UNIT - 1

CHAPTER 1

Introduction to Quality

Introduction to Quality

Syllabus Topic : Historical Perspective of Quality

1.1 Historical Perspective of Quality

Q. 1.1.1 What is Quality? Explain its history. (Ref. Sec. 1.1) (5 Marks)

- As per the traditional thinking, quality is conformance to specification i.e. whether the product does what it is designed for.
- The quality movement has its roots back in medieval Europe, where craftsmen began organizing into unions called guilds in the late 13th century.
- Until the early 19th century, manufacturing in the industrialized world tended to follow this craftsmanship model. The factory system, with its emphasis on product inspection, started in Great Britain in the mid-1750's and grew into the Industrial Revolution in the early 1800's.
- In the early 20th century, manufacturers began to include quality processes in quality practices. After the United States entered World War II, quality became a critical component of the war effort: Bullets manufactured in one state, for example, had to work consistently in rifles made in another.
- The armed forces initially inspected virtually every unit of product; then to simplify and speed up this process without compromising safety, the military began to use sampling techniques for inspection.
- The birth of total quality took place in Japan where Japanese rather than concentrating on inspection, focused on improving all organizational processes through the people who used them.

Syllabus Topic : What is Quality? (Is It a Fact or Perception?)

1.2 What is Quality ? (Is It a Fact or Perception ?)

Q. 1.2.1 Is quality based on perception? (Ref. Sec. 1.2) **(5 Marks)**

- Quality means different things to different people and it is highly context dependent. Sometimes it can be recognized through experience but is difficult to define in some traceable form. Generally it is perceived as fitness for use i.e. while evaluating quality we must ask a question "Does the product satisfy user need and experience?"
- However quality is not only viewed in terms of what it can deliver but also by the experience it provides. Here the quality is tied to internal characteristics of the product thus we can say quality is a fact provided by the product.
- People generally think high priced products offer better quality and quality in this perspective depends upon the amount the customer is willing to pay for it hence at times it can also be a perception.

Syllabus Topic : Quality View

1.3 Quality View

Q. 1.3.1 List and explain different views of Quality. (Ref. Sec. 1.3) **(5 Marks)**

David Garvins described software quality from five different perspectives :

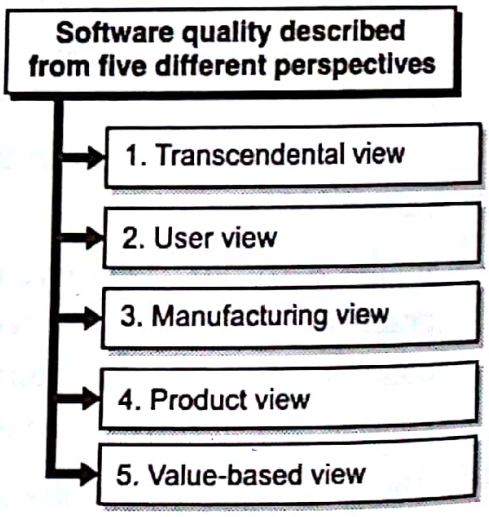

Fig. 1.3.1

➜ **1. Transcendental view**

Quality is something that can be recognized but not defined.

➜ **2. User view**

Quality is fitness for purpose. This view of quality evaluates how the product meets the user's needs. It can thus be a highly personalized view. This view is often measured by reliability, usability.

➜ **3. Manufacturing view**

Quality is conformance to specification. It focuses on how well the product was constructed, in an effort to avoid the costs associated with rework during development and after delivery. Defect counts and rework costs are two characteristics to measure.

➜ **4. Product view**

Quality is tied to inherent product characteristics. It assumes that measuring and controlling internal product properties (internal quality indicators) will result in improved external product behaviour (quality in use).

➜ **5. Value-based view**

Quality depends on the amount the customer is willing to pay for it. Software quality is customer satisfaction. Indeed customer satisfaction should be a result of delivering a quality product, but satisfaction can be influenced by many things and is not the same as quality. Moreover, customers routinely are satisfied by poor quality and not satisfied by high quality.

Syllabus Topic : Financial Aspect of Quality

1.4 Financial Aspect of Quality

Q. 1.4.1 Explain the costs associated with quality. (Ref. Sec. 1.4) (5 Marks)

☞ **Cost of quality**

It is the sum of the costs related to prevention and detection of defects and costs due to occurrences of defects.

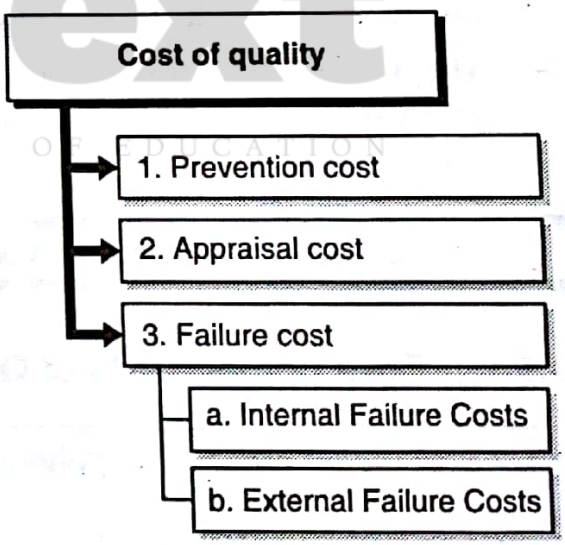

Fig. 1.4.1

➜ **1. Prevention cost**

The cost that arises from efforts to prevent defects. Quality planning, formal technical reviews, test equipment, training etc.

➜ **2. Appraisal cost**

The cost that arises from efforts to detect defects. In-process and inter-process inspection, equipment calibration and maintenance, testing etc.

➜ **3. Failure cost**

- **Internal Failure Costs:** The cost arises from defects identified internally and efforts to correct them (rework, repair, failure mode analysis).
- **External Failure Costs:** The cost arises from defects identified by the client or end-users and efforts to correct them (complaint resolution, product return and replacement, help line support, warranty work).

Formula for calculating cost of quality

Cost of Quality (COQ) = Cost of Control + Cost of Failure of Control

Where ,

Cost of Control = Prevention Cost + Appraisal Cost

and

Cost of Failure of Control = Internal Failure Cost + External Failure Cost

Syllabus Topic : Definition of Quality

1.5 Definition of Quality

Q. 1.5.1 Define Quality. (Ref. Sec. 1.5) (5 Marks)

- The degree to which a component, system or process meets specified requirements and/or user/customer needs and expectations. When we talk about software quality, we are actually talking about the evaluation of the software based on certain attributes.
- A software quality is defined based on the study of external and internal features of the software. The external quality is defined based on how software performs in real time scenario in operational mode and how useful it is for its users.
- The internal quality on the other hand focuses on the intrinsic aspects that are dependent on the quality of the code written.
- The user focuses more on how the software works at the external level, but the quality at external level can be maintained only if the coder has written a meaningful good quality code.

Syllabus Topic : Core Components of Quality

1.5.1 Core Components of Quality

Q. 1.5.2 What are the components of quality? (Ref. Sec. 1.5.1) (5 Marks)

Core components of quality include :

1. Functionality

Functionality is the conformity of the software with actual requirements and specifications. An application should provide all the required functionality as specified in the requirements.

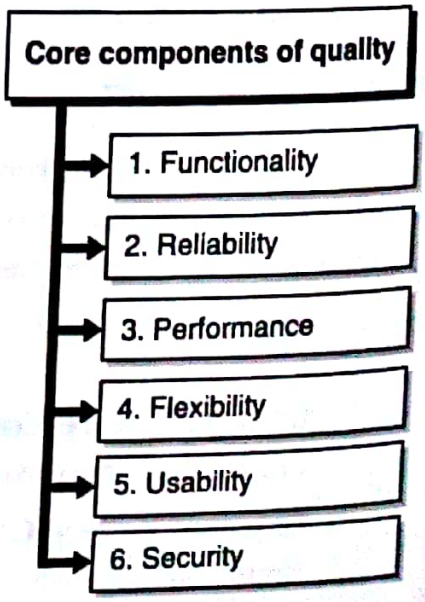

Fig. 1.5.1

2. Reliability

The ability of an application or system to consistently perform its intended or required function on demand and without degradation or failure.

3. Performance

Performance is mostly about response time of the software. This response time should be in acceptable intervals (e.g. max. a few seconds), and should not increase if transaction count increases.

4. Flexibility

It normally refers to the ability of the application or system to adapt to possible or future changes in its requirements. Flexibility is the ability of software to add/modify/remove functionality without damaging current system. Change is inevitable so flexibility become an important aspect of quality.

5. Usability

Usability is the ease of use and learning ability of an application or system.

6. Security

The degree to which the application is protected against malicious attacks and other potential risks. Security is very important especially when the application is dealing with sensitive user data like bank details.

Syllabus Topic : Customers, Suppliers and Processes

1.6 Customers, Suppliers and Processes

Q. 1.6.1 Define Customer. Supplier and processes. (Ref. Sec. 1.6) (5 Marks)

Customer

- A customer (sometimes known as a client, buyer, or purchaser) is the recipient of a good, service, product or an idea - obtained from a seller, vendor, or supplier via a financial transaction or exchange for money or some other valuable consideration. Customer is end user of a product.
- In Software testing consumer or end user is very important. The entire testing revolves around the expectation of the user from the product or service. Testing is done to give confidence to the customer about the reliability and satisfaction in the application.

Supplier

A supplier is a person, company or organization that sells or supplies goods or equipment to other organizations. The organization adds some value to the goods or equipment supplied and converts them into a product.

Processes

- Processes are the activities carried or steps taken in order to achieve a particular end. In software testing defined processes to achieve a milestone are important because :
  - streamlines the work you are doing
  - brings consistency across the team the way everyone works
  - creates a standard of working
  - following process will result in better quality product
  - makes future maintenance of the product easier

Syllabus Topic : Total Quality Management (TQM)

1.7 Total Quality Management (TQM)

Q. 1.7.1 What is TQM? Explain the principles of TQM. (Ref. Sec. 1.7) (5 Marks)

- High quality service has become a critical factor for a successful business. This gave birth to Total Quality Management concept. TQM describes a management approach to long-term success through customer satisfaction.
- In a TQM effort, all members of an organization participate in improving processes, products, services, and the culture in which they work.

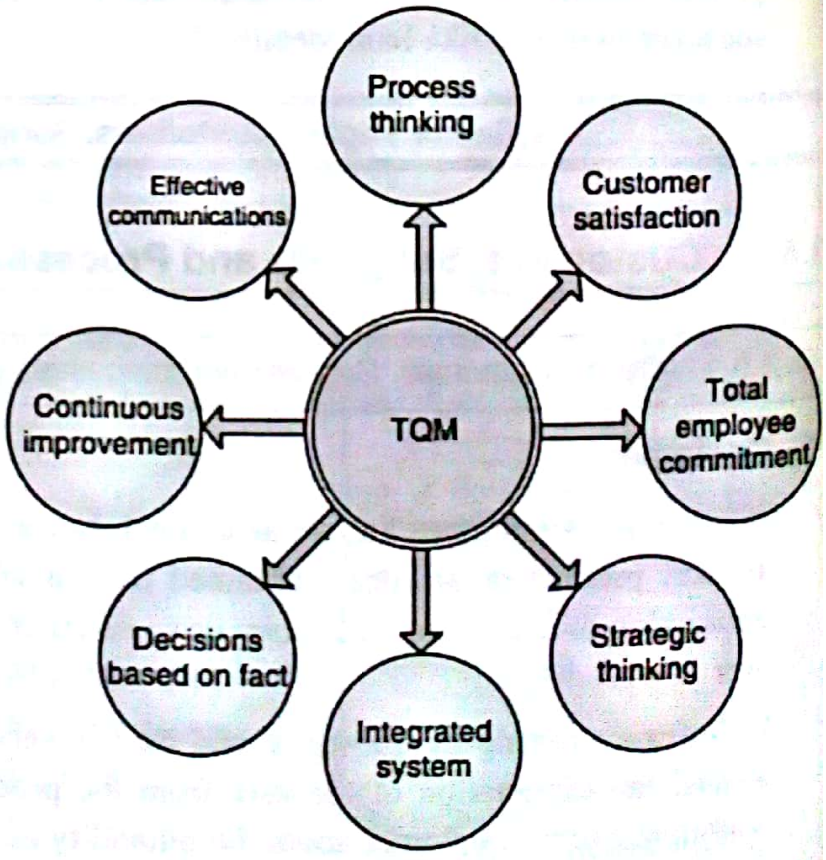

Fig. 1.7.1

Syllabus Topic : Quality principles of TQM

1.7.1 Quality principles of TQM

Q. 1.7.2 Explain the costs associated with Quality. (Ref. Sec. 1.7.1) (5 Marks)

The Quality principles of TQM are :

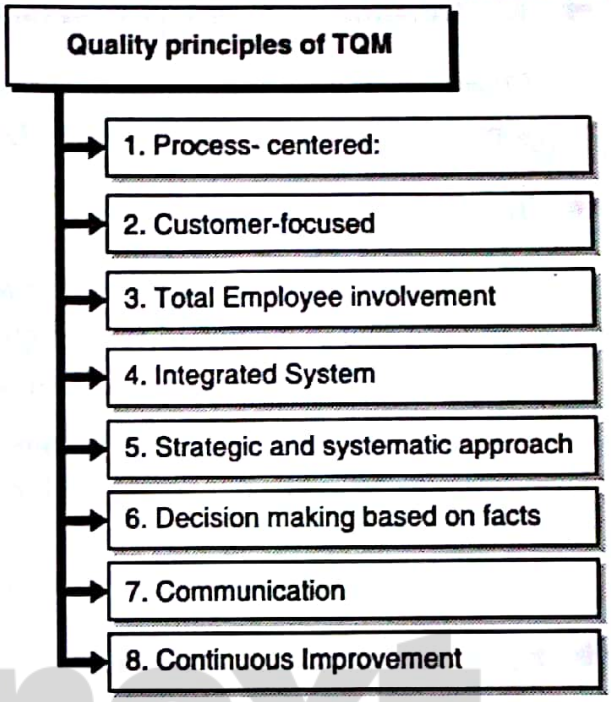

Fig. 1.7.2

➜ **1. Process- centered**

- A process is a series of steps that take inputs from suppliers (internal or external) and transforms them into outputs that are delivered to customers.
- The steps required to carry out the process are defined, and performance measures are continuously monitored in order to detect unexpected variation.

➜ **2. Customer-focused**

- The customer ultimately determines the level of quality.
- No matter what an organization does to improve quality-training employees, integrating quality into the design process, upgrading computers or software, or buying new measuring tools-the customer determines whether the efforts were worthwhile.

➜ **3. Total Employee involvement**

- Employee involvement in the development of products or services of an organization largely determines the quality of these products or services.
- Organization should ensure to create a culture in which employees feel they are involved with the organization and its products and services.

➜ **4. Integrated System**

- There are different departments and functional specialty units in an organization, it is important to have an integrated organization system.
- Everyone must understand the vision, mission, and guiding principles as well as the quality policies, objectives, and critical processes of the organization.
- All processes aggregate into the business processes required for defining and implementing strategy.

5. Strategic and systematic approach

A critical part quality management is the strategic and systematic approach to achieving an organization's vision, mission, and goals. Strategic planning or strategic management includes the formulation of a strategic plan that integrates quality as a core component.

6. Decision making based on facts

Decision-making within the organization must only be based on facts and not on opinions (emotions and personal interests). Data should support this decision-making process.

7. Communication

- During times of organizational change, as well as part of day-to-day operation, effective communications plays a large part in maintaining morale and in motivating employees at all levels. Communications involve strategies, method, and timeliness.
- A communication strategy must be formulated in such a way that it is in line with the mission, vision and objectives of the organization. This strategy comprises the stakeholders, the level within the organization, the communications channels, the measurability of effectiveness, timeliness, etc.

8. Continuous Improvement

Continual improvement drives an organization to be both analytical and creative in finding ways to become more competitive and more effective at meeting stakeholder expectations.

Syllabus Topic : Quality Management through Statistical Process Control

1.8 Quality Management through Statistical Process Control

Q. 1.8.1 How can quality be managed using statistical process control? (Ref. Sec. 1.8) (5 Marks)

- **Statistical process control (SPC)** is a method of quality control which employs statistical methods to monitor and control a process.
- The most successful SPC tool is the control chart. Quality data in the form of Product or Process measurements are obtained in real-time during manufacturing. This data is then plotted on a graph with pre-determined control limits.
- **Control limits** are determined by the capability of the process, whereas **specification limits** are determined by the client's needs.

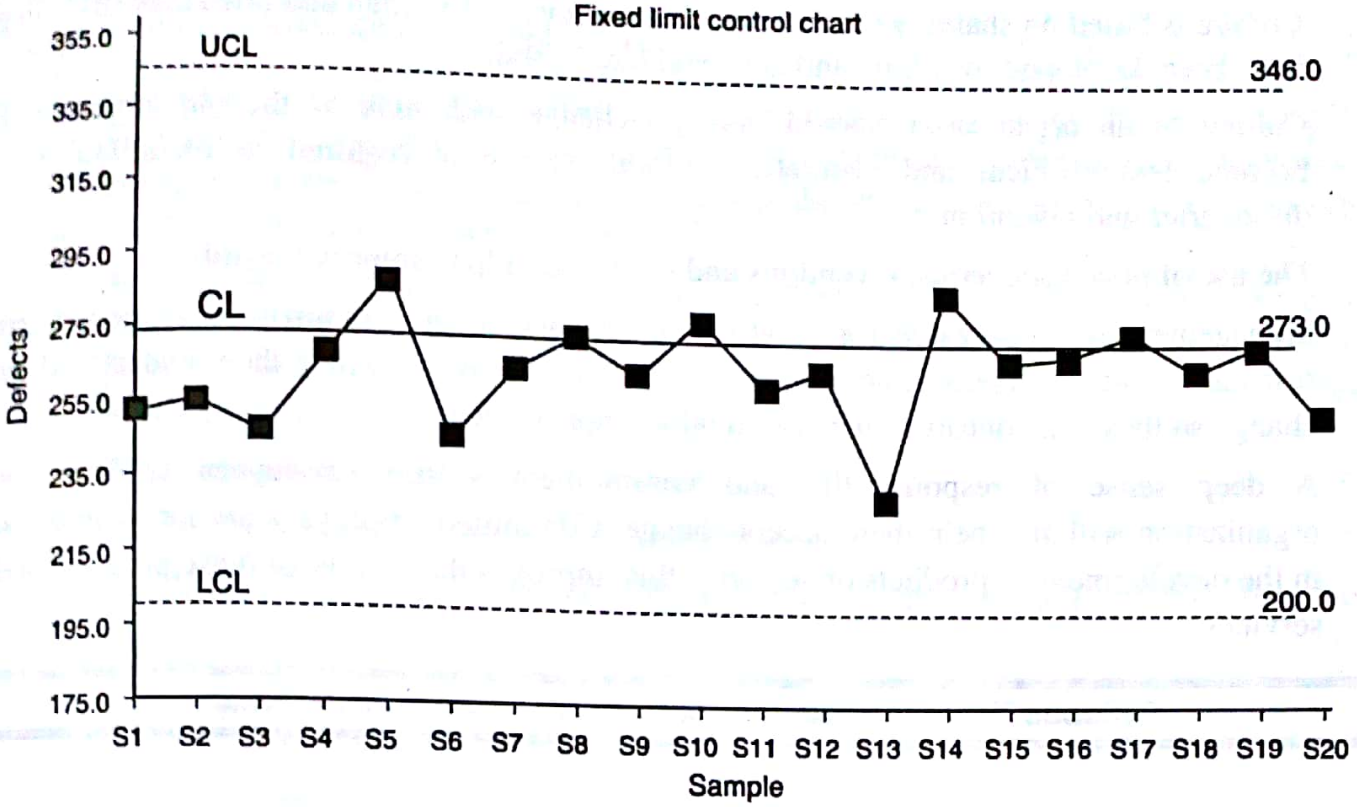

Fig. 1.8.1

- Data that falls within the control limits indicates that everything is operating as expected. Any variation within the control limits is likely due to a common cause-the natural variation that is expected as part of the process.
- If data falls outside of the control limits, this indicates that an assignable cause is likely the source of the product variation, and something within the process should be changed to fix the issue before defects occur.
- An advantage of SPC is that it emphasizes early detection and prevention of problems, rather than the correction of problems after they have occurred.

Syllabus Topic : Quality Management through Cultural Change

1.8.1 Quality Management through Cultural Change

Q. 1.8.2 Explain the statement: Cultural change in an organization can increase quality of a product. (Ref. Sec. 1.8.1) **(5 Marks)**

- Organization culture i.e. the pattern of shared values, beliefs and assumptions influence the way to think and act within an organization.
- Organizational culture includes an organization's expectations, experiences, philosophy, as well as the values that guide member behavior, and is expressed in member self-image, inner workings, interactions with the outside world, and future expectations.

- Culture is based on shared attitudes, beliefs, customs, and written and unwritten rules that have been developed over time and are considered valid.
- Culture of an organization should undergo change with time as the old culture will become less efficient and obsolete. Cultural change is required to assimilate new discoveries and invention.
- The use of new discoveries, inventions and tools will help to improve quality.
- Management is accountable for creating this change in the organization as people are resistant to change. Training should be provided to employees to help them understand the change so they can contribute better in quality improvement.
- A deep sense of responsibility and commitment within employees towards the organization will also help them accept change. Committed employees are more involved in the development of products or services; this improves the quality of these products and services.

Syllabus Topic : Continual(Continuous) Improvement Cycle

1.9 Continuous Improvement Cycle

Q. 1.9.1 How can we continuously improve quality? (Ref. Sec. 1.9) (5 Marks)

A continuous improvement cycle is an ongoing effort to improve products, services, or processes.

The continuous improvement cycle consists of three phases :

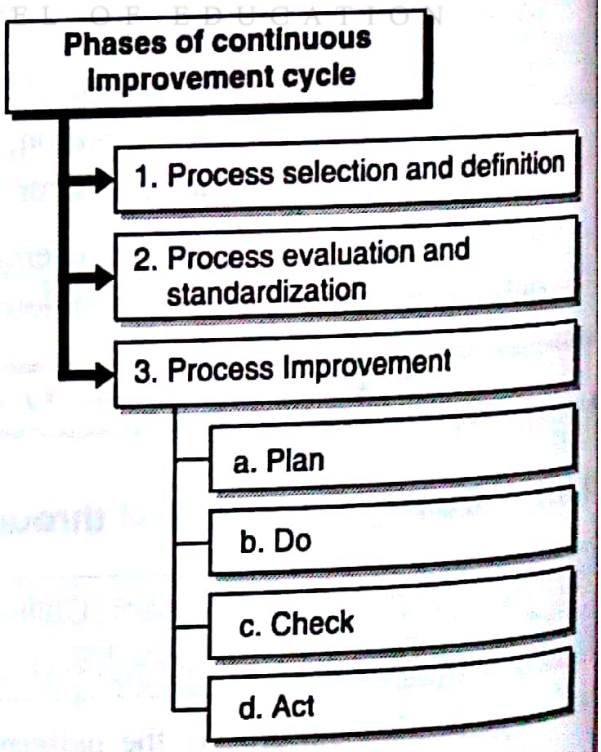

Fig. 1.9.1

➜ **1. Process selection and definition**

This involves selecting and defining a critical process (related to the business strategy) having scope for continuous improvement. It also includes forming a steering group who will focus on the critical process. Management actively participates in this. The he process owner and the team members define the chosen process. Defining the process is an essential prerequisite for improving the process. Process definition includes :

(a) internal/external customers (b) Inputs (c) process and (d) outputs.

2. Process evaluation and standardization

- The selected process is described in detail to check whether the process is clearly understood or not. This requires standardising the process, measuring and reviewing the performance of the process, identifying the short comings of the process and analysing the problems with the process.
- By standardising the process, the current best way to perform that process is established. This is done to ensure that everyone understands the process and the process are used consistently in the same way every time. The process standards provide the base line for continuous process improvement.
- The important activities in this phase are :
  (i) Defining and describing the selected process
  (ii) Measuring the performance of the process on the basis of the described performance measure
  (iii) Analysing the available process data
  (iv) Performing cause and effect analysis
  (v) Identifying root causes

3. Process Improvement:

PDCA cycle is used to improve the selected process.

a. **Plan :** The improvement plan formulated during the "process selection and definition" phase is updated. The improvement objectives are determined, improvement actions and how they are related to the organizational goals are indicated and solutions which address the stated requirements and problem causes are developed.

b. **Do :** Improvement plan is executed, Solutions are tested. Strategies are implemented on smaller scale.

c. **Check :** The effects of the process change are measured, the results of these improvement actions are reviewed based on performance measures, the extent to which improvement objectives can be realised with these actions is checked and the results are compared with the norms.

d. **Act :** The results are implemented i.e. the process improvement is introduced, the process is brought under control, the final process change is made, the results assessed, the process continuously improved and monitored, the improved performance documented, and the possible process changes standardised.

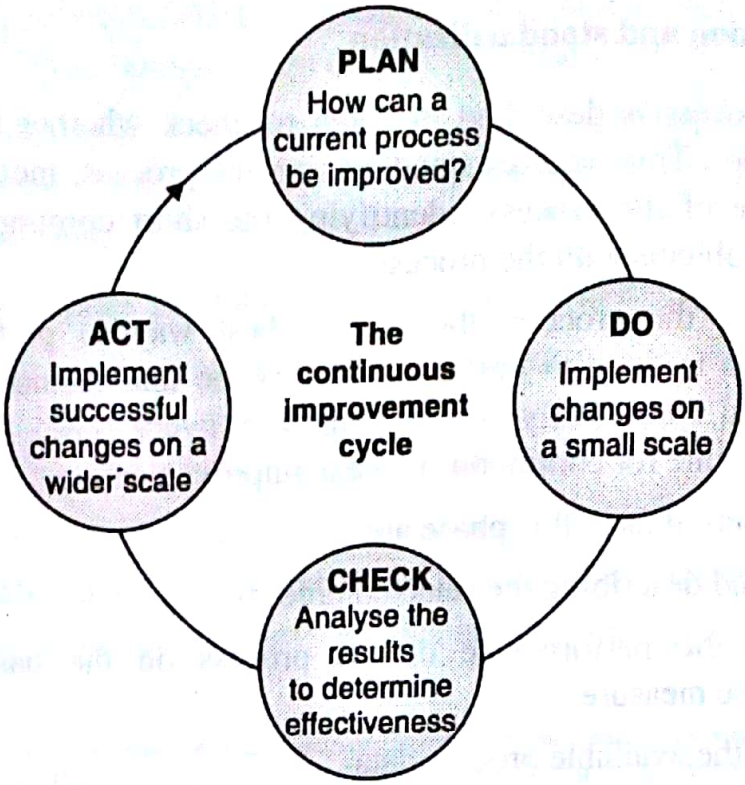

Fig. 1.9.2

Syllabus Topic : Quality in Different Areas

1.9.1 Quality in Different Areas

Q. 1.9.2 Explain how quality affects different sectors? (Ref. Sec. 1.9.1) (5 Marks)

- **Quality** is the value of things relative to their purpose. Any product, service, experience or asset can be described in terms of its quality or lack of quality. Quality includes both tangible aspects such as features and intangible aspects such as the taste of food.
- The following are types of quality.

☞ **Product Quality**

It is related to the quality of Products that fit customer needs and fulfill customer expectations.

☞ **Service Quality**

Services involve intangible elements of quality such as environments, customer service and customer experience such as accessibility, personalization etc.

☞ **Experience Quality**

It includes quality of experiences in areas such as medicine, education, entertainment, events and services such as design, delivery etc

Quality of Life

The health, happiness, security, prosperity and fulfillment of individuals and communities.

Ex. : Air Quality, Freedom of speech etc.

Syllabus Topic : Benchmarking

1.10 Benchmarking

Q. 1.10.1 Define Benchmarking. List its advantages. (Ref. Sec. 1.10) (5 Marks)

- Benchmarking plays an important role in determining some of the major qualities of the software. Benchmark can be defined as "something that serves as a standard by which others may be measured or judged".
- To put it simply, it is a metric or a point of reference against which products or services can be compared, which further assess the quality of the said product or service.
- Similarly, benchmark testing is the process of load testing a component or an entire end to end IT system to determine the performance characteristics of the application. It is mainly used to compare the performance of the software or hardware.

1.10.1 Advantages of Benchmarking

The advantages of Benchmarking are :

- Validates that the software components are in an accurate working condition.
- It helps build an application whose characteristics are well understood and that can stand up to the demands of the real world users.
- With its assistance software developers are able to launch a software or application with confidence, as they are sure that their users will be satisfied with the performance as well as the effectiveness of the released product.
- Benchmarking is one stop solution for software developers and testers who want to develop a quality product that offers remarkable services.

Syllabus Topic : Metrics

1.11 Metrics

Q. 1.11.1 Define metrics in software testing with some examples. (Ref. Sec. 1.11) (5 Marks)

- A software metric is a standard of measure of a degree to which a software system or process possesses some property. The goal of software metrics is to identify and control essential parameters that affect software development.

- Other objectives of using software metrics are listed below :
  - Measuring the size of the software quantitatively.
  - Assessing the level of complexity involved.
  - Assessing the testing techniques.
  - Specifying when to stop testing.
  - Determining the date of release of the software.
  - Estimating cost of resources and project schedule.
- Software metrics help project managers to gain an insight into the efficiency of the software process, project, and product. This is possible by collecting quality and productivity data and then analyzing and comparing these data with past averages in order to know whether quality improvements have occurred.
- Also, when metrics are applied in a consistent manner, it helps in project planning and project management activity. For example, schedule-based resource allocation can be effectively enhanced with the help of metrics.
- Example of metrics :
  - Total number of tests executed
  - Total number of tests not executed
  - Total number of tests passed
  - Total number of test failed
  - Total number of defects identified

Syllabus Topic : Problem Solving Techniques

1.12 Problem Solving Techniques

Q. 1.12.1 Explain some problem solving techniques. (Ref. Sec. 1.12) **(5 Marks)**

The process of solving a software problem can be divided into four steps :

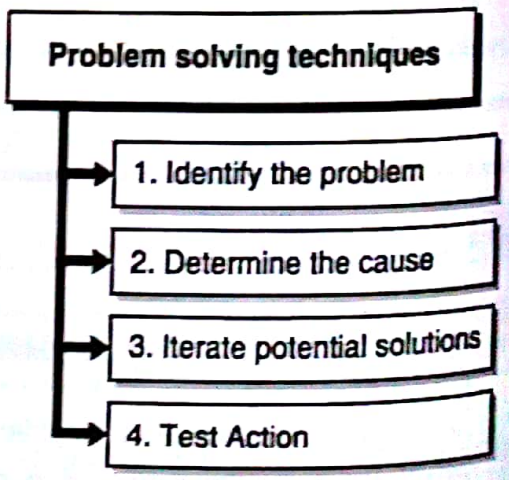

Fig. 1.12.1

➜ 1. Identify the problem

Define the problem and identify the type of problem - the size, scope and nature of problem. Be clear about what the problem is. How well you define the problem determine how well you will solve the problem.

➜ **2. Determine the cause**

- After defining the problem we need to determine what is causing it. If you consider the problem as a gap between where you are now and where you want to be, the causes of the problem are the obstacles that are preventing you from closing that gap immediately.
- This level of analysis is important to make sure your solutions address the actual causes of the problem instead of the symptoms of the problem. If your solution fixes a symptom instead of an actual cause, the problem is likely to reoccur since it was never truly solved.

➜ **3. Iterate potential solutions**

- Brainstorm and develop possible solutions to the problem. There is a lot of room for creativity here. List all the solutions. After you come up with several ideas that can solve the problem, one problem solving technique you can use to decide which one is the best solution to your problem is a simple **trade-off analysis.**
- To perform the trade-off analysis, define the critical criteria for the problem that you can use to evaluate how each solution compares to each other. The evaluation can be done using a simple matrix. The highest ranking solution will be your best solution for this problem.

➜ **4. Test Action**

After deciding on the solution it is a good idea to create an action plan and implement it.

Syllabus Topic : Problem Solving Software Tools

1.12.1 Problem Solving Software Tools

Q. 1.12.2 What is a flowchart? (Ref. Sec. 1.12.1) **(5 Marks)**

Q. 1.12.3 Explain Fishbone chart. (Ref. Sec. 1.12.1) **(5 Marks)**

➜ **1. Flowchart**

A flowchart is a diagram that represents a process; they can be simple or quite complex. A flowchart usually incorporates questions and decisions and as such it is a good troubleshooting tool, it ensures that certain options have been investigated and that nothing obvious has been left out.

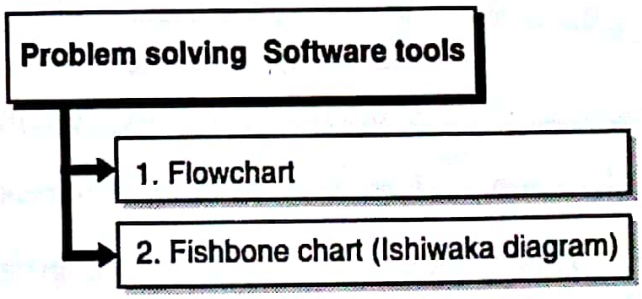

Fig. 1.12.2

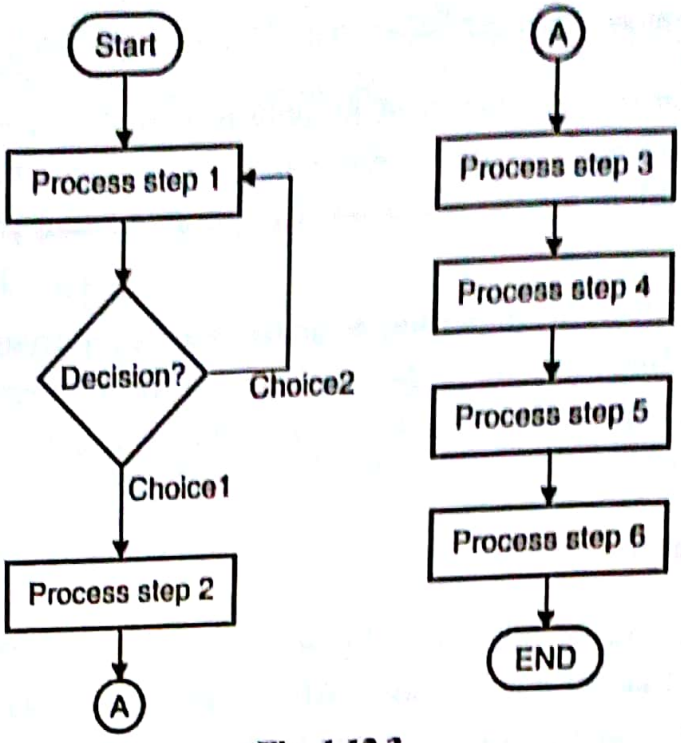

Fig. 1.12.3

➜ 2. Fishbone chart (Ishiwaka diagram)

- A fishbone diagram, also called a cause and effect diagram or Ishikawa diagram, is a visualization tool for categorizing the potential causes of a problem in order to identify its root causes.
- A fishbone diagram is useful in brainstorming sessions to focus conversation. After the group has brainstormed all the possible causes for a problem, the facilitator helps the group to rate the potential causes according to their level of importance and diagram a hierarchy.
- How to create a fish diagram :
  - Create a head, which lists the problem or issue to be studied.
  - Create a backbone for the fish (straight line which leads to the head).
  - Identify at least four "causes" that contribute to the problem. Connect these four causes with arrows to the spine. These will create the first bones of the fish.
  - Brainstorm around each "cause" to document those things that contributed to the cause. Use the 5 Whys or another questioning process such as the 4P's (Policies, Procedures, People and Plant) to keep the conversation focused.
  - Continue breaking down each cause until the root causes have been identified.
- This example illustrates how a group might begin a fish diagram to identify all the possible reasons a web site went down in order to discover the root cause.

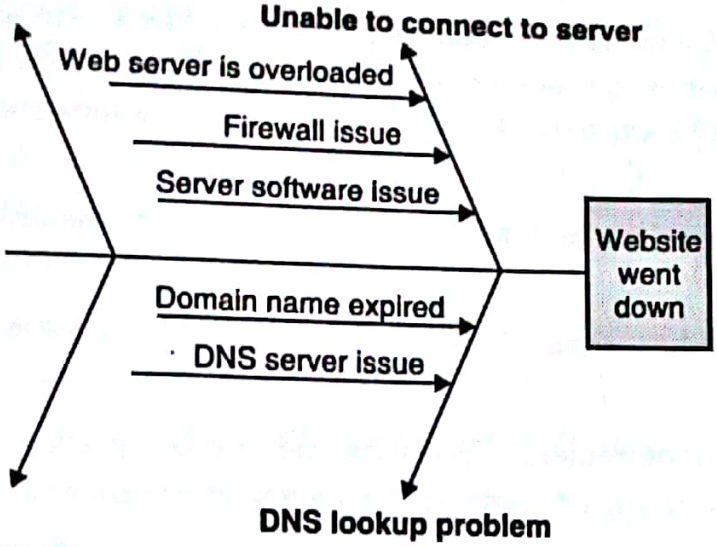

Fig. 1.12.4

Software Quality

Syllabus Topic : Introduction to Software Quality

1.13 Introduction to Software Quality

Q. 1.13.1 What is software quality? (Ref. Sec. 1.13) (5 Marks)

- Software quality is the degree of conformance to explicit or implicit requirements and expectations.
- A software quality is defined based on the study of external and internal features of the software.
- The external quality is defined based on how software performs in real time scenario in operational mode and how useful it is for its users. The internal quality on the other hand focuses on the intrinsic aspects that are dependent on the quality of the code written.
- The user focuses more on how the software works at the external level, but the quality at external level can be maintained only if the coder has written a meaningful good quality code.

Syllabus Topic : Constraints of Software Product Quality Assessment

1.13.1 Constraints of Software Product Quality Assessment

Q. 1.13.2 What affects quality assessment of software products? (Ref. Sec. 1.13.1) (5 Marks)

- Testing is difficult to institute in small organizations where available resources to perform the necessary activities are not present. A smaller organization tends not to have the required resources like manpower, capital etc. to assist in the process of Software Quality Assurance.
- Hence, some organizations may be less willing to include the cost of implementing SQA into their budget.
- Testing requires 30-40% time of SDLC. In most projects there is not enough time for through testing.
- There is continuous requirement of learning and upgrading knowledge on multiple testing tools. This requires training of employees which again consumes time and money.

Syllabus Topic : Customer is King

1.13.2 Customer is King

Q. 1.13.3 Explain customer is king. (Ref. Sec. 1.13.2) **(5 Marks)**

- The key to measuring quality or testing for quality is to have a QA team that is very customer-oriented.
- Features of software are based on customer requirements.
- Customer is the end user of the software hence testing software from customer point of view is very important.
- We may develop a product which meets the entire requirement but is not useful to the customer, such product will be rejected by the customer in the market.
- Customer will want value for their money. They will buy software and expect return on their investment. Thus the software should be designed keeping this in mind.
- **Transcendent feeling** - This is a feeling of an individual or group of individual towards a product. Customer like the software and feel it is fun and it's the latest thing- so what if it has few problems. They enjoy working with this.

Syllabus Topic : Quality and Productivity Relationship

1.13.3 Quality and Productivity Relationship

Q. 1.13.4 How are quality and productivity related? (Ref. Sec. 1.13.3) **(5 Marks)**

- **Productivity** is the relationship between the amount of outputs and amount of inputs needed to produce a product. In other words, management measures productivity by comparing the amount of a product produced to the amount of raw materials and manpower needed to produce a product.

- If less raw materials and manpower are used to produce more of a product, then productivity is considered high.
- **When quality increases productivity also improves.** This is because wastes and rework are reduced. Higher productivity enables an organization to reduce price and gain competitive advantage both in terms of price and quality.
- Customers also feel happy as they get value for their money. Since organization's bottom line improves, it raises the satisfaction level of all stakeholders, including employees.
- Saving the cost of quality will have an immediate effect on the enhanced productivity of an organization. All these establish that quality and productivity are indirectly related.

Syllabus Topic : Requirements of a Product

1.13.4 Requirements of a Product

Q. 1.13.5 What requirements should a product fulfil? (Ref. Sec. 1.13.4) (5 Marks)

The following are example of product requirement :

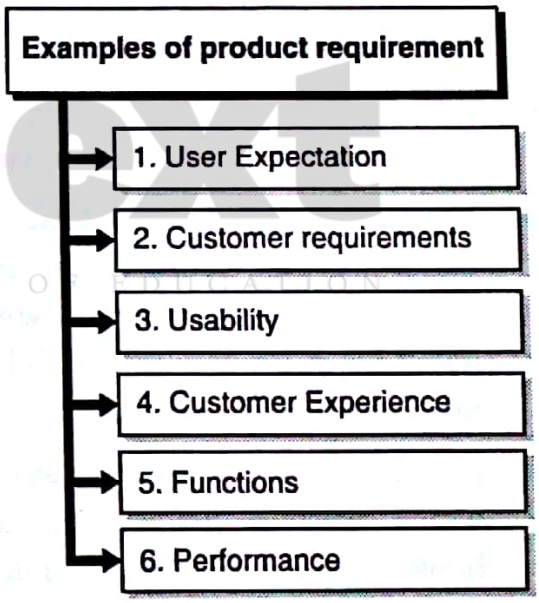

Fig. 1.13.1

➜ **1. User Expectation**

Requirements should capture expectations of product (customer expectation). For example: As a customer, I want the shirt to be free of tags that rub against the skin.

➜ **2. Customer requirements**

Requirements contributed by a customer. For example, "I want to be able to choose from hundreds of bright colours."

➜ **3. Usability**

Ease of use. For example, "this button works when user's finger is slightly off target."

➜ **4. Customer Experience**

Requirements intended to improve the end-to-end customer experience such as "beeps and other feedback sounds are off by default."

➜ **5. Functions**

Specifications of goals that can be accomplished with the product. For example, "As a customer, I want to be able to effortlessly carry a bag of groceries with the bicycle."

6. Performance

Performance targets for the product such as a figure of merit. For example, "the solar panels shall have a maximum conversion efficiency of at least 20%."

Syllabus Topic : Organisation Culture

1.14 Organisation Culture

Q. 1.14.1 Explain Organisation Culture. (Ref. Sec. 1.14) (5 Marks)

- Organization culture i.e. the pattern of shared values, beliefs and assumptions influence the way to think and act within an organization.
- Organizational culture includes an organization's expectations, experiences, philosophy, as well as the values that guide member behaviour, and is expressed in member self-image, inner workings, interactions with the outside world, and future expectations.
- Culture is based on shared attitudes, beliefs, customs, and written and unwritten rules that have been developed over time and are considered valid.
- The culture decides the way employees interact at their workplace. A healthy culture encourages the employees to stay motivated and loyal towards the management.
- Every organization must have set guidelines for the employees to work accordingly. The culture of an organization represents certain predefined policies which guide the employees and give them a sense of direction at the workplace. Every individual is clear about his roles and responsibilities in the organization and know how to accomplish the tasks ahead of the deadlines.
- No two organizations can have the same work culture. It is the culture of an organization which makes it distinct from others. The work culture goes a long way in creating the brand image of the organization.
- The work culture gives an identity to the organization. In other words, an organization is known by its culture.
- The organization culture brings all the employees on a common platform. The employees must be treated equally and no one should feel neglected or left out at the workplace. It is essential for the employees to adjust well in the organization culture for them to deliver their level best.
- The work culture unites the employees who are otherwise from different back grounds, families and have varied attitudes and mentalities. The culture gives the employees a sense of unity at the workplace.

Syllabus Topic : Characteristics of Software

1.14.1 Characteristics of Software

Q. 1.14.2 What are characteristics of software? (Ref. Sec. 1.14.1) (5 Marks)

The Characteristics of Software are as follows :

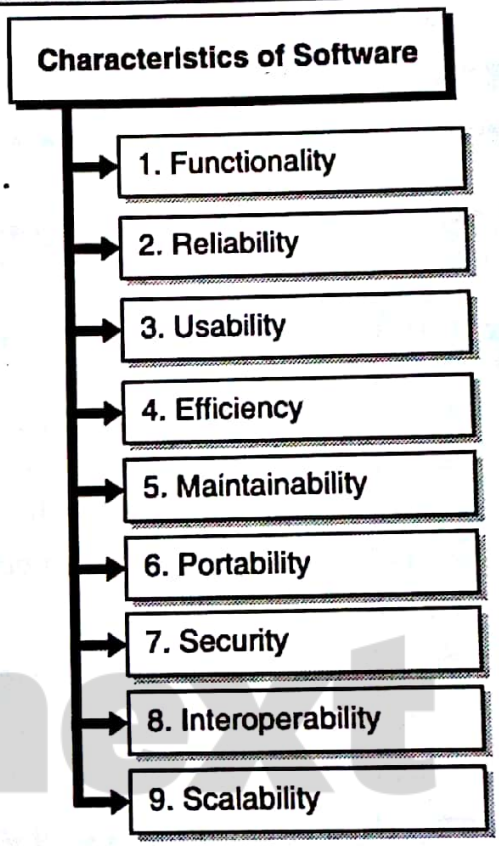

Fig. 1.14.1

1. Functionality

Refers to the degree of performance of the software against its intended purpose.

2. Reliability

Refers to the ability of the software to provide desired functionality under the given conditions.

3. Usability

Refers to the extent to which the software can be used with ease.

4. Efficiency

Refers to the ability of the software to use system resources in the most effective and efficient manner.

5. Maintainability

Refers to the ease with which the modifications can be made in a software system to extend its functionality, improve its performance, or correct errors.

6. Portability

Refers to the ease with which software developers can transfer software from one platform to another, without (or with minimum) changes. In simple terms, it refers to the ability of software to function properly on different hardware and software platforms without making any changes in it.

7. Security

With the increase in security threats nowadays, this factor is gaining importance. The software shouldn't have ill effects on data / hardware. Proper measures should be taken to keep data secure from external threats.

8. Interoperability

Interoperability is the ability of software to exchange information with other applications and make use of information transparently.

9. Scalability

It should be very easy to upgrade it for more work(or for more number of users).

Syllabus Topic : Software Development Process

1.15 Software Development Process

Q. 1.15.1 Explain SDLC. (Ref. Sec. 1.15) **(5 Marks)**

- It is a framework that defines activities that are performed throughout the software development process. The development process adopted for a project will depend on project aims and goals.
- There are numerous development life cycles that have been developed in order to achieve different required objectives.
- The most appropriate development process should be applied to each project. The models specify the various stages of the process and the order in which they are carried out.

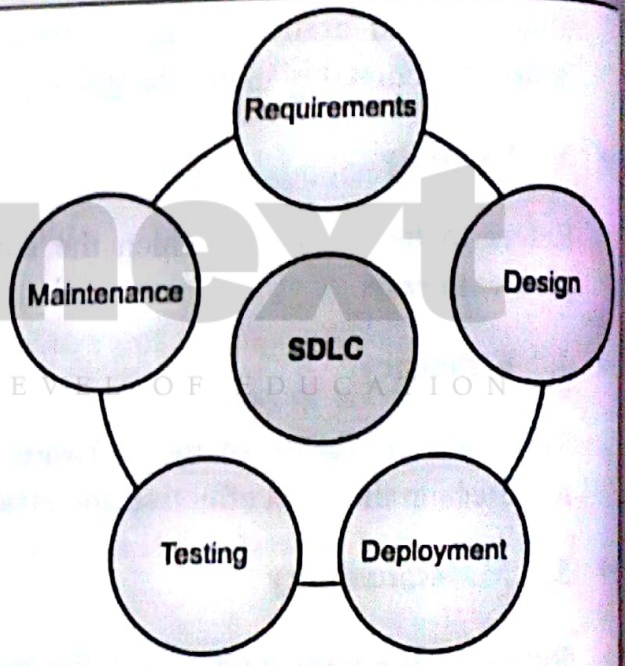

Fig. 1.15.1

- Phases of SDLC are as follows :

Requirement Gathering

- This is the most important phase in software development life cycle. In this phase requirement for the software are collected from the customer/client.
- These requirements are provided in a document called Business Requirement Specification (BRS) or System Requirement Specification (SRS). All details and specifications of the product must be discussed with the customer.
- The development team analyses the requirements keeping in view the design and coding of the software. The requirements so gathered are then analyzed for their validity and possibility of incorporating them into the software system.
- The aim of requirement analysis is to capture the detail of each requirement so that everyone understands how each requirement is to be worked.

- **Design**

  It has two steps :

  HLD - High Level Design - It gives the architecture of the software product to be developed and is done by architects and senior developers.

  LLD - Low Level Design - It is done by senior developers. It describes how each and every feature in the product should work and how every component should work. Here, only the design will be there and not the code. The outcome from this phase is High Level Document and Low Level Document which works as an input to the next phase.

- **Coding**

  It means translating the design into a computer readable language. Development team does the actual coding based on designed software and writes unit tests for each component to test the new codes written by them. This is the longest phase of the software development life cycle.

- **Testing**

  After the code is developed, testing is carried out to verify the entire requirement specified by customer has been implemented. The aim of tester is to find out the gaps or defects within the system and also to verify that the software works as expected according to the requirements. It includes Unit testing, Integration testing and System testing.

- **Deployment**

  After successful testing, the product is delivered/deployed to the customer for their use. The size of the project will determine the complexity of the deployment. The users can be trained on, or aided with the documentation on how to operate the software. A small round of testing is also performed on production to make sure of any environmental issues and any impact of new release.

- **Maintenance**

  Once when the customers start using the developed system then the actual problems will come up and needs to be solved from time to time. Fixing the issues found by the customer comes in the maintenance phase.

Syllabus Topic : Types of Products

1.15.1 Types of Products

Q. 1.15.2 What are categories of a software? (Ref. Sec. 1.15.1) (5 Marks)

Software is often divided into two categories :

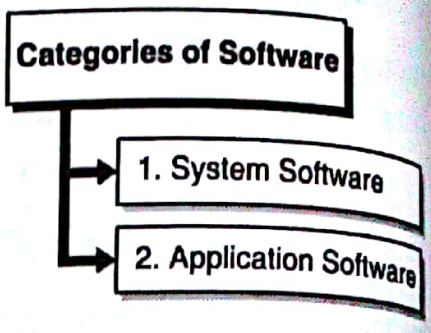

Fig. 1.15.2

1. System Software

It is a base for application software which responsible for managing hardware. In other words we can say that system software is an intermediate between user and hardware. System software is also known as Operating System.

Examples : MS-Windows, UNIX, Linux, Sun Solaris

2. Application Software

Application Software - or simply applications are often called productivity programs or end-user programs because they enable the user to complete tasks, such as creating documents, spreadsheets, databases and publications, doing online research, sending email, designing graphics, running businesses, and even playing games! Application software is specific to the task it is designed for and can be as simple as a calculator application or as complex as a word processing application.

Syllabus Topic : Schemes of Criticality Definitions

1.16 Schemes of Criticality Definitions

Q. 1.16.1 What is severity of a defect? (Ref. Sec. 1.16) **(5 Marks)**

- **Severity** is defined as the degree of impact a defect has on the development or operation of a component application being tested.
- Defect severity is categorized into four classes :

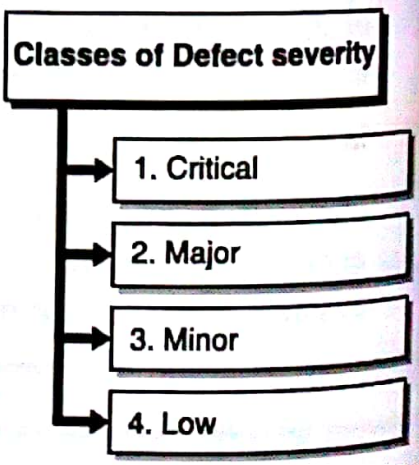

Fig. 1.16.1

1. Critical

This defect affects critical functionality of the software. It indicates complete shut-down of the process, nothing can proceed further

E.g. : Restarting of the phone, complete failure of a feature etc.

2. Major

Major defects are of high severity which affects functionality of the software. There is a workaround but is not easy and obvious for users to understand.

E.g. : functionality is not working from a module but the same functionality can be accessed from other module after performing some number of additional steps from that module.

➜ **3. Minor**

This defect affects minor functionality which is not very critical. It cause some undesirable behaviour, but the system is still functional

E.g. : functionality is not working from a module but the same functionality can be accessed from other module easily i.e. Minor defects have easy workaround

➜ **4. Low**

These defects do not affect functionality. These are generally cosmetic defects and does not impact productivity or efficiency of the software.

E.g. : Spelling mistake, grammatical error etc.

Syllabus Topic : Problematic Areas of Software Development Life Cycle

1.16.1 Problematic Areas of Software Development Life Cycle

Q. 1.16.2 What are problematic areas of software development? (Ref. Sec. 1.16.1)(5 Marks)

➜ **1. Lack of Flexibility**

- It is based on the assumption that detailed requirements and a plan can be laid out at the beginning of the project without a significant amount of change as the project is in progress.
- That just doesn't work well in an uncertain environment where it is very difficult, if not impossible, to lay out detailed requirements for a project upfront.
- And applying the typical level of change control to control changes can create a very inflexible approach.

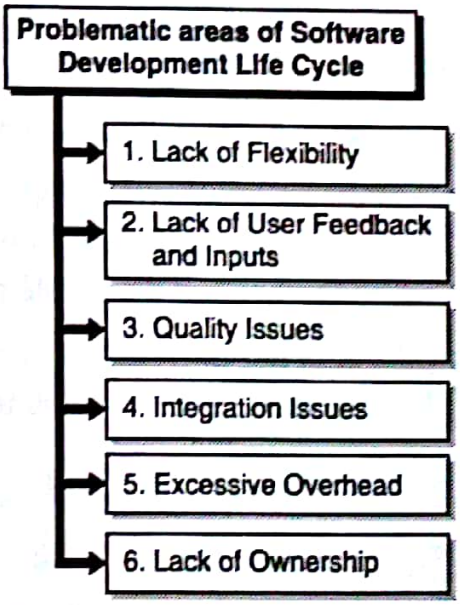

Fig. 1.16.2

- It forces the project team to make a lot of assumptions and many times those assumptions are wrong which can cause a lot of unnecessary problems and rework later.

➜ 2. Lack of User Feedback and Inputs

- The user doesn't typically see the solution until the very end of the project and by that time, it may be either too late or very expensive to make any corrections.
- The alternative to both of these problems is a more empirical and iterative approach where the detailed requirements are further elaborated based on user feedback and inputs as the project is in progress.

➜ 3. Quality Issues

In the typical SDLC process, quality testing is typically done by a separate QA organization in a sequential phase after development is complete. There are several problems inherent in that approach :

(a) Developers feel that "quality is someone else's problem"

(b) There is a relatively long cycle time for finding problems and correcting them

(c) It is a very reactive process to try to find and fix defects after development is complete and can result in a lot of rework. A much better approach is to make quality testing more of an integral part of the development process and make everyone on the team feel responsibility for the quality of the product that they produce.

➜ 4. Integration Issues

- Many times final integration of the overall solution doesn't take place until the very end of the project and that can lead to some very big and nasty surprises. In the early 1990's, I worked on a large communications project where the company invested over $150 million in development costs.
- A lot of the development was done by separate groups and when they tried to integrate the whole thing together at the end, they couldn't get all the pieces to work together and the whole project had to be scrapped.
- A better approach is to use an incremental and iterative development approach and do continuous integration throughout the project to discover any problems very early.

➜ 5. Excessive Overhead

- The typical SDLC relies heavily on documentation to carry the results of one phase over to the next phase. There's a lot of unnecessary overhead to try to rely on documenting everything that there is to know about a project.
- A much better approach is to use a much more integrated team approach to development that relies much more heavily on direct face-to-face communication and collaboration and less on written documentation.

6. Lack of Ownership

The team in a typical SDLC project is typically not well-integrated; and, as a result, there may not be a strong feeling of ownership for the results of the project.

Syllabus Topic : Software Quality Management

1.17 Software Quality Management

Q. 1.17.1 Differentiate between Quality Assurance and Quality Control. (Ref. Sec. 1.17) **(5 Marks)**

Q. 1.17.2 What is Quality Planning? (Ref. Sec. 1.17) **(5 Marks)**

- Software Quality Management is a process that ensures the required level of software quality is achieved when it reaches the users, so that they are satisfied by its performance.
- Software quality management is split into three main activities :

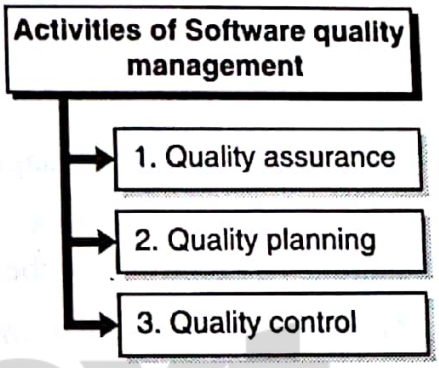

Fig. 1.17.1

1. Quality assurance

The development of a framework of organizational procedures and standards that lead to high quality software.

2. Quality planning

The selection of appropriate procedures and standards from this framework and adapt for a specific software project.

3. Quality control

Definition of processes ensuring that software development follows the quality procedures and standards.

Quality management provides an independent check on the software and software development process. It ensures that project deliverables are consistent with organizational standards and goals.

- **Quality Assurance (QA)** is the maintenance of a desired level of quality in an application or software by means of giving attention to every stage of the software development process.
- It ensures that the approaches, techniques, methods and processes are designed for the projects are implemented correctly.

- Quality assurance activities monitor and verify that the processes used to manage and create the deliverables have been followed and are operative.
- It is a procedure that focuses on providing assurance that quality requested will be achieved. It is process oriented and mainly focuses on defect prevention.
- **Quality planning** is the process of developing a quality plan for a project. The quality plan defines the quality requirements of software and describes how these are to be assessed.
- The quality plan selects those organizational standards that are appropriate to a particular product and development process. Quality plan has the following parts :
  1. Introduction of product.
  2. Product plans.
  3. Process descriptions.
  4. Quality goals.
  5. Risks and risk management.
- The quality plan defines the most important quality attributes for the software and includes a definition of the quality assessment process.
- **Quality Control (QC)** ensures that the approaches, techniques, methods and processes are designed in the project is following correctly.
- It is a procedure or set of procedures intended to ensure that an application or system adheres to a defined set of quality criteria or meets the requirements of the client or customer.
- QC activities monitor and verify that the project deliverables meet the defined quality standards. It is product oriented and focuses on defect identification.

Syllabus Topic : Why Software Has Defects?

1.17.1 Why Software Has Defects?

Q. 1.17.3 Why does defect exists in a software? (Ref. Sec. 1.17.1) (5 Marks)

- Software is created by human being. Programmer and tester can make mistakes which could lead to an error. These errors can produce defects in the system.
- Below are the reasons which cause defects in the software :
  - **Miscommunication of requirements introduces error in code :** This may introduce from the first step of SDLC means from requirement gathering to development of the product. If the requirements are incomplete or vague then the development leads to situation where developers facing in developing software application based on

incomplete requirements and this cause the issue in testing of such incomplete application. Also communication errors are introduced in case if application developed by other developer and modified by other developer.

- **Unrealistic time schedule for development :** Many of the times developers not get enough time to design, develop and unit test the application in development environment prior to move to tester to test complete application. There are chances that the changes made the last minutes have introduced errors
- **Lack of designing experience :** How good your design is decides the overall software application development
- **Poorly documented code :** It's tough to maintain and modify code that is badly written or poorly documented; the result is bugs. In many organizations management provides no incentive for programmers to document their code or write clear, understandable, maintainable code. In fact, it's usually the opposite: they get points mostly for quickly turning out code, and there's job security if nobody else can understand it ('if it was hard to write, it should be hard to read').
- Human factors introduces errors in code
- Buggy third-party tools
- Last minute changes in the requirement introduce error
- Poor Software testing skill

Syllabus Topic : Processes Related to Software Quality

1.18 Software Quality Assurance (SQA)

Q. 1.18.1 Write a short note on Software Quality Assurance. (Ref. Sec. 1.18) **(5 Marks)**

Q. 1.18.2 Does the software adequately meet its quality factors ? (Ref. Sec. 1.18) **(5 Marks)**

Q. 1.18.3 Has software development been conducted according to pre-established standards? (Ref. Sec. 1.18) **(5 Marks)**

Q. 1.18.4 Have technical disciplines performed their SQA roles properly? (Ref. Sec. 1.18) **(5 Marks)**

- Software Quality Assurance (SQA) is a process that ensures that developed software meets and complies with defined or standardized quality conditions.
- It is an ongoing process within the software development life cycle (SDLC) that routinely checks the developed software to ensure it meets desired quality measures. SQA is an umbrella activity that is applied throughout the software process.

- Rather than checking for quality after completion, SQA processes test for quality in each phase of development until the software is complete. It is implemented in all types of software development project regardless of which software development Life cycle model is used.
- SDLC moves to the next phase only when the current or the previous phase complies with all the quality standards.

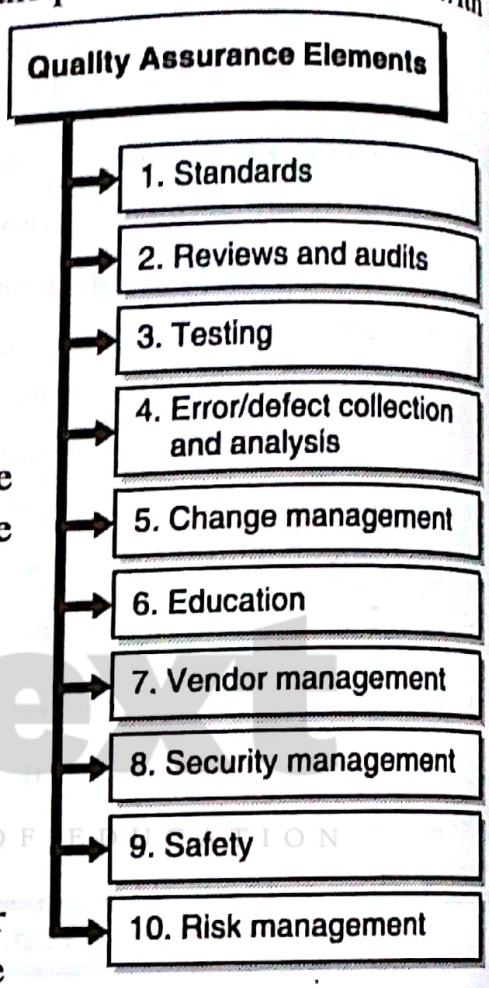

Fig. 1.18.1

1.18.1 Quality Assurance Elements

➜ 1. Standards

Ensure that standards are adopted and followed.

➜ 2. Reviews and audits

Audits are reviews performed by SQA personnel to ensure that quality guidelines are followed for all software engineering work.

➜ 3. Testing

Ensure that testing is properly planned and conducted.

➜ 4. Error/defect collection and analysis

Collects and analyses error and defect data to better understand how errors are introduced and can be eliminated.

➜ 5. Change management

Ensures that adequate change management practices have been instituted.

➜ 6. Education

Takes lead in software process improvement and educational program.

➜ 7. Vendor management

Suggests specific quality practices vendor should follow and incorporate quality mandates in vendor contracts.

➜ 8. Security management

Ensures use of appropriate process and technology to achieve desired security level.

9. Safety

Responsible for assessing impact of software failure and initiating steps to reduce risk.

10. Risk management

Ensures risk management activities are properly conducted and that contingency plans have been established.

1.18.2 SQA Tasks

- Prepare SQA plan for the project.
- Participate in the development of the project's software process description.
- Review software engineering activities to verify compliance with the defined software process.
- Audit designated software work products to verify compliance with those defined as part of the software process.
- Ensure that any deviations in software or work products are documented and handled according to a documented procedure.
- Record any evidence of noncompliance and reports them to management.

1.18.3 SQA Goals

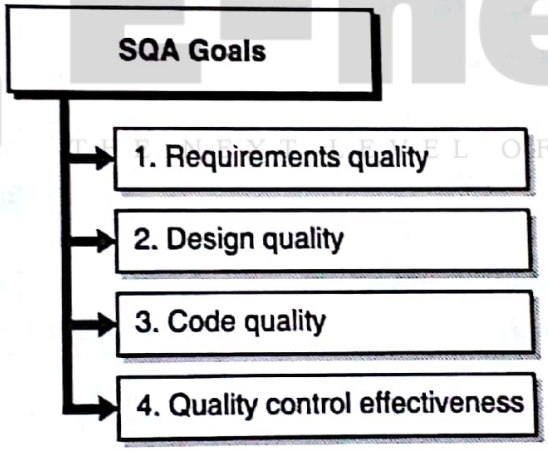

Fig. 1.18.2

1. Requirements quality

- Ambiguity
- Volatility
- Model clarity
- Completeness
- Traceability

2. Design quality

- Architectural integrity
- Interface complexity
- Component completeness
- Patterns

3. Code quality

- Complexity
- Understandability
- Documentation
- Maintainability
- Reusability

4. Quality control effectiveness

- Resource allocation
- Review effectiveness
- Completion rate
- Testing effectiveness

Syllabus Topic : Quality Management System Structure

1.19 Quality Management System Structure

Q. 1.19.1 What is QMS structure? (Ref. Sec. 1.19) **(5 Marks)**

- A QMS can be defined as :

  "A set of co-ordinated activities to direct and control an organisation in order to Continually improve the effectiveness and efficiency of its performance."

- The main thrust of a QMS is in defining the processes, which will result in the production of quality products and services, rather than indetecting defective products or services after they have been produced.

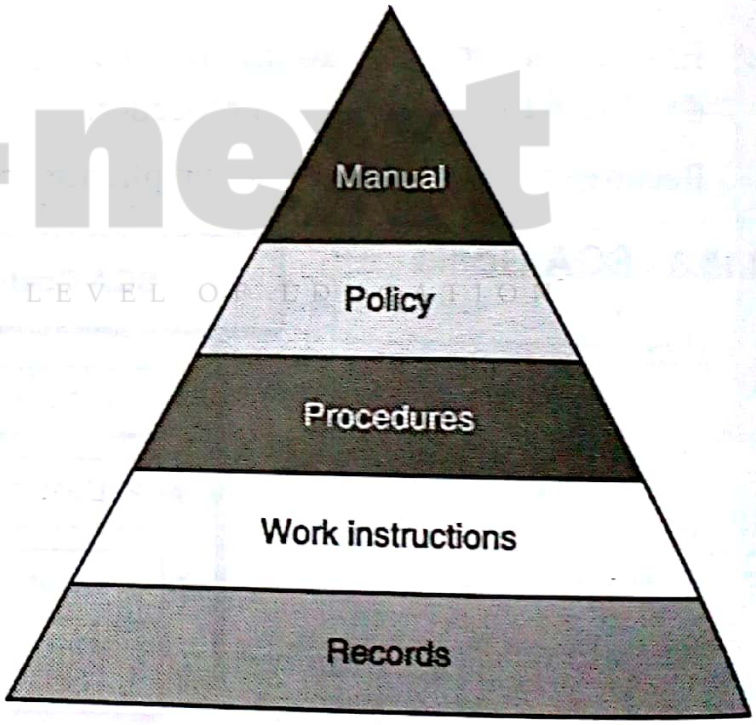

Fig. 1.19.1

1. Quality manual

Small organizations can document the entire QMS in one manual. On the other side, large international organizations may have several different quality manuals. Generally, the manual includes the QMS scope, exclusions from the standard, references to relevant documents, and the business process model. The quality policy and the objectives can be part of the manual as well.

2. Quality policy

- A policy represents a declarative statement by an organization. A Quality policy should state the commitment of the organization to quality and continual improvement. Usually, this policy is used for promotional purposes and should be displayed in the organization's premises and posted on websites, so a clear and short quality policy is convenient and is the general practice.
- The Quality policy defines the quality objectives to which the organization strives. The quality goals of organizations are defined by quantifying the quality objectives.

3. Quality procedures

- Quality procedures can have different formats and structures. They can be narrative, i.e., described through text; they can be more structured by using tables; they can be more illustrative, i.e., flow charts; or they can be any combination of the above.
- Quality procedures should include the following elements :
  - Title - for identification of the procedure;
  - Purpose - describing the rationale behind the procedure;
  - Scope - to explain what aspects will be covered in the procedure, and which aspects will not be covered;
  - Responsibilities and authorities of all people/functions included in any part the procedure;
  - Records that result from the activities described in the procedure should be defined and listed;
  - Document control - identification of changes, date of review, approval and version of the document should be included in accordance with the established practice for document control;
  - Description of activities - this is the main section of the procedure; it relates all the other elements of the procedure and describes what should be done, by whom and how, when and where. In some cases, "why" should be clarified as well. Additionally, the inputs and the outputs of the activities should be explained, including the needed resources.
  - Appendices may be included, if needed.

4. Work Instructions

Work instructions can be part of a procedure, or they can be referenced in a procedure. Generally, work instructions have a similar structure to the procedures and cover the same elements; however, the work instructions include details of activities that need to be realized, focusing on the sequencing of the steps, tools, and methods to be used and required accuracy.

Syllabus Topic : Pillars of Quality Management System

1.19.1 Pillars of Quality Management System

Q. 1.19.2 What are pillars of Quality Management System? (Ref. Sec. 1.19.1) **(5 Marks)**

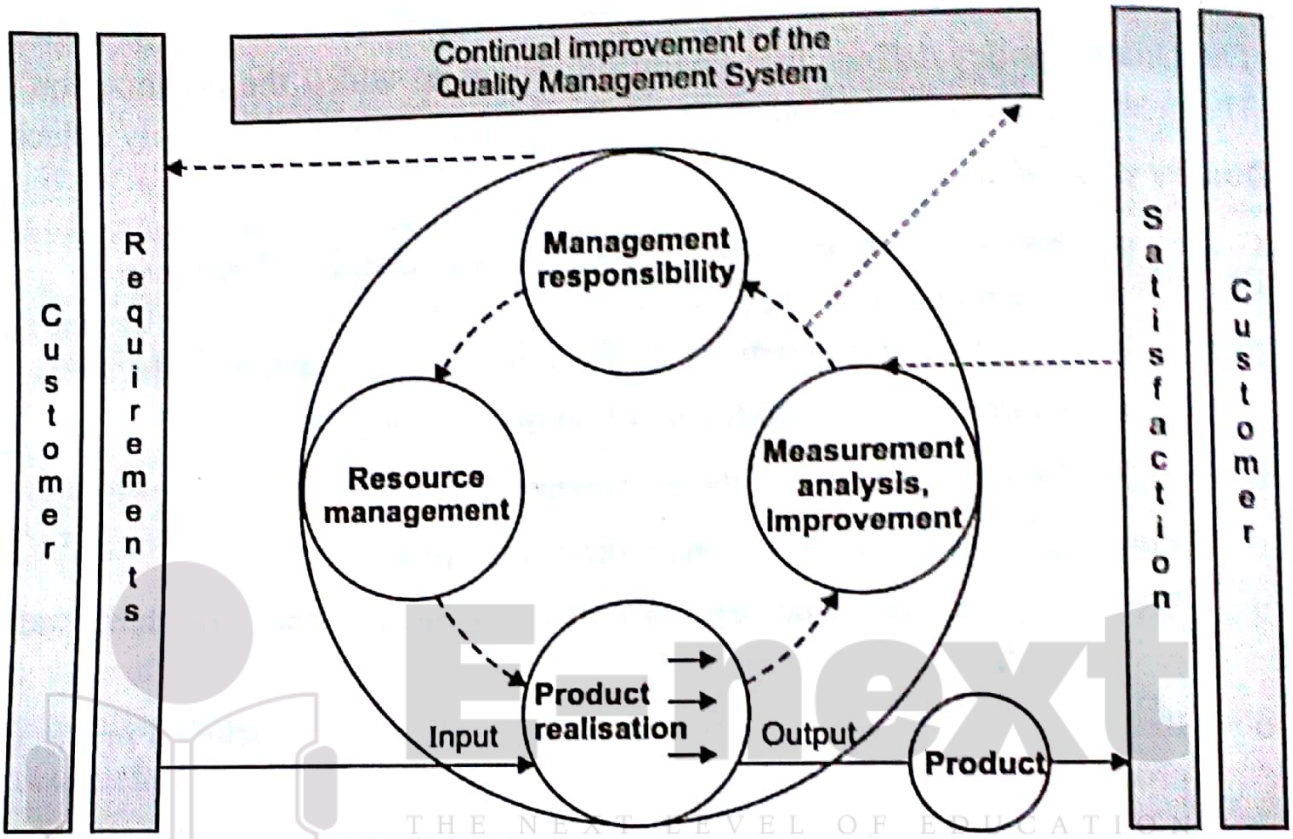

Fig. 1.19.2

1. **Management responsibility**
   - Management commitment
   - Customer focus
   - Quality policy
   - Planning
   - Responsibility, authority and communication
   - Management review
2. **Resource management**
   - Provision of resources
   - Human resources
   - Infrastructure
   - Work environment

3. **Product realisation**
   - Planning of product realisation
   - Customer-related processes
   - Design and/or development
   - Purchasing
   - Production and service operations
   - Control of measuring and monitoring devices
4. **Measurement, analysis and improvement**
   - General
   - Planning
   - Monitoring and measurement
   - Control of non-conforming product
   - Analysis of data
   - Improvement

Syllabus Topic : Important Aspects of Quality Management

1.19.2 Important Aspects of Quality Management

Q. 1.19.3 Explain important aspects of quality Management. (Ref. Sec. 1.19.2) **(5 Marks)**

A QMS must ensure that the products/services conform to customer needs and expectations, and the objectives of the organisation. Issues to be considered when setting up a QMS include its :

1. **Design** and **build** includes the structure of the quality management system, the process and its implementation. Its design must be led by senior managers to suit the needs of the organisation, and this is ideally done using a framework to lead the thinking. Design of the QMS should come from determining the organisation's core processes and well-defined goals and strategies, and be linked to the needs of one or more stakeholders.

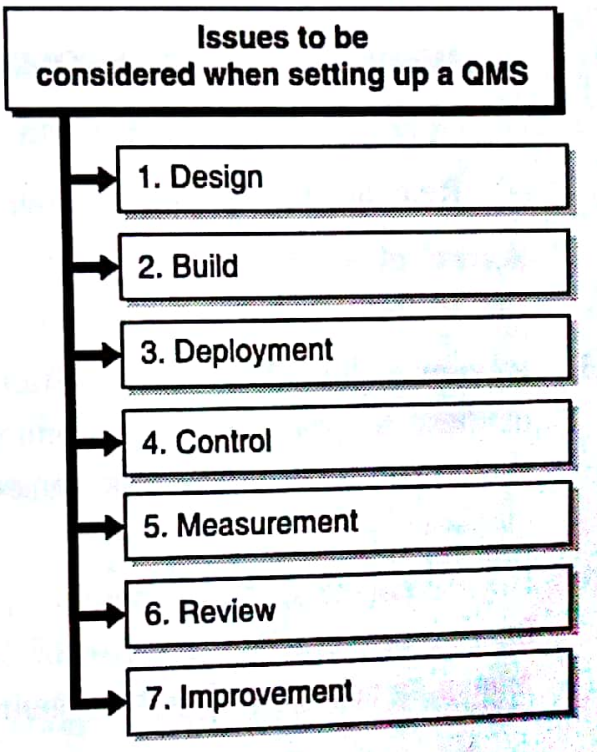

Fig. 1.19.3

The process for designing and building the QMS must also be clear, with the quality function playing a key role, but involvement and buy-in to the system must also come from all other functions.

2. **Deployment** and **implementation** is best achieved using process packages, where each core process is broken down into sub-processes, and described by a combination of documentation, education, training, tools, systems and metrics. Electronic deployment via Intranets is increasingly being used.

3. **Control** of the QMS will depend on the size and complexity of the organisation. ISO is a site-based system, and local audits and reviews are essential even if these are supplemented by central reviews.

   Local control, where possible, is effective, and good practice is found where key stakeholders are documented within the process and where the process owner is allowed to control all of the process.

   Ideally, process owners/operators are involved in writing procedures.

4. **Measurement** is carried out to determine the effectiveness and efficiency of each process towards attaining its objectives. It should include the contribution of the QMS to the organisation's goals; this could be achieved by measuring the following :

   - Policy definition completeness
   - Coverage of business
   - Reflection of policies
   - Deployment
   - Usage
   - Whether staffs finds the QMS helpful in their work
   - Speed of change of the QMS
   - Relevance of QMS architecture to the job in hand

   A form of scorecard deployed through the organisation down to individual objective level can be employed, and the setting of targets at all levels is vital.

5. **Review** of the effectiveness, efficiency and capability of a QMS is vital, and the outcome of these reviews should be communicated to all employees. Reviewing and monitoring should be conducted whether or not improvement activities have achieved their expected outcomes.

6. **Improvement** should follow as a result of the review process, with the aim of seeking internal best practice. It is part of the overall improvement activities and an integral part of managing change within the organisation.

A good QMS will :

- Set direction and meet customers' expectations
- Improve process control
- Reduce wastage
- Lower costs
- Increase market share
- Facilitate training
- Involve staff
- Raise morale

1.20 Exam Pack (Review Questions)

☞ **Syllabus Topic : Historical Perspective of Quality**

Q. 1 What is Quality? Explain its history. *(Ans. : Refer section 1.1)* **(5 Marks)**

☞ **Syllabus Topic : What is Quality? (Is It a Fact or Perception?)**

Q. 2 Is quality based on perception? *(Ans. : Refer section 1.2)* **(5 Marks)**

☞ **Syllabus Topic : Quality View**

Q. 3 List and explain different views of Quality. *(Ans. : Refer section 1.3)* **(5 Marks)**

☞ **Syllabus Topic : Financial Aspect of Quality**

Q. 4 Explain the costs associated with Quality. *(Ans. : Refer section 1.4)* **(5 Marks)**

☞ **Syllabus Topic : Definition of Quality**

Q. 5 Define Quality. What are the core components of Quality? *(Ans. : Refer section 1.5)* **(5 Marks)**

☞ **Syllabus Topic : Core Components of Quality**

Q. 6 What are the components of quality? *(Ans. : Refer section 1.5.1)* **(5 Marks)**

☞ **Syllabus Topic : Customers, Suppliers and Processes**

Q. 7 Define Customer. Supplier and processes. *(Ans. : Refer section 1.6)* **(5 Marks)**

☞ **Syllabus Topic : Total Quality Management (TQM)**

Q. 8 What is TQM? Explain the principles of TQM. *(Ans. : Refer section 1.7)* **(5 Marks)**

☞ **Syllabus Topic : Quality principles of TQM**

Q. 9 Explain the costs associated with Quality. *(Ans. : Refer section 1.7.1)* **(5 Marks)**

☞ **Syllabus Topic : Quality Management through Statistical Process Control**

Q. 10 How can quality be managed using statistical process control? *(Ans. : Refer section 1.8)* **(5 Marks)**

☞ **Syllabus Topic : Quality Management through Cultural Change**

Q. 11 Explain the statement: Cultural change in an organization can increase quality of a product. *(Ans. : Refer section 1.8.1)* **(5 Marks)**

☞ **Syllabus Topic : Continual(Continuous) Improvement Cycle**

Q. 12 How can we continuously improve quality? *(Ans. : Refer section 1.9)* **(5 Marks)**

☞ **Syllabus Topic : Quality in Different Areas**

Q. 13 Explain how quality affects different sectors? *(Ans. : Refer section 1.9.1)* **(5 Marks)**

☞ **Syllabus Topic : Benchmarking**

Q. 14 Define Benchmarking. List its advantages. *(Ans. : Refer section 1.10)* **(5 Marks)**

☞ **Syllabus Topic : Metrics**

Q. 15 Define metrics in software testing with some examples. *(Ans. : Refer section 1.11)* **(5 Marks)**

☞ **Syllabus Topic : Problem Solving Techniques**

Q. 16 Explain some problem solving techniques. *(Ans. : Refer section 1.12)* **(5 Marks)**

☞ **Syllabus Topic : Problem Solving Software Tools**

Q. 17 What is a flowchart? *(Ans. : Refer section 1.12.1)* **(5 Marks)**

Q. 18 Explain Fishbone chart. *(Ans. : Refer section 1.12.1)* **(5 Marks)**

☞ **Syllabus Topic : Introduction to Software Quality**

Q. 19 What is software quality? *(Ans. : Refer section 1.13)* **(5 Marks)**

☞ **Syllabus Topic : Constraints of Software Product Quality Assessment**

Q. 20 What affects quality assessment of software products? *(Ans. : Refer section 1.13.1)* **(5 Marks)**

☞ **Syllabus Topic : Customer is King**

Q. 21 Explain - Customer is king. *(Ans. : Refer section 1.13.2)* **(5 Marks)**

☞ **Syllabus Topic : Quality and Productivity Relationship**

Q. 22 How are quality and productivity related? *(Ans. : Refer section 1.13.3)* **(5 Marks)**

☞ **Syllabus Topic : Requirements of a Product**

Q. 23 What requirements should a product fulfil? *(Ans. : Refer section 1.13.4)* **(5 Marks)**

☞ **Syllabus Topic : Organisation Culture**

Q. 24 Explain Organisation Culture. *(Ans. : Refer section 1.14)* **(5 Marks)**

☞ **Syllabus Topic : Characteristics of Software**

Q. 25 What are characteristics of software? *(Ans. : Refer section 1.14.1)* **(5 Marks)**

☞ **Syllabus Topic : Software Development Process**

Q. 26 Explain SDLC. *(Ans. : Refer section 1.15)* **(5 Marks)**

☞ **Syllabus Topic : Types of Products**

Q. 27 What are categories of a software? *(Ans. : Refer section 1.15.1)* **(5 Marks)**

☞ **Syllabus Topic : Schemes of Criticality Definitions**

Q. 28 What is severity of a defect? *(Ans. : Refer section 1.16)* **(5 Marks)**

☞ **Syllabus Topic : Problematic Areas of Software Development Life Cycle**

Q. 29 What are problematic areas of software development? *(Ans. : Refer section 1.16.1)* **(5 Marks)**

☞ **Syllabus Topic : Software Quality Management**

Q. 30 Differentiate between Quality Assurance and Quality Control. *(Ans. : Refer section 1.17)* **(5 Marks)**

Q. 31 What is Quality Planning? *(Ans. : Refer section 1.17)* **(5 Marks)**

☞ **Syllabus Topic : Why Software Has Defects?**

Q. 32 Why does defect exists in a software? *(Ans. : Refer section 1.17.1)* **(5 Marks)**

☞ **Syllabus Topic : Processes Related to Software Quality**

Q. 33 Write a short note on Software Quality Assurance. *(Ans. : Refer section 1.18)* **(5 Marks)**

Q. 34 Does the software adequately meet its quality factors ? *(Ans. : Refer section 1.18)* **(5 Marks)**

Q. 35 Has software development been conducted according to pre-established standards? *(Ans. : Refer section 1.18)* **(5 Marks)**

Q. 36 Have technical disciplines performed their SQA roles properly? *(Ans. : Refer section 1.18)* **(5 Marks)**

☞ **Syllabus Topic : Quality Management System Structure**

Q. 37 What is QMS structure? *(Ans. : Refer section 1.19)* **(5 Marks)**

☞ **Syllabus Topic : Pillars of Quality Management System**

Q. 38 What are pillars of Quality Management System? *(Ans. : Refer section 1.19.1)* **(5 Marks)**

☞ **Syllabus Topic : Important Aspects of Quality Management**

Q. 39 Explain important aspects of quality Management. *(Ans. : Refer section 1.19.2)* **(5 Marks)**

❑❑❑