Safety and Reliability of Embedded Systems

(Sicherheit und Zuverlässigkeit eingebetteter Systeme)

Quality Assurance and Quality Management

Content

- Organizational aspects of quality assurance and quality management
- Separation of development and quality assurance personnel
- Documentation and evaluation of quality inspection activities
- Standards
  - Relevance of standards
  - Process-oriented standards
  - Technical standards
- Literature

- Alternative organizational forms of quality management
  - Total quality management (TQM)
    - Distributed quality responsibility
    - No independent quality assurance
  - Classic quality assurance
    - Clear distinction between the role of the developer and the role of the quality assurance personnel
    - The quality assurance personnel is responsible for quality. Thus, they need a strong and independent position
    - In this organizational form, the quality assurance is usually not subordinated to the project management

- The majority of software companies use organizational forms which are a compromise between TQM and classic quality assurance
- In safety-critical applications: Independent quality assurance is done in addition to the quality assurance measures for which the developers are responsible (required by related standards)
  - E.g. DIN EN 50128 „Bahnanwendungen – Software für Eisenbahnsteuerungs- und Überwachungssysteme“ distinguishes the roles of verification and validation personnel
    - Verification: Check of the failure-free implementation of the requirements of a phase by the result of the phase
    - Validation: Demonstration that the product fulfills its requirements

The standard requires the following

- For software which is not safety-critical (software safety integrity level 0), it is allowed that design, implementation, verification and validation are done by the same person
- For software of safety integrity levels 1 and 2, it is allowed that verification and validation is done by the same person. But it is required that this person must not be identical with the designer/programmer. This arrangement assures the “4-eye-principle”: Development and quality assurance are realized by different persons. However, both groups are allowed to report to the same project leader. Hence, the project leader has the possibility to ignore warnings of the quality assurance personnel
- For software of high safety integrity levels 3 and 4, there exist two alternative organizational forms of quality assurance
  - Verification and validation can be done by the same person, but these persons are not allowed to be a designer/programmer at the same time. Verification and validation personnel do not report to the project management and must have the possibility to prevent the release of the software
  - Design/implementation, verification and validation are all done by different persons. Designer/programmer and verification personnel report to the project management. Validation personnel must have an independent way to report and must be able to prevent the release of the software

- These rules are based on the following principle: The independency of quality assurance with respect to its organization and staff has to increase with increasing safety criticality of the software
- For safety critical software, the classical organization of quality assurance is preferred over the principles of Total Quality Management. This rule is simplified because the existence of an independent quality assurance does not exclude a quality assurance which is integrated in the development. In contrast: especially in a safety critical development it is important to make sure that the developers know the quality objectives and that appropriate methods are used to accomplish them. Furthermore, it is important to check whether these objectives have been reached. Quality has to be actively developed – it cannot be tested “into” the software

- At first sight, the separation of development and verification/validation personnel seems to make sense. However, on closer examination it proves to be not correct
  - If e.g. the module test is accomplished by an independent person, the sole duty of the programmer is to compile his module failure free. When all syntactical errors are corrected, the responsibility is then taken over by the independent module tester. He is able to recognize certain failures which cannot be recognized by the programmer. Such failures might be caused by e.g. misunderstandings of the module specification, which do not occur to an independent tester in the same way. It is a disadvantage that the independent tester has not the precise knowledge of the structure of the module compared to the programmer. The programmer knows the reason for a certain control structure and the tasks of the variables, and how certain test cases have to be processed. This knowledge of the programmer is not used since the independent tester does not have it. The example shows that there are arguments for verification by the developer and for verification by another person. Systematic test techniques offer a solution for this problem

Sample rule for assigning responsibilities

- If the module test requires a branch coverage test, the programmer has to reach a branch coverage of e.g. 80%. This procedure uses the know-how of the programmer at the beginning of the test phase and assures that a module is passed on to the independent tester which basically works. Afterwards, an independent tester takes over the responsibility. This can be e.g. the same person which also does the integration test for the module. In this case, the responsibility is taken over already before the end of the phase. The independent tester finishes the test and thereby assures the „4-eye-principle“. The integration test and the system test are usually done by independent persons. Furthermore, in large software projects, the system test is not subordinated to the project management

- It make sense to integrate those persons performing verification/validation activities in associated development phases. The system tester should be integrated in the analysis, the integration tester in the design and the module tester in the implementation phase. The same rule applies also vice versa. Someone who was involved in the analysis should be integrated into the planning of the system test cases. Also, a designer should be integrated into the planning of the integration test and a programmer into the planning of the module test
- In a mature organisation there exist systematic workflows, defined goals and possibilities to check whether these goals have been reached. The developer is thereby responsible for achieving the defined goals. The task of a strong and organisationally independent quality assurance is simply to check whether the defined goals have been reached. This is done independently from the project manager. Such an organization of quality assurance combines the advantages of the classic quality assurance approach and Total Quality Management

- All standards regarding quality management and quality assurance emphasize the importance of quality inspection activities done in a systematic fashion. Quality inspections have to be planned, performed, checked, evaluated, and documented systematically. The standard DIN EN ISO 9000-3 /DIN EN ISO 9000-3 97/ requires – similar to numerous other standards – the existence of a quality assurance plan which should contain the following topics:
  - Measurable quality goals
  - Criteria for the inputs and results of each development phase
  - Defined types of quality inspections
  - Detailed planning of the quality inspections including dates, tools, and bodies responsible for acceptance
  - Responsibilities

- Quality inspections have to be documented particularly to confirm that they have been done correctly
- For dynamic testing the documentation usually comprises
  - the test plan
  - a confirmation that the test cases have been executed
  - the documentation of the test results
  - the failure report
- Depending on the applied test technique, these documents may be structured differently
- Function-oriented testing
  - Test plan: List of equivalence classes with assigned test cases
  - Report: Confirmation of test execution and related test results
- Structure-oriented testing
  - Report generated by test tool

- Failure tracking
  - Timestamp
  - Test case
  - Rating of failure severity
  - Failure classification

No.	Date	Test Case	Severity (1-4)	Failure Classification	Correction Date	Fault Classification	Correction Effort (MD)
1	05.08.2002	218	1	Total loss	12.08.2002	Programming fault	0.5
2	08.08.2002	279	3	Time requirement violated			
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

- In case of doubt, standards decide which procedures, methods, and techniques can be considered as state of technology and/or state of the art
- Standards and norms
  - No legal norm but anticipated expertises
- Regulations by law
  - e.g. Product Liability Act, compensation of damages according to BGB
- European guidelines
  - have the character of a law because they have to be mandatorily converted to national law by the member states
- Regulations
  - in most cases issued by public authorities (the executive) and are usually mandatory

- Standardisation in Germany is the systematic unification of material and immaterial objects for the benefit of the general public, collaboratively done by interested society circles. German standards are created in an association under private law by interested society circles (e.g. DIN Deutsches Institut für Normung e.V., Verband Deutscher Elektrotechniker (VDE) e.V.). Standards and norms are not legal norms. The are – in contrast to laws – not legally binding, but they can be understood as anticipated expertises. A manufacturer can assure by compliance to relevant standards that he has reached the state of technology and thus has fulfilled his duty to take care

Process-Oriented Standards

- Rules for e.g. processes, activities, tasks, and responsibilities in software development and software quality assurance
- Basically they contain organisational requirements
- They barely enclose detailed technical requirements
- Examples
  - DIN ISO 9000 series
  - V-Model
  - ISO/IEC TR 15504 for SPICE assessments
  - AQAP-Century-Standards for military applications

Technical Standards

- Technical standards may concern either a certain field of application – e.g. aviation or rail traffic – or certain kinds of systems which can appear in numerous fields of application
- Often contain explicit rules of to-be-applied techniques
- Examples
  - IEC 61508 /IEC 61508 98/ is a very enclosing standard related to safety of electrical/electronic/programmable electronic safety-related systems. Software is particularly discussed in IEC 61508-3
  - The standards DIN EN 50128 /DIN EN 50128 01/ and Mü 8004 /Mü 8004 99/ are used for rail traffic systems
  - The standard /RTCA/DO-178B 92/ concerns software requirements for avionics

Literature

/DIN EN 50128 01/: DIN EN 50128, Bahnanwendungen – Telekommunikationstechnik, Signaltechnik und Datenverarbeitungssysteme – Software für Eisenbahnsteuerungs- und Überwachungssysteme, Berlin: Beuth Verlag 2001

/DIN EN ISO 9000-3 97/: DIN EN ISO 9000-3, Normen zum Qualitätsmanagement und zur Qualitätssicherung/QM-Darlegung – Teil 3: Leitfaden für die Anwendung von ISO 9001 auf Entwicklung, Lieferung, Installation und Wartung von Computer-Software, Berlin: Beuth Verlag 1997

/IEC 61508 98/: IEC 61508, Functional safety of electrical/electronic/programmable electronic safety-related systems: Parts 1 – 7, International Electrotechnical Commission, 1998

/RTCA/DO-178B 92/: RTCA/DO-178B, Software Considerations in Airborne Systems and Equipment Certification, RTCA, Inc., 1992