TECHNISCHE UNIVERSITÄT KAISERSLAUTERN

0101seda010100

software engineering dependability

Quality Management of Software and Systems: Organization of Tests

Contents

- Organization of Quality Assurance and Quality Management
  - Separation of Development and QA
- Test Documentation and Evaluation
- Standards
  - Importance of standards
  - Process-oriented standards
  - Technical standards
- Literature

Organization of Quality Assurance and Quality Management

- Alternative organizational forms of quality management
- Total Quality Management (TQM)
  - Distributed QA responsibilities
  - No independent QA
- Classical quality assurance
  - Strict separation between developers and testers
  - In principle, the tester is responsible for quality. Therefore, he requires to have an independent position
  - Normally, quality assurance is not subordinate to the project management

Organization of Quality Assurance and Quality Management

- The majority of software companies uses a compromise between TQM and classical QA
- For safety critical applications: independent QA, in addition to the quality assuring techniques performed by the developers (requirement of relevant norms)
- E.g., DIN EN 50128 „Bahnanwendungen – Software für Eisenbahnsteuerungs- und Überwachungssysteme“ (train applications – software for train control and monitoring systems) differentiate the roles of the verifier and the validater
  - Verification: Test whether the input of a specific phase is transformed into the output without introducing faults.
  - Validation: Demonstrate that the product fulfills its requirements

Organization of Quality Assurance and Quality Management: Standard DIN EN 50128

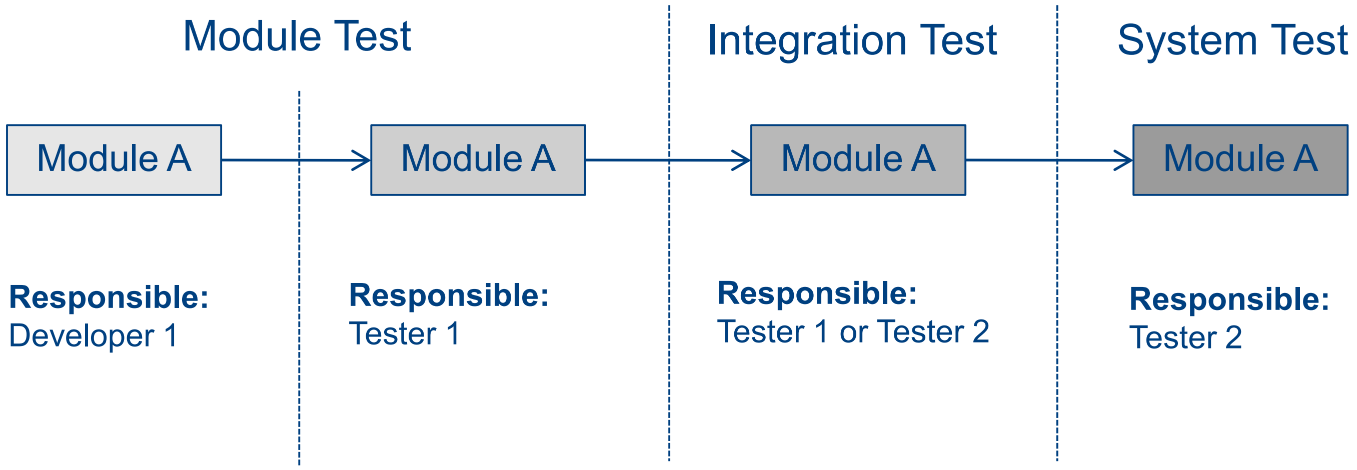
- The basis for these requirements is the following principle: With an increasing safety integrity level the independency of development and quality assurance must increase, too
- For safety-critical software the classical organization is often preferred against TQM. An independent QA is required, but may be enriched by TQM techniques, that are used throughout the development. Especially for safety-critical software, the developers should be aware about quality goals and should use techniques to reach and proof these goals. Quality can not be tested into the software at the end. Quality must be taken into account during all development phases.

Organization of Quality Assurance and Quality Management: Separation of Development and QA

- A separation of the developer role and the tester role makes only sense at the first glance. It is not always recommended, though
  - If, e.g., the unit test is done by an independent person, the only task of the developer is to compile his module/unit without faults. If all syntax errors have been eliminated, the responsibility passes to the independent unit tester. He can identify certain faults that the developer can not identify, e.g., faults resulting from a misinterpretation of the specification by the developer. On the other hand, the developer knows much about the software structure. There are reasons why the tests should be performed by the developer, and there are other reasons why the tests should be performed by an independent person. Systematic test techniques offer solutions to this problem.

Organization of Quality Assurance and Quality Management : Separation of Development and QA

- Example of a responsibility rule
  - If a branch coverage test is made during the module test, the developer has to achieve a certain branch coverage (e.g., 80%). This uses the developer's expertise and ensures a basic quality of the module that is handed over to the tester. Afterwards, an independent tester is responsible. This may be the same person performing the integration test later on. In this case, the responsibility changes before the completion of the unit test phase. The tester finishes the test and ensures an independent test. Normally, the integration test and the system test are performed by independent persons.



Organization of Quality Assurance and Quality Management: Separation of Development and QA

- It makes sense to involve the testers in the related development phases. The system tester should be involved in the analysis phase, the integration tester should be involved in the design phase, and the module tester should be involved in the implementation phase.
- In a mature company well-defined goals exist. Developers are responsible for attaining these goals. The task of the independent quality assurance is to check and confirm whether these goals have been reached. This would be a combination of TQM used to develop high-quality software and the classical QA.

Test Documentation and Evaluation

- All standards about quality management and quality assurance underline the importance of a systematic approach to testing. Tests have to be systematically planned, performed, controlled, evaluated, and documented. The standard ISO/IEC 90003:2004 /ISO/IEC 90003 04/ demands, as numerous other standards, the existence of a quality assurance plan, containing the following items
  - Measurable quality goals
  - Criteria for the requirements and the results of each development phase
  - Specification of test types
  - Detailed test plan, including deadlines, resources, and permission authorities
  - Responsibilities

Test Documentation and Evaluation

- Tests have to be documented
- For dynamic tests, the documentation normally consists of
  - Test strategy
  - Test execution document
  - Evaluation of test results
  - Failure / fault description
- This documents might differ depending on the used techniques:
  - Function-oriented test
    - Test strategy, e.g., equivalence classes with corresponding test cases
    - Test execution document and test results: Manual protocol
  - Structure-oriented test
    - Automatic protocol provided by a test tool

Test Documentation and Evaluation

- Recording of failures
  - Date
  - Test case
  - Failure classification
  - Fault classification

No.	Date	Test Case	Severity	Failure Classification	Date of correction	Fault Classification	Effort of correction (man*days)
1	04.01.2010	218	1	Total Failure	12.01.2010	Programming fault	0,5
2	07.01.2010	279	3	Time Requirement violated			
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

Standards
Importance of standards

- Standards determine which procedures, methods, and techniques are state-of-the-practice and state-of-the-art, respectively

- Standards
  - Not law, but anticipated expert opinions

- Law
  - E.g., product liability act, and other laws, e.g, civil code

- European directives
  - Are laws, since all EU member states must put them into their national legislation

- Directives
  - Normally decreed by administrative institutions, no law!

Standards
Importance of standards

- In Germany, standardization is the planned unification of physical and insubstantial subjects for general benefit. German standards are developed within a society under private law (e.g., DIN Deutsches Institut für Normung e.V., Verband Deutscher Elektrotechniker (VDE) e.V.). Standards are not laws. They are, in contrast to laws, not legally binding, but can be seen as anticipated expert opinions. By compliance with relevant standards, manufacturers can ensure the use of the state-of-the-practice technology.

Standards
Process-Oriented Standards

- Rules (e.g., procedures, workflows, tasks, and responsibilities) for software engineering and quality assurance
- Basically, organizational requirements
- Hardly no technical requirements
- Examples
  - DIN ISO 9000
  - V-Model XT
  - ISO/IEC 15504 (SPICE)
  - AQAP-Century-Standards for the military domain

Standards
Technical Standards

- Technical standards may be applicable to a specific application domain (e.g., aviation, railways) or to a specific type of systems, e.g., programmable E/E-Systems
- Contain explicit recommendations of techniques
- Examples
  - IEC 61508 /IEC 61508 98/ is a broad standard for safety of electric or electronic programmable, safety-critical systems. Software is dealt with especially in IEC 61508-3
  - The standard DIN EN 50128 /DIN EN 50128 01/ is intended for the application to railway transportation
  - The standard /RTCA/DO-178C 12/ deals with software requirements for aviation systems

Literature

- /DIN EN 50128 01/: DIN EN 50128, Bahnanwendungen – Telekommunikationstechnik, Signaltechnik und Datenverarbeitungssysteme – Software für Eisenbahnsteuerungs- und Überwachungssysteme, Berlin: Beuth Verlag 2001
- /ISO/IEC 90003 04/: ISO/IEC 90003, Software Engineering – Guidelines for the application of ISO 9001:2000 to computer software. 2004
- /IEC 61508 98/: IEC 61508, Functional safety of electrical/electronic/programmable electronic safety-related systems: Parts 1 – 7, International Electrotechnical Commission, 1998
- /RTCA/DO-178C 12/: RTCA/DO-178C, Software Considerations in Airborne Systems and Equipment Certification, RTCA, Inc., 2012