5. **Support Multiple Data Combination :** WinRunner has an ability to use numerous data combinations for one test. The DataDriver Wizard has been designed for automatic processing of large amounts of data.
6. **Multiple Verification :** It can offer checkpoints for text, URL, GUI and databases, and this will give the testers the ability to compare the expected outcomes with the outcomes that occur. In addition to this, it is able to find specific problems with various objects that are GUI based.

6.16.2 Load Runner

HP Load runner is a software testing tool from Hewlett-Packard. It is used to test applications, measuring system behaviour and performance under load. HP Load runner can simulate thousands of users concurrently using application software, recording and later analysing the performance of key components of the application.

Features

1. It has excellent **monitoring and analysis interface** where tester can see reports in easy to understand colored charts and graphics.
2. It uses C as a default programming language. However, it also **supports** other languages like **Java and Visual Basic.**
3. **No need to install** it on the server under test. It uses native monitors.
4. It has a support for most of the **commonly used protocols.**
5. It has **GUI generated scripts** which can be modified as per the requirements.
6. This tool can quickly point out the effect of the wide area network (WAN) on application **reliability, performance,** and **response time.**

Review Questions

1. *Explain the concept of automated testing tool along with advantages and disadvantages of it.*
2. *Write a note on - 1) WinRunner 2) LoadRunner.*

❑❑❑

7 Quality Assurance and Management

Syllabus

Quality concepts and software quality assurance, Software reviews (Formal technical reviews), Software reliability, The quality standards : ISO 9000, CMM, Six sigma for SE, SQA plan.

Contents

7.1 Quality Concepts

GTU : Summer-2019, Marks 3

There are many definitions of software quality. In simple words, the software quality means 'how well the software works'. Furthermore we can also state that controlling the variation or differences is the **key** to high quality software product. Let us see ***"what is software quality ?"***

7.1.1 Quality

Software quality can be defined as "the conformance to explicitly stated functional and performance requirements, explicitly documented development standards and implicit characteristics that are expected of all professionally developed software".

- There are two kinds of quality

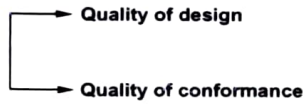


Fig. 7.1.1

Quality of design is the characteristics of the item which is specified for the designer. For example, if a temperature control system is designed, then it should display the temperature with maximum limit of 100 degree centigrade. This is what the basic characteristic of that system is, and at the time of design of the product this issue must be focused.

Quality of conformance is the degree to which the design specifications are followed during manufacturing. If the degree of conformance is **more** then it indicates **higher quality**.

Thus in software development process **quality of design** is concerned towards requirements, specifications and design of the system and **quality of conformance** is concerned with implementation.

- Along with quality of design and quality of conformance, **customer's satisfaction** is very important factor in any software product.
- According to **Robert Glass** - an authority in software field,

User satisfaction = Compliant product + Good quality + Delivery within budget.

7.1.2 Quality Control

- Quality control is a process in which activities are conducted in order to maintain the quality of product. These activities are series of inspections, reviews and tests

used throughout the software process. These activities ensure whether each work product is satisfying the requirements imposed on it.

- While applying the quality control there should be a **feedback loop** to the process which generates the work product. With the help of such feedback we can tune the process if it does not satisfy the requirements. The feedback loop helps in minimizing the defects in the software product.
- The quality control activities can be fully automated or it can be completely manual or it can be a combination of automated tools and manual procedures.

7.1.3 Quality Assurance

Definition of quality assurance : *It is planned and systematic pattern of activities necessary to provide a high degree of confidence in the quality of a product. It provides quality assessment of the quality control activities and determines the validity of the data or procedures for determining quality.*

- The quality assurance consists of set of reporting and auditing functions.
- These functions are useful for assessing and controlling the effectiveness and completeness of quality control activities.
- The goal of quality assurance is to ensure the management of data which is important for product quality.

Compare Quality Control with Quality assurance

Sr. No.	Quality Control	Quality Assurance
1.	This is an activity with the primary goal as to prevent the defects.	This is an activity with the primary goal as to identify and correct the defects.
2.	This process is intended to provide the assurance that the quality request will be achieved.	This process is intended to focus on quality being requested.
3.	This method is to manage the quality verification.	This method is for quality validation.
4.	It does not involve executing of the program.	During this method the program is executed.
5.	It is a preventive technique.	It is a corrective technique.
6.	It is a proactive measure	It is a reactive measure.
7.	It involves the full software development life cycle.	It involves testing phase of software development life cycle.
8.	It's main goal is to prevent defects in the system. It is less time consuming activity.	It's main goal is to correct the defects in the system. Hence it is more time consuming activity.

7.1.4 Cost of Quality

The cost of quality can be defined as the total cost required to obtain the quality in the product and to conduct the quality related activities.

The cost of quality has various components such as

1. Prevention cost - This is the cost of quality required for conducting quality planning, formal technical reviews, test equipments and training.
2. Appraisal cost - This is the cost of quality required for gaining the insight into the product. It includes the cost required for in-process and inter process inspection, maintenance and testing.
3. Failure cost - Failure cost means the cost required to remove the defects in the software product before delivering it to customer. There are two types of failure costs.
   a. *Internal failure cost* - Internal failure is nothing but the cost of defects occurred in the product before delivering it to customer e.g. repair in networking, repairing of communication network.
   b. *External failure cost* - External failure is the cost of defects occurred in the product after delivering it to the customer e.g. product repair/replace, complaint processing, warranty work.

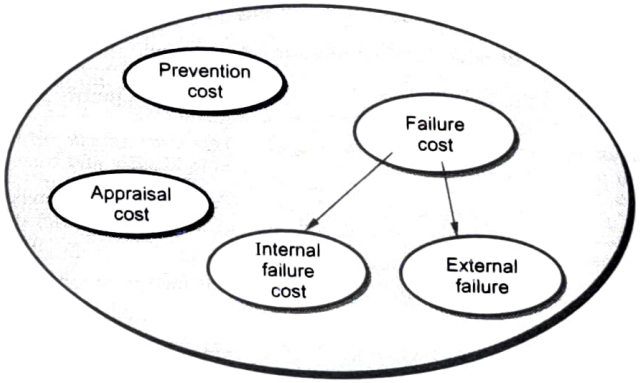


Fig. 7.1.2 Cost of quality

Review Question

1. *Compare quality control with quality assurance.* **GTU : Summer-2019, Marks 3**

7.2 Software Quality Assurance (SQA)

GTU : Winter-2011, 2013, 2018, Summer-2012, 2014, 2016, 2018, Marks 7

We have already discussed the definition of software quality in section 7.1.1 which says that quality is the conformance to functional and non functional requirements of the product.

There are three main reasons for ***why software quality gets failed ?***

1. Software **requirements** must be well understood before the software development process begins.
2. Similar to **explicit requirements** it is also essential to understand the implicit requirements of the software. If the software confirms the explicit requirements but not satisfying the implicit requirements then surely quality of software being developed is poor.
3. The set of **development criteria** has to be decided in order to specify the standards of the product. This will ultimately help the software engineer during development. If such a criteria is not been fixed then definitely the software product will lack the quality. Software quality assurance is the process in which conformance to the requirements of the product is made.

7.2.1 Software Quality Assurance (SQA) Activities

Let us list out the SQA activities conducted by SQA group.

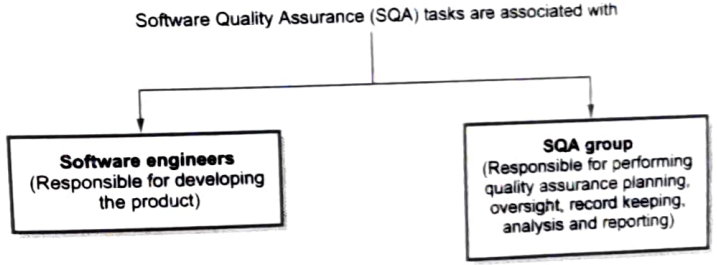


Fig. 7.2.1

1. Create a SQA plan.

A SQA plan is developed while planning the project. Quality assurance activities are conducted that are indicated in this plan. This plan basically

- Identifies evaluations to be performed.
- Audits and reviews to be performed, standards that should be adopted for the project.
- Procedures for error reporting and tracking.

- It also specifies documents to be produced by SQA group.
- Amount of feedback provided to the software project team.

2. Participates in description of software process.

The process selected by the software team is reviewed by the SQA group. This review is for

- Process description to ensure that it follows the organizational policy.
- Internal software standards.
- Some standards that are adopted by the organization.

3. Reviews software engineering activities.

The SQA group identifies and documents the processes. The group also verifies the correctness of software process.

4. Authenticate designated software work products.

The SQA group performs following tasks -

- Reviews selected work product
- Identifies the process
- Documents them
- Tracks deviations
- Verifies the correctness made in the processes
- Regular reporting of results of its work to the project manager.

5. Ensure the deviations in software work. Document work products

The deviations in software work are identified from project plan. These processes are identified and handled according to documented procedure.

6. Identify any noncompliance and reports to senior management.

Non compliance items are identified and pursued until they get resolved. The periodic reporting about it is done to project manager.

Review Questions

1. *List software related activities.* **GTU : Winter-2011, Marks 7**
2. *Define quality for software. List and explain SQA activities.* **GTU : Summer-2012, 2018, Winter-2018, Marks 7**
3. *Explain importance of SQA.* **GTU : Winter-2013, Marks 7**
4. *Explain software quality assurance techniques.* **GTU : Summer-2014, Marks 7**
5. *What is the importance of SQA ? Explain the SQA activities.* **GTU : Summer-2016, Marks 7**

7.3 Software Reviews

GTU : Winter-2018, 2019, Summer-2019, Marks 3

Software reviews are filter to software engineering process. Such reviews are applied at various points during software development life cycle. The objective of software reviews is to uncover errors and defects that can be removed. The software reviews are conducted for **following reasons -**

Reasons for Reviews

- Point out needed improvements in the product of a single person or team.
- Confirm those parts of the product in which improvement is not desired.
- Achieve technical work of more uniform quality than can be achieved without reviews, in order to make technical work more manageable.

There are three different ways by which software review can be conducted.

1. **Informal meeting** an informal meeting can be conducted outside the working environment and an informal discussion of technical issues can be held.
2. **Formal presentations** can be conducted for customer, management and technical staff.
3. **Formal Technical Reviews (FTR)** sometimes called as **walkthrough** or an inspection is the most effective way of software review. This helps a lot for uncovering the software errors and to improve the quality of software.

Benefit of formal technical reviews or walkthrough is that errors can be discovered in early stage before they become defects in the next release of software.

7.3.1 Formal Technical Reviews (FTR)

Formal Technical Review is a **software quality assurance activity** performed by software engineer.

Objectives of FTR

1. FTR is useful to **uncover errors** in logic, function and implementation for any representation of the software.
2. The purpose of FTR is to ensure that software **meets specified requirements.**
3. It also ensures that the software is represented according **to predefined standards.**
4. It helps to review the **uniformity** in software development process.
5. It makes the project more **manageable.**

Besides the above mentioned objectives, the purpose of FTR is to enable junior engineers to observe the analysis, design, coding and testing approaches more closely.

Each FTR is conducted as a **meeting** and is considered successful only if it is properly planned, controlled and attended.

7.3.1.1 The Review Meeting

- Every review meeting should be conducted by considering the following constraints -
  - **Involvement of people** - Between 3 and 5 people should be involved in the review.
  - **Advance preparation** - Advance preparation should occur but it should be very short i.e. at the most 2 hours of work for each person can be spent in this preparation.
  - **Short duration** - The duration of the review meeting should be less than 2 hours.
- Rather than attempting to review the entire design, **walkthroughs** are conducted for modules or for small groups of modules.
- The **focus** of the FTR is on a **work product** (a software component to be reviewed).
- The review meeting is attended by the review leader, all reviewers and the producer.
- The *review leader* is responsible for evaluating the product for its readiness. The copies of product material is then distributed to reviewers.
- The *producer* organizes a "walkthrough" the product, explaining the material, while the *reviewers* raise issues based on their advance preparation.
- One of the reviewers becomes *recorder* who records all the important issues raised during the review. When errors are discovered, the recorder notes each.
- At the end of the review, the attendees decide whether to accept the product or not, with or without modifications.

7.3.1.2 Review Reporting and Record Keeping

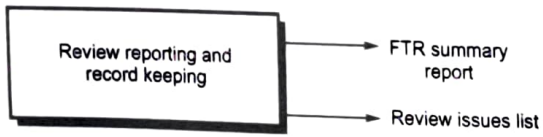


Fig. 7.3.1

During the FTR, the reviewer actively records all issues that have been raised. At the end of meeting these all raised issues are consolidated and *review issues list* is prepared. Finally, a *formal technical review summary report* is produced.

What is a purpose of producing review issues list ?

1. *It helps in identifying problematic areas within the product.*
2. *From this issue list, a check list can be prepared which guides the producer for making the corrections.*

The review issue list is normally attached to formal technical summary report.

7.3.1.3 Review Guidelines

Guidelines for the conducting of formal technical reviews must be established in advance. This guideline must be distributed to all reviewers, agreed upon, and then followed. For example - Guideline for review may include following things.

1. Concentrate on work product only. That means review the product, not the producer.
2. Set an agenda of review and maintain it.
3. When certain issues are raised then debate or arguments should be limited. Reviews should not ultimately result in some hard-feelings.
4. Find out problem areas, but don't attempt to solve every problem noted.
5. Take written notes (it is for the recorder).
6. Limit the number of participants and insist upon advance preparation.
7. Develop a checklist for each product that is likely to be reviewed.
8. Allocate resources and time schedule for FTRs in order to maintain time schedule.
9. Conduct meaningful trainings for all reviewers in order to make the reviews effective.
10. Review earlier reviews which servers as the base for the current review being conducted.

Review Question

1. *Explain formal technical review.* **GTU : Winter-2018, 2019, Summer-2019, Marks 3**

7.4 Software Reliability

Software reliability is defined as the probability of failure free operation of a computer program in a specified environment for a specified time.

The software reliability can be measured, directed and estimated.

7.4.1 Measure of Reliability and Availability

Normally there are two measures of software reliability.

1. ***MTBF :*** mean-time-between-failure is a simple measure of software reliability which can be calculated as

$$MTBF = MTTF + MTTR$$

where MTTF means mean-time-to-failure

and MTTR stands for mean-time-to-repair.

Many software researchers feel that MTBF is more useful measure of software reliability than defects/KLOC or defects/FP.

2. ***Availability*** : It's another measure of software reliability software availability is defined as the probability that the program is working according to the requirements at a given points in time. It is measured as

$$Availability = (MTTF/(MTTF+MTTR))^* 100\ \%$$

MTBF is equally sensitive to MTTF and MTTR but *availability* is more sensitive to MTTR.

7.4.2 Software Safety

Software safety is a quality assurance activity in which **potential hazards** are identified and assessed. These hazards may bring the total failure of the system. If such hazards are identified and specified in early stage of software development then such hazards can be eliminated or controlled in order to make the software safe. Modeling and analysis process is conducted as a part of software safety.

For example : In a computer based automobile system software hazards are

1. Uncontrolled acceleration that cannot be stopped.
2. Does not respond to slow the system when breaks are applied.
3. Slowly gains the speed.

How to handle the system level hazards?

Following are the steps that can be applied to preserve the software safety.

Step 1 : The hazards are identified.

Step 2 : Analysis techniques are used to assign severity of these hazards. The probability of occurrence of such hazards is also analyzed with the help of analysis techniques. The commonly used analysis techniques are fault-tree analysis, real-time logic and Petri-net models. These techniques basically predict the chain of events that can cause hazards.

Step 3 : Once hazards are identified, safety related requirements can be specified for the software. This specification basically includes list of undesirable events and desired system response. The template for safty related requirement specification is given below -

Safety related requirement specification

Identified hazards		
1. 2. 3. ...		
Name of the event that can cause hazard	**Severity of hazard**	**Probability of occurrence**
1. 2. 3. 4.		
Undesired event	**Desired system response**	
1. 2. 3. ...		

Step 4 : Finally the role of software in managing undesirable event is specified.

What is the difference between software reliability and software safety?

- Software reliability and software safety are closely related to each other. However, the difference between them lies in degree and not the type.
- Software reliability uses statistical analysis made to determine the occurrence of software failure. These failures will cause simply dissatisfaction of customer

requirements. But the software safety examines the ways in which failure results in conditions that can lead to hazards.

- Software reliability does not detect the failures in depth. But the software safety detects the failures in context of an entire computer based system.

Review Question

1. *Write a note on - Software Reliability.*

7.5 Quality Standards

GTU : Summer-2014, Winter-2017, 2018, Marks 7

7.5.1 ISO 9000

In order to bring quality in the product and service, many organizations are adopting the *quality assurance system*. The *quality assurance systems* are the organizational structures that are used to bring quality in responsibilities, procedures, processes and resources.

ISO 9000 is a family of *quality assurance system*. It can be applied to all types of organizations. It doesn't matter what size they are or what they do. It can help both product and service oriented organizations to achieve standards of quality. ISO 9000 is maintained by ISO, the **International Organization for Standardization** and is administered by accreditation and certification bodies. In ISO 9000, company's quality system and operations are scrutinized by third-party auditors for a complaince to the standard and effective operation. This process is called registration to ISO 9000. On successful registration, the company gets a certification from accreditation bodies of ISO. Such a company is then called "ISO certified company".

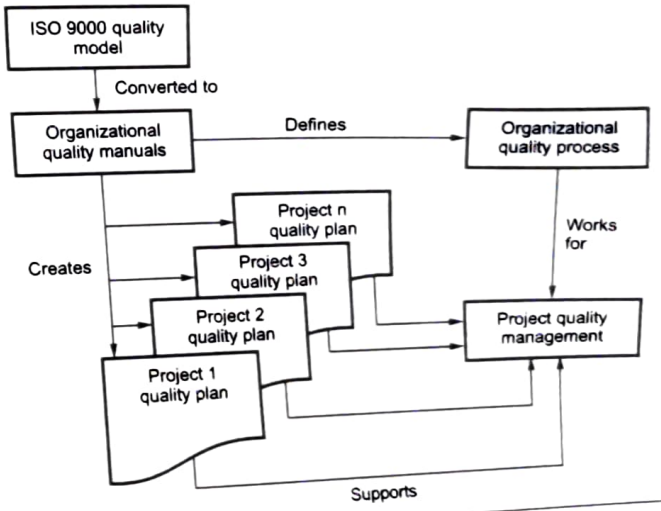
ISO 9001:2000 is a quality assurance standard which is applied to software engineering systems. It focuses on process flows, customer satisfaction, and the continual improvement of quality management systems. ISO 9001:2000 specifies requirements for a quality system that can be applied to any size or type of organization. The guideline steps for ISO 9001:2000 are

- *Establish quality management system* - Identify and manage the processes in the quality management system.
- *Document the quality management system*
- *Support the quality*
- *Satisfy the customers*
- *Establish quality policy*
- *Conduct quality planning*
- *Control quality systems*

- *Perform management reviews*
- *Provide quality resources*
- *Provide quality personnel*
- *Provide quality infrastructure*
- *Provide quality environment*
- *Control realization planning*
- *Control customer processes*
- *Control product development*
- *Control purchasing functions*
- *Control operational activities*
- *Control monitoring devices*
- *Control non confirming products*
- *Analyze quality information*
- *Make quality improvement*

The ISO 9000 helps in creating organisational quality manuals. These quality manuals identify the organisational quality processes. Using these quality manuals, the project quality plan can be prepared for every individual project. Thus project quality management can be done.

This is illustrated by following figure -



7.5.2 CMM

- The Software Engineering Institute (SEI) has developed a comprehensive process meta-model emphasizing process maturity. It is predicated on a set of system and software capabilities that should be present when organizations reach different levels of process capability and maturity.
- The Capability Maturity Model (CMM) is used in assessing how well an organization's processes allow to complete and manage new software projects.
- Various process maturity levels are

Level 1 : Initial - Few processes are defined and individual efforts are taken.

Level 2 : Repeatable - To track cost schedule and functionality basic project management processes are established. Depending on earlier successes of projects with similar applications necessary process discipline can be repeated.

Level 3 : Defined - The process is standardized, documented and followed. All the projects use documented and approved version of software process which is useful in developing and supporting software.

Level 4 : Managed - Both the software process and product are quantitatively understood and controlled using detailed measures.

Level 5 : Optimizing - Establish mechanisms to plan and implement change. Innovative ideas and technologies can be tested.

Thus CMM is used for improving the software project.

7.5.3 Six Sigma

Six sigma is widely used statistical software quality assurance strategy. It is a business driven approach to process improvement, reduced costs and increased profit. The word "six sigma" is derived from six standard deviations - 3.4 defects per million occurrences. Six Sigma originated at Motorola in the early 1980s.

Fig. 7.5.1 Six sigma framework

There are three core steps in six sigma method -

Define - The customer requirements, project goals and deliverables are defined by communicating the customers.

Measure - The existing process and its output is measured in order to determine current quality performance.

Analyze - In this phase defect metrics are analyzed in order to determine the few causes.

If an improvement is needed to an existing software then there are additional two methods in six sigma -

Improve - By eliminating the root causes of defects the process can be improved.

Control - The process can be controlled in such a way that the causes of defects can not be reintroduced.

These steps can sometimes be referred as DMAIC.

For a newly developing software, some organizations are suggesting following two alternating steps -

Design - In this step avoid root causes of defects and meet the customer requirements.

Verify - To verify the process, avoid defects and meet customer requirements.

These steps can sometimes be referred as DMADV.

Review Questions

1. *Explain six sigma method.*
2. *Write a note on ISO 9000.*
3. *Explain different quality standards.* **GTU : Summer-2014, Marks 7**
4. *List quality standards. Explain any one.* **GTU : Winter-2017, 2018, Marks 3**

7.6 SQA Plan

The SQA plan is a document created for summarizing all the SQA activities conducted for the software project. The SQA plan specifies the goal and tasks that can be performed in order do conduct all the SQA activities. Such plan should be developed by SQA group. The standard for this plan is published by IEEE standard. The template for this plan is as given below –

SQA Plan
1. Purpose and scope of the plan
2. Description of work product
3. Applicable standards
4. SQA activities
5. Tools and methods/standards used

6. SCM procedures for managing change
7. Methods for maintaining SQA related records
8. Organizational roles and responsibilities

The SQA plan is a document aimed to give confidence to developers and customers that the specified requirements will be met and final product will be a quality product.

Review Question

1. *Give the template for SQA Plan.*

❑❑❑

8 Software Maintenance and Configuration Management

Syllabus

Types of software maintenance, Re-engineering, Reverse engineering, Forward engineering, The SCM process, Identification of objects in the software configuration, Version control and change control.

Contents