Software Quality Assurance

Software Engineering - I

What is Software?

According to the IEEE

Software is:

"Computer programs, procedures, and possibly associated documentation and data pertaining to the operation of a computer system".

What is Software Quality ?

According to the IEEE

- Software quality is:
  1. *The degree to which a system, component, or process meets specified requirements.*
  2. *The degree to which a system, component, or process meets customer or user needs or expectations.*

What is Software Quality?

- “Achieving high levels of user satisfaction, portability, maintainability, robustness, and fitness for use” by Dr. Barry Boehm.
- Quality means “conformance to user requirements” by Phil Crosby.
- Edwards Deming considers quality to be “striving for excellence” in reliability and functions by continuous improvement in the process of development, support by statistical analysis of the causes of failure.

What is Software Quality?

- Watts Humphrey, of the SEI, tends to speak of quality as “achieving excellent levels of fitness for use, conformance to requirements, reliability and maintainability.”
- James Martin said that software quality means being on time, within budget and meeting user needs
- Tom McCabe, the software complexity specialist, defines quality as “high level of user satisfaction and low defect levels, often associated with low complexity

What is Software Quality?

- John Musa of Bell Laboratories states that quality means combination of “low defect levels, adherence of software functions to users needs, and high reliability”
- Bill Perry, head of Quality Assurance Institute has defined quality as “high levels of user satisfaction and adherence to requirements”.

Why Quality is Important?

- Quality is critical for survival and success.
- Customers demand quality.
- Everybody seems to understand quality.
- Everybody wants quality
- Everybody has a different perception of quality.
- Essentially quality means satisfying customer.

Why Quality is Important?

- Why business should be concerned with quality:
  - Quality is competitive issue now
  - Quality is a must for survival
  - Quality gives you the global reach
  - Quality is cost effective
  - Quality helps retain customers and increase profits
  - Quality is the hallmarks of world-class business

Importance of Software Quality

- Software is a major component of computer systems
- Software defects are extremely costly in term of
  - money
  - reputation
  - loss of life

Importance of Software Quality

- Several historic disasters attributed to software
  - 1988 shooting down of Airbus 320 by the USS Vincennes cryptic and misleading output displayed by tracking software
  - 1991 patriot missile failure inaccurate calculation of time due to computer arithmetic errors.
  - London Ambulance Service Computer Aided Dispatch System – several deaths
  - On June 3, 1980, the North American Aerospace Defense Command (NORAD) reported that the U.S. was under missile attack.

Software Quality Factors

- Correctness
  - accuracy, completeness of required output
  - upto-dateness, availability of the information
- Reliability
  - Minimum failure rate
- Efficiency

  resources needed to perform software function
- Integrity
  - software system security, access rights
- Usability
  - ability to learn, perform required task

Software Quality Factors

- Maintainability
  - effort to identify and fix software failures (modularity, documentation, etc)
- Flexibility
  - degree of adaptability (to new customers, tasks, etc)
- Testability
  - support for testing (e.g. log files, automatic diagnostics, etc)
- Portability
  - adaptation to other environments (hardware, software)
- Reusability
  - use of software components for other projects
- Interoperability
  - ability to interface with other components/systems

Elements of Quality Definition

Quality factors	Predictable	Measurable
Defect Level	Yes	Yes
Defect Origins	Yes	Yes
Defect Severity	Yes	Yes
Defect Removal Efficiency	Yes	Yes
Product Complexity	Yes	Yes
Project Reliability	Yes	Yes
Project Maintainability	Yes	Yes
Project schedules	Yes	Yes
Project budgets	Yes	Yes
Portability	Yes	Yes
Conformance to requirements	No	Yes
User Satisfaction	No	Yes
Fitness for Use	No	Yes
Robustness	No	No

Software Quality Challenges

- The measures for quality differ from project to project and organization to organization
  - Quality measures used for small systems may not be appropriate for the large ones.
- Criteria for quality vary as a function of the specific characteristics of the project, the needs of the users and stakeholders, and the application requirements of the system and software.
  - Criteria for quality applied to real-time applications are not always relevant when dealing with systems that are not real-time.

What is Software Quality Assurance?

According to the IEEE

Software quality assurance is:

1. *A planned and systematic pattern of all actions necessary to provide adequate confidence that an item or product conforms to established technical requirements.*
2. *A set of activities designed to evaluate the process by which the products are developed or manufactured. Contrast with: quality control.*

Objectives of SQA

Development:

- Assuring an acceptable level of confidence that the software will conform to functional technical requirements.
- Assuring an acceptable level of confidence that the software will conform to managerial scheduling and budgetary requirements.
- Initiation and management of activities for the improvement and greater efficiency of software development and SQA activities.

Maintenance:

- Assuring an acceptable level of confidence that the software maintenance activities will conform to the functional technical requirements.
- Assuring an acceptable level of confidence that the software maintenance activities will conform to managerial scheduling and budgetary requirements.
- Initiate and manage activities to improve and increase the efficiency of software maintenance and SQA activities.

Three General Principles of QA

- Know what you are doing
- Know what you should be doing
- Know how to measure the difference

Three General Principles of QA

Know what you are doing

- understand **what** is being built**, how** it is being built and what it currently **does.**
- suppose a software development process with
  - management structure (milestones, scheduling)
  - reporting policies
  - tracking

Three General Principles of QA

- Know what you should be doing
  - having explicit **requirements** and **specifications**
  - suppose a software development process with
    - requirements analysis,
    - acceptance tests,
    - frequent user feedback

Three General Principles of QA

- Know how to measure the difference
  - having explicit measures comparing what is being done from what should be done
  - four complementary methods:
    - **formal methods** – verify mathematically specified properties
    - **testing** – explicit input to exercise software and check for expected output
    - **inspections** – human examination of requirements, design, code, ... based on checklists
    - **metrics** – measures a known set of properties related to quality.

Software Quality Challenges

- Complex Software requires different monitoring procedures than trivial applications.
- Quality criteria vary dramatically depending on the phase of the project at which the evaluation takes place
- The measures of the quality must be specific to the project being evaluated and must assess the effectiveness of the entire development process, not just individual segments.

Software Quality Challenges

- Quality cannot be directly checked in the product; it must planned right from the beginning.
- Quality goals must be clearly defined, effectively monitored, and rigorously enforced.
- The project must focus on the quality issues of the project from the beginning, ensuring that quality criteria are consistent with defined requirements.
- Quality must be planed into the project structure, constantly evaluated, and corrections applied when deficiencies are identified.

Changing View of Quality

Past	Present
Quality is the responsibility of blue collar workers and direct labor employees working on the product	Quality is everyone's responsibility, including, white-collar workers, the indirect labor force and the overhead staff
Quality defects should be hidden from the customers and management	Defects should be highlighted and brought to the surface for corrective actions
Quality problems lead to blame, faulty justification and excuses	Quality problems lead to cooperative solutions
Corrections-to-quality problems should be accompanied with minimum documentation	Documentation is essential for "lessons learnt" so the mistakes are not repeated.

Changing View of Quality

Past	Present
Increased quality will increase project costs	Improved quality saves money and increase business
Quality is internally focused	Quality is customer focused
Quality will not occur without close supervision of people	People want to produce quality products
Quality occurs during project execution	Quality occurs at project initiation and must be planned for within the project

Quality Control v/s Quality Assurance

- Quality means meeting requirements and meeting customer needs, which means a defect-free product from both the producer's and the customer's viewpoint.
- Both quality control and quality assurance are used to make quality happen.
- Quality is an attribute of a product. A product is something produced, such as a requirement document, test data, source code etc.

Quality Control v/s Quality Assurance

Quality Assurance	Quality Control
Quality Assurance (QA) is the set of activities (including facilitation, training, measurement and analysis) needed to provide adequate confidence that processes are established and continuously improved in order to produce products or services that conform to requirements and are fit for use.	Quality Control (QC) is defined as the processes and methods used to compare product quality to requirements and applicable standards, and the action taken when a nonconformance is detected.

Quality Control v/s Quality Assurance

Quality Assurance	Quality Control
QA is an activity that establishes and evaluates the processes that produce the products. If there is no process, there is no role for QA.	QC is an activity that verifies whether or not the product produced meets standards.

Quality Control v/s Quality Assurance

Quality Assurance	Quality Control
QA helps establish processes	QC relates to a specific product or service
QA sets up measurement programs to evaluate processes	QC verified whether particular attributes exist, or do not exist, in a specific product or service
QA identifies weakness in processes and improves them	QC identifies defect for the primary purpose of correcting defects.

Quality Control v/s Quality Assurance

Quality Assurance	Quality Control
QA is a management responsibility, frequently performed by a staff function	QC is the responsibility of the worker
QA evaluates whether or not quality control is working for the primary purpose of determining whether or not there is weakness in the process	

Quality Control v/s Quality Assurance

Quality Assurance	Quality Control
Defining Processes	Walkthrough
Quality Audit	Testing
Selection of Tools	Inspection
Training	Checkpoint review

Quality Assurance at different phase of SDLC

- **Requirements Analysis Phase:**
- Three major activities that foster quality
  - Measurement of process attributes
  - Verification and Validation
  - Management
- Managing quality in the analysis stage is a challenge.

Quality Assurance at different phase of SDLC

- Good Quality Requirements
  - They are precise, with no room for misinterpretation by users or implementers
  - They specify just what the system must do, not how to do it. They avoid specifying implementation details
  - They show conceptual integrity, building on a simple set of facilities that interact well with each other

Quality Assurance at different phase of SDLC

- Major management deficiencies in most software development projects:
  - Incorrect schedules
  - Incorrect cost estimates
  - Inadequate project accountability procedures
  - Inadequate quality assurance procedures
  - Imprecise goals and success criteria

Quality Assurance at different phase of SDLC

- **Design phase:**
- A lack of quality in the design process can invalidate good requirements specification and can make correct implementation impossible.
- Industry practice shows that use of checklist during design helps improve design quality.