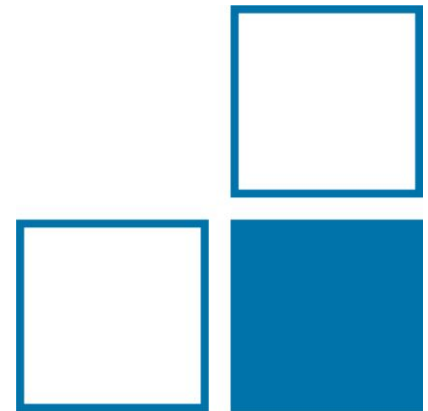


Software Quality Assurance for Metrology

Daniel Peters, Norbert Greif



Consequences of
incorrect software

Butler Group (2007): In
German industry, incorrect
software applications cause a
damage of €4.7 billion p.a.

Declining user
confidence

(incorrect)
Software

■ Specific features of software

- immaterial, digital product
- faulty
- distributed
- fast moving
- easily changeable
- quickly transferable
- minimal changes cause large-scale consequences (side effects)
- missing transparency

Severe software bugs related to measurements

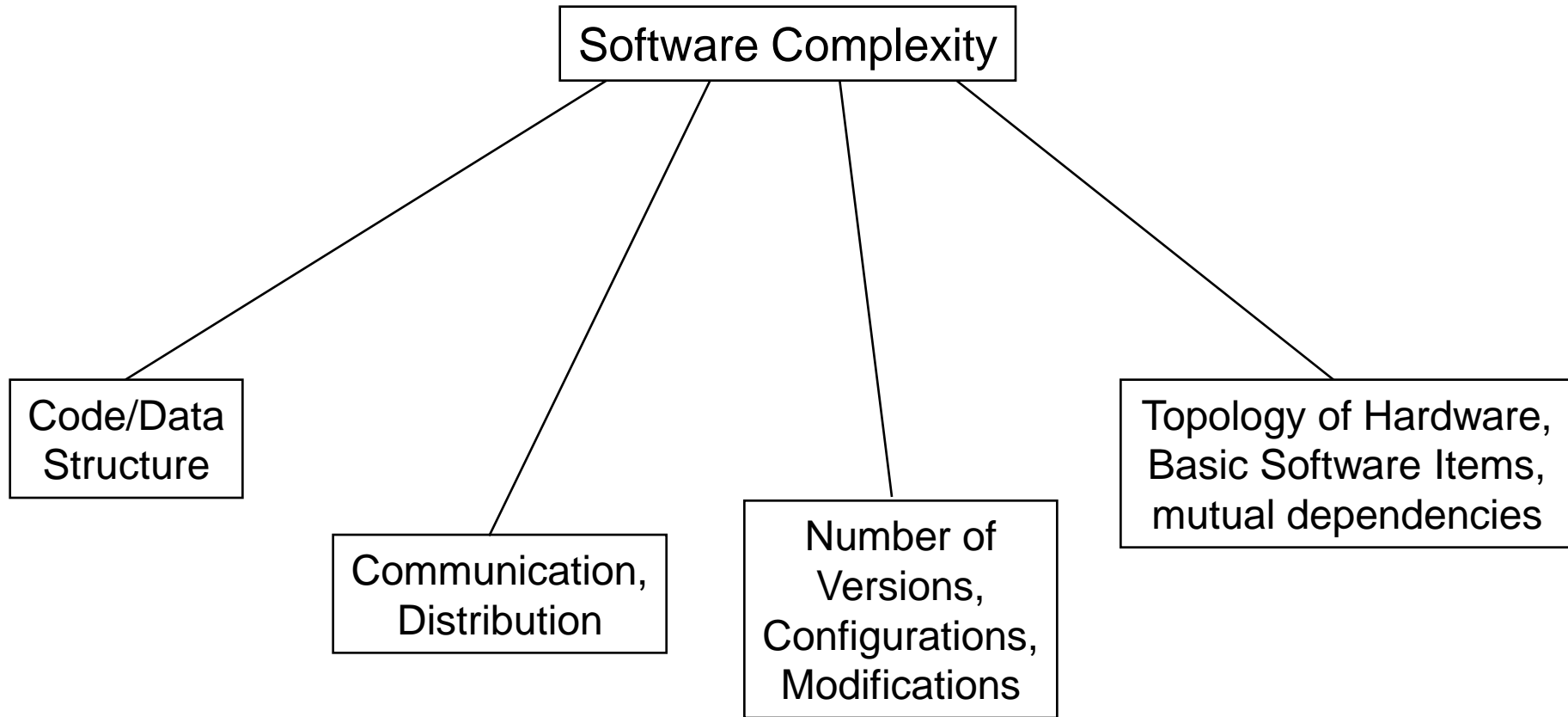


- Complete loss of Mars Climate Orbiter at NASA Mission 2000
- Damage: 125 Million US Dollar for Orbiter, ...
- Reason: Mix-up between metric und English measures

- Thunderstorm „Lothar“: More than 60 dead person
- 25.12.99: Forecasting predicted a thunderstorm, error in interpretation: measurement error
- 26.12.99: Weather prediction failed because of ignored outlier data, wind speed 90 km/h instead of 215 km/h



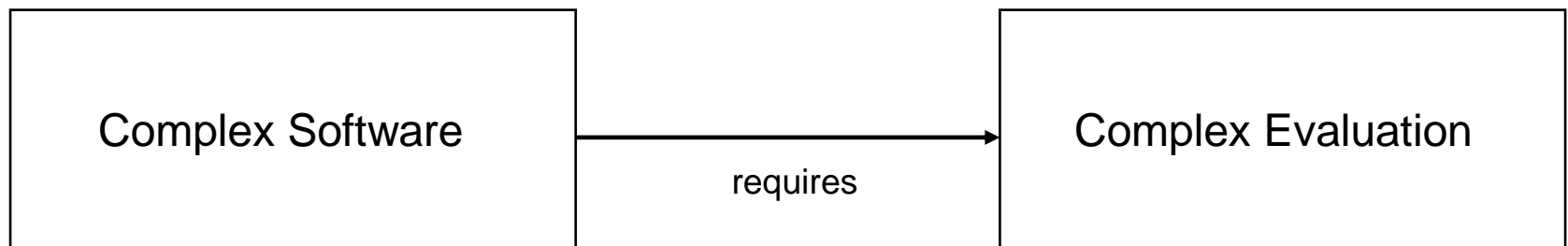
- Organisation and management:
 - Insufficient sensitivity to software problems on behalf of the management,
 - no support for the software development departments in companies,
 - no strategic plans
 - Process models are not used for the development of a software



- Increasing complexity of software in embedded systems

BUT

- There is no standardised software architecture
- There are no standardised software development methods
- There are no standardised software assessment methods



- 85% of functionality of automobiles is implemented and controlled by software.
- 70 - 90 μ -controller with mutual dependencies are integrated in one car.
- Approximately 1 Gigabyte Onboard Software is implemented in one car.

- **Constructive (preventive) Measures:**
 - Specification of technical (and organizational) rules for software development
- **Analytic Measures:**
 - Audits, reviews, evidence of a low-error implementation, control of constructive measures
- **Organizational Measures:**
 - Creation of good surrounding conditions by the management

Levels of Quality Characteristics	analytical	constructive	organizational
physical-metrological	data analysis, physical interpretation	physical modelling, standards, guidelines	quality manuals, audits, reviews
mathematical-algorithmic	reference algorithms, functional tests, statistical data analysis	mathematical modelling, numerical algorithms	algorithm libraries, experimental design
software-oriented	dynamic tests, static analysis, code inspections, data flow analysis	life-cycle models, design methods, programming guidelines, documentation rules	quality systems, configuration and change management, program libraries

Analytical Measures

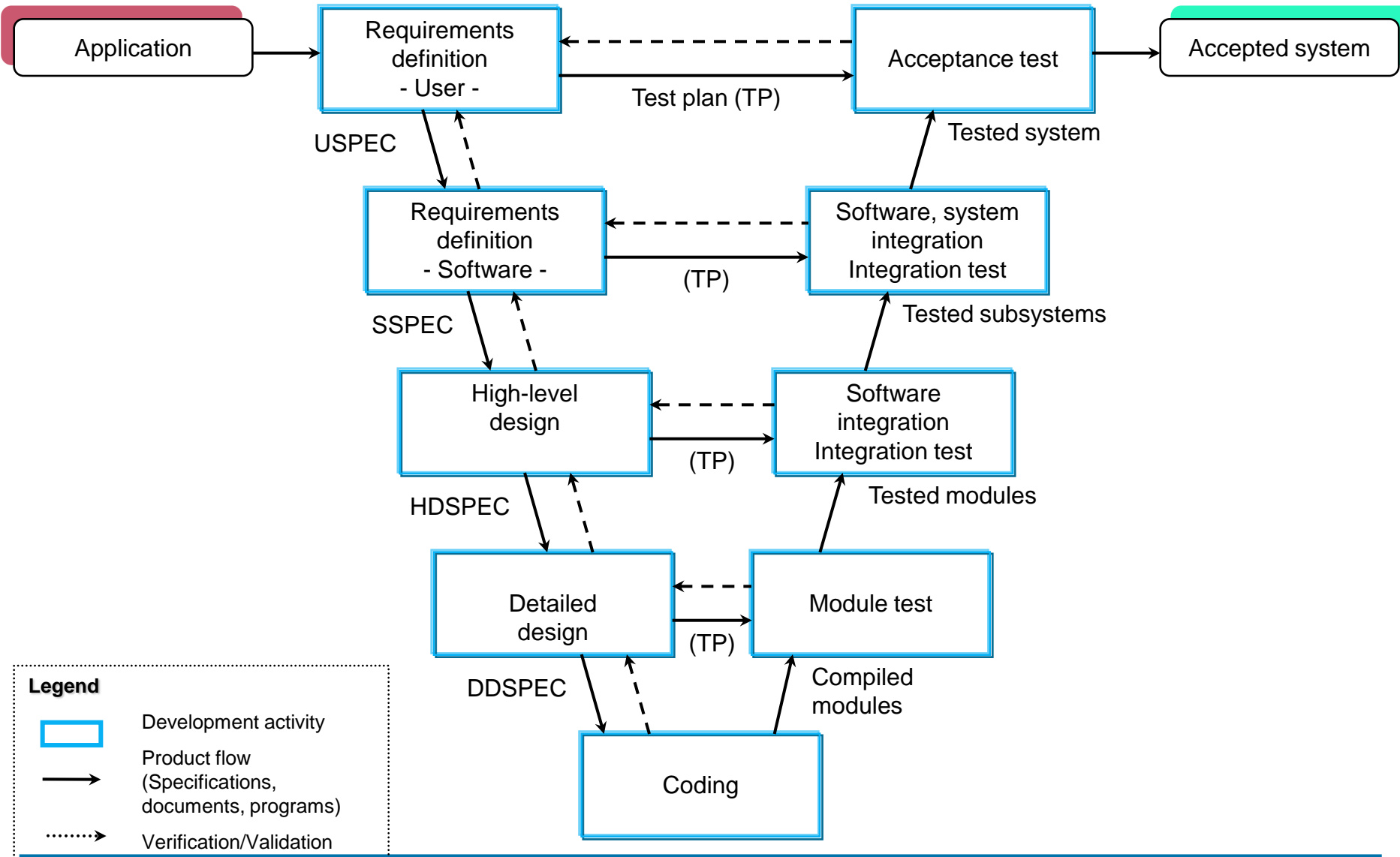
Testing methods, techniques and tools to detect failures/defects, to evaluate software products and processes

- Dynamic testing
- Static analysis
- Inspections
- Reviews
- Usability testing
- Performance testing
- Auditing of processes

Preventive Measures

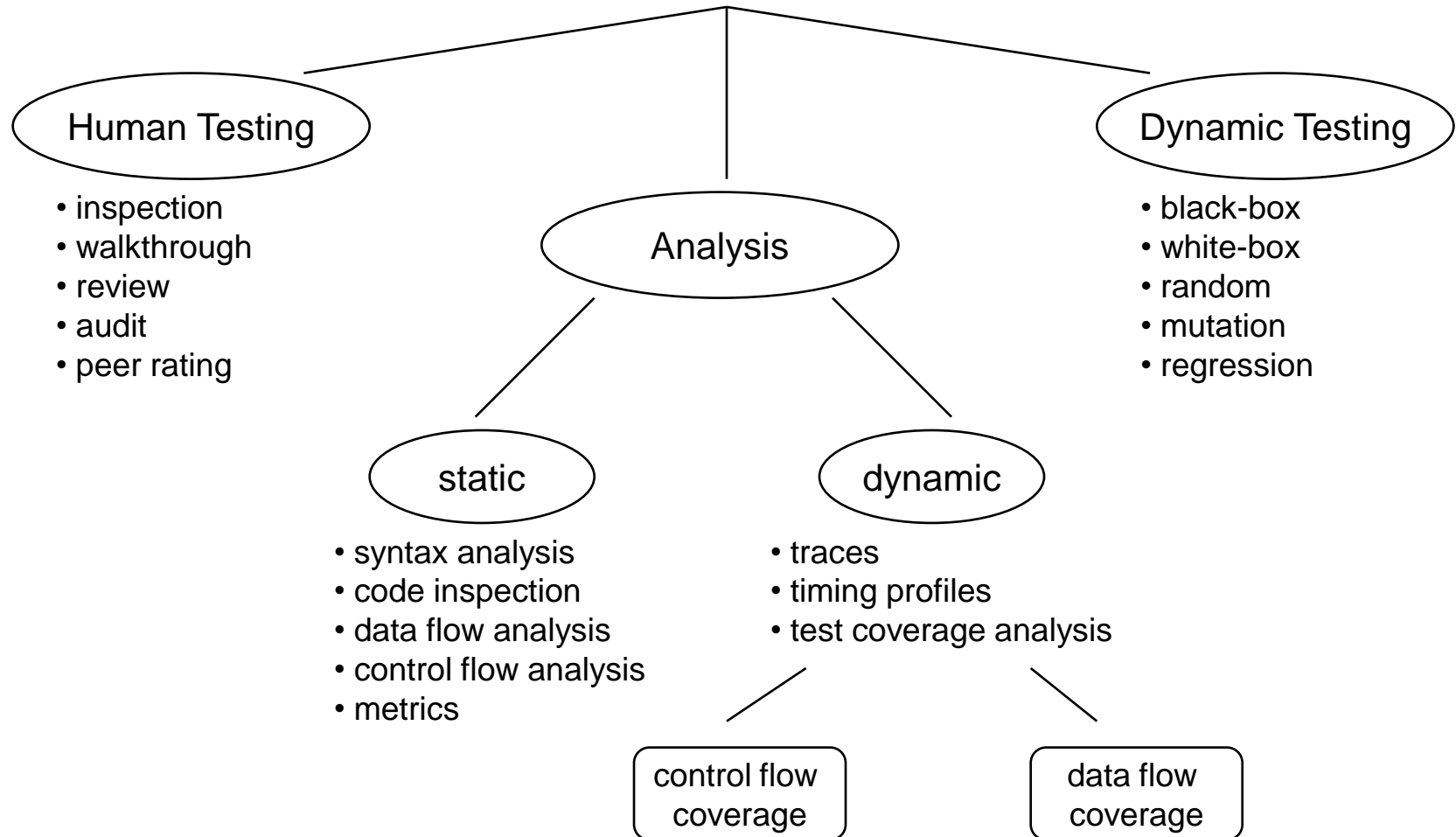
Constructive methods, techniques and tools to avoid failures/defects in software development processes

- Establishing compliance with standards and guidelines (evidence of conformity)
- Systematic application of software life-cycle models/best practices
- Testing/documenting at an early stage
- Elaboration of harmonised documents

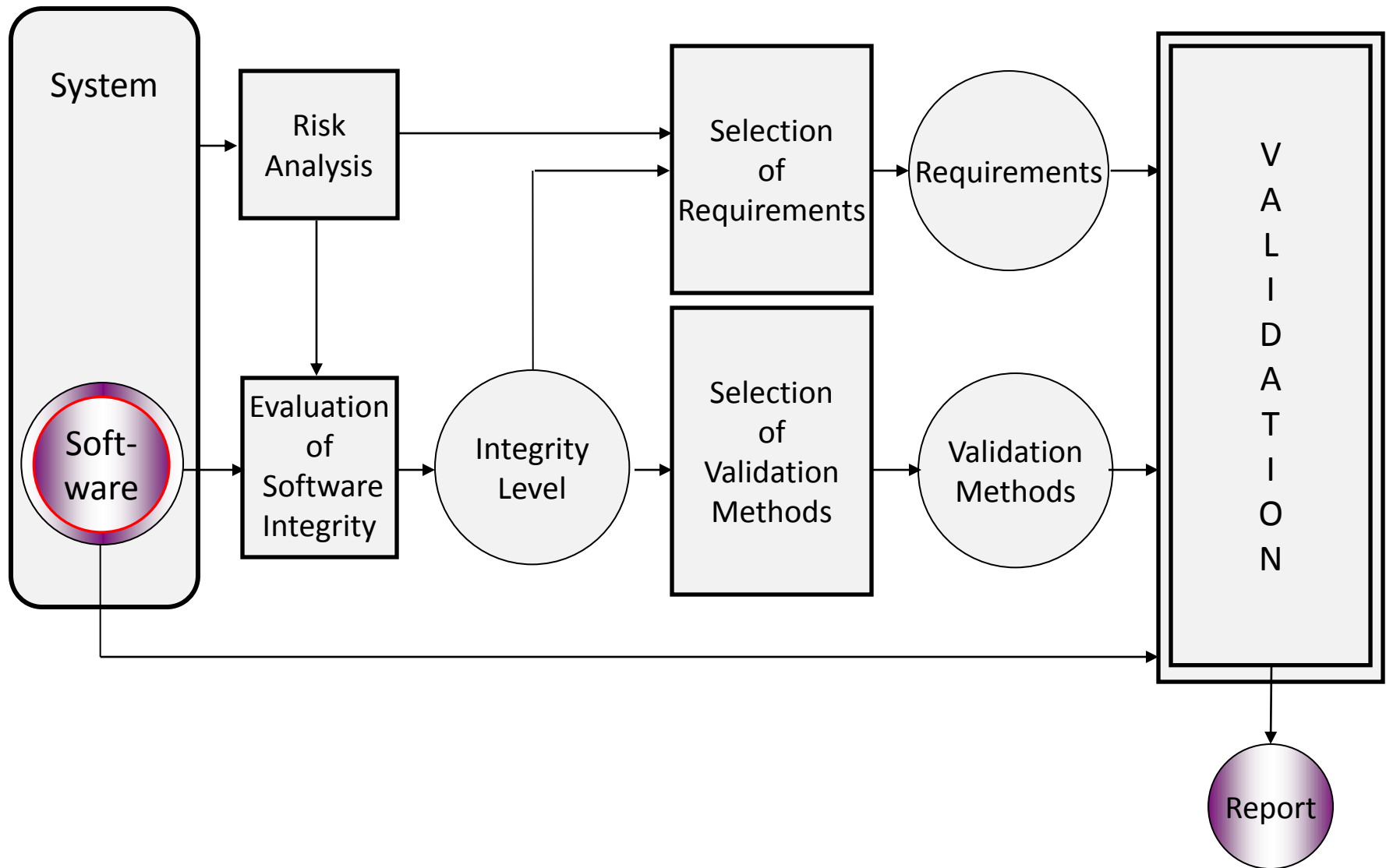


- ISO/IEC 17025, 5.4.5.1
- „Validation is the confirmation by examination and the provision of objective evidence that the particular requirements for a specific intended use are fulfilled.“
- ISO 9000:2000, 3.8.5
- Validation: Confirmation, through the provision of objective evidence, that the requirements for a specific intended use or application have been fulfilled.

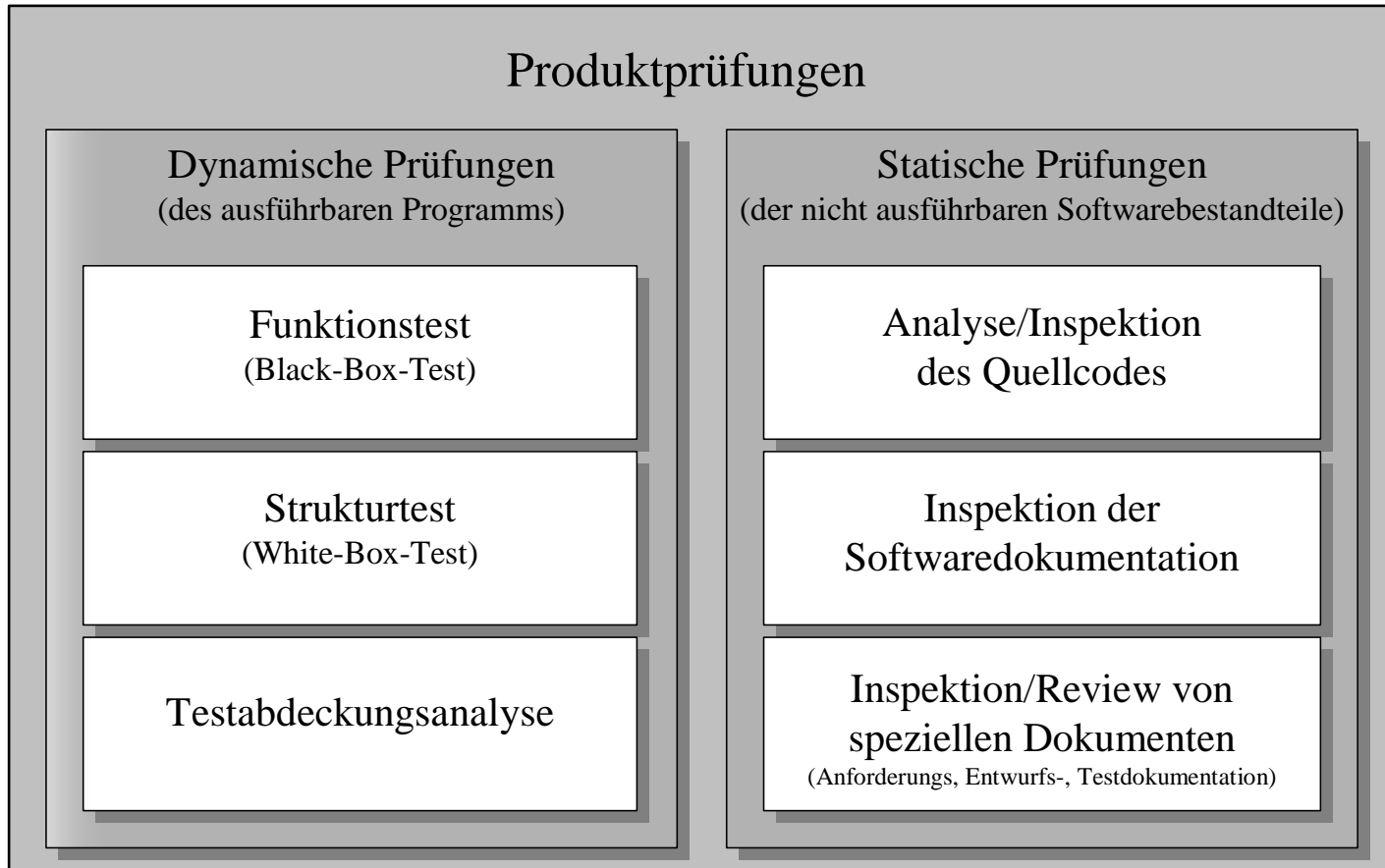
Software Testing Methods

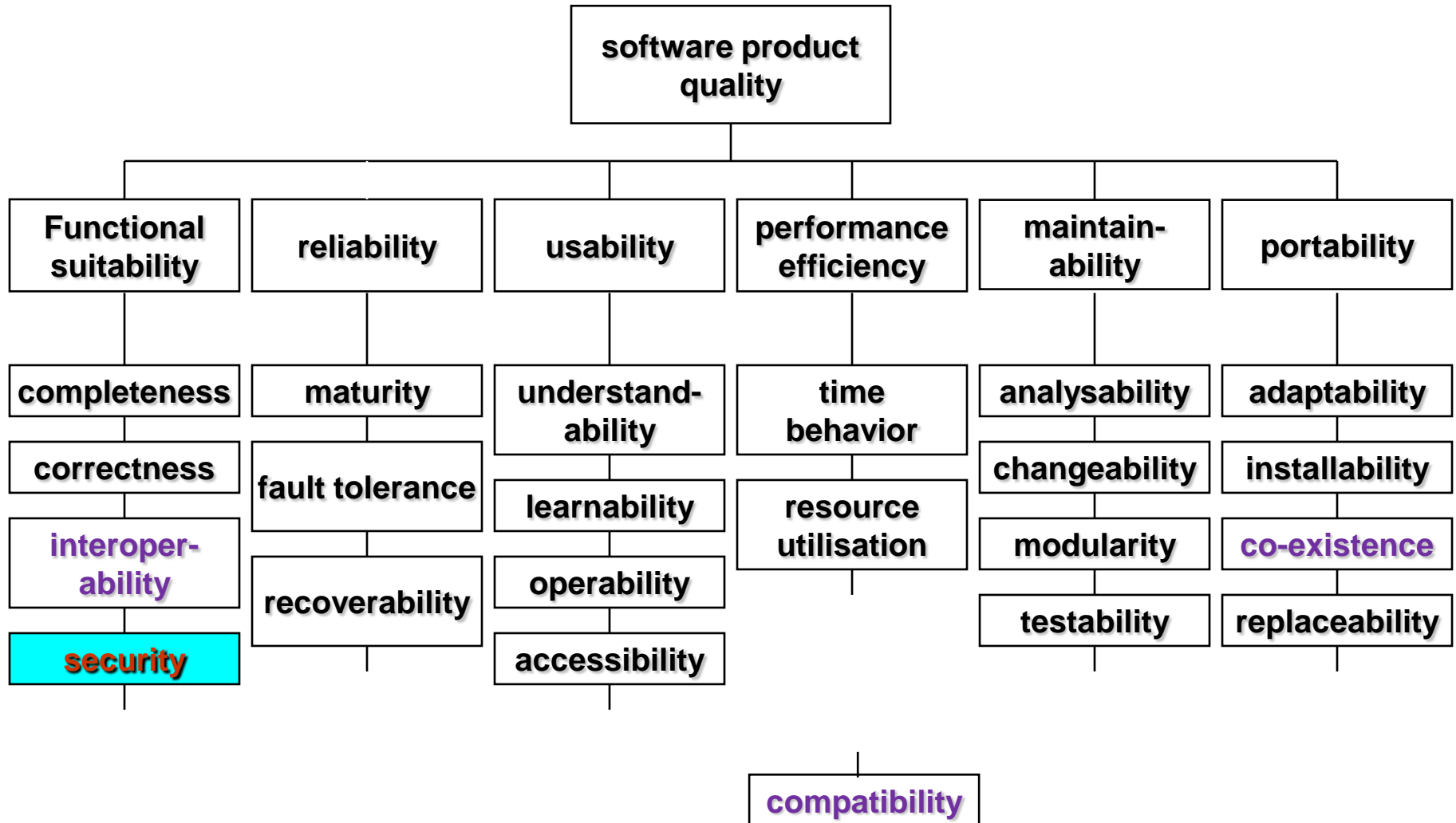


- Simple (informal) reviews of documents
- Manual conformity tests regarding guidelines, definitions, naming conventions, style guides, etc.
- Meetings/discussions with the software developer
- Systematic assessments of factors influencing the software results
- Comparison of software results with results achieved with other (software) methods
- Attestation of long-term correct operation
- Acceptance of certificates, test reports, process audit reports, self-declarations, etc.



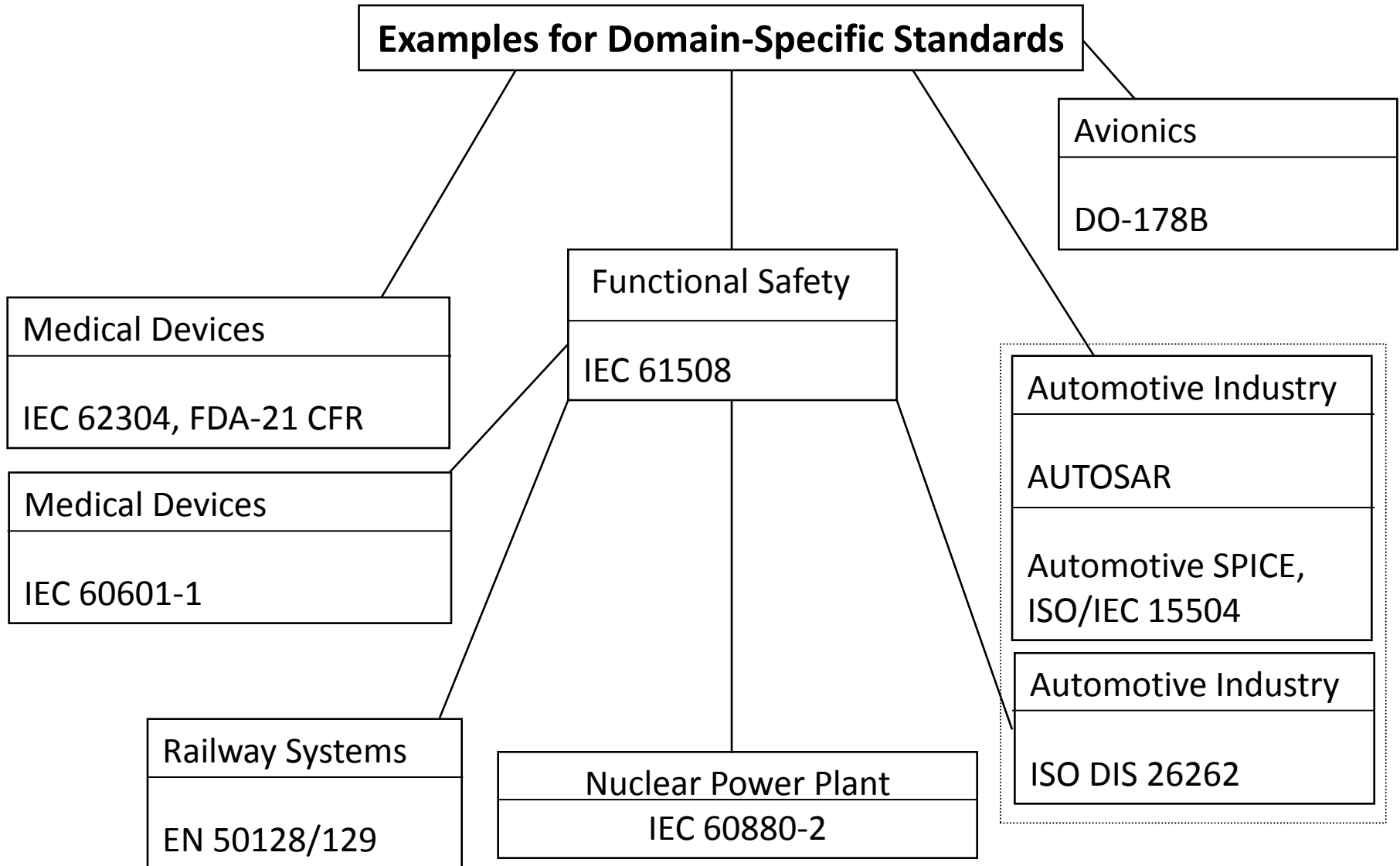
Methoden der Softwareprüfung

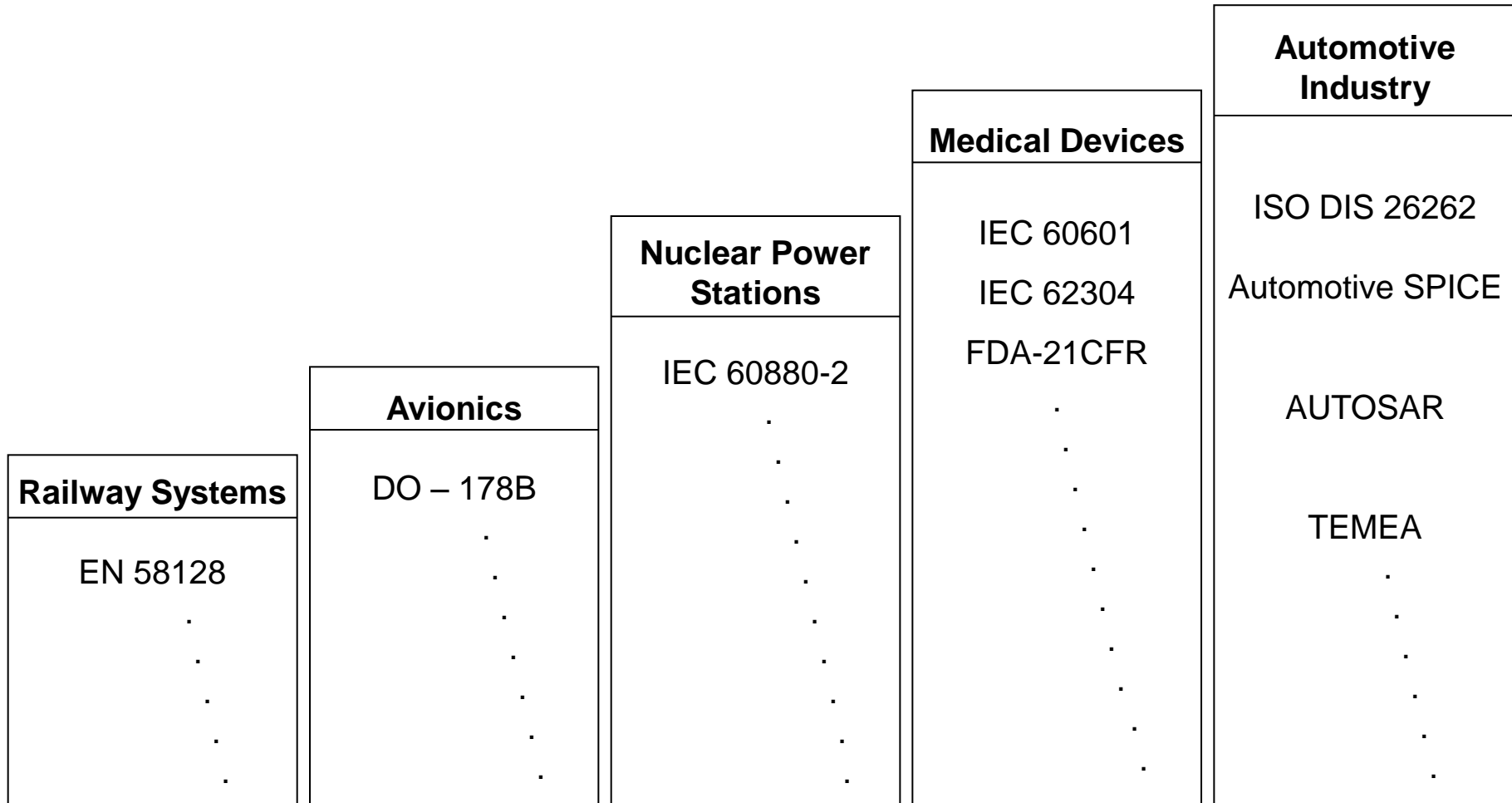




- Subcharacteristics:

Availability	Data and programs must at any time be available to authorised users.
Confidentiality	Information shall be available to authorised users only (access protection).
Integrity	Data and programs must be protected from unintended or unauthorised modifications (including protection from complete loss).
Authenticity	Programs must clearly identify the communication partner (user, process) of protected transaction.





Progress of the standardisation process

for software architecture, development and assessment methods



**Physikalisch-Technische Bundesanstalt
Braunschweig and Berlin**

Abbestraße 2 - 12

10587 Berlin



Daniel Peters

Telefon: +49 30 3481-7916

E-Mail: daniel.peters@ptb.de

www.ptb.de



Stand: 09/15