Quality Management of Software and Systems

Organization of Tests

Contents

- Organization of Quality Assurance and Quality Management
  - Separation of Development and QS
- 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
  - Responsible for the quality is the tester. He needs to be independent
  - Normally, quality assurance is not subordinate to the project management

Organization of Quality Assurance and Quality Management

- The majority of software producing 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, if the requirements of a specific phase are transformed without faults in the final product of that phase
  - Validation: Demonstration, that the product fulfills its requirements

Organization of Quality Assurance and Quality Management
Norm DIN EN 50128

The Norm include the following requirements

- For non safety critical software (safety integrity level 0) the designer, the developer, the verifier, and the validater may be the same person
- For software of safety integrity level 1 or 2 the verifier and the validater may be the same person, but must not be the same person as the designer/developer. This requirement ensures a 'second set of eyes'. The development and the quality assurance is performed by different groups, but both groups may report to the same project manager. Thus, the project manager may ignore warnings from the quality assurance
- For software of safety integrity level 3 and 4, two alternative organizational forms exists
  - The verifier and the validater may be the same person. They must not be simultaneously the designer or the developer. The verifier and the validater do not report to the project management and must be able to prohibit the release
  - The designer/developer, the verifier, and the validater are different persons. The designer/developer and the verifier report to the project management, but the validater uses an alternative reporting. He must be able to prohibit the release

Organization of Quality Assurance and Quality Management
Norm DIN EN 50128

- The basis for these requirements is the following principle: With an increasing safety integrity level the independency of the human resources and the organization performing the QA must increase, too
- For safety-critical software the classical organization is often preferred against TQM. But the existence of an independent QA does not forbid a development included QA. 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, but must be developed into it

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

- A stringent separation of the developer role and the tester role makes sense only at the first glance. It is not always recommended, though
  - If, e.g., the module test is done by an independent person, the only task of the developer is to compile his module without faults. If all syntax errors have been eliminated, the responsibility passes to the independent module tester. He can identify certain faults the developer can not identify, e.g., faults resulting in a misinterpretation of the module specification by the developer, not likely to be done the same way by the tester. On the other hand the testers' lacking knowledge about the module structure is detrimental. The developer knows why a certain control structure is used, which functions certain variables fulfill, and how test cases have to be handled. This knowledge is not used by an independent tester. This example shows, that there are reasons the tests should be performed by the developer and reasons 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 meet a certain branch coverage (e.g., 80%). This uses the developer's knowledge and ensures a basically functional module is handed over to the tester. Afterwards, an independent tester is responsible. This may be the same person performing the integration test. In this case, the responsibility changes before the completion of this phase. The independent tester finishes the test and ensures a 'second set of eyes'. Normally, the integration test and the system test are performed by independent persons. In a large software development project, the system test is not subordinate to the project management

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. This rule also applies backwards. For the planning of the system tests a person involved in the analysis should be involved. Similarly, a designer should be involved in the integration test planning and a developer should be involved in the module test.
- In a mature company systematic workflows, well-defined goals, and possibilities to check their attainments exist. Developers are responsible for attaining these goals. The task of an organizational independent quality assurance is merely to check if this goals are attained independently to the project management. Such an organization of the QA combines the advantages of the classical QA and TQM

Test Documentation and Evaluation

- All norms about quality management and quality assurance stresses the importance of a systematic approach to testing. Tests have to be systematically planned, performed, controlled, evaluated, and documented. The norm DIN EN ISO 9000-3 /DIN EN ISO 9000-3 97/ demands, as numerous other norms, 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, particularly to proof their correct realization
- For dynamic tests, the documentation normally consists of
  - Test strategy
  - Proof of test realization
  - Evaluation of test results
  - Fault Description
- This documents can be differ according to the used techniques
- Function-oriented test
  - Test strategy basis: Equivalence classes with corresponding test cases
  - Proof of test realization and test results: Protocol
- Structure-oriented test
  - Protocol of a test tool

Test Documentation and Evaluation

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

Nr.	Datum	Testfall	Schwere (1 - 4)	Klassische Fehlverhalten	Korrektur-datum	Klassische Fehler	Korrektur-aufwand (MT)
1	05.08.2002	218	1	Totalausfall	12.08.2002	Programmier-Fehler	0,5
2	08.08.2002	279	3	Zeitanfordung verletzt			
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

Standards
Importance of standards

- In case of doubt, standards determine which procedures, methods, and techniques are state-of-the-practice and state-of-the-art, respectively
- Standards and norms
  - No legal norm, but anticipated expert opinions
- Legal regulation
  - E.g., product liability act, compensation according to civil code
- European directives
  - Have law character, as the member states have to implement them to national law
- Regulations
  - Normally decreed by the administration – the executive – and as a general rule mandatory

Standards
Importance of standards

- In Germany, standardization is the planned unification of physical and insubstantial subjects for general benefit by interested circles. 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 and norms are no legal norms. They are, in contrast to laws, not legally binding, but can be comprehend as anticipated expert opinions. By compliance with relevant standards, manufacturers can ensure the use of the state-of-the-art and, therefore, has satisfied his due diligence

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 chain
  - V-Model
  - ISO/IEC TR 15504 about the assessment procedure SPICE
  - AQAP-Century-Standards for the military domain

Standards
Technical Standards

- Technical standards may be for a specific application domain (e.g., aviation, railway transportation) or for a specific type of systems, coming from different application domains
- Contain explicit regulation for the techniques to use
- 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 standards DIN EN 50128 /DIN EN 50128 01/ and Mü 8004 /Mü 8004 99/ are intended for the application to railway transportation
  - The standard /RTCA/DO-178B 92/ affects 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
- /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