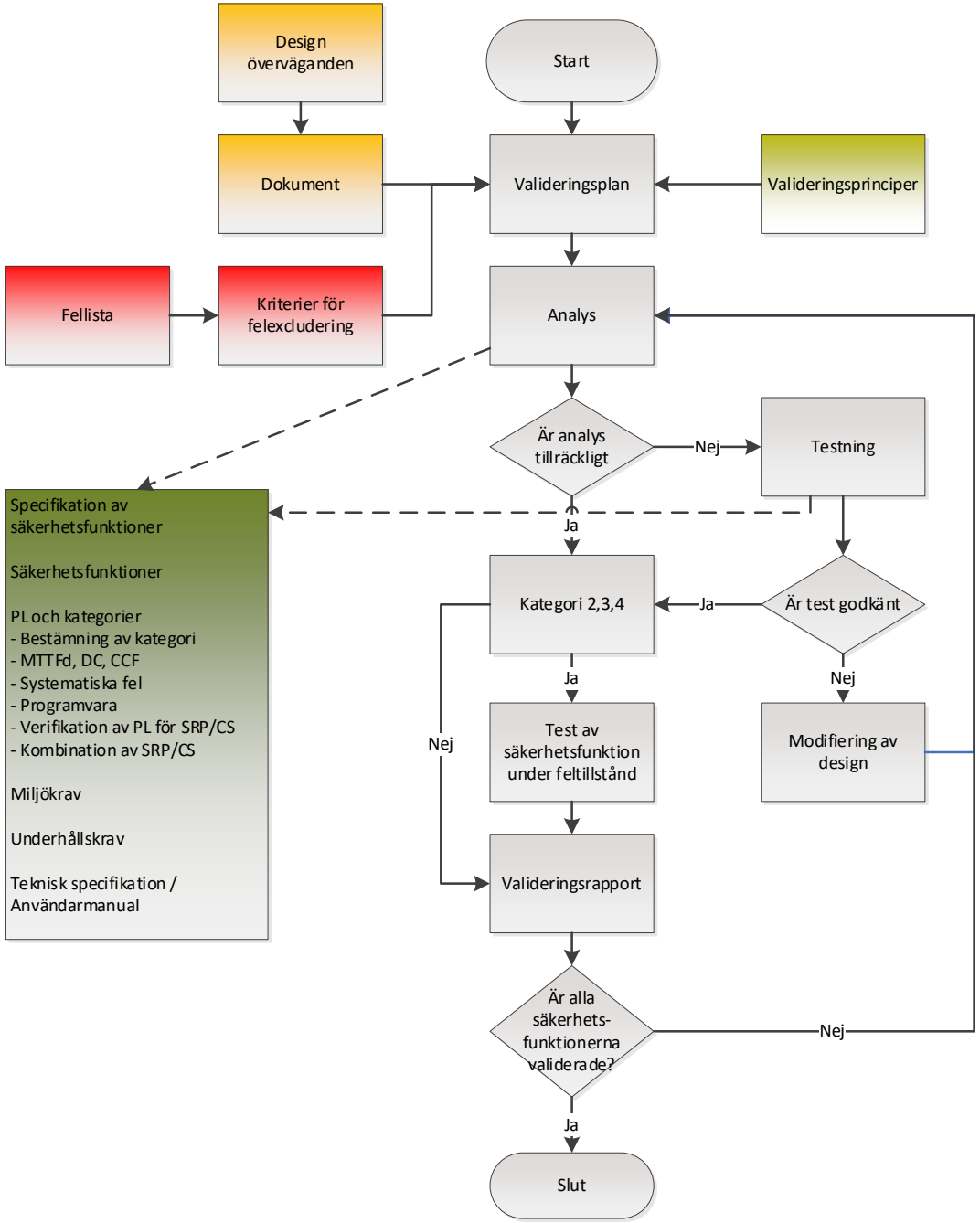
SÄKRA STYRSYSTEM
ELEKTRIFIERING & PÅLITLIGHET

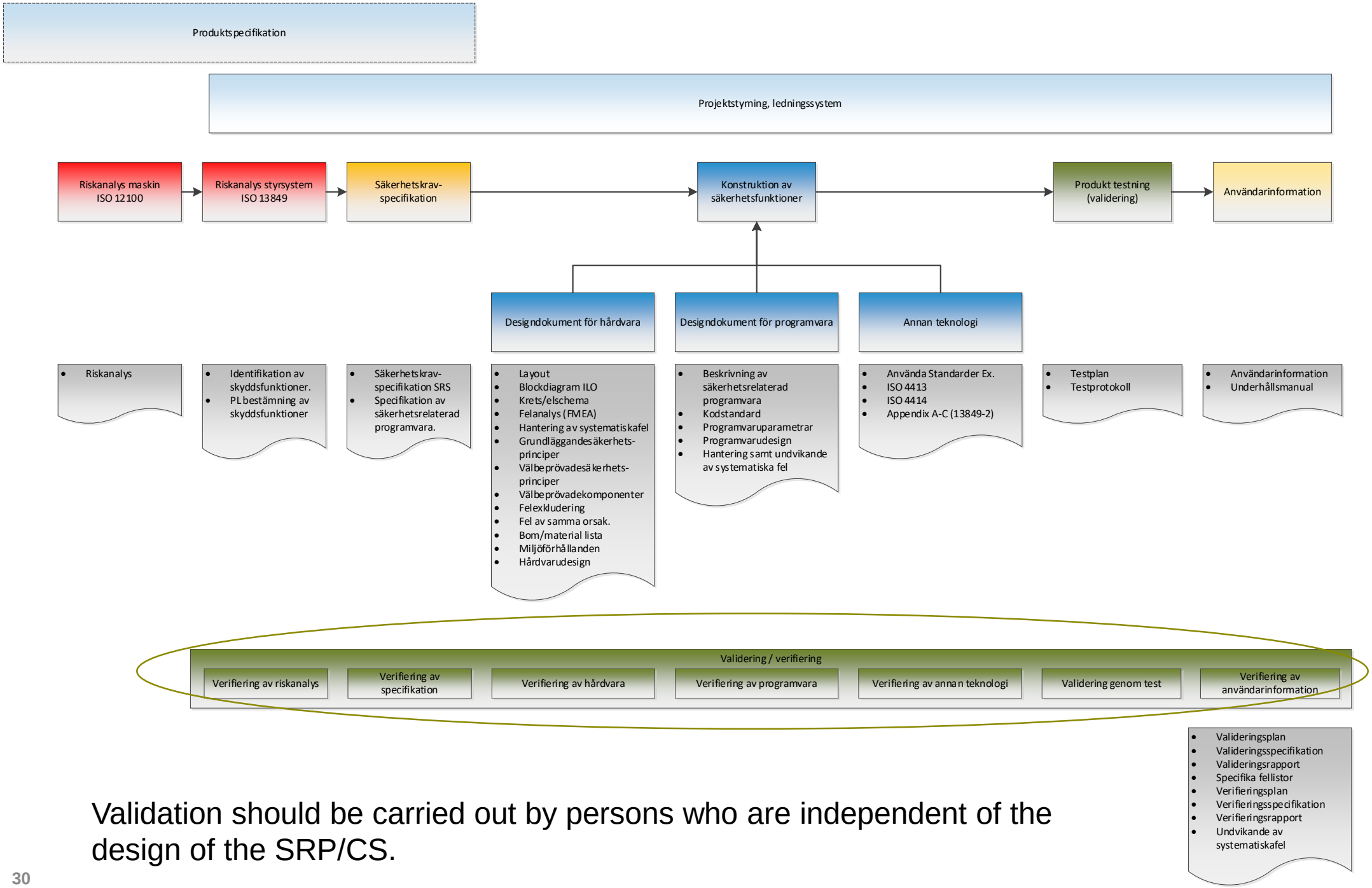
Validering av säkerhetsfunktioner

3.1.54
validation
confirmation, through the provision of objective evidence, that the requirements for a specific intended use or application have been fulfilled by the SRP/CS

Note 1 to entry: The objective evidence needed for a validation is the result of a test or other form of determination such as performing alternative calculations or reviewing documents.

I.e.: The purpose of the validation process is to confirm that the SRP/CS meets the overall safety requirements specification


Design
överväganden
Start
Dokument
Valideringsplan
Valideringsprinciper
Fellista
Kriterier för
felexcludering
Analys
Är analys
tillräckligt
Nej
Testning
Specifikation av
säkerhetsfunktioner
Säkerhetsfunktioner
PL och kategorier
- Bestämning av kategori
- MTTFd, DC, CCF
- Systematiska fel
- Programvara
- Verifikation av PL för SRP/CS
- Kombination av SRP/CS
Miljökrav
Underhållskrav
Teknisk specifikation /
Användarmanual
Ja
Kategori 2,3,4
Ja
Är test godkänt
Ja
Nej
Nej
Test av
säkerhetsfunktion
under feltillstånd
Modifiering av
design
Valideringsrapport
Är alla
säkerhets-
funktionerna
validerade?
Nej
Ja
Slut



Validation should be carried out by persons who are independent of the design of the SRP/CS.

Validation tools

- Validation is carried out by a combination of:
  - Analysis, and/or
  - Testing
- Provides guidance for the validation of:
  - The SRS (..*shall be verified to ensure consistency and completeness for its intended use.*)
  - The safety functions (...*functional testing w.r.t. their specified characteristics*)
  - Perfomance levels and categories (...*fault injection testing*)
  - Measures against systematic failures (..*inspection of documentation and FMEA*)
  - Safety related software (..*reviews and black-box testing, etc.*)
  - Combination of SRP/CS, Maintenance, Information for use, Etc.
- Checking/verification of safety integrity

Dokumentation (ISO/DIS 13849-1)

Table 2 — Documentation requirements for categories in respect of performance levels

Documentation requirement	Category for which documentation is required				
	B	1	2	3	4
Basic safety principles	X	X	X	X	X
Expected operating stresses	X	X	X	X	X
Influences of processed material	X	X	X	X	X
Performance during other relevant external influences	X	X	X	X	X
Well-tried components	—	X	—	—	—
Well-tried safety principles	—	X	X	X	X

Dokumentation (ISO/DIS 13849-1)

Table 2 *(continued)*

Documentation requirement	Category for which documentation is required				
	B	1	2	3	4
Mean time to dangerous failure ($MTTF_d$) of each channel	X	X	X	X	X
The check procedure of the safety function(s)	—	—	X	—	—
Diagnostic measures performed, including fault reaction	—	—	X	X	X
Checking intervals, when specified	—	—	X	X	X
Diagnostic coverage (DC_{avg})	—	—	X	X	X
Foreseeable single faults considered in the design and the detection method used	—	—	X	X	X
Common-cause failures (CCF) identified and how to prevent them	—	—	X	X	X
Foreseeable single faults excluded	—	—	—	X	X
Faults to be detected	—	—	X	X	X
How the safety function is maintained in the case of each of the faults	—	—	—	X	X
How the safety function is maintained for each of the combinations of faults	—	—	—	—	X
Measures against systematic faults	X	X	X	X	X
Measures against software faults	X	—	X	X	X

X documentation required

— documentation not required

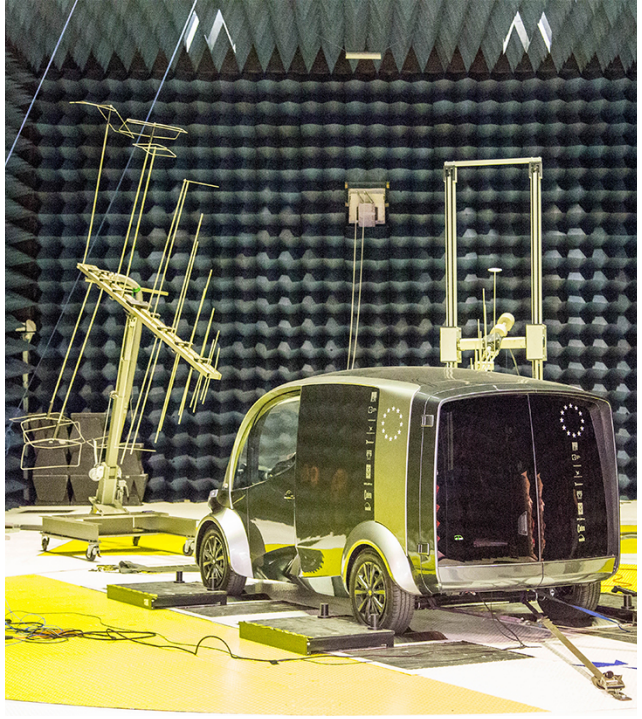
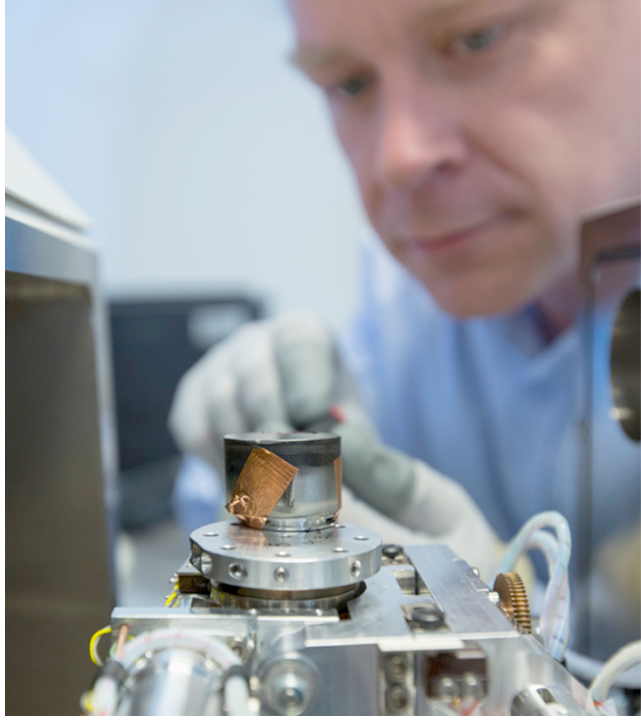
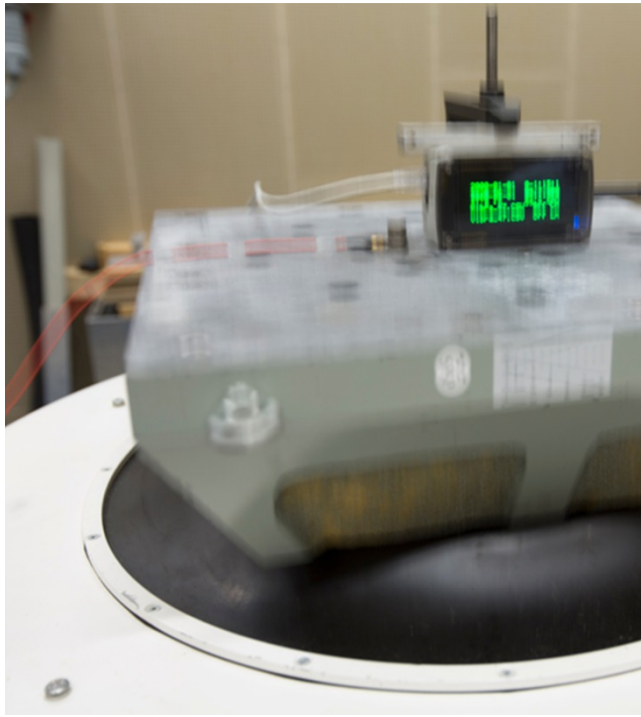
NOTE The categories are those given in ISO 13849-1:2006.

Validering EN ISO 13849-2

- Annex A-D Valideringsverktyg för system
- Grundläggande säkerhetsprinciper
- Väl beprövade säkerhetsprinciper
- Väl beprövade komponenter
- Felsätt och möjliga felexkluderingar för olika komponenter

TACK!

Andreas Söderberg

Research Institutes of Sweden

SÄKRA STYRSYSTEM
ELEKTRIFIERING & PÅLITLIGHET