Software Quality Assurance

COMPUTER SCIENCE DEPERTMENT (M-TECH)
GURU NANAK DEV ENGINEERING COLLEGE (AN AUTONOMOUS COLLEGE UNDER UGC ACT), LUDHIANA.

What is Software Quality?

According to the IEEE, Software quality is:

- The degree to which a system, component, or process meets specified requirements.
- The degree to which a system, component, or process meets customer or user needs or expectations.

Importance of Software Quality

Software is a major component of computer systems (a bout 80% of the cost) – used for

– communication (e.g. phone system, email system)

– health monitoring,

– transportation (e.g. automobile, aeronautics),

– economic exchanges (e.g. e-commerce),

– entertainment , etc.

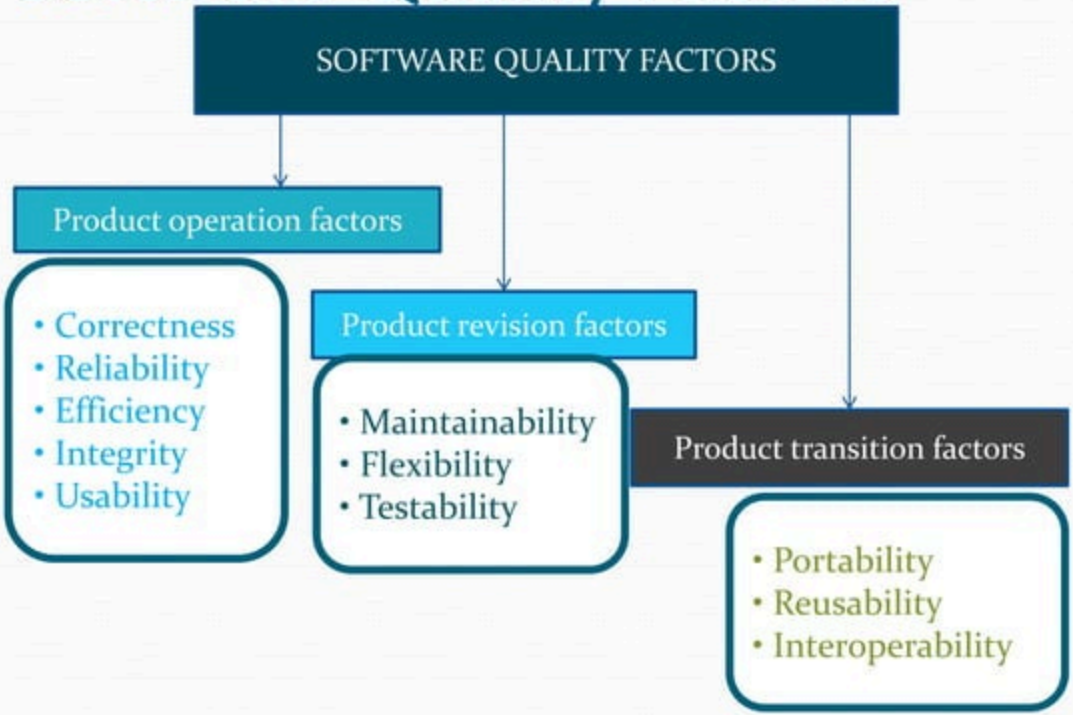
- Software defects are extremely costly in term of

– money

– reputation

– loss of life

Software Quality Factors



Software Quality Factors

- Correctness
  - accuracy, completeness of required output
  - up-to-dateness, availability of the information
- Reliability
  - maximum 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)

Software Quality Factors

- Portability
  –
adaptation to other environments (hardware, software)
- Reusability
  – use of software components for other projects
- Interoperability
  – ability to interface with other components/systems

What is Software Quality Assurance?

According to the IEEE , Software quality assurance is:

- A planned and systematic pattern of all actions necessary to provide adequate confidence that an item or product conforms to established technical requirements.
- A set of activities designed to evaluate the process by which the products are developed or manufactured.

What is Software Quality Assurance?

A systematic, planned set of actions necessary to provide adequate confidence that the software development process or the maintenance process of a software system product conforms to established functional technical as well as with the managerial requirements of keeping the schedule and operating within the budgetary confines.

Objectives of SQA in development

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

Objectives of SQA in maintenance

(1) Assuring an acceptable level of confidence that the software maintenance activities will conform to the functional technical requirements.
(2) Assuring an acceptable level of confidence that the software maintenance activities will conform to managerial scheduling and budgetary requirements.
(3) 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 Assurance

SQA: Comprehensive life-cycle approach concerned with every aspect of the software product development process

- Includes
  - comprehensive set of quality objectives
  - measurable quality attributes (quality metrics) to assess progress toward the objectives
  -
  quantitative certification targets for all component of the software development processes.
- Takes into account:
  - customer product requirements,
  - customer quality requirements, and
  - corporate quality requirements.

SQA

SQA includes

- Verification

– are we building the product right ?

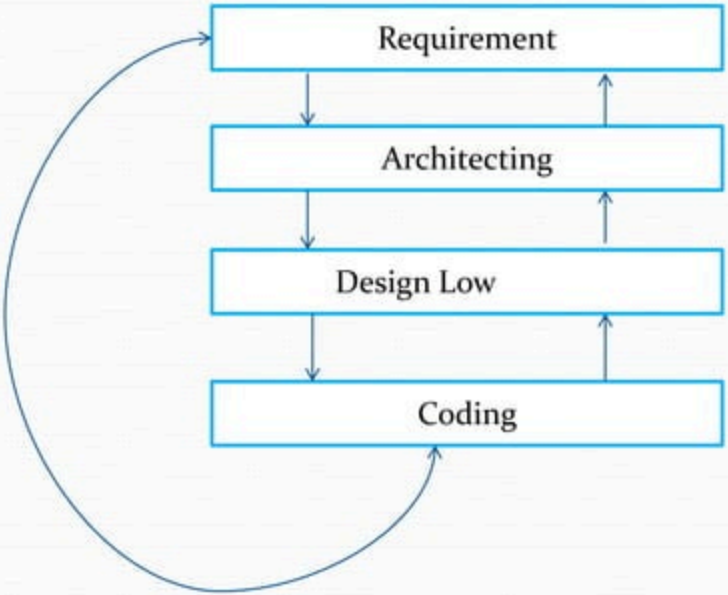
– performed at the end of a phase to ensure that requirements established during previous phase have been met

- Validation

– are we building the right product ?

– performed at the end of the development process to ensure compliance with product requirements

SQA



SQA

SQA includes

- Defect Prevention
  - prevents defects from occurring in the first place
  - Activities: training, planning, and simulation
- Defects detection
  - finds defects in a software artifact
  - Activities: inspections, testing or measuring
- Defects removal
  - isolation, correction, verification of fixes
  - Activities: fault isolation, fault analysis, regression testing

SQA

Typical activities of a SQA process

- Requirements validation.
- Design verification.
- Static code checking (inspection/reviews).
- Dynamic testing.
- Process engineering and standards.
- Metrics and continuous improvement

Levels of SQA

-Testing

-Validation

- Certification

Levels of SQA

Testing

- The common view is to eliminate errors.
- All that can be done by putting the system through a "Fail Test" cycle to determine what will make it fail.
- A successful test finds error. All strategies implemented in system testing can be used here.

SQA levels

Validation

- Checks validity of system in both simulated and live environment as follows-

1) Software undergoes through alpha testing in which errors and failures based on simulated user requirements are verified and studied.

2) The modified software is then subjected to beta testing in actual user's site or live environment.

3) The system is then used regularly with live transactions.

4) After a scheduled time failures and errors are documented and final corrections and enhancements are made before package is released for use.

SQA levels

Certification

-A package that is certified goes through a team of specialists who test, review and determine how well it meets the user's requirement.

-Certification is actually issued after the package passes the test.

-Certification, however, does not assure that it is the best package to adopt; it only attests that it will perform what the user claims.

Technique used in SQA

Audit

- Used to:
  - Review Management
  - Technical Processes
  - Assurance Processes
- Provide an indication of the Quality and Status of the Software Product
- SQA Product is an Audit Report to Management consisting of Findings and Recommendations to bring development up to par with Standards and Procedures

Tools

Many different Tools on the market today for SQA

Each Tool works differently but accomplishes same goal:

- Help Improve the Development Process of a Computer System

- Look at two different Tools:
  - Configuration and Problem Management
    - Borland StarTeam and Lucent Technologies Sablime
  - Testing Software
    - Programming Research QA C++ and Parasoft Jtest

Benefits of SQA in projects

- Essential to the Development Process
- Without SQA, many Development Groups would not reach their release goals/deadlines on time
- Spend too much time Revisiting Requirements, Design, Code, and Documentation without SQA
- Lowers time spent on mundane areas and lets more time be focused on important areas
- Decreases the time from Development to Deployment
- Can help catch errors before they are too costly to fix
- Standards can be used across many different Projects