

Génie Logiciel Software quality

Sylvain Lobry

23/09/2022



Definition of software quality

- ISO 9000:2015 standard
 - "degree to which a set of inherent characteristics of an object fulfils requirements"
 - Requirements can be implicit or explicit
- Side note: ISO = International Organization for Standardization



Examples of desirable properties

- **Fiability**: the software is valid (obtains desired outcome) and robust (able to cope with errors during the execution or erroneous input)
- **Safety**: when software is safety-critical, additional requirements
- **Performances**: expression of requirements in execution or processing speed and in terms of resources (e.g. memory)
- **Compatibility**: the software can be used with other softwares (either defined or meeting a specific format)
- **Portability**: the software can be used in various environments (e.g. OS)
- **Ergonomics**: the software can be easily (or naturally) used by the end-user
- Many more in few slides



I want it all!

- Quality comes at a cost. Non qualitative software as well -> Notion of "Cost of Software Quality"
- Often "level of quality in a software product can be inferred from the cost of activities related to dealing with the consequences of poor quality"
- 4 types of cost of quality:
 - Prevention (training, infrastructure, tools)
 - Appraisal (review, testing)
 - Internal failure (before delivery of the product)
 - External failure (after delivery of the product)



Software Quality Assurance (SQA)

- Set of activities to ensure that the software is of suitable quality (at every stage of the development)
- In other words, SQA will check the conformity of the product to standards
- Often, people from SQA are independent from the project
- NOT testing, e.g.:
 - Checking that 90% of the code is unit tested -> SQA
 - Writing the unit tests -> testing



Standardization bodies

- International System Organization (ISO)
- Institute of Electrical and Electronic Engineers (IEEE)
- Association Française de Normalisation (AFNOR)
- American National Standard Institute (ANSI)



Quality standards - ISO 25000 series

- Systems and software engineering Systems and software Quality Requirements and Evaluation (SQuaRE)
- Published from 2005
- Framework for the evaluation of software product quality
- 5 parts:
 - 2500x: Quality Management
 - 2501x: Quality Model
 - 2502x: Quality Measurements
 - 2503x: Quality Requirements
 - 2504x: Quality Evaluation



Quality standards - ISO 25010

- Follows ISO 9126
- 8 **product quality characteristics**, each with *sub characteristics*:
 - Functional suitability: how well the software provides functions satisfying explicit and implicit needs. Functional Completeness, Functional Correctness, Functional Appropriateness
 - Reliability: under specific conditions, what functionalities? Maturity, Availability, Recoverability, Fault tolerance
 - Performance Efficiency: performances vs resources. Time behaviour, Resource utilization, Capacity
 - **Usability**: Effort needed for use and assessment of such use by users. *Appropriateness Recognizability, Learnability, Operability, User error protection, UI aesthetics, Accessibility*
 - **Security**: How well the system protects user and data from vulnerabilities. *Confidentiality, Integrity, Non-repudiation, Accountability, Authenticity*
 - Compatibility: Degree to which a product, system or component can exchange information with other products, systems or components, and/or perform its required functions while sharing the same hardware or software environment. Co-existence, Interoperability
 - **Maintainability**: to which extent the software can be modified to improve it, correct it or adapt it to changes in environment, and in requirements. *Modularity, Reusability, Analysability, Modifiability, Testability*
 - **Portability**: Can the software be transferred from one environment to another? *Adaptability, Installability, Replaceability*



Formal software inspection

- Definition from NASA-STD-8739.9: "[...] Technical evaluation process during which a product is examined with the purpose of finding and removing defects and discrepancies as early as possible in the software life cycle"
- In general, formal inspection has the following characteristics:
 - Control is made by technically competent persons
 - Product's author is actively involved
 - Involves the following persons:
 - Moderator
 - Reader
 - Recorder
 - Author



Formal software inspection

- 5 stage process:
- 1. Planification and presentation by the author
- 2. Preparation (inspectors study the product, potentially with control list(s))
- 3. Inspection
- 4. Rework
- 5. Follow-up



Control lists

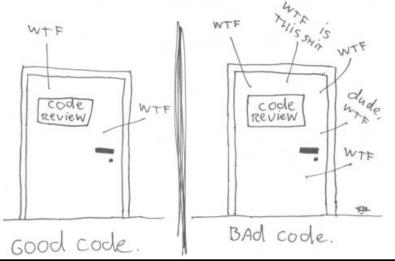
- List of (technical) elements to be verified during the actual inspection
- For instance, one list per programming language
- Errors found should belong to the control list and identified
- Needs to be concise



Elements of a software project

To conclude...

The ONLY VALID MEASUREMENT OF Code QUALITY: WTFs/minute



(c) 2008 Focus Shift/OSNews/Thom Holwerda - http://www.osnews.com/comics