L3 Mention Informatique
Parcours Informatique et MIAGE

Génie Logiciel Avancé - Advanced Software Engineering

Standards and Legal Constraints

Burkhart Wolff
wolff@lri.fr

Plan of the Chapter

- Introduction: The Role of Standards in SE
- Objectives:
  - Addressing System "Quality", "Safety", or "Security"
- Types of Standards / Norms
  - Generic Process Standards
  - Domain-specific Standards
    (Automative, Railway, Avionics, Medicine, Security)
  - Specific Standards to address phases in Processes
    (attempting assure overall "Quality", or "Tests")

The Role of Norms in Software Engineering

- Reminder: What is it, when I talk about Software Engineering ? Writing:
  - "Write-once, throw away" programs ?
  - Programs written by a small team with 50 KLoCs?
  - Our U-Paris-Saclay Website-Service ?
  - Open-Source Software ?
  - ... or :

The Role of Norms in Software Engineering

- Reminder: What is it, when I talk about Software Engineering ? Writing:

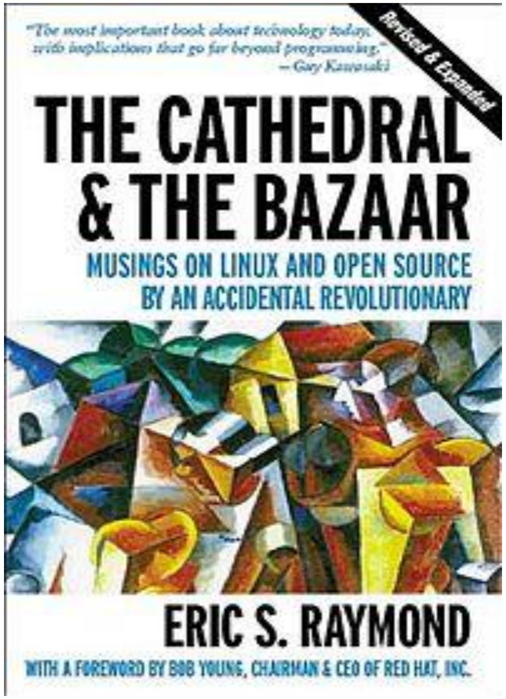
  - ...
  - programs in industrial context with 10 MLoCs, millions of users or large institutions (states, companies) commissioning it ?

The Role of Norms in Software Engineering

Amusing Book: Raymonds Cathedral-Bazaar

Metaphor for (Open-Source) Processes:

- ... The *Bazaar* model, in which the code is developed over the Internet in view of the public. Raymond credits Linus Torvalds, leader of the Linux kernel project, as the inventor of this process.


"The most important book about technology today, with implications that go far beyond programming."
—Guy Kawasaki
Revised & Expanded
THE CATHEDRAL & THE BAZAAR
MUSINGS ON LINUX AND OPEN SOURCE BY AN ACCIDENTAL REVOLUTIONARY
ERIC S. RAYMOND
WITH A FOREWORD BY BOB YOUNG, CHAIRMAN & CEO OF RED HAT, INC.

contrasted to the

- ... The *Cathedral* model, in which source code is available with each software release, but code developed between releases is restricted to an exclusive group of software developers.

(Which is the standard case in industrial projects …)

The Role of Norms in Software Engineering

While it can be argued, if Open-Source Developments are really Bazaar-style or not, Industrial Developments follow clearly the *Cathedral* model

- ... for reasons of legal responsibility
- ... for having a contractual basis
  between partners in industrial developments
- ... for having a control on the
  timing and he investment of a development process.

Modern societies try to establish legal standards if safety, security, economic stability is concerned.

Standardisation organisations can be legal orgs (BIPM, ANSI,…) or industrial consortia (ISO, OMG, …)

The Role of Norms in Software Engineering

Some truths on Software Development Standards

- ... as such, they are usually not the beloved ones by companies and developers (exception: company intern standards to control investment risks)
- ... usually, they give an advantage over a competitor or are required by the contractor ...
- ... require an own management process (quality management, risk assessment, ..., “governance”)
- ... few empirical data over the actual improvement of a process

Objectives: Safety vs. Security vs. Quality

Safety is the condition of protecting human beings against harmful conditions or events, or the control of hazards to reduce risk.

Conference of Computer Safety: … **dependable** application of computers in safety-related and safety-critical systems. SAFECOMP is an annual event covering the state-of-the- art, experience and new trends in the areas of **safety**, **…** and **reliability** of **critical computer applications**.

„Safety Critical System“:

- Energy networks, Aviation, Medicine, Nuclear Power-plants, Military
- Cars, Railway and Signalling Systems
- More and more: Networks and Telecommunication

Objectives: Safety vs. Security vs. Quality

> **Computer security** (also known as **cybersecurity** or **IT security**) is information security as applied to computing devices such as computers and smartphones, as well as computer networks such as private and public networks, including the whole Internet.

Computer security is a branch of information technology known as information security which is intended to protect computers. Computer security has three main goals:

- Confidentiality: Making sure people cannot acquire information they should not (*keeping secrets*)
- Integrity: Making sure people cannot change information they should not (*protecting data*)
- Availability: Making sure people cannot stop the computer from doing its job.

Objectives: Safety vs. Security vs. Quality

Note: Slightly different to the french definition:

La **sécurité des systèmes d’information** (**SSI**) est l’ensemble des moyens techniques, organisationnels, juridiques et humains nécessaire et mis en place pour conserver, rétablir, et garantir la sécurité du système d'information. Assurer la sécurité du système d'information est une activité du management du système d'information.)

Attention: Confusion avec l'usage en français courant:

Les différents types de sécurité correspondent aux modes de transport :

... Sécurité routière ... Sécurité ferroviaire … Sécurité aérienne ... Sécurité en mer ..

Objectives: Safety vs. Security vs. Quality

In software engineering, software quality refers to two related but distinct notions:

- Software functional quality reflects how well it complies with or conforms to a given design, based on functional requirements or specifications. That attribute can also be described as the fitness for purpose of a piece of software or how it compares to competitors in the marketplace as a worthwhile product.[1]

  It is the degree to which the correct software was produced**.**
- Software structural quality refers to how it meets extra-functional requirements that support the delivery of the functional requirements, such as robustness or maintainability. It has a lot more to do with the degree to which the software works as needed.

Hm, a) correctness, but also “fitness to market”

b) extra-functional requirements such as maintainability

Objectives: Safety vs. Security vs. Quality

- Criticism: This classical distinction between safety and security is somewhat outdated ! Security is Safety !
- Story: Sasser Worm spreading April 30, 2004. Named Sasser because it spreads by exploiting a buffer overflow in the component known as LSASS (Local Security Authority Subsystem Service) on the affected operating systems Windows XP /2000.
- Effect: Affected within hours several million machines . . .
  - Agence France Press had all its satellite connections blocked
  - Delta Airlines cancelled Cross-Atlantic Flights
  - Insurance company *If* and *Sampo Bank* had to shut down services
  - British Coastguard had its electronic mapping service disabled
  - Lund University Hospital : no X-Rays possible
  - University of Missoury had to unplug its network
  - ... experts estimated 100 casualties world-wide ...

Objectives: Safety vs. Security vs. Quality

- Criticism: This classical distinction between safety and security is somewhat outdated !

Security is Safety!

- ➢ Renewed Discussion on military exploitation of Viruses after Stuxnet Virus (discovered June 2010, designed to attack the Iran Nuclear Centrifuge Program)
- ➢ Cyber-Warfare developed in the Armies of many Countries

Still, you will find a lot of people disputing over this difference ...

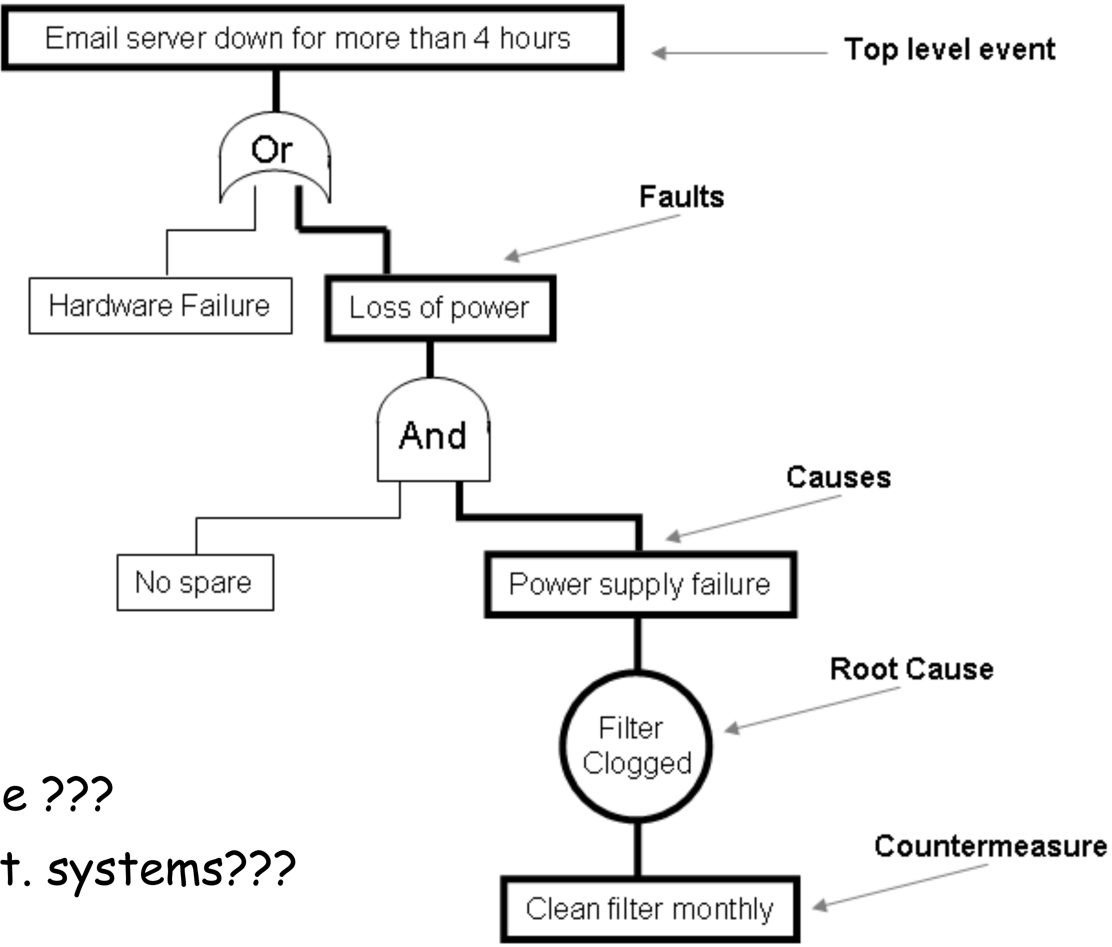
Domain Specific Safety Standards

- A bunch of Safety Standards have their roots in hardware - and systems design, and are therefore centred around probabilistic notions:
  - PFD Probability of Failure on Demand
    PFH Probability of Failure per Hour (Cont. Service)
  - Risk Analysis
- A Certifications must provide:
  - A rigorous definition of 'dangerous failure' for the system in question,
  - Fault Tree Models
  - Likelyhood of Demand, Complexity of Device

Domain Specific Safety Standards

- Example: A Fault-Tree Model

- Criticism:
  - Models and probabilities difficult to justify (risks independent?)
  - Applicable to software ??? To digital, determinist. systems???



Domain Specific Safety Standards

- Core notion:

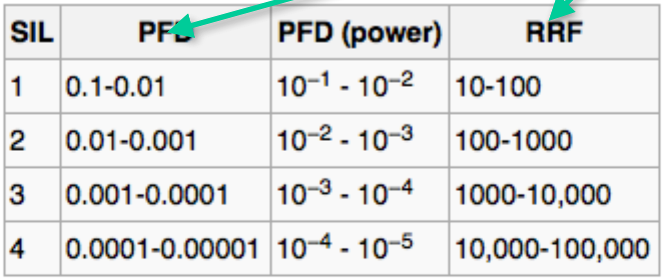
MTBF (Mean Time Between Failures)

RRF (risk reduction factor)

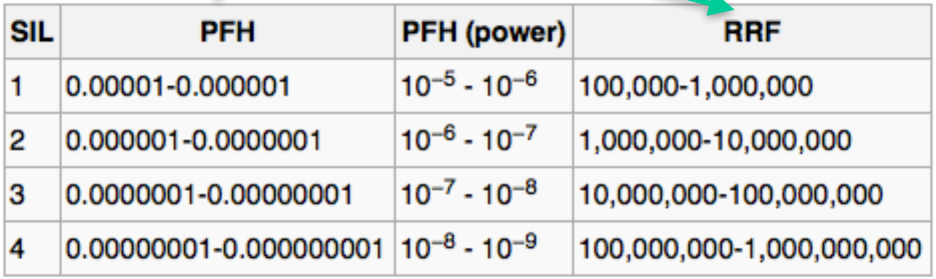
Safety Integrity Level (SIL)

PFH (Probability of failure per hour)

PFD (probability of failure on demand)



SIL	PFD	PFD (power)	RRF
1	0.1-0.01	10^{-1} - 10^{-2}	10-100
2	0.01-0.001	10^{-2} - 10^{-3}	100-1000
3	0.001-0.0001	10^{-3} - 10^{-4}	1000-10,000
4	0.0001-0.00001	10^{-4} - 10^{-5}	10,000-100,000



SIL	PFH	PFH (power)	RRF
1	0.00001-0.000001	10^{-5} - 10^{-6}	100,000-1,000,000
2	0.000001-0.0000001	10^{-6} - 10^{-7}	1,000,000-10,000,000
3	0.0000001-0.00000001	10^{-7} - 10^{-8}	10,000,000-100,000,000
4	0.00000001-0.000000001	10^{-8} - 10^{-9}	100,000,000-1,000,000,000

SIL in Safety Standards

D. Smith, K. Simpson, "Safety Critical Systems Handbook - A Straightforward Guide to Functional Safety, IEC 61508 (2010 Edition) and Related Standards" (3rd Edition, ISBN 978-0-08-096781-3, 270 Pages).

Domain Specific Safety Standards

- The following standards use SIL as a measure of reliability and/or risk reduction
  - ANSI/ISA S84 (Functional safety of safety instrumented systems for the process industry sector)
  - IEC EN 61508 (Functional safety of electrical/electronic/programmable electronic safety related systems)
  - IEC 61511 (Safety instrumented systems for the process industry sector)
  - IEC 61513 (Nuclear Industry)
  - IEC 62061 (Safety of machinery)
  - EN 50128 (Railway applications - Software for railway control and protection)
  - EN 50129 (Railway applications - Safety related electronic systems for signalling
  - EN 50402 (Fixed gas detection systems)

Domain Specific Safety Standards

- The following standards use SIL as a measure of reliability and/or risk reduction
  - ...
  - EN 50402 (Fixed gas detection systems)
  - ISO 26262 (Automotive industry)
  - MISRA, various (Guidelines for safety analysis, modelling, and programming in automotive applications)
  - Defence Standard 00-56 Issue 2 - accident consequence

The use of a SIL in specific safety standards may apply different number sequences or definitions to those in IEC EN 61508.

Domain Specific Safety Standards

- Even from these «soft» probabilistic models, hard «digital» requirements arise:

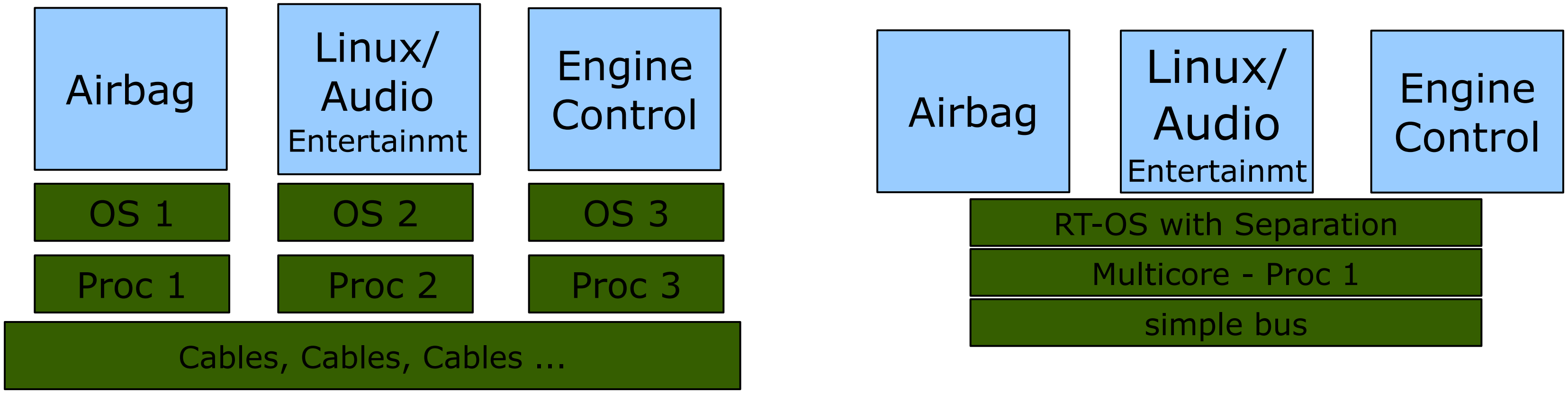
  The international standard on functional safety for software development of road vehicles ISO26262-6 requires the

  freedom from interference by software partitioning

- Thus it is aimed at providing a trusted embedded real-time operating system, which is oriented to ECUs (Electronic Control Units) in automotive industry. (avionics similarly)

Security Standards : Consequences

- Example: A current industrial challenge resulting from the requirement «Freedom of interference»
  - Real-time Operating System Kernels assuring not only memory protection, but « Non-interference » (PikeOS, Sel4, INTEGRITY-178B, RTOS Wind River Systems...)



Airbag

Linux/ Audio Entertainmt

Engine Control

OS 1

OS 2

OS 3

Proc 1

Proc 2

Proc 3

Cables, Cables, Cables ...

Airbag

Linux/ Audio Entertainmt

Engine Control

RT-OS with Separation

Multicore - Proc 1

simple bus

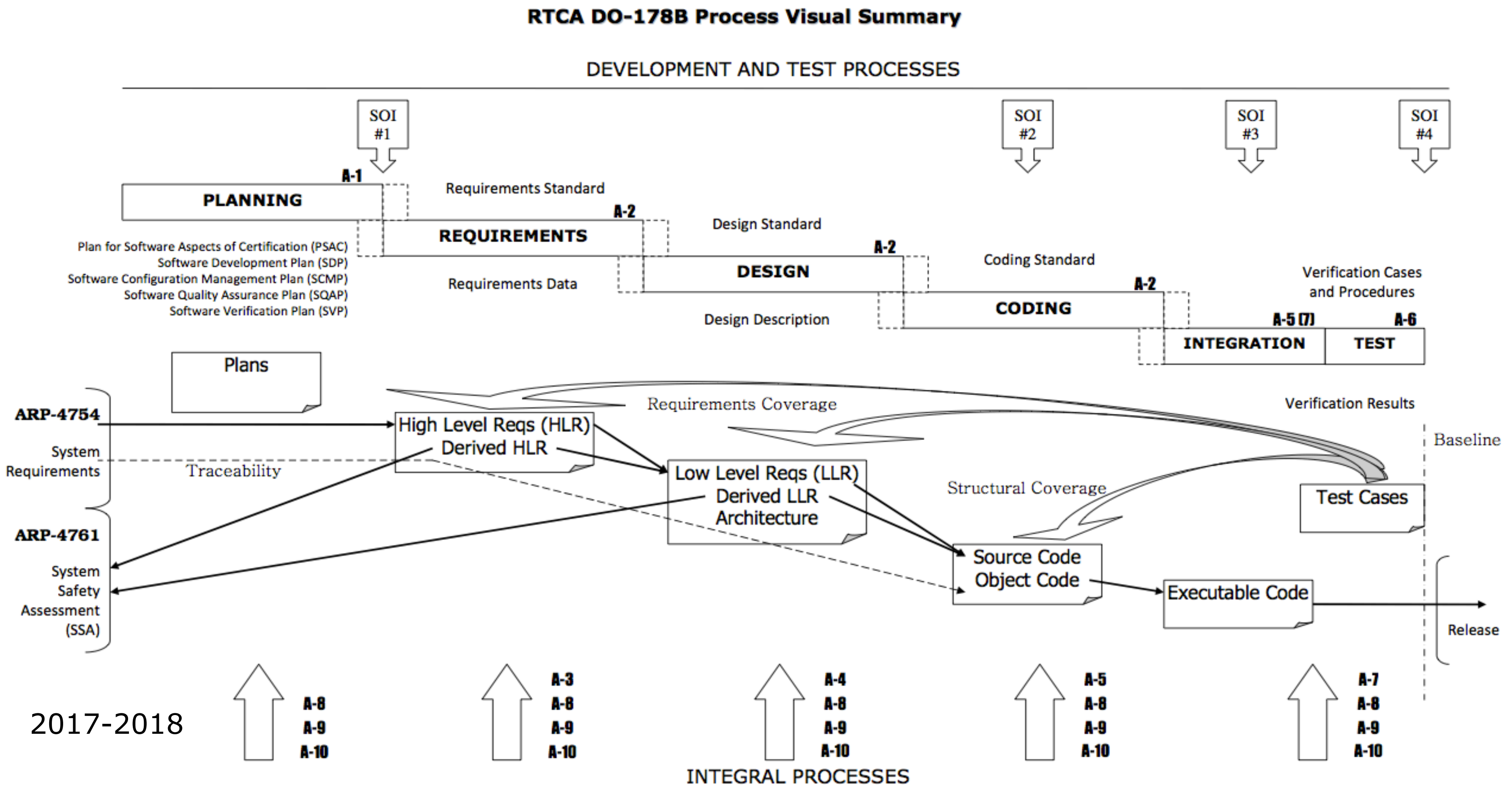
In-between Generic and Specific SE Standards : DO 178B

- ... stems from the Avionics Context (FAA certifications)
- ... but adresses explicitly the needs of software:

The FAA applies DO-178B as the document it uses for guidance to determine if the software will perform reliably in an airborne environment,[1] when specified by the Technical Standard Order (TSO) for which certification is sought. The introduction of TSOs into the airworthiness certification process, and by extension DO-178B, is explicitly established in 14 Code of Federal Regulations (CFR) Part 21, Subpart O.

In-between Generic and Specific SE Standards : DO 178B

- DO 178B makes explicit requirements
  - on the SE Development process and its documentation



2017-2018

In-between Generic and Specific SE Standards : DO 178B

- DO 178B makes explicit requirements
  - on the SE Development process and its documentation
    (with an emphasis on traceability of Requirements
    to Code and Tests)
  - Verification. The process is required to produce documents on
    - Software verification procedures
    - Software verification results
      (review of requirements and design,
      tests of object code coverage analysis)
    - Required:Unit testing, Integration testing,
      Black-box and acceptance testing
  - Configuration Management
  - Quality Assurance
  - Tools: Must be certified as well! (Except Blackbox-Testing)

Generic Software Engineering Standards

A truly generic SE Norm:

ISO/IEC 250XX for of software quality.

(SQuaRE -- System and software quality models). It replaces: **ISO/IEC 9126** *Software engineering — Product quality Software engineering — Product quality* as well as **ISO/IEC 14598**)

http://www.iso.org/iso/catalogue_detail.htm?csnumber=35733

ISO/IEC 25030:2007 helps to improve the quality of software quality requirements. It does this by providing requirements and recommendations for quality requirements, and guidance for the processes used to define and analyse quality requirements. It applies the quality model defined in ISO/IEC 9126-1 [ISO/IEC 25010] and it complies with the requirement processes defined in ISO/IEC 15288.

Generic Software Engineering Standards

A truly generic SE Norm:

ISO/IEC 250XX for of software quality.

Software product quality requirements are needed for:

- specification (including contractual agreement and call for tender);
- planning (including feasibility analysis);
- development (including early identification of potential quality problems during development); and
- evaluation (including objective assessment and certification of software product quality).

Generic Software Engineering Standards

ISO/IEC/IEEE 29119 for software testing.

A not undisputed norm - the International Organisation for Software testing rejected it (see: http://en.wikipedia.org/wiki/ISO/IEC_29119)

- **ISO/IEC 29119-1:** Concepts & Definitions,[2] published September 2013
- **ISO/IEC 29119-2:** Test Processes,[3] published September 2013
- **ISO/IEC 29119-3:** Test Documentation,[4] published September 2013
- **ISO/IEC 29119-4**: Test Techniques
- **ISO/IEC 29119-5**: Keyword Driven Testing

Part 4 : Test Techniques:

This covers many common dynamic software testing techniques from the Specification based and Structure based areas, such as Equivalence Partitioning (=DNF), Classification Tree, Error Guessing (Specification based) and Statement testing, Decision testing, Data flow testing (Structure based).

Part 4 also includes definitions for quality related testing types, such as Usability, Disaster Recovery, Conversion, Compatibility testing.

Generic Software Engineering Standards

ISO/IEC/IEEE 15408 for computer security certification:

«Common Criteria» (CC)

- Framework where users can specify security functional and assurance requirements (SFR and SAR) by Protection Profiles (PP)
- Vendors/Developers can implement and/or claim security attributes of their products
- Evaluators (usually test labs) evaluate the products and determine if they actually meet the claims.
- A certification authority (France: ANSI, Germany: BSI) issues certificate

Common criteria provides assurance that the process of specification, implementation and evaluation of a computer security product has been conducted in a rigorous and repeatable manner.

Generic Software Engineering Standards

ISO/IEC/IEEE 15408 for computer security certification:

«Common Criteria» (CC)

- Evolved Terminology:
  - EAL: Evaluation Assurance Level
  - PP: Protection Profile
  - SAR: Security Assurance Requirement
  - SF: Security Function
  - SFR: Security Functional Requirement
  - SFP: Security Function Policy
  - SOF: Strength of Function
  - ST: Security Target
  - TOE: Target of Evaluation
  - TSP: TOE Security Policy
  - TSF: TOE Security Functionality
  - TSC: TSF Scope of Control
  - TSFI: TSF Interface

Generic Software Engineering Standards

ISO/IEC/IEEE 15408 for computer security certification:

«Common Criteria» (CC)

Documentation process and assurance levels:

- EAL1: Functionally Tested
- EAL2: Structurally Tested
- EAL3: Methodically Tested and Checked
- EAL4: Methodically Designed, Tested and Reviewed
- EAL5: Semi-formally Designed and Tested
  (Smart-Cards, Tenix Interactive Link, XTS-400 (an OS))
- EAL6: Semi-formally Verified Design and Tested
  (Green Hills INTEGRITY-178 RTOS)
- EAL7: Formally Verified Design and Tested
  (Fox Data Diode, Gemplus Smart Card).

Conclusion

- Attempts to control development processes and software products by standards (norms)
- Attempts to assure and certify software quality.
- Most serious and relevant standards (in France):
  - DO 178B (Avionics)
  - CENELEC 50128 (Avionics)
  - ISO 9000 (Processes)
  - **ISO/IEC/IEEE** 29119 (Software Test)
  - **ISO/IEC/IEEE** 15408 «Common Criteria» for computer security certification requiring formal models as well as proof techniques for EAL 6 and EAL 7.