Deepraj Bhosale
181105016

Date : 24-4-2021

STQA ASSIGNMENT 1

1. What is software quality? On which factors the quality of the system is defined? Explain.

Software quality is defined as a field of study and practice that describes the desirable attributes of software products. There are two main approaches to software quality: defect management and quality attributes.

ISO-9 126 (ISO, 2001) provides a hierarchical framework for quality definition, organized into quality characteristics and sub-characteristics. There are six top-level quality characteristics, with each associated with its own exclusive (non-overlapping) sub-characteristics, as summarized below:

1. ***Functionality***: A set of attributes that bear on the existence of a set of functions and their specified properties. The functions are those that satisfy stated or implied needs.The sub-characteristics include:
    - Suitability
    - Accuracy
    - Interoperability
    - Security

2. ***Reliability***: A set of attributes that bear on the capability of software to maintain its level of performance under stated conditions for a stated period of time. The sub-characteristics include:
    - Maturity
    - Fault tolerance
    - Recoverability

3. Usability: A set of attributes that bear on the effort needed for use, and on the individual assessment of such use, by a stated or implied set of users. The sub-characteristics include:
    - Understandability
    - Learnability
    - Operability

4. Efficiency: A set of attributes that bear on the relationship between the level of performance of the software and the amount of resources used, under stated conditions. The sub-characteristics include:
    - Time behavior
    - Resource behavior

5. Maintainability: A set of attributes that bear on the effort needed to make specified modifications. The sub-characteristics include:
    - Analyzability
    - Changeability
    - Stability
    - Testability

6. Portability: A set of attributes that bear on the ability of software to be transferred from one environment to another. The sub-characteristics include:
    - Adaptability
    - Installability
    - Conformance
    - Replaceability

2. What is Quality Assurance?

Software Quality Assurance (SQA) is simply a way to assure quality in the software. It is the set of activities which ensure processes, procedures as well as standards suitable for the project and implemented correctly.

Software Quality Assurance is a process which works parallel to development of a software. It focuses on improving the process of development of software so that problems can be prevented before they become a major issue. Software Quality Assurance is a kind of an Umbrella activity that is applied throughout the software process.

Software Quality Assurance have:

1. A quality management approach
2. Formal technical reviews
3. Multi testing strategy
4. Effective software engineering technology
5. Measurement and reporting mechanism

3. List and explain different types of QA activities.

Software quality assurance is composed of a variety of functions associated with two different constituencies. The software engineers who do technical work and an SQA group that has responsibility for quality assurance planning, record keeping, analysis, and reporting.

Following activities are performed by an independent SQA group:

1. **Prepares an SQA plan for a project:** The program is developed during project planning and is reviewed by all stakeholders. The plan governs quality assurance activities performed by the software engineering team and the SQA group. The plan identifies calculation to be performed, audits and reviews to be performed, standards that apply to the project, techniques for error reporting and tracking, documents to be produced by the SQA team, and amount of feedback provided to the software project team.

2. **Participates in the development of the project's software process description:** The software team selects a process for the work to be performed. The SQA group reviews the process description for compliance with organizational policy, internal software standards, externally imposed standards (e.g. ISO-9001), and other parts of the software project plan.

3. **Reviews software engineering activities to verify compliance with the defined software process:** The SQA group identifies, reports, and tracks deviations from the process and verifies that corrections have been made.

4. **Audits designated software work products to verify compliance with those defined as a part of the software process:** The SQA group reviews selected work products, identifies, documents and tracks deviations, verify that corrections have been made, and periodically reports the results of its work to the project manager.

5. **Ensures that deviations in software work and work products are documented and handled according to a documented procedure:** Deviations may be encountered in the project method, process description, applicable standards, or technical work products.

6. **Records any noncompliance and reports to senior management:** Non- compliance items are tracked until they are resolved.

4. On which areas one should focus more on at the time of education and training?

Education and training provide people-based solutions for error source elimination. Education and training of software professionals can help them control, manage, and improve the way they work. Such activities can also help ensure that they have few, if any, misconceptions related to the product and the product development.

The education and training effort for error source elimination should focus on the following areas:

- ***Product and domain specific knowledge***. If the people involved are not familiar with the product type or application domain, there is a good chance that wrong solutions will be implemented. For example, developers unfamiliar with embedded software may design software without considering its environmental constraints, thus leading to various interface and interaction problems between software and its physical surroundings.

- ***Software development knowledge and expertise*** plays an important role in developing high-quality software products. For example, lack of expertise with requirement analysis and product specification usually leads to many problems and rework in subsequent design, coding, and testing activities.

- ***Knowledge about Development methodology, technology, and tools*** also plays an important role in developing high-quality software products. For example, in an implementation of Cleanroom technology (Mills et al., 1987b), if the developers are not familiar with the key components of formal verification or statistical testing, there is little chance for producing high-quality products.

- ***Development process knowledge.*** If the project personnel do not have a good understanding of the development process involved, there is little chance that the process can be implemented correctly. For example, if the people involved in incremental software development do not know how the individual development efforts for different increments fit together, the uncoordinated development may lead to many interface or interaction problems.

5. List and briefly explain two major activities of defect reduction.

For most large software systems in use today, it is unrealistic to expect the defect prevention activities surveyed above to be 100% effective in preventing accidental fault injections. Therefore, we need effective techniques to remove as many of the injected faults as possible under project constraints.

The 2 major activities of defect reduction are:

1. **Inspection: Direct fault detection and removal**
   Software inspections are critical examinations of software artifacts by human inspectors aimed at discovering and fixing faults in the software systems. Inspection is a well-known QA alternative familiar to most experienced software quality professionals. The basic ideas of inspection are outlined below:

- Inspections are critical reading and analysis of software code or other software artifacts, such as designs, product specifications, test plans, etc.
- Inspections are typically conducted by multiple human inspectors, through some coordination process. Multiple inspection phases or sessions might be used.
- Faults are detected directly in inspection by human inspectors, either during their individual inspections or various types of group sessions.
- Identified faults need to be removed as a result of the inspection process, and their removal also needs to be verified.
- The inspection processes vary, but typically include some planning and follow-up activities in addition to the core inspection activity.
- The formality and structure of inspections may vary, from very informal reviews and walkthroughs, to fairly formal variations of Fagan inspection, to correctness inspections approaching the rigor and formality of formal methods.

2. **Testing: Failure observation and fault removal**
   Testing is one of the most important parts of QA and the most commonly performed QA activity. Testing involves the execution of software and the observation of the program behaviour or outcome. If a failure is observed, the execution record is then analyzed to locate and fix the fault(s) that caused the failure. Individual testing activities and techniques can be classified using various criteria and examined accordingly.