Software Quality Assurance

CHAPTER 1

INTRODUCTION

DR. JAMIL S. ALAGHA

Outline

Quality

Quality Management

Quality Assurance

Software

Software quality assurance

Software Testing

Mission

The mission of a software company is to develop high-quality innovative products and services at a competitive price to its customers, and to do so ahead of its competitors.

Quality

Quality, simplistically, “meeting the requirements of the customer”.

That means:

"delivery of reliable product or service within an agreed span of time under agreed conditions, at agreed costs, and with necessary aftercare"

Quality

Traditionally, efforts to improve quality have centered around the end of the product development cycle by emphasizing the detection and correction of defects.

Quality

On the contrary, the new approach to enhancing quality encompasses all phases of a product development process—from a requirements analysis to the final delivery of the product to the customer. Every step in the development process must be performed to the highest possible standard.

Quality

An effective quality process must focus on :

- Paying much attention to customer’s requirements
- Making efforts to continuously improve quality
- Integrating measurement processes with product design and development
- Pushing the quality concept down to the lowest level of the organization
- Eliminating waste through continuous improvement

Quality

Importance of Quality

1) Producing quality products has been identified as a key factor in the long term success (i.e. profitability) of organization

Quality

2) Quality is not just a marketing and perception issue, it is a moral and legal requirement - we have a professional responsibility associated with the software we create

- Professionals must be able to demonstrate, and to have confidence, that they are using "best practices"

Quality is…..

Invisible when **GOOD**

Impossible to ignore when **BAD**

Quality

In principle, three levels of organization of these activities can be distinguished. From the top down these levels are:

1. Quality Management (QM)

2.Quality Assurance (QA)

3. Quality Control (QC)

Outline

Quality

Quality Management

Quality Assurance

Software

Software quality assurance

Software Testing

Quality Management

Quality Management may be defined as :

an act of performing all the activities and tasks which are needed to maintain a desired level of excellence. This includes creating and implementing quality planning and assurance, as well as quality control and quality improvement.

Cost of effective quality management

The total cost of effective quality management is the sum of four component costs: prevention, inspection, internal failure, and external failure.

Cost of effective quality management

Prevention costs consist of actions taken to prevent defects from occurring in the first place.

prevention cost:

- quality planning

- formal technical reviews

- testing equipment

- training

Cost of effective quality management

Inspection costs consist of measuring, evaluating, and auditing products or services for conformance to standards and specifications.

Inspection cost:

- in-process and inter-process inspection

- equipment calibration and maintenance

- testing

Cost of effective quality management

Internal failure costs are those incurred in fixing defective products before they are delivered.

Internal failure costs

- rework, repair, and failure mode analysis

Cost of effective quality management

External failure costs consist of the costs of defects discovered after the product has been released.

External failure costs

- complaint resolution

- product return and replacement

- help line support

- warranty work

Cost of effective quality management

The greatest payback is with prevention. Increasing the emphasis on prevention costs reduces the number of defects that go to the customer undetected, improves product quality, and reduces the cost of production and maintenance.

External failure can be devastating because they may damage the organization’s reputation or result in the loss of future sales.

Outline

Quality

Quality Management

Quality Assurance

Software

Software quality assurance

Software Testing

Quality assurance

In developing products and services, quality assurance is any systematic process of **checking** to see whether a product or service being developed is meeting specified requirements.

Quality assurance systems emphasize catching defects before they get into the final product

Quality assurance

The word *assurance* means that if the processes are followed, management can be assured of product quality.

Successful quality assurance managers know how to make people quality aware and to make them recognize the benefits of quality to themselves and to the organization.

Quality assurance

Many companies have a separate department devoted to quality assurance. A quality assurance system is said to increase customer confidence and a company's credibility, to improve work processes and efficiency, and to enable a company to better compete with others.

Quality Control

Quality control is concerned with activities to ensure that the end product satisfies the functional and non-functional requirements and is fit for purpose.

It includes inspections and testing to verify that the deliverables produced satisfy their requirements.

Inspections typically consist of a formal review of a deliverable by independent experts, and the objective is to identify defects within the work product, and to provide confidence in its correctness.

Testing means Validation and verification of the system.

Outline

Quality

Quality Management

Quality Assurance

Software

Software quality assurance

Software Testing

Software

Software – IEEE definition

Software is:

1) Computer programs, (code)
2) procedures, and
3) possibly associated documentation
4) And data pertaining to the operation of a computer system. (Data necessary for operating the software system)

Software

All four components are needed in order to assure the quality of the software development process and the coming years of maintenance services for the following reasons:

1) Computer programs (the “code”) are needed because, obviously, they activate the computer to perform the required applications.

Software

2) Procedures are required, to define the order and schedule in which the programs are performed, the method employed, and the person responsible for performing the activities that are necessary for applying the software.

Software

3) Various types of documentation are needed for developers, users and maintenance personnel. The development documentation (the requirements report, design reports, program descriptions, etc.) allows efficient cooperation and coordination among development team members and efficient reviews and inspections of the design and programming products.

The user's documentation (the "user's manual", etc.) provides a description of the available applications and the appropriate method for their use.

Software

The maintenance documentation (the “programmer’s software manual”, etc.) provides the maintenance team with all the required information about the code, and the structure and tasks of each software module. This information is used when trying to locate causes of software failures (“bugs”) or to change or add to existing software

Software

4) Data including parameters, codes and name lists that adapt the software to the needs of the specific user are necessary for operating the software.

Another type of essential data is the standard test data, used to ascertain that no undesirable changes in the code or software data have occurred, and what kind of software malfunctioning can be expected.

Outline

Quality

Quality Management

Quality Assurance

Software

Software quality assurance

Software Testing

Software Quality Assurance

Definition

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.

Software Quality Assurance

Monitoring *processes* and *products* throughout the software development lifecycle to ensure the quality of the delivered product(s)

Software Quality Assurance

Monitoring the processes

- Provides management with objective feedback regarding process compliance to approved plans, procedures, standards, and analyses

Monitoring the products

- Focus on the quality of product within each phase of the software lifecycle
  - e.g., requirements, test plan, architecture, etc.
- Objective: identify and remove defects throughout the lifecycle, as early as possible

Software Quality Assurance

Product and software quality does not happen by accident, and is not something that can be added on after the fact.

To achieve quality, we must plan for it from the beginning, and continuously monitor it day to day

Software Quality Assurance

Software quality assurance is a strategy for risk management. It exists because software quality is typically costly and should be incorporated into the formal risk management of a project. Some examples of poor software quality include the following:

Software Quality Assurance

Delivered software frequently fails.

Consequences of system failure are unacceptable, from financial to life threatening scenarios.

Systems are often not available for their intended purpose.

System enhancements are often very costly.

Costs of detecting and removing defects are excessive.

Software Quality Assurance

Three General Principles of SQA

Know what you are doing

Know what you should be doing

Know how to measure the difference

Software Quality Assurance

QA Principle 1: Know What You Are Doing

In the context of software quality, this means continuously understanding what it is you are building, how you are building it and what it currently does

This requires organization, including having a management

structure, reporting policies, regular meetings and reviews, frequent test runs, and so on

We normally address this by following a software process

with regular milestones, planning, scheduling, reporting

and tracking procedures

Software Quality Assurance

QA Principle 2: Know What You Should be Doing

In the context of software quality, this means having explicit

requirements and specifications

These must be continuously updated and tracked as part of the software development and evolution cycle

We normally address this by requirements and use-case

analysis, explicit acceptance tests with expected results,

explicit prototypes, frequent user feedback

Particular procedures and methods for this are usually part

of our software process

Software Quality Assurance

QA Principle 3: Know How to Measure the Difference

In the context of software quality, this means having explicit measures comparing what we are doing to what we should be doing Achieved using four complementary methods

- Formal Methods
- Testing
- Inspection
- Metrics

Formal Methods

- Formal methods include formal verification (proofs of correctness),

  abstract interpretation simulated execution using a mathematical model to keep track of state transitions and other mathematical methods

- In practice, formal methods are used directly in software quality assurance in only a small (but important) fraction of systems

- Primarily safety critical systems such as onboard flight control systems, nuclear reactor control systems, embedded systems such as automobile braking systems and medical equipment, and so on

Formal Methods

Use of formal methods requires mathematically sophisticated programmers, and is necessarily a slow and careful process, and very expensive

For these reasons, the vast majority (over 95%) of software quality assurance uses testing, inspection and metrics instead

Testing

Testing includes a wide range of methods based on the idea of running the software through a set of example inputs or situations and validating the results

Includes methods based on requirements (acceptance testing), specification and design (functionality and interface testing), history (regression testing), code structure (path testing), and many more

Inspection

Inspection includes methods based on a human or automated review of the software artifacts

Includes methods based on requirements reviews, design reviews, scheduling and planning reviews, code walkthroughs, and so on

- Helps discover potential problems before they arise in practice

Metrics

Software metrics includes methods based on using tools to count the use of features or structures in the code or other software artifacts, and compare them to standards

Includes methods based on code size (number of source lines), code complexity (number of parameters, decisions, function points, modules or methods), structural complexity (number or depth of calls or transactions), design complexity, and so on

- Helps expose anomalous or undesirable properties that may reduce reliability and maintainability

The uniqueness of software quality assurance

(1) **Product complexity**. **Product complexity can be measured by the number** of operational modes the product permits. An industrial product, even an advanced machine, does not allow for more than a few thousand modes of operation, created by the combinations of its different machine settings.

Looking at a typical software package one can find millions of software operation possibilities. Assuring that the multitude of operational possibilities is correctly defined and developed is a major challenge to the software industry.

The uniqueness of software quality assurance

(2) **Product visibility. Whereas the industrial products are visible, software** products are invisible. Most of the defects in an industrial product can be detected during the manufacturing process. Moreover the absence of a part in an industrial product is, as a rule, highly visible (imagine a door missing from your new car). However, defects in software products (whether stored on diskettes or CDs) are invisible, as is the fact that parts of a software package may be absent from the beginning.

The uniqueness of software quality assurance

(3) Nature of development and production process

Opportunities to detect defects arise in only one phase namely product development

a) Product development. During this phase, efforts of the development teams and software quality assurance professionals are directed toward detecting inherent product defects. At the end of this phase an approved prototype, ready for reproduction, becomes available.

The uniqueness of software quality assurance

b) Product production planning. This phase is not required for the software production process, as the manufacturing of software copies and printing of software manuals are conducted automatically. This applies to any software product, whether the number of copies is small, as in custom-made software, or large, as in software packages sold to the general public.

The uniqueness of software quality assurance

c) Manufacturing. **As mentioned previously, the manufacturing of** software is limited to copying the product and printing copies of the software manuals. Consequently, expectations for detecting defects are quite limited during this phase

Opportunities to detect defects arise in only one phase namely product development

SOFTWARE QUALITY RISK ALL INDUSTRIES

1. Software is blamed for more major business problems than any other man-made product.

2. Poor software quality has become one of the most expensive topics in human history: > $150 billion per year in U.S.; > $500 billion per year world wide.

3. Projects cancelled due to poor quality >15% more costly than successful projects of the same size and type.

4. Software executives, managers, and technical personnel are regarded by many CEO’s as a painful necessity rather than top professionals.

5. Improving software quality is a key topic for all industries.

Outline

Quality

Quality Management

Quality Assurance

Software

Software quality assurance

Software Testing

Software Testing

Definition:

Software testing is the process of executing a software system to determine whether it matches its specification and executes in its intended environment.

Software Testing

Goals of testing

Executing a program with the intent of finding an *error*.

To check if the system meets the requirements and be executed successfully in the Intended environment.

To check if the system is “ Fit for purpose”.

To check if the system does what it is expected to do.

Software Testing

“Program testing can be a very effective way to show the presence of bugs, but it is hopelessly inadequate for showing their absence”

Software Testing

Testing is a process of executing software with the intent of finding errors

- Good testing has a high probability of finding as-yet-undiscovered errors
- Successful testing discovers unknown errors
- If did not find any errors, need to ask whether our testing approach is good

Software Testing

Software testing should establish confidence that the software is fit for purpose

This does NOT mean completely free of defects

Rather, it must be good enough for its intended use and the type of use will determine the degree of confidence that is needed

OBJECTIVES OF TESTING

Different stakeholders view a test process from different perspectives as explained below:

1) **It does work: While implementing a program unit, the programmer may** want to test whether or not the unit works in normal circumstances. The programmer gets much confidence if the unit works to his or her satisfaction. The same idea applies to an entire system as well—once a system has been integrated, the developers may want to test whether or not the system performs the basic functions. Here, for the psychological reason, the objective of testing is to show that the system works, rather than it does not work.

OBJECTIVES OF TESTING

2) **It does not work:** **Once the programmer (or the development team) is** satisfied that a unit (or the system) works to a certain degree, more tests are conducted with the objective of finding faults in the unit (or the system). Here, the idea is to try to make the unit (or the system) fail.

OBJECTIVES OF TESTING

3) **Reduce the risk of failure: Most of the complex software systems contain** faults, which cause the system to fail from time to time. This concept of “failing from time to time” gives rise to the notion of *failure rate. As* faults are discovered and fixed while performing more and more tests, the failure rate of a system generally decreases. Thus, a higher level objective of performing tests is to bring down the risk of failing to an acceptable level.

OBJECTIVES OF TESTING

4) **Reduce the cost of testing: The different kinds of costs associated with a** test process include the cost of designing, maintaining, and executing test cases, the cost of analyzing the result of executing each test case, the cost of documenting the test cases, and the cost of actually executing the system and documenting it.

Software Testing

Ultimate goal for software testing

Quality Assurance