UNIT IV

CHAPTER 4

Software Quality Assurance and Quality Control

University Prescribed Syllabus for the Academic Year 2022-2023

Software Quality Assurance : Introduction, Constraints of Software Product Quality Assessment, Quality and Productivity Relationship, Requirements of a Product, Characteristics of Software, Software Development Process, Types of Products, Schemes of Criticality Definitions, Software Quality Management, Why Software Has Defects? Processes Related to Software Quality, Quality Management System Structure, Pillars of Quality Management System, Important Aspects of Quality Management.

Software Quality Control : Software quality models, Quality measurement and metrics, Quality plan, implementation and documentation, Quality tools including CASE tools, Quality control and reliability of quality process, Quality management system models, Complexity metrics and

Customer Satisfaction, International quality standards – ISO, CMM

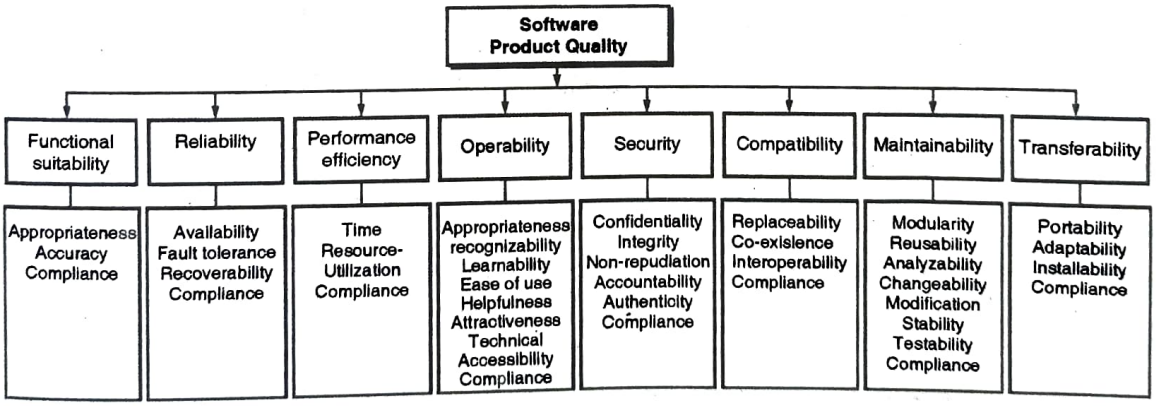
4.1 SOFTWARE QUALITY ASSURANCE -INTRODUCTION

The degree to which a scheme, component, or process chances definite requirements.

1. Quality is suitability for use.
2. Conformance to specification.
3. Transparency of service delivery
4. Achieving desired results
5. Continuous Improvement
6. Competitive advantage
7. Added value for society Best value for price
8. Cost effectiveness
9. Performance measurement
10. Satisfaction of stakeholders
11. Doing the correct things
12. Doing things right & doing the right effects right

4.1.1 Quality in Software Engineering

- In broader terms, the software quality definition of "fitness for purpose" refers to the satisfaction of requirements. But what are requirements? Necessities, also called user levels in today's agile terms, can be considered as practical and non-functional. Functional requirements mention to precise functions that the software should be able to complete.
- For example, the facility to print on an HP Inkjet 2330 printer is a useful prerequisite. However, just since the software has a confident function or a user can comprehensive a task consuming the software, does not mean that the software is of respectable quality.
- There are probably many instances where you've used software and it did what it was supposed to do, such as find you a flight or make a hotel reservation, but you thought it was poor quality. This is because of how the function was executed. The displeasure with "how" signifies the non-functional necessities not being met.
- For this purpose the International Organization for Standardization (ISO) developed as a model for specifying non-functional necessities. The classical shown below exemplifies the classification of non-functional requirements.



(1A10)**Fig. 4.1.1 : Software product Quality Requirements and Evaluation (SQuaRE) Quality model and guide**

Why Non-Functional Quality Components Are Important ?

Adequate non-functional necessities such as presentation, ease of use and learn aptitude first necessitates requiring and important. Only then can they be contented, and sufficient them can be even more problematic than substantial functional necessities. So, what does this mean for you? Let's examine the following non-functional characteristics using the same printing example :

- **Functional Suitability (functional appropriateness)** Does the function facilitate the completion of the user's task(s) and objectives? If the user doesn't want to print on that printer or wants to print a PDF but isn't given those options, then maybe not.
- **Performance Efficiency (time behaviour)** – Does the printer purpose return inside three seconds?
- **Compatibility(interoperability)** – Can the operator print over a variety of networks and printers and on processors with unlike operative schemes(Windows and Mac)?
- **Usability (learnability)** – Can the operative figure out in what way to print or will it income a rocket expert?
- **Consistency (recoverability)** – When the printer is released in the internal of printing a task, is the user learned?
- **Security (nonrepudiation)** – Is contiguous a record that the printer published the file efficiently?
- **Maintainability (testability)** – Can test events be measured for the print function?
- **Portability (adaptability)** – Can the software mechanically familiarize to new printer models, or an update in printer driver software ? Can the print purpose deliver shortcuts for highly refined users?

Now that we have an accepting of non-functional requests, let's examine the excellence lifecycle in the diagram beneath. Looking at the three circles under, internal excellence represents quality that you wouldn't realize and is stately by internal possessions such as code feature. Exterior worth signifies what we have conversed above in the non-functional superiority model and is typically measured by actual execution of the code and examination of software behaviour.

4.2 CONSTRAINTS OF SOFTWARE PRODUCT QUALITY ASSESSMENT

Requirement specification is made by business analyst and system analyst. Tester may or may not have direct access to the customer and may get information though requirement statements, queries answered etc. either from customer or business analyst. There are few limitations of product quality assessment in this.

- Software is virtual in nature. Software products cannot be touched or heard.
- There is huge communication gap between users of software and developers/testers of the product.
- Software product is unique in nature. Similarities between any two products are superficial ones.
- All aspects of software cannot be tested fully as member of permutations and combinations for testing all possibilities tend to infinity.
- A software program in the same way every time when it is executing some instruction.

Quality Tool

- Quality function deployment (QFD)
- Process charts
- Taguchi techniques
- Cause & effect diagrams
- Pareto charts
- Statistical process control (SPC)

4.3 QUALITY & PRODUCTIVITY RELATIONSHIP

- Productivity is the relationship between a given amount of output and the amount of input needed to produce it. Quality affects productivity.
- Productivity is tool of quantity that defines the competence of the association in relations of the ratio of output created with esteem to inputs used.
- Quality must improve productivity by reducing wastage.
- Improvement in quality directly leads to improved productivity.

- Cost reduction is possible by improved quality.
- Proper communication between management and employee is essential.
- Quality improvement leads to cost reduction.
- Employee involvement in quality improvement.

4.4 REQUIREMENTS OF PRODUCT

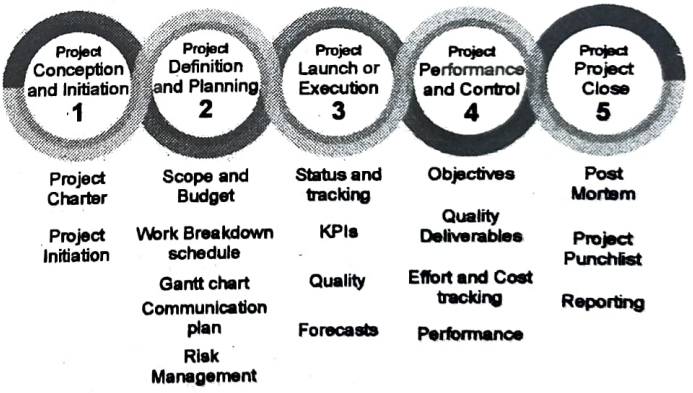
1. **Stated/Implied Requirements:** Functional and non-functional requirements stated by customer.
2. **General/Specific Requirement :** Requirements are generic in nature.
3. **Present/Future Requirements :** Present when application is used and future for required after some time span.
4. **Primary Requirements:** Must /must not be requirements. Customer pay for this
5. **Secondary Requirements:** Should/Should not be requirements.
6. **Tertiary Requirements:** Could/could not be requirements.

4.5 CHARACTERISTICS OF SOFTWARE

Quality culture is set of group standards that guide how developments are made to average working practices and resultant outputs. An organization's values can support entities at all levels make improved and more accountable choices involving issues of quality.

4.5.1 Following Features Emerged as Indicative of a Quality Culture Project Management

- Academic Ownership of quality.
- Quality culture is primarily about the behavior of stakeholders rather than the operation of a quality.
- A quality culture places students as center.
- Quality policy is more like a philosophy of any organization. Unless it is implemented, it is of no value.
- Quality policy is like a recipe book with a list of ingredients but no cooking instructions. Thus, any quality initiative needs to be managed as a project.
- A quality product project can use various tools, techniques, methodologies of project management. The success of a quality program depends the way it has been implemented.



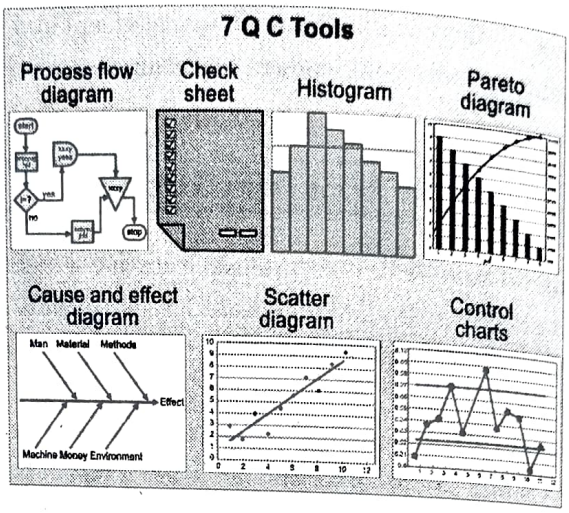
(1A11)**Fig. 4.5.1 : Steps in Project Management**

4.5.2 Sustainability and Quality Management

- Process efficiency, quality metrics, reduced waste etc. have started fascinating Sustainability practitioners nowadays. Sustainability teams are undergoing training & certifications to make themselves conversant with quality management.
- They are keen to use various tools & techniques of quality management in sustainability.

Seven Quality Control Tools

- Like Quality, Sustainability also has a strong attention on people. It takes into account quality of working life and employee satisfaction also.
- ISO 26000 brands a more thoughtful joining between people and excellence organization systems.
- Quality management is going to create value in Sustainability space in a big way.
- Thus, the trend of convergence between quality management and Sustainability in the offing.



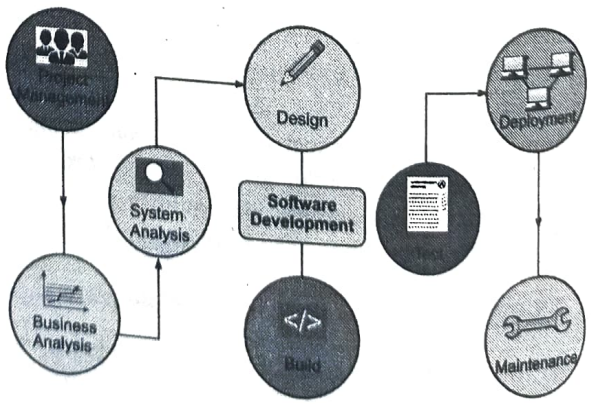
(1A12)**Fig. 4.5.2 : 7 QC Tools**

4.6 SOFTWARE DEVELPOMENT PROCESS

UQ. What is impact of defect in different phases of software development ?

SPPU - Nov./Dec.19 (End Sem)

- When we talk about the software development process, it requires establishing a connection between the idea and the approach to implement that idea.
- Thus, a proper process needs to be followed while implementing that idea for software development process. Over the years, the evolution of software development process came into existence for building customers business to reach success level.
- Since its evolution, the software industry is growing at a very fast pace because of the global level increase in market demand for software. Many players entered in the software industry and offered various IT solutions to their clients.
- Many companies around the globe design custom software to address their client's particular business problem, called custom software development services.
- Businesses introduced software application development services into the market and claimed to be the best quality provider of that service.
- Software industry rectifies the challenges that have been taken to identify the gap and risk in managing software development.
- These steps are taken to confirm the quality of software processing at each level. It basically refers to attempt for creating the software application to meet the need of client's business.



(1A13)**Fig. 4.6.1 : The process of Software Development**

4.6.1 Different Phases Included in Software Development Process

When it emanates to software development services, there are several approaches and models included in it. Here, we are conversing few major periods of software development methodologies :

Phases in Software Development

1. **Considerate the requirement :** In order to grow fully useful software, one necessity obviously appreciate the necessities of its customers. This is the most significant concern that should be taken, before start planning & working on the entire development process. A developer faces many challenges & alterations before coming up with the final plan as per the requirement & brief shared by the customer.
2. **Feasibility Analysis :** This phase involves scrutinizing the project whether it is feasible to work or not. Hence, you must build a strong interconnection between the project requirements and the need of your customer's project.
3. **Design :** Design plays a very imperative role in attracting visitors and generating more traffic. This includes production the complete architecture of the software. Henceforth, the design must be formed highly creatively and exclusively so that regulars can get best consequences.
4. **Coding :** This period contracts with the designer or programmers. It completely be contingent upon customer's choice in which programming language do they need to establish their project. It involves the transformation of design into coding through the help of a programmer.
5. **Software Testing :** Once the code is generated; it undergoes through various testing phases. This determines whether the product established is original or not. At this phase, any kind of bug or glitches found can be fixed.
6. **Maintenance :** Last but not the least, high maintenance is required in the project before & after it is being delivered to the user. The developer checks if the software is working fine before delivering it to the users & provides after sales service support & assistance in relations of the working of the software to the end users.

4.6.2 Seven Ways to Develop Software Development Process

- There are some ways through which the software development processes can be improved effectively such as:-
- Evolution in software development came with years of refined, tested and innovative processes. The demand for software development process has a view of the market and results in the growth of businesses.
- Life has never been better, therefore software provides human with the technology that helps in better and efficient accomplishment of tasks, functions, and goals.
  - No obligation for minuscule detail
  - Obtain the feedback from the user
  - Fewer people to attend meetings
  - Small projects and team
  - Empower your user
  - Implementation of features
  - Detain of price & flexibility

(1A14)**Fig. 4.6.2 : Seven ways to develop Software Development Process**

- Following Software Development Process Models are used :
  1. Waterfall Development Approach/Model.
  2. Iterative Development Approach/Model
  3. Incremental Development Approach/Model
  4. Spiral Development Approach/Model
  5. Prototyping Development Approach/Model
  6. Rapid Development Approach/Model
  7. Agile Development Approach/Model

4.6.3 Types of Product

- Life Affecting Products.
- Product Affecting Huge Some of Money.
- Products Which can be Tested only by Simulators: Example Space Research
- Other Products.

4.7 SCHEMES OF CRITICALITY DEFINATIONS

From User's Perspective

Failure of product disrupts entire business. Products failure affects business.

From Developer's Perspective

This classification defines the complexity of the system on the basis of development capabilities required.

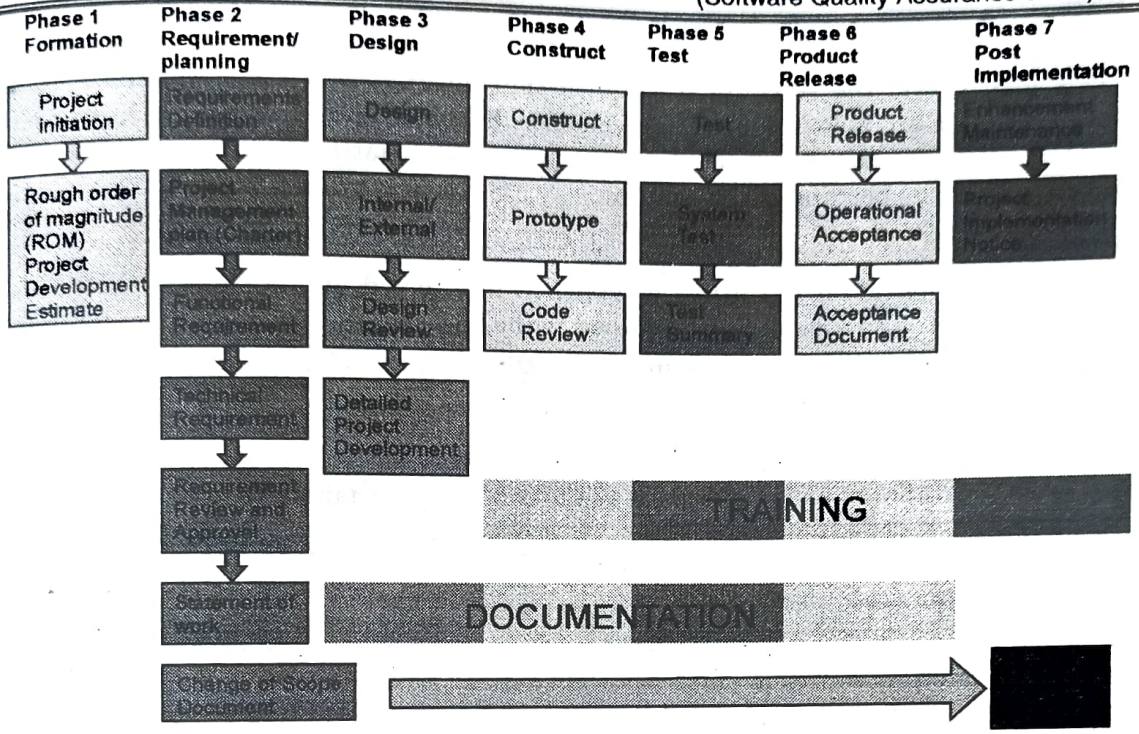
4.7.1 Problematic Areas of SDLC

Software Improvement Life Cycle Organizations

What is SDLC ?

- Every day, software engineers and specialists similar have to submerge themselves into the subtleties of the best **Software Development Lifecycle (SDLC)** procedure and method to progress and transport software in optimal environments. But, what is SDLC?
- In the meekest positions, SDLC practices deliver a methodical context to initiative, progress and offer software requests, from start to end. It is a sequence of phases that proposal a basis for the software growth process.
- Having a organization to grow software is important, which is why there are numerous software development organizations obtainable to choose from.
- It is gradually significant for software engineers to excellent the accurate **SDLC** model that meets precise necessities and concerns of the scheme to drive achievement. In this article, we go into the particulars of **SDLC** organizations, their significance, their benefits, difficulties, and everything in between.
- To a definite extent, **SDLC** organizations can be assumed of like a checklist of the dissimilar stages that must be completed to progress and deliver effective software requests.
- All **SDLC** procedures share a common ground of separate stages that include development, analysis, strategy, structure, difficult, organizing, and conservation. These **SDLC** stages provide the outline of what a software request project requires.

In the succeeding section, we are going to travel how software development lifecycles impact the software growth process.



(1A15)**Fig. 4.7.1 : SDLC phases**

☞ The Software Development Process

- The software development method, as with all excessive projects, starts with an idea. It earnings planning, preparation, and management of stages and team members to influence a goal.
- SDLC is a mapped out, planned context that classically surveys the following universal stages to transport high quality software application.

☞ Formation phase

This basic, initial phase is the beginning of an idea for a resolution that advances an existing solution or develops an entirely new one. It helps define the magnitude of the project to plan resources.

☞ Requirement/Planning Phase

In this phase, supplies are gathered to formulate a design plan for the software application solution. This phase involves a systematic examination to evaluate user requests, feasibility, growth, enhancements, and more. It is very significant to contain documentation to refine necessities and keep a record of the solution's development. This phase includes the formation of a project charter which describes technical and functional necessities.

☞ Design Phase

This phase is intensive on the design feature of the software application resolution in relations of the designated technical and functional necessities and the effects of the exhaustive enquiry of the software's feasibility.

☞ Development Phase

This phase is the meat of the software growth process. In this phase, software engineers are exclusively attentive on building an example of the solution to attain a code evaluation and finally create the solution itself. The team everything on converting software conditions into a employed and dependable solution.

Testing Phase

This serious phase tests the software to confirm that entirety works as it planned. In the analysis phase, software engineers are able to identify deficiencies, bacteria, and errors in the software solution and eventually have a quality product that meets business prospects. **Quality Assurance (QA)** experts perform a series of tests to assess the position of the solution.

☞ **Release Phase**

Once the software application is entirely established and tested, it transfers to the release phase. In this phase, the software goes aware and is unconfined to the end operator for definite use of the product.

In essence, the software is fully operative in a live setting where end users operate it.

☞ **Maintenance Phase**

- This post issue phase is tasked with observance the software entirely operational, informing it to meet value standards, and improving it through its life to confirm it remains to entice and recall users.
- The software development process sets the tone and describes a goal from which developer's kick-start a project. Eventually, succeeding a software development process is proposed to develop software earlier and with a few hiccups as possible.
- Now that we've enclosed the universal **SDLC** phases, let's measure how imperative it is to follow software development procedures in an IT environment.
  - Problems with requirement phase.
  - Requirements change very frequently.
  - Unique product is built in any time.
  - Product nature is intangible.
  - Inspection can be exhaustive.
  - Requirements are not easily communicated.
    - Technical Requirements.
    - Economical Requirements.
    - Operational Requirements.
    - System Requirements.

4.8 SOFTWARE QUALLITY MANAGEMENT

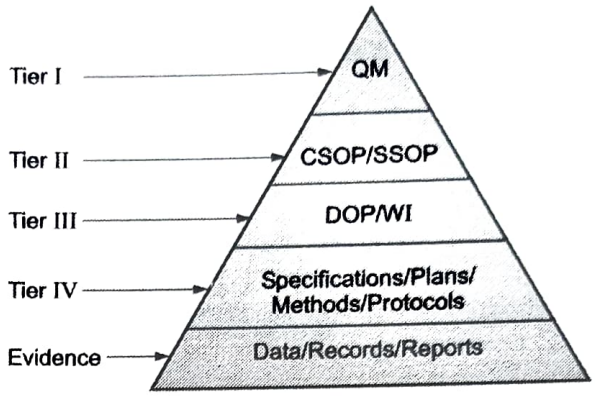
UQ. With respect Quality Management Systems explain the following ?
(i) Quality Management Systems Structure (ii) Pillars of Quality Management System
(iii) Important aspects of quality management **SPPU - Aug. 18(In Sem)**

- Quality management involves management of all inputs and processing defined so that the output from the process as per defined criteria.
- It handles three levels of problems :
  - Correction.
  - Corrective Actions.
  - Preventive Actions.
- Quality Management System Structure
  - 1st Tier-Quality policy
  - 2nd Tier-Quality objectives
  - 3rd Tier-Quality Manual

The Importance of Hierarchical Organization

An organization is spirited when working with controlled documents. A suggested hierarchy for QMS documentation management is:

1. Quality Manual
2. Policies
3. Procedures
4. Work Instructions
5. Lists
6. Checklists
7. Forms



(1A16)**Fig. 4.8.1 : QMS Tier structure**

4.9 PILLARS OF QUALITY MANAGEMENT SYSTEM

UQ. What are the pillars of Quality Management System ? SPPU - Oct. 19 (In Sem)

- Quality processes/Quality Procedures/work instructions
- Guidelines and standards
- Formats and Templates

☞ **Steps for the Creation of an Effective QMS**

The steps required for the conceptualization and implementation of a QMS include the following :

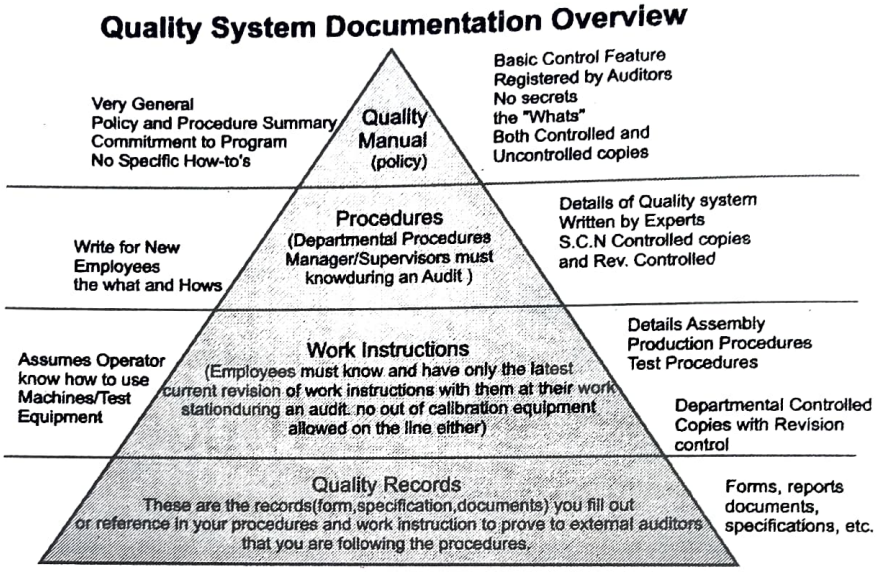
1. **Define and Map Your Processes**
   - Process maps creation will force the organization to visualize and define their processes. In the process, they will define the interaction sequence of those processes.
   - Process maps are vital for appreciating the responsible person. Define your main business process and converse the flow.
2. **Define Your Quality Policy**
   - Your Quality Policy communicates the duty of the organization as it is about the quality. The mission may be what customers need, a quality mission.
   - When constructing quality management system, consider the commitment towards customer focus. It may be Quality, Customer Satisfaction, and Continuous Improvement.
3. **Define Your Quality Objectives**

UQ. What are types of Requirements and Product? Also point out relationship between Quality and Productivity ? SPPU - Aug 18(In Sem)

All Quality management systems must have objectives. Each employee must appreciate their influence on quality. Quality objectives are derivative of your quality policy. It is measurable and set up throughout the organization. The objective may be in the form of critical success factors. This helps an organization in emphasizing the journey towards accomplishing its mission. These performance-based measures deliver a gauge to determine compliance with its objectives.

Some Critical Success Factors are:

1. Financial Performance
2. Product Quality
3. Process Improvement
4. Customer Satisfaction
5. Market Share
6. Employee Satisfaction



(1A17)**Fig. 4.9.1 : Quality System Documents (QSD) Overview Levels**

4. Develop Metrics to Track and Monitor CSF Data

- Once critical success factors are known, measurements and metrics keep track of advancement.
- This can be complete complete a data reporting procedure used to collect specific data. Share the processed information with leaders. A process goal is to improve customer approval index score.
- There requests to be a goal and a quantity to inaugurate realization of that goal.

4.10 IMPORTANT ASPECTS OF QUALITY MANAGEMENT

1. Quality planning at organisation level
2. Quality planning at project level
3. Resource management
4. Work Environment
5. Customer Related Processes
6. Quality management system document and data control
7. Verification and validation
8. Software project management
9. Software configuration management
10. Software metrics and measurement
11. Software Quality Audits

Software Quality Attributes are : Correctness, Reliability, Adequacy, Learn facility, Robustness, Maintainability, Readability, Extensibility, Testability, Competence, and Portability.

1. **Correctness :** The perfection of a software system mentions to:
   - Promise of program code with conditions
   - Individuality of the definite request of the software system.
   - The correctness of a program develops especially critical when it is implanted in a composite software system.

2. **Reliability :** Reliability of a software system develops from

(a) Correctness (b) Accessibility

- The conduct over time for the contentment of a given requirement be contingent on the consistency of the software system.
- Reliability of a software system is definite as the prospect that this system achieves a function (resolute by the provisions) for a definite number of input trials under detailed input situations in a definite time interval (presumptuous that hardware and input are allowed of errors).
- A software system can be seen as consistent if this test produces a low error rate (i.e., the possibility that an error will occur in a definite time interval.)
- The error rate is contingent on the frequency of inputs and on the possibility that a separate input will lead to an error.

3. **Acceptability :** Factors for the prerequisite of Capability:
   - The input compulsory of the user should be imperfect to only what is essential. The software system should assume information only if it is essential for the purposes that the user needs to transmit out.
   - The software system should allow plastic data input on the part of the user and should carry out credibility checks on the input. In dialog ambitious software systems, we vest specific standing in the consistency, clearness and ease of the interchanges.
   - The presentation accessible by the software system should be altered to the needs of the operator with the thought given to extensibility; i.e., the purposes should be incomplete to these in the requirement.
   - The results created by the software system : The outcomes that a software system distributes should be output in a clear and well organized form and be informal to understand.
   - The software system should afford the user flexibility with respect to the scope, the degree of detail, and the form of presentation of the results. Error messages must be provided in a form that is comprehensible for the user.
4. **Learn ability :** Learn ability of a software system depends on:
   - The design of user interfaces.
   - The clarity and the simplicity of the user instructions (tutorial or user manual).
   - The user interface should present information as close to reality as possible and permit efficient utilization of the software's failures.
   - The user manual should be structured clearly and simply and be free of all dead weight.
   - It should clarify to the user anything the software system should do, in what way the separate functions are motivated, what relations exist between functions, and which exclusions strength arise and how they can be modified.
   - In adding, the user guide should serve as a orientation that maintenances the user in quickly and securely definition the correct responses to queries.
5. **Robustness :** Robustness decreases the impact of working mistakes, specious input data, and hardware errors.
   - A software system is robust if the significances of an error in its process, in the input, or in the hardware, in relative to a given application, is contrariwise comparative to the possibility of the amount of this error in the given request.
   - Frequent errors (e.g. erroneous commands, typing errors) must be touched with particular care.
   - Less recurrent errors (e.g. power letdown) can be touched more laxly, but still must not lead to permanent consequences.

6. **Maintainability :** Maintainability = suitability for correcting (localization and correction of errors) and for modification and extension of functionality.

   The maintainability of a software system be contingent on its:

   - Readability
   - Extensibility
   - Testability

7 **Readability :** Readability of a software system be contingent on its:

   - Form of illustration
   - Programming style
   - Consistency
   - Readability of the application programming languages
   - Structureless of the system
   - Quality of the certification
   - Tools available for inspection

8. **Extensibility :** Extensibility sanctions compulsory alterations at the suitable locations to be made without undesirable side effects. Extensibility of a software system depends on its :

   - Structureless (modularity) of the software system
   - Opportunities that the application language delivers for this resolution
   - Readability (to find the suitable location) of the code
   - Availability of comprehensible program documentation

9. **Testability :** appropriateness for permitting the programmer to survey program implementation (runtime performance under given conditions) and for correcting.

10. **The testability** of a software system be contingent on its Modularity Structureness Modular; well-structured databases prove more appropriate for systematic, stepwise difficult than monumental, structured programs.

11. **Testing tools** and the opportunity of expressing constancy situations (declarations) in the source code decrease the testing effort and afford important fundamentals for the general, organized testing of all system mechanisms.

12 **Efficiency :** ability of a software system to accomplish its determination with the best possible operation of all essential properties (time, storage, transmission channels, and peripherals).

13. **Portability :** the easiness with which a software system can be altered to run on computers other than the one for which it was intended.

14. The movability of a software system be contingent on:

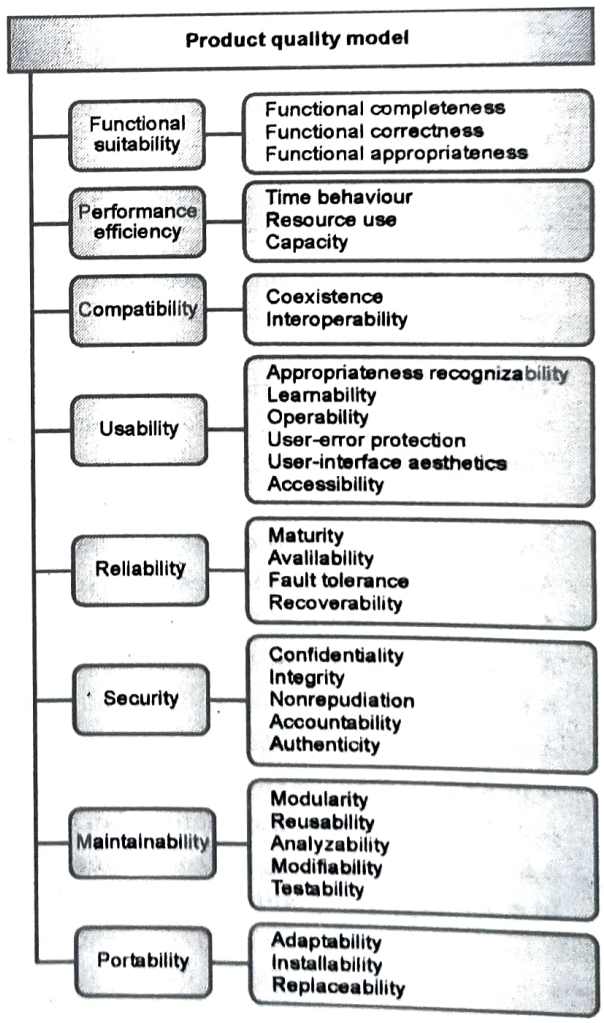
   - Degree of hardware independence
   - Implementation language
   - Magnitude of manipulation of dedicated system functions
   - Hardware possessions
   - Structureless : System needy elements are composed in easily substitutable program mechanisms.
   - A software system can be said to be transferable if the effort required for porting it shows meaningfully less than the effort essential for a new implementation.

4.11 SOFTWARE QUALITY CONTROL

- Quality control popularly abbreviated as QC. It is a Software Engineering process used to ensure quality in a product or a service. It does not deal with the processes used to create a product; rather it examines the quality of the "end products" and the final outcome.
- The main aim of Quality control is to check whether the products meet the specifications and requirements of the customer. If an issue or problem is identified, it needs to be fixed before delivery to the customer.
- QC also evaluates people on their quality level skill sets and imparts training and certifications. This evaluation is required for the service-based organization and helps provide "perfect" service to the customers.

4.12 SOFTWARE QUALITY MODEL

- This approach to software quality is best exemplified by fixed quality models, such as ISO/IEC 25010:2011. This standard describes a hierarchy of eight quality characteristics, each composed of sub-characteristics :
  1. Functional suitability
  2. Reliability
  3. Operability
  4. Performance efficiency
  5. Security
  6. Compatibility
  7. Maintainability
  8. Transferability
- Additionally, the standard defines a quality-in-use model composed of five characteristics :
  1. Effectiveness
  2. Efficiency
  3. Satisfaction
  4. Safety
  5. Usability
- A fixed software quality model is often helpful for considering an overall understanding of software quality.
- In practice, the relative importance of particular software characteristics typically depends on software domain, product type, and intended usage. Thus, software characteristics should be defined for, and used to guide the development of, each product.
- Quality function deployment provides a process for developing products based on characteristics derived from user needs.
- Software quality product is defined in term of its fitness of purpose. That is, a quality product does precisely what the users want it to do. For software products, the fitness of use is generally explained in terms of satisfaction of the requirements laid down in the SRS document.



(1E2) **Fig. 4.12.1 : Software Quality Model**

- Although "fitness of purpose" is a satisfactory interpretation of quality for many devices such as a car, a table fan, a grinding machine, etc. For software products, "fitness of purpose" is not a wholly satisfactory definition of quality.
- **Example :** Consider a functionally correct software product. That is, it performs all tasks as specified in the SRS document. But, has an almost unusable user interface. Even though it may be functionally right, we cannot consider it to be a quality product.
- The modern view of a quality associated with a software product several quality methods such as the following:

4.13 QUALITY MEASUREMENT AND METRICS

- **Portability :** A software device is said to be portable, if it can be freely made to work in various operating system environments, in multiple machines, with other software products, etc.
- **Usability :** A software product has better usability if various categories of users can easily invoke the functions of the product.
- **Reusability :** A software product has excellent reusability if different modules of the product can quickly be reused to develop new products.
- **Correctness :** A software product is correct if various requirements as specified in the SRS document have been correctly implemented.
- **Maintainability :** A software product is maintainable if bugs can be easily corrected as and when they show up, new tasks can be easily added to the product, and the functionalities of the product can be easily modified, etc.

4.14 QUALITY PLAN IMPLEMENTATION AND DOCUMENTATION

- Quality plan is an organized set of plans, policies, procedures, standards, and practices to achieve consistent, high-quality products and services for customers. It includes all aspects of product development, production, distribution, service delivery, and customer support.
- The quality plan provides a framework for managing quality within an organization. A quality plan should be developed by senior management with input from the entire organization.

The quality plan must include :

- A definition of what constitutes high-quality products or services.
- An explanation of how quality will be measured (quality objectives).
- A description of how quality will be improved through continuous improvement initiatives.
- A description and justification of the organizational structure and processes used to manage quality.
- A summary of the key stakeholders involved in the quality process.
- A description of the roles and responsibilities of each stakeholder group.
- A description of the tools and techniques used to measure and improve quality.
- A process to manage changes
- A description of how the quality plan will be communicated to employees.
- A statement indicating the commitment of the organization to achieving high quality.
- In summary, the Quality Plan is a roadmap to achieve quality.

☞ **The purpose of a Quality Plan**

Quality Plan

GQ. Defines quality expectations.

Focus on meeting customer expectations

- Establishes a mechanism to check compliance with the requirements. (inspection, tests and audits)
- Provides confidence to customers Sometimes customers ask to provide a copy of the Quality Plan for complex products or projects.
- A quality plan is a document, or several documents, that together specify quality standards, practices, resources, specifications, and the sequence of activities relevant to a particular product, service, project, or contract. Quality plans should define:
- Objectives to be attained (for example, characteristics or specifications, uniformity, effectiveness, aesthetics, cycle time, cost, natural resources, utilization, yield, dependability, and so on)
- Steps in the processes that constitute the operating practice or procedures of the organization
- Allocation of responsibilities, authority, and resources during the different phases of the process or project
- Specific documented standards, practices, procedures, and instructions to be applied
- Suitable testing, inspection, examination, and audit programs at appropriate stages
- A documented procedure for changes and modifications to a quality plan as a process is improved
- A method for measuring the achievement of the quality objectives
- Other actions necessary to meet the objectives
- At the highest level, quality goals and plans should be integrated with overall strategic plans of the organization. As organizational objectives and plans are deployed throughout the organization, each function fashions its own best way for contributing to the top-level goals and objectives.
- At lower levels, the quality plan assumes the role of an actionable plan. Such plans may take many different forms depending on the outcome they are to produce. Quality plans may also be represented by more than one type of document to produce a given outcome.

4.14.1 Quality Plan Documentation and Deployment

- Quality plans result from both deployed strategic quality policies (which are linked to organizational strategic plans) and from the specific legal regulations, industry standards, organization policies and procedures, internal guidelines, and good practices needed to meet customers' requirements for products or services.
- Strategic-level quality plans are developed and deployed through the strategic planning process. These broad-based quality plans become the guideline for each function's or department's supporting quality plan. Where appropriate, each function or department may develop and internally deploy operating-level quality plans.
- Operating-level quality plans often are the resulting document(s) from a production scheduling function. As such, this documentation often includes blueprints, a copy of the customer's order, references to applicable standards, practices, procedures, and work instructions, and details on how to produce the specific product or service.
- When the product or service is produced, the planning documents may be augmented by inspection documentation, statistical process control (SPC) charts, and copies of shipping documents and customer-required certifications.

- In the process, the plans are transformed from documents to records. In a fully computerized system, the documents mentioned may well be interactive computer screens accessed at operators' workplaces and control points. These screens, internally, become records when operators, inspectors, shippers, and others make computer entries to the screens.
- A completed set of matrices, developed by a quality function deployment (QFD) process, may fulfill a component of an organization's quality plan. The purpose of QFD is to capture and deploy the customers' needs and requirements throughout the organization.

Documenting the quality plan(s) has multiple uses, such as :

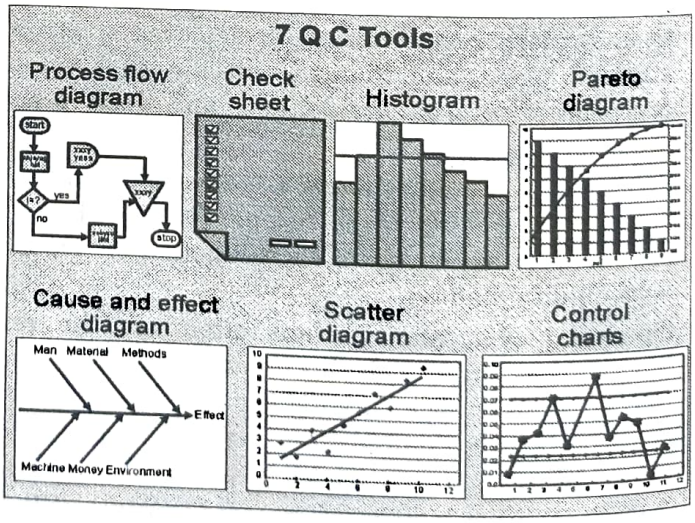
- Ensuring conformance to customer requirements
- Ensuring conformance to external and internal standards and procedures
- Facilitating traceability
- Providing objective evidence
- Furnishing a basis for training
- Together with multiple plans for the organization's products, services, and projects, providing a basis for evaluating the effectiveness and efficiency of the quality management system (QMS).

4.15 QUALITY TOOLS INCLUDING CASE TOOLS

UQ. What are 7 QC Tools and Modern Tools in detail ? (SPPU - May 18 Endsem)

UQ. What are Ishikawa's 7 basic tools ? (SPPU - May 18 Endsem)

- Dr. Kaoru Ishikawa was first total quality management guru, who has been associated with the development and advocacy of using the seven Quality Control (QC) tools in the organizations for problem solving and process improvements.
- Seven old quality control tools are a set of the QC tools that can be used for improving the performance of the production processes, from the first step of producing a product or service to the last stage of production. So, the general purpose of this paper was to introduce these 7 QC tools.
- This study found that these tools have the significant roles to monitor, obtain, analyze data for detecting and solving the problems of production processes, in order to facilitate the achievement of performance excellence in the organizations.
- There are seven basic quality tools, which can assist an organization for problem solving and process improvements.
- The first guru who proposed seven basic tools was Dr. Kaoru Ishikawa in 1968, by publishing a book entitled "Gemba no QC Shuho" that was concerned managing quality through techniques and practices for Japanese firms.
- It was intended to be applied for "self-study, training of employees by foremen or in QC reading groups in Japan.
- It is in this book that the seven basic quality control tools were first proposed. valuable resource when applying the seven basic tools (Omachonu and Ross, 2004).



(1D3) **Fig. 4.15.1 : Ishikawa7 Basic Tool for Quality Control**

- These seven basic quality control tools, which introduced by Dr. Ishikawa, are :
  1. Check sheets;
  2. Graphs (Trend Analysis);
  3. Histograms;
  4. Pareto charts;
  5. Cause-and-effect diagrams;
  6. Scatter diagrams;
  7. Control charts.
- Fig. 4.15.1 indicates the relationships among these seven tools and their utilizations for the identification and analysis of improvement of quality.

4.16 QUALITY CONTROL AND RELIABILITY OF QUALITY PROCESS

- Export of quality products is one of the important determinants of success in international marketing. Inferior quality of the product spoils the image of not only the product but also the nation. So, exporters should be quality conscious and their governments should insist on quality control.
- What is Quality Control ?
- Quality control is a deliberate and planned activity in order to determine the quality of a product with a view to accepting it as such. If it does not satisfy these requirements, then appropriate remedial measures are taken to correct the process or activity. Quality control is best exercised during the course of production of an article – actually starting with the raw materials, going through the various processing stages and ending up with the final product and paying due attention to packing, storage and transport.

Objectives of Quality Control

The important objectives of the quality control are as follows :

1. To promote and ensure the image of Indian goods exported to other countries.
2. To ensure goods of assured quality alone are sent to the export market.
3. To sustain foreign markets where Indian goods are well received and develop new markets with competitive edge.
4. To instill confidence in the minds of overseas buyers with the assurance provided by the third party guarantee.
5. To adhere strictly to technological requirements of the product accepted by foreign buyers.
6. To ensure sound and safe performance of the products without causing health hazards.
7. To observe conformity of rules and regulations of the importing countries.
8. To maintain proper packaging for the safety of the product during transit.
9. To eliminate the causes of complaints from the foreign buyers and to improve the overall quality of the Indian products.
10. To maximize production by achieving economies of scale.

Quality Standards in quality control

- Specification of quality is a must for quality control. Only when quality characteristics are assessed, specified and measured, quality control can be implemented. Generally, buyers give specifications of products which they intend to buy.
- However, standards are specified by quality control and inspection institutions such as, The Bureau of Indian Standards and also national standards of the importing country, such as International Standards Organization (ISO) and International Electro-chemical Commission.
- The exporters should conform to the specifications stated by these organizations. Exports of products are allowed only when the standards specified are complied with. Apart from this, the Export inspection council has laid down minimum standards for a number of products.

Methods of quality control and inspection

There are two methods of quality control, namely,

(1) In-process quality control and (2) Consignment-wise inspection.

(1) In-process quality control

- In-process quality control covers certain products like paints, and allied products, linoleum, ceramic sanitary ware, printing ink, chrome figments, etc. Under this method, the manufacturers themselves are responsible for producing export consignments conforming to the standard specification.
- Proper control should be exercised at various levels like raw materials control, process control, preservation control and packing control. Adequate controls are taken by periodic inspection and testing of export consignments at random.

(2) Reliability and Quality Control

- Reliability refers to the extent to which the product may perform its functions. A product's useful life span or the expected duration of its performance constitutes its reliability.
- There is a close relationship between reliability and quality control. In some cases, e.g., electronic generating stations, steel plants, etc., reliability is not only desirable but essential. Any downtime of a power plant or steel plant is very expensive.
- Reliability embraces the entire production cycle from product design to product use. Quality control involves checking of standards established during the design stage. But improved workmanship is possible only when production staff are motivated and take pride in workmanship.
- A well planned and organized quality control system starts controlling quality from design stage and continues to work until a trouble free product is supplied to the customer. Customer satisfaction can be achieved only when the customer is happy with the performance, reliability and serviceability of the product.
- Reliability can be measured as the probability of performing without failure, a specified function under given conditions for specified period of time.

4.17 QUALITY MANAGEMENT SYSTEM MODELS

- A quality management system (QMS) is defined as a formalized system that documents processes, procedures, and responsibilities for achieving quality policies and objectives. A QMS helps coordinate and direct an organization's activities to meet customer and regulatory requirements and improve its effectiveness and efficiency on a continuous basis.
- ISO 9001:2015, the international standard specifying requirements for quality management systems, is the most prominent approach to quality management systems. While some use the term "QMS" to describe the ISO 9001 standard or the group of documents detailing the QMS, it actually refers to the entirety of the system. The documents only serve to describe the system.

Benefits of Quality Management Systems

- Implementing a quality management system affects every aspect of an organization's performance. Benefits of a documented quality management system include:
- Meeting the customer's requirements, which helps to instill confidence in the organization, in turn leading to more customers, more sales, and more repeat business
- Meeting the organization's requirements, which ensures compliance with regulations and provision of products and services in the most cost- and resource-efficient manner, creating room for expansion, growth, and profit

These benefits offer additional advantages, including :

(1) Defining, improving, and controlling processes
(2) Reducing waste
(3) Preventing mistakes
(4) Lowering costs
(5) Facilitating and identifying training opportunities
(6) Engaging staff
(7) Setting organization-wide direction
(8) Communicating a readiness to produce consistent results

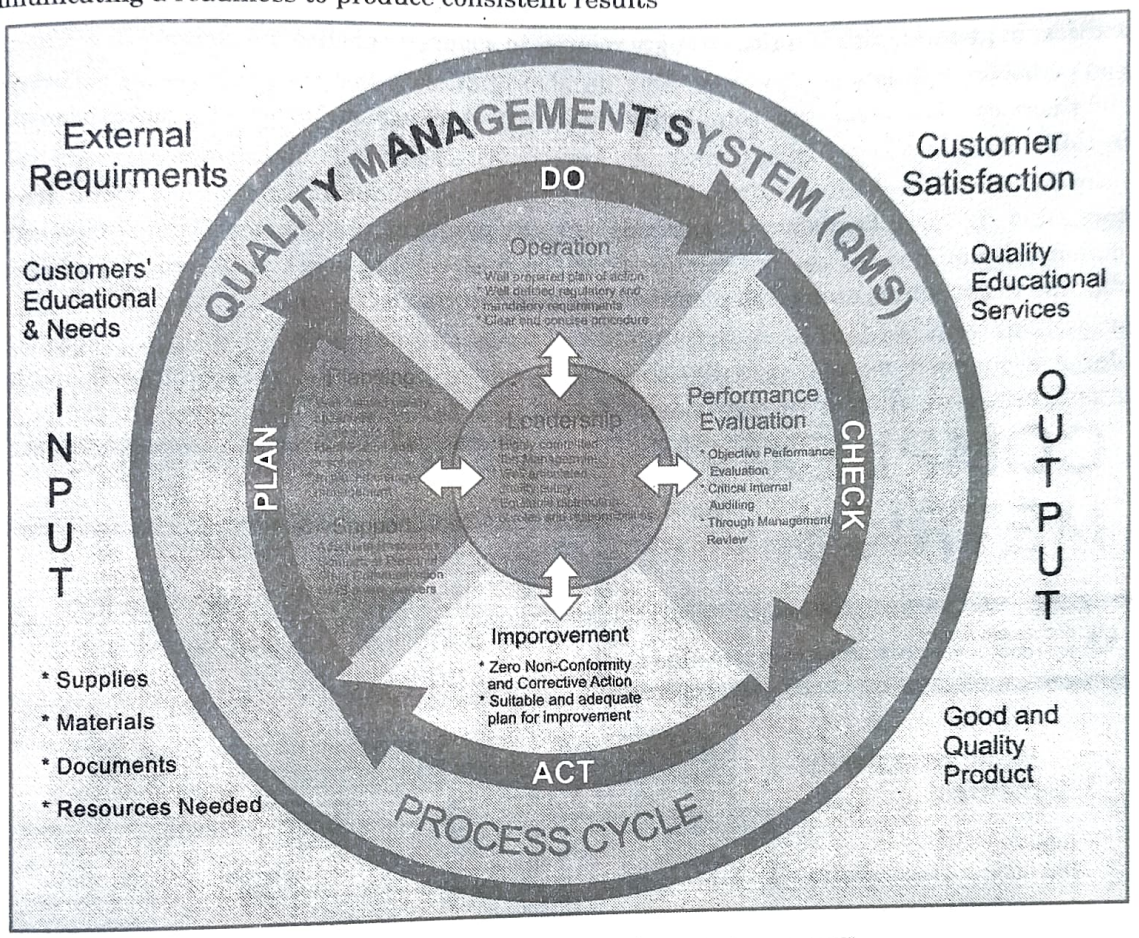


Fig. 4.17.1 : Quality Management System (QMS)

4.18 COMPLEXITY METRICS AND CUSTOMER SATISFACTION

- The productivity, if measured only in terms of lines of code per unit of time, can vary a lot depending on the complexity of the system to be developed.
- A programmer will produce a lesser amount of code for highly complex system programs, as compared to a simple application program. Similarly, complexity has a great impact on the cost of maintaining a program.
- To quantify complexity beyond the fuzzy notion of the ease with which a program can be constructed or comprehended, some metrics to measure the complexity of a program are needed.
- A complexity measure is a cyclomatic complexity in which the complexity of a module is the number of independent cycles in the flow graph of the module. A number of metrics have been proposed for quantifying the complexity of a program, and studies have been done to correlate the complexity with maintenance effort. In this article, we will discuss a few complexity measures. Most of these have been proposed in the context of programs, but they can be applied or adapted for detailed design as well.

- Size Measures: A complexity measure tries to capture the level of difficulty in understanding a module. In other words, it tries to quantify a cognitive aspect of a program. It is well known that, in general, the larger a module, the more difficult it is to comprehend.
- Hence, the size of a module can be taken as a simple measure of the complexity of the module. It can be seen that, on average, as the size of the module increases, the number of decisions in it is likely to increase. It means on average, as the size increases, the cyclomatic complexity also increases.
- Though it is clearly possible that two programs of the same size have substantially different complexities, in general, size is quite strongly related to some complexity measures.
- Halstead's Measure: Halstead also proposed a number of other measures based on his software science. Some of these can be considered complexity measures. A number of variables have been defined to explain this.
- These are n1 (number of unique operators), n2 (number of unique operands), N1 (total frequency of operators), and N2 (total frequency of operands). As any program must have at least two operators : one for function call and one for end of the statement the ratio n1/2 can be considered the relative level of difficulty due to the larger number of operators in the program.
- The ratio N2/n2 represents the average number of times an operand is used. In a program in which variables are changed more frequently, this ratio will be larger. As such programs are harder to understand, ease of reading or writing is defined as:

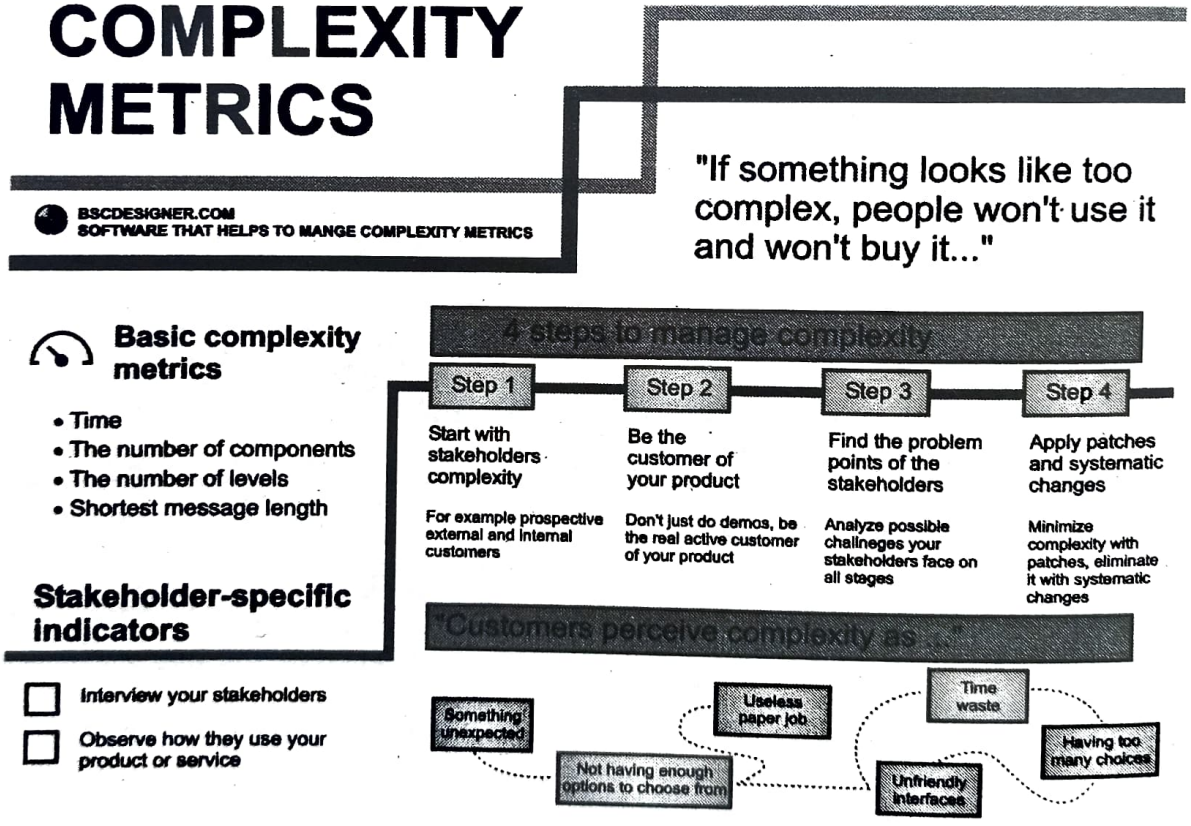


Fig. 4.18.1

- **Customer satisfaction metrics reveal crucial data that helps you understand how delighted customers are with your products, services, or the overall business. In other words, customer satisfaction metrics reveal the quality of a customer's relationship with your brand.**
- **It must be noted that customer metrics reveal more than just numerical scores. They reveal the valuable opinions, attitudes, behavioral patterns, and feedback of your audience.**

- Some examples of customer satisfaction KPIs include Customer Effort Score (CES), Net Promoter Score (NPS), Customer Churn Rate (CCR), etc.
- Although each and every metric has an important role to play, you need to ensure they align well with your business goals. For instance, if your goal is to create a simple and easy-to-use product, you can prioritize a metric such as the Customer Effort Score.

GQ. Why Do You Need to Measure Customer Satisfaction ?

- Customer satisfaction is no longer just a buzzword that brands use to attract customers. A great emphasis on customer satisfaction can give your business an edge over your competitors.
- Create the Right Culture of Communication & Feedback
- Modern customers want to do business with a brand that values their opinions and is always striving to improve.
- Monitoring customer satisfaction is a great way to keep customers engaged with your brand. It gives the message that your business is always open for communication and cares about the well-being of its customers.
- Moreover, with the help of customer feedback, you can get a good idea about their preferences. For instance, if CSAT surveys reveal that customers are more satisfied watching online video tutorials compared to blogs, then you can focus on creating more such videos.

☞ **Forecast Business Growth & Revenue**

- Happy and satisfied customers not just buy more but also act as brand ambassadors and recommend your products or services to others. In short, they do the job of your marketing team, and that too for free.
- According to a study by Temkin Group, around 77% of customers would recommend a brand to a friend after having a single positive experience.
- By adopting metrics such as the NPS, you can easily measure the likelihood of new and repeat business. You can identify the proportion of your customer base that is loyal to you and therefore, forecast cash flow, repeat purchases, referrals, and business growth.

4.19 INTERNATIONAL QUALITY STANDARDS – ISO, CMM

UQ. Write in brief ISO-9000 series, CMM, CMMI, Test Maturity Models ? SPPU - Aug. 18 Insem

GQ. What is the ISO 9001 Standard ?

- The ISO 9001 standard is a document that describes all of the requirements needed in order to create and maintain a quality management system as described in ISO 9000. This is a subtle difference between ISO 9000 and ISO 9001 that some fail to recognize.
- So, to explicitly point it out, the difference between the two (ISO 9000 vs 9001) is summarized as the definition of quality management system (ISO 9000) and requirements needed to meet that definition (ISO 9001).
- Both the ISO 9000 and 9001 standards are based on a number of quality management principles including a strong customer focus, the motivation, and implication of top management, the process approach and continual improvement.

The seven quality management principles include the following as described by the ISO :

1. **Customer focus :** Quality management primarily focuses on meeting customer requirements and striving to exceed customer expectations.
2. **Leadership :** Helping leaders to establish unity of purpose and direction at all levels and to create conditions to engage members of the organization in achieving the organization's quality objectives.

3. **Engagement of people :** Obtaining and maintaining (at all levels throughout the organization) competent, empowered, and engaged people to enhance the organization's capability to create and deliver value.
4. **Process approach :** Delivering consistent and predictable results through the use of effective and efficient activities that are understood and managed as interrelated processes that function as a coherent system.
5. **Improvement :** Maintaining an ongoing, organization-wide focus on improvement.
6. **Evidence-based decision making :** Using the analysis and evaluation of data and information in the decision making process to produce desired results.
7. **Relationship management :** Managing the organization's relationships with related parties, such as partners or vendors, for sustained success.

- You can learn more about the quality management principles from the ISO's 2015 publication. Also, a copy of ISO 9001:2015 can be purchased at the ISO online store.

4.19.1 Why ISO 9000 or 9001 ?

- One misconception is that ISO 9000 or 9001 is only for manufactures or large organizations. As a principles-based standard, ISO 9001 can be applied to any organization regardless of what type or size it may be.
- The standard defines the requirements, but it does not dictate the method of application.
- The latest version of the standard has been specifically designed to be more accessible to organizations outside the manufacturing sector.
- As with anything, there are ISO 9000/9001 pros and cons. The application of ISO 9001 when implementing a quality management system can provide the following benefits the organizations:
- Clear understanding of your objectives and new business opportunities. Identifying and addressing the risks associated with your organization. Renewed emphasis on putting your customers first. Meeting the necessary statutory and regulatory requirements. Organizational and process alignment to increase productivity and efficiency.

4.19.2 What is an ISO 9000 Certification ?

- If you are researching the ISO 9000 requirements or how to become ISO 9000 certified, you should really be focused on ISO 9001. You see an organization cannot become ISO 9000 certified.
- First issued in 1987 and last updated in 2015, ISO 9001 is the standard that sets out the criteria for a quality management system and is also the only standard within ISO 9000 that an organization can certify to. Therefore, it is incorrect to say that an organization is ISO 9000 compliant.
- However, a business can be ISO 9001 certified or compliant.
- While an ISO 9001 certification is not regulatory requirement, ISO reports that "over one million companies and organizations in over 170 countries have certified to ISO 9001."

An organization must demonstrate the following in order to be ISO 9001 certified :

- The company follows the guidelines within the ISO 9001 standard;
- The company meets its own requirements;
- The company meets its customer requirements and statutory and regulatory requirements; and
- The company maintains documentation of its performance.
- An ISO 9001 certification can enhance an organization's credibility as it shows customers that the organization's products and services meet quality expectations.

- Additionally, there are some instances where an ISO 9001 certification is required or legally mandated for businesses in some industries.

4.19.3 How to become ISO 9000 Certified ?

- The ISO 9001 certification process requires an organization to implement ISO 9001:2015 requirements. Once implemented, an organization must successfully complete registrar's audit to confirm that the organization system meets those requirements.
- The auditor will interview management and staff within the organization to determine whether or not they understand their role and responsibilities in complying with the ISO 9001 standards.
- The auditor will also examine the organization's documentation to validate compliance with the ISO 9001 requirements.
- The auditor will then prepare a detailed report that details the parts of the standard that the organization did not meet.
- The organization will need to agree to correct any problems within a specified time frame. The organization executes remedial activities to ensure that all problems are corrected.
- Once these gaps are addressed and confirmed by the auditor, the organization can then be certified.
- In order to maintain the ISO 9001 certification, the organization must continue with regular surveillance and recertification audits.
- Capability Maturity Model is used as a benchmark to measure the maturity of an organization's software process.
- CMM was developed at the Software engineering institute in the late 80's. It was developed as a result of a study financed by the U.S Air Force as a way to evaluate the work of subcontractors. Later based on the CMM-SW model created in 1991 to assess the maturity of software development, multiple other models are integrated with CMM-I they are

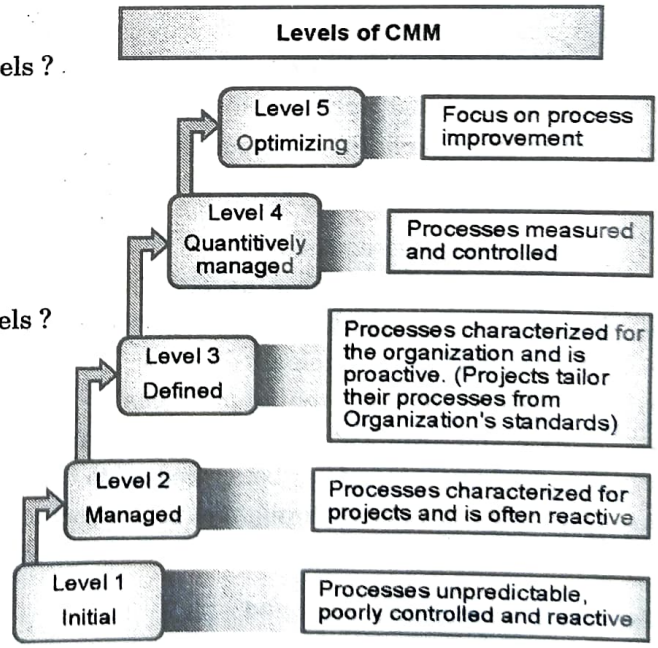
4.20 CAPABILITY MATURITY MODEL (CMM) & CMM LEVELS

In this tutorial, we will learn,

- What is Capability Maturity Model (CMM) Levels ?
- What happens at different levels of CMM?
- How long does it Take to Implement CMM?
- Internal Structure of CMM

Limitations of CMM Models

1. Why Use CMM ?
2. What is Capability Maturity Model (CMM) Levels ?
3. Initial
4. Repeatable/Managed
5. Defined
6. Quantitatively Managed
7. Optimizing



(1E3)**Fig. 4.20.1**

4.20.1 Capability Maturity Model (CMM) With Levels

GQ. What happens at different levels of CMM ?

Levels

- Activities
- Benefits

Level 1 Initial

- At level 1, the process is usually chaotic and ad hoc.
- A capability is characterized on the basis of the individuals and not of the organization.
- Progress not measured.
- Products developed are often schedule and over budget
- Wide variations in the schedule, cost, functionality, and quality targets.
- None. A project is Total Chaos.

Level 2 Managed

- Requirement Management
- Estimate project parameters like cost, schedule, and functionality.
- Measure actual progress.
- Develop plans and process.
- Software project standards are defined.
- Identify and control products, problem reports changes, etc.
- Processes may differ between projects.
- Processes become easier to comprehend.
- Managers and team members spend less time in explaining how things are done and more time in executing it.
- Projects are better estimated, better planned and more flexible.
- Quality is integrated into projects.
- Costing might be high initially but goes down overtime.
- Ask more paperwork and documentation.

Level-3 Defined

- Clarify customer requirements
- Solve design requirements, develop an implementation process.
- Makes sure that product meets the requirements and intended use.
- Analyse decisions systematically.
- Rectify and control potential problems.
- Process Improvement becomes the standard.
- Solution progresses from being "coded" to being "engineered".
- Quality gates appear throughout the project effort with the entire team involved in the process.
- Risks are mitigated and don't take the team by surprise.

Level-4 Quantitatively Managed

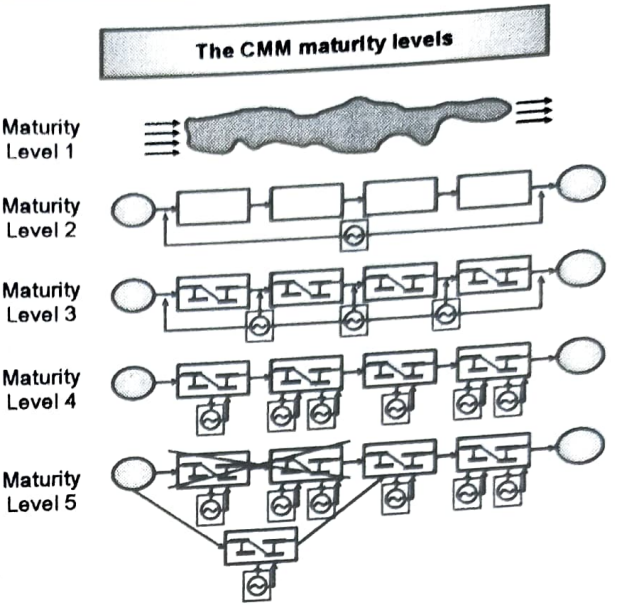
- Manages the project's processes and sub-processes statistically.

- Understand process performance, quantitatively manage the organization's project.
- Optimizes Process Performance across the organization
- Fosters Quantitative Project Management in an organization.

Level-5 Optimizing

- Detect and remove the cause of defects early.
- Identify and deploy new tools and process improvements to meet needs and business objectives.
- Fosters Organizational Innovation and Deployment.
- Gives impetus to Causal Analysis and Resolution.

Following diagram, gives a pictorial representation of what happens at different CMM level :



(1E4)**Fig. 4.20.2**

4.20.2 Capability Maturity Model (CMM) with Maturity Level

How long does it Take to Implement CMM ?

- CMM is the most desirable process to maintain the quality of the product for any software development company, but its implementation takes little longer than what is expected.
- CMM implementation does not occur overnight.
- It's just not merely a "paperwork."
- Typical times for implementation is
- 3-6 months -> for preparation
- 6-12 months -> for implementation
- 3 months -> for assessment preparation
- 12 months ->for each new level

4.20.3 Internal Structure of CMM

- Each level in CMM is defined into key process area or KPA, except for level-1.
- Each KPA defines a cluster of related activities, which when performed collectively achieves a set of goals considered vital for improving software capability
- For different CMM levels, there are set of KPA's, for instance for CMM model-2, KPA are
  - REQM - Requirement Management
  - PP - Project Planning
  - PMC - Project Monitoring and Control
  - SAM - Supplier Agreement Management
  - PPQA - Process and Quality Assurance
  - CM - Configuration Management
- Likewise, for other CMM models, you have specific KPA's. To know whether implementation of a KPA is effective, lasting and repeatable, it is mapped on following basis :
  - Commitment to perform

- Ability to perform
- Activities perform
- Measurement and Analysis
- Verifying implementation

4.20.4 Limitations of CMM Models

1. CMM determines what a process should address instead of how it should be implemented.
2. It does not explain every possibility of software process improvement.
3. It concentrates on software issues but does not consider strategic business planning, adopting technologies, establishing product line and managing human resources.
4. It does not tell on what kind of business an organization should be in CMM will not be useful in the project having a crisis right now

Why Use CMM ?

- Today CMM act as a "seal of approval" in the software industry. It helps in various ways to improve the software quality.
- It guides towards repeatable standard process and hence reduce the learning time on how to get things done.
- Practicing CMM means practicing standard protocol for development, which means it not only helps the team to save time but also gives a clear view of what to do and what to expect.
- The quality activities gel well with the project rather than thought of as a separate event.
- It acts as a commuter between the project and the team.
- CMM efforts are always towards the improvement of the process.

Summary

CMM was first introduced in late 80's in U.S Air Force to evaluate the work of subcontractors. Later on, with improved version, it was implemented to track the quality of the software development system. The entire CMM level is divided into five levels.

1. **Level 1 (Initial) :** Where requirements for the system are usually uncertain, misunderstood and uncontrolled. The process is usually chaotic and ad-hoc.
2. **Level 2 (Managed) :** Estimate project cost, schedule, and functionality. Software standards are defined.
3. **Level 3 (Defined) :** Makes sure that product meets the requirements and intended use.
4. **Level 4 (Quantitatively managed) :** Manages the project's processes and sub-processes statistically.
5. **Level 5 (Maturity) :** Identify and deploy new tools and process improvements to meet needs and business objectives.

...Chapter Ends

□□□