Anna University

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Software Project Management NOTES

Subject Name: IT2403 Software Project Management

Year / Sem : IV / VIII
UNIT 1 Software Quality

Contents:

· Views of Quality
· Hierarchical Modeling
· Boehm and McCall’s Models
· Quality Criteria
· Interrelation
· Measuring Quality
· Quality Metrics
· Overall Measures of Quality

Quality:

   Ability of the product/service to fulfill its function

   Hard to define

   Impossible to measure

   Easy to recognize in its absence

   Transparent when present

Definition of Quality:
Characteristics of Quality:

- Quality is not absolute
- Quality is multidimensional
- Quality is subject to constraints
- Quality is about acceptable compromises
- Quality criteria are not independent, but interact with each other causing conflicts.

Software Quality:

Kitchen ham (1989 b) refers to software quality “fitness for needs” and claims quality involves matching expectations.

Two features of a piece of quality software:

- Conformance to its specification
- Fitness for its intended purpose.
The Department of Defense (DOD, 1985) in the USA defines software quality as “the degree to which the attributes of the software enable it to perform its intended end use”.

Software was particularly problematical for the following reasons:

- Software has no physical existence
- The lack of knowledge of client needs at the start
- The change of client needs over time
- The rapid rate of change on both hardware and software
- The high expectations of customers, particularly with respect to adaptability.

Within the software quality area, the need to provide a solution that matches user needs is often considered as “design quality”, whilst ensuring a match to the specification is considered as “manufacturing quality”.

**Views of Quality:**

Quality is a multidimensional construct. It may therefore be considered using a polyhedron metaphor. Within this metaphor, a three-dimensional solid represents quality. Each face represents a different aspect of quality such as correctness, reliability, and efficiency.
It has been classified according to a number of ‘views’ or perspective. These views are often diverse and may conflict with each other. Each view comes from a particular context.

The views are generally presented in adversarial pairs such as versus designers.

The software project has the following roles

- Project manager
- Business analyst
- Implementation programmer
- Quality auditor
- End user
- Line manager
- Project sponsor

### Views of Quality

<table>
<thead>
<tr>
<th>User</th>
<th>Designer</th>
</tr>
</thead>
<tbody>
<tr>
<td>What I Want</td>
<td>Good Specification</td>
</tr>
<tr>
<td>Fast response</td>
<td>technically correct</td>
</tr>
<tr>
<td>Control information</td>
<td>Fits within systems structure</td>
</tr>
<tr>
<td>Easy to use help menus</td>
<td>Easy to maintain</td>
</tr>
<tr>
<td>Available as required</td>
<td>Difficult for user to manage</td>
</tr>
<tr>
<td>Exception data</td>
<td>Fast development</td>
</tr>
<tr>
<td>Reacts to business change</td>
<td>Low maintenance</td>
</tr>
<tr>
<td>Input data once</td>
<td>well documents</td>
</tr>
</tbody>
</table>
In an attempt to classify different and conflicting views of quality, Garvin (1984) has suggested five different views of quality

1. The transcendent view
   - Innate excellence
   - Classical definition

2. The product-based view
   - Higher the quality higher the cost
   - Greater functionality
   - Greater care in development

3. The user-based view
   - Fitness for purpose
   - Very hard to quantify

4. The manufacturing view
   - Measures quality in terms of conformance
   - Zero defects

5. The value-based view
   - Provides the data with what the customer requires at a price.

Quality is people:

Quality is determined by people because
· It is people and human organizations who have problems to be solved by computer software
· It is people who define the problems and specify the solutions
· It is still currently people who implement designs and product code.
· It is people who test code

**HIERARCHICAL MODEL OF QUALITY:**

To compare quality in different situations, both qualitatively and quantitatively, it is necessary to establish a model of quality.

Many model suggested for quality.

Most are hierarchical in nature.

A quantitative assessment is generally made, along with a more quantified assessment.

Two principal models of this type, one by Boehm (1978) and
one by McCall in 1977. A hierarchical model of software quality is based upon a set of quality criteria, each of which has a set of measures or metrics associated with it.

The issues relating to the criteria of quality are:

- What criteria of quality should be employed?
- How do they inter-relate?
- How may the associated metrics be combined into a meaningful overall measure of Quality?

THE HIERARCHICAL MODELS OF BOEHM AND MCCALL

THE GE MODEL (MCCALL, 1977 AND 1980) / (McCall Model)

- This model was first proposed by McCall in 1977.

- It was later revised as the MQ model, and it is aimed by system developers to be used during the development process.

- In early attempt to bridge the gap between users and developers, the criteria were chosen in an attempt to reflect user’s views as well as developer’s priorities.

- The criteria appear to be technically oriented, but they are described by a series of questions which define them in terms to non specialist managers.
The three areas addressed by McCall’s model (1977):

Product operation: requires that it can be learnt easily, operated efficiently And it results are those required by the users.

Product revision: it is concerned with error correction and Adaptation Of the system and it is most costly part of software development.

Product transition: it is an important application and it is distributed processing and the rapid rate of change in hardware is Likely to increase.

McCall’s criteria of quality defined

   Efficiency is concerned with the use of resources e.g. processor time, storage. It falls into two categories: execution efficiency and storage efficiency.

   Usability is the ease of use of the software.

   Integrity is the protection of the program from unauthorized access.

   Correctness is the extent to which a program fulfils its specification.

   Reliability is its ability not to fail.
Maintainability is the effort required to locate and fix a fault in the program within its operating environment.

Flexibility is the ease of making changes required by changes in the operating environment.

Testability is the ease of testing the programs, to ensure that it is error-free and meet its specification.

Portability is the effort required to transfer a program from one environment to another.

Reusability is the ease of refusing software in a different context.

Interoperability is the effort required to couple the system to another system.

The GE model after McCall (1977)

The Boehm model (1978)
It is to provide a set of well-defined, well-differentiated characteristics of software quality.

It is hierarchical in nature but the hierarchy is extended, so that quality criteria are subdivided.

According to the uses made of the system and they are classed into ‘general’ or ‘as is’ and the utilities are a subtype of the general utilities, to the product operation.

There are two levels of actual quality criteria, the intermediate level being further split into primitive characteristics which are amenable to measurement.

This model is based upon a much larger set of criteria than McCall’s model, but retains the same emphasis on technical criteria.

The two models share a number of common characteristics are,

- The quality criteria are supposedly based upon the user’s view.
- The models focus on the parts that designers can more readily analyze.
- Hierarchical models cannot be tested or validated. It cannot be shown that the metrics accurately reflect the criteria.
- The measurement of overall quality is achieved by a weighted summation of the characteristics.

Boehm talks of modifiability where McCall distinguishes expandability from adaptability and documentation, understandability and clarity.
HOW THE QUALITY CRITERIA INTERRELATE

The individual measure of software quality provided do not provide an over all measure of software quality.

The individual measures must be combined.

The individual measures of quality may conflict with each other.

Some of these relationships are described below;

________________________________________________________________________

Integrity vs. efficiency (inverse) the control of access to data or software requires additional code and processing leading to a longer runtime and additional storage requirement.

________________________________________________________________________

Usability vs. efficiency (inverse) Improvements in the human / computer interface may significantly increase the amount of code and power required.

________________________________________________________________________

Maintainability and testability vs. efficiency (inverse) Optimized and compact code is not easy to maintain.

www.Annauniversityplus.com
Portability vs. efficiency (inverse) the use of optimized software or system utilities will lead to decrease in probability.

Flexibility, reusability and interoperability vs. efficiency (inverse) the generally required for a flexible system, the use if interface routines and the modularity desirable for reusability will all decrease efficiency.

Flexibility and reusability vs. integrity (inverse) the general flexible data structures required for flexible and reusable software increase the security and protection problem.

Interoperability vs. integrity (inverse) Coupled system allow more avenues of access to more and different users.

Reusability vs. reliability (inverse) reusable software is required to be general: maintaining accuracy and error tolerance across all cases is difficult.

Maintainability vs. flexibility (direct) maintainable code arises from code that is well structured.

Maintainability vs. reusability (direct) well structured easily maintainable code is easier to reuse in other programs either as a library of routines or as code placed directly within another program.

Portability vs. reusability (direct) portable code is likely to be free of environment-specific features.

www.Annauniversityplus.com
Correctness vs. efficiency (neutral) the correctness of code, i.e. its conformance to specification does not influence its efficiency.

MEASURING SOFTWARE QUALITY

MEASURING QUALITY

Quality measurement, where it is considered at all, is usually expressed in terms of metrics.

Software metric is a measurable property which is an indicator of one or more of the quality criteria that we are seeking to measure. As such, there are a number of conditions that a quality metric must meet. It must:

- Be clearly linked to the quality criterion that it seeks to measure
- Be sensitive to the different degrees of the criterion
- Provide objective determination of the criterion that can be mapped onto a suitable scale.
- Metrics are not same as direct measures.

Measurement techniques applied to software are more akin to the social sciences, where properties are similarly complex and ambiguous.

A typically measurable property on which a metric may be based is structured ness.

The criteria of quality related to product revision, maintainability, adaptability and reusability are all related to structured ness of the source code.
Well-structured code will be easier to maintain or adapt than so called “spaghetti code”.

Structured ness as it simplest may be calculated in terms of the average length of code modules within the programs.

\[
\text{Structured ness} \propto \text{modularity} \times \frac{\text{lines of code}}{\text{Number of modules}}
\]

SOFTWARE METRICS

Metrics are classified into two types according to whether they are predictive or descriptive.

A predictive metric is used to make predictions about the software later in the lifecycle. Structured ness is used to predict the maintainability of the software product in use.

A descriptive metric describes the state of the software at the time of measurement.

Different authors have taken different approaches to metrics.

Structured ness is measured by questions such as:

- Have the rules for transfer of control between modules been followed?(y/n)
- Are modules limited in size?(y/n)
- Do all modules have only one exit point ?(y/n)
- Do all modules have only one entry point?(y/n)
A well-structured program will produce positive answers to such questions.

McCall’s approach is more quantities, using scores derived from equations such as

$$\text{McCall's structured ness metric} = \frac{n01}{ntot}$$

Where: $n01 =$ no of modules containing one or zero exit points only
$ntot =$ total number of modules

Generally, in this approach, scores are normalized to a range between 0 and 1, to allow for easier combination and comparison.

This appears attractive, to give unjustified credibility to the results obtained.

To validate this relationship and determine whether it is a linear relationship or more complex in nature.

It is also possible to validate whether the dependence of maintainability structured ness in identical to that of adaptability or reusability.

What makes a good metric?

Seven criteria for a good metric, after Watts (1987)
Objectivity the results should be free from subjective influences. It must not matter who the measurer is.

Reliability the results should be precise and repeatable.

Validity the metric must measure the correct characteristic.

Standardization the metric must be unambiguous and allow for comparison.

Comparability the metric must be comparable with other measures of the same Criterion.

Economy the simpler and therefore, the cheaper the measure is to use, the Better.

Usefulness the measure must address a need, not simply measure a property for its own sake.

A further important feature is consistency.

Automation is also desirable.

Metrics cited in the literature:

Metrics available for each criterion (after Watts, 1987) \[\]

<table>
<thead>
<tr>
<th>Quality criteria</th>
<th>number of metrics cited</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maintainability</td>
<td>18</td>
</tr>
<tr>
<td>Reliability</td>
<td>12</td>
</tr>
<tr>
<td>Usability</td>
<td>4</td>
</tr>
</tbody>
</table>

[www.Annauniversityplus.com](http://www.Annauniversityplus.com)
The metrics cited depends to a very large extent upon just seven distinct measurable properties: readability, error prediction, error detection, complexity, and mean time to failure (MTTF), modularity, testability.

1. Readability as a measure of usability may be applied to documentation in order to assess how such documentation may assist in the usability of a piece of software.

2. Error prediction as a measure of correctness this measure is depends upon the stable software development environment.

3. Error detection as measure of correctness

4. Mean time to failure (MTTF) as a measure of reliability

5. Complexity as a measure of reliability the assumption underpinning these measures is that as complexity increases, so reliability decrease.
6. Complexity as a measure of maintainability is also indicative of maintainability.

7. Readability of code as a measure of maintainability has also been suggested as a measure of maintainability.

8. Modularity as a measure of maintainability increased modularity is generally assumed to increase maintainability. Four measures have been suggested. Yau and Collofello (1979) measured “stability” as the number of modules affected by program modification. Kentger (1981) defined a four-level hierarchy of module types:

    _________________ Control modules.
    _________________ Problem-oriented modules.
    _________________ Management modules for abstract data.
    _________________ Realization modules for abstract data.

9. Testability as a measure of maintainability the ease and effectiveness of testing will have an impact upon the maintainability of a product.

An overall measure of quality

Much of the work in this area has been concerned with simple reduction of a set of scores to a single ‘figure-of-merit’.

www.Annauniversityplus.com
Five such methods are detailed by Watts (1987) as part of the MQ approach.

1. Simple scoring: In this method, each criterion is allocated a score. The overall quality is given by the mean of the individual scores.

2. Weighted scoring: This scheme allows the user to weight each criterion according to how important they consider them to be. Each criterion is evaluated to produce a score between 0 and 1. Each score is weighted before summation and the resulting figure reflects the relative importance if the different factors.

3. Phased weighting factor method: This is an extension of weighted scoring. A weighting is assigned to a group characteristics before each individual weighting is considered.

**Example of simple and weighted scoring methods**

<table>
<thead>
<tr>
<th>Quality criteria</th>
<th>metric</th>
<th>weight</th>
<th>product</th>
</tr>
</thead>
<tbody>
<tr>
<td>Usability</td>
<td>0.7</td>
<td>0.5</td>
<td>0.35</td>
</tr>
<tr>
<td>Security</td>
<td>0.6</td>
<td>0.2</td>
<td>0.12</td>
</tr>
<tr>
<td>Efficiency</td>
<td>0.4</td>
<td>0.2</td>
<td>0.08</td>
</tr>
<tr>
<td>Correctness</td>
<td>0.8</td>
<td>0.5</td>
<td>0.40</td>
</tr>
<tr>
<td>Reliability</td>
<td>0.6</td>
<td>0.4</td>
<td>0.24</td>
</tr>
<tr>
<td>Maintainability</td>
<td>0.6</td>
<td>0.4</td>
<td>0.24</td>
</tr>
<tr>
<td>Adaptability</td>
<td>0.7</td>
<td>0.1</td>
<td>0.07</td>
</tr>
<tr>
<td>Expandability</td>
<td>0.7</td>
<td>0.1</td>
<td>0.07</td>
</tr>
<tr>
<td>Total</td>
<td>5.10</td>
<td>2.40</td>
<td>1.57</td>
</tr>
</tbody>
</table>

Simple score = 5.10/8 = 0.64
Weighted score = 1.57/2.40 = 0.65
The phased weighting factor method

Product operation weighted mean = 0.660

Product transition weighted mean = 0.633

Overall measure by PWF method = \(((2/3) \times 0.660) + ((1/3) \times 0.633) = 0.65 \)

4. The Kepner-Tregoe method (1981): The criteria are divided into ‘essential’ and ‘desirable’. A minimum value is specified for each essential criterion and any software failing to reach these scores is designated unsuitable.

‘Suitable’ software is then judged by use of the weighting factor method.

5. The Cologne combination method (Schmitz, 1975): This method is designed with comparative evaluation in mind. Using the chosen criteria, each product is ranked in order.

POLARITY PROFILING:

In this scheme, quality is represented by series of ranges from -3 to +3.

The required quality may be represented and compared to the actual quality achieved.

It is a common problem amongst software developers that they focus upon particular aspects of quality.

When a user complains of poor quality, they tend to improve the product further in these areas.
Often the product has already exceeded the user’s expectations in these areas, and a further improvement does not improve their overall view of the quality of the product.

This effort wasted.

Worse, the user’s needs still have not been met in other critical areas, leading to tensions between the developers and users.

Two different outcomes result.

In the first case, usability is improved. Unfortunately, reliability and efficiency are still not up to the required standard, and usability was already considered satisfactory.

In the second case, improvements in reliability and efficiency are traded for a reduction in adaptability and maintainability, perhaps by ‘tweaking’ the code.

The consequence is that all criteria are now at the required level, resulting in an overall perception of quality and satisfied users.

UNIT – I TEST QUESTIONS

PART – A  (5 X 2 = 10 Marks)

1. Define software quality.
2. What are five different views of quality suggested by Garvin?
3. What are the common characteristics of both Boehm and McCall models?

www.Annauniversityplus.com
5. What makes a good metric?

PART – B

6. I Explain briefly about views of quality (7)
   II State and explain, McCall’s Model of quality (8)

(Or)

7. I Explain how the quality criterions are interrelated (8)
   II Explain briefly about overall measure of quality (8)

Question Bank:

2 Marks Questions:

1. Define Quality
2. Define software quality.
3. What are all the problems in measuring the quality of software?
4. What are the two quality factors that fall under software quality area?
5. What is design quality?
6. What is meant by manufacturing quality?
7. What should be the aim of any software production process?

9. Give some insights about quality.

10. What are five different views of quality suggested by Garvin?

11. What are the methodologies used in the manufacturer’s view?

12. What is value-based view?

13. What is the purpose of hierarchical modeling?

14. Give some examples of quality criteria employed in software quality.

15. What are the metrics associated with reliability?


17. Give any two examples of hierarchical models.

18. Write about GE model.

19. What are the three areas addressed by McCall’s model?

20. What are all McCall’s criteria of quality?

21. What is portability?

22. Define interoperability

23. Explain Boehm’s model.

24. What are the common characteristics of both Boehm and McCall models?

25. Give the interrelationships between quality criteria.
26. Give few relationships that have inverse relationships.

27. Give examples of direct relationship

28. Give an example of neutral relationship - suggest.

29. Correctness Vs efficiency - Differentiate

30. Define software metric.

31. What are two types of software metric?

32. What is meant by predictive metric?

33. Define descriptive metric.

34. What makes a good metric?

35. What is the objectivity criterion for a good metric?

36. Write down the problems with metrics

37. What are the methods that are used in overall measure of quality?

38. Explain the simple scoring method.

39. How is weighted scoring method used?

40. How does phased weighting factor method differ from weighted scoring method?

42. How is cologne combination method used?

43. What is the role of a project manager?

44. Define Structured ness

45. Who is called the implementation programmer?

46. What is the role of a quality auditor?

47. What do you mean by the transcendent view?

48. What is product - based view?
49. What does user-based view mean?

50. Why is software quality important?

51. Define Software quality assurance.

52. What are the five methods detailed by watts (1987) as part of the MQ approach?

53. Define Quality as defined by International Standard Organization.

54. What is meant by software quality?

55. Quality is conformance to requirements to both implicit and explicit. Explain the terms “explicit” and “implicit” requirements in the context of Garvin’s views of quality.

16 Marks Questions:

1) McCall suggest that simplicity, modularity, instrumentation and self descriptiveness are software quality criteria that is internal characteristics that promote external quality testability.

   (I). Explain each of the above four criteria.

   (II). Describe the possible measures for each of the criteria.

   (III). Describe the possible ways in which the measures could be assessed.

2) Explain Garvin’s five views of quality.

3) State and explain, McCal’ s Model of quality.

www.Annauniversityplus.com
4) Explain briefly about views of quality.

5) What are the quality metrics, measuring quality available? Explain.

6) Explain briefly about overall measure of quality.

7) Explain about the interrelation between quality criteria and software process.

8) Explain briefly about hierarchical modeling.

**UNIT II:-**

**The purpose of standard**

Standards are generally defined in terms of a model of a best practice against which all others are compared.

- It is used to build better products
- Also used to ensure whether the products are conformance to a standard

**Types of accreditation**

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>First party</td>
<td>internal monitoring</td>
</tr>
<tr>
<td>Second party</td>
<td>external monitoring by a customer</td>
</tr>
<tr>
<td>Third party</td>
<td>external monitoring by an independent standards body</td>
</tr>
</tbody>
</table>

**Benefits of accreditation**

- It provides external validation to see whether the investment made in the QMS is being effective
• It gives credibility to the supplier and their quality system
• It allows the supplier to sell to those customers who insist on accreditation as a condition of tender
• It qualifies the supplier to be included in the buyers guides

❖ The ISO9000 series: a generic quality management standard
   • The ISO9000 series of standards are the international standards defined for quality management systems.
   • The series dates from 1979, when BS 5750 r introduced in the UK. In 1987, the corresponding ISO, BS and EN standard were harmonized to produce three identical series of standards. In this text, shall use the ISO numbers for consistency.
   • Minor modifications were introduced in 1994. The corresponded European and British standards are given in Table 7.3, which also lists the future of each standard.
   • Applied within software development. ISO9002 is intended for many manufacturing situations where the product is produced to a predefined specification and ISO9 for easy applications where the quality can be determined by a simple final inspection and testing procedure.
   • ISO9000 provides guidance on which standard to adopt and ISO9004 assistance on how to establish a QMS which meets the requirements of the ISO9000 series.
The contents of the standard

- In this section, we shall deal with the requirements of the ISO 9001 standard.
- The ISO 9002 and ISO 9003 standards may be thought of as subsets of the ISO 900 standard, and in any case most software applications will require the full range of ISO 9001 activity.
- The standard is based around a model specification for a quality management System.
- This underlying model is based around two fundamental principles:
  - Right first time.
  - Fitness for purpose
- The standard is intended to be realistic and implementable and, therefore, sets no prescriptive quality performance targets, referring instead to standards agreed as part of the contract with the customer and acceptable to them.
- The standard focuses upon ensuring that procedures are carried out in a systematic. Way and that the results are documented, again in a systematic manner.
- The main requirements are dealt with in Clause 4 of the standard under 20 subclasses the headings of each sub clause in Clause 4 are summarized in Table.
- Those clauses also found in ISO 9002 and ISO 9003 are marked with a tick in the right-hand columns. It should be noted that in ISO 9003, some of the clauses are simplified Standards.
Table 7.4 Comparison of the requirements of the three principal standards

<table>
<thead>
<tr>
<th>Clause</th>
<th>ISO9001</th>
<th>ISO9002</th>
<th>ISO9003</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1</td>
<td>Management responsibility</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>4.2</td>
<td>Quality system</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>4.3</td>
<td>Contract review</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>4.4</td>
<td>Design control</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>4.5</td>
<td>Document control</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>4.6</td>
<td>Purchasing</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>4.7</td>
<td>Purchaser supplied product</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>4.8</td>
<td>Product identification and traceability</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>4.9</td>
<td>Process control</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>4.10</td>
<td>Inspection and testing</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>4.11</td>
<td>Inspection, measuring and testing equipment</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>4.12</td>
<td>Inspection and test status</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>4.13</td>
<td>Control of non-conforming product</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>4.14</td>
<td>Corrective action</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>4.15</td>
<td>Handling, storage, packaging and delivery</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>4.16</td>
<td>Quality records</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>4.17</td>
<td>Internal quality audits</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>4.18</td>
<td>Training</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>4.19</td>
<td>Servicing</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>4.20</td>
<td>Statistical techniques</td>
<td>✔</td>
<td>✔</td>
</tr>
</tbody>
</table>

The function of each section is detailed below.

**Clause 4.1: Management responsibility**

- The model recognizes the importance of management responsibility for quality throughout the organization whilst it is impossible for senior management to oversee everything personally, the standard explicitly provides for a management representative who is directly responsible for quality and is accountable to senior management.
- Clause also sets out the basic principles for establishing the quality system within the organization and sets out many of its function which are then described in greater detail in later sections.

**Clause 4.2: Quality system**

- The model requires the organization to set up a quality system. The system should be documented and a quality plan and manual prepared.
• The scope of the plan is determined by the activities undertake and consequently the standard (1S09001/213) employed.
• Focus of the plan should be to ensure that activities are carried out in a systematic way and documented

Clause 4.3: Contract review

Contract review specifies that each customer order should be regarded as a contract. Order entry procedures should be developed and documented aim of these procedures is to:
• Ensure that customer requirements are clearly defined in writing.
• Highlight differences between the order and the original quotation, so that they may be agreed.
• Ensure that the requirements can be met.
• The aim of this clause is to ensure that both the supplier and customer understand the specified requirements of each order and to document this agreed specification to prevent misunderstandings and conflict at a later date

Clause 4.4: Design control

Design control procedures are required to control and verify design activity to take the results from market research through to practical designs. Key activities covered are: Planning for research and development.
• Assignment of activities to qualified staff.
• Identify interfaces between relevant groups.
• Preparation of a design brief.
• Production of technical data.
• Verification that the outputs for the design phase meets the input requirements.
• Identification and documentation of all changes and modifications.

The aim of this section is to ensure that the design phase is carried out effectively and to ensure that the output from the design phase accurately reflects the input requirements. The importance of this process in a software context cannot be underestimated.
Clause 4.5: Document control

Three levels of documentation are recognized by the standard:

- **Level 1**: planning and policy documents.
- **Level 2**: procedures.
- **Level 3**: detailed instructions

- The top level documents the quality plan and sets out policy on key quality issues.
- Level adds more detail to the documentation. Where possible, existing documentation should be incorporated.
- The aim should be to provide systematic documentation, rather than simply to provide more documents.
- It is important that each level of documentation is consistent with the one above it, providing greater detail as each level is descended.
- It is a common complaint that the standard requires a prohibitive amount of documentation to be produced.
- Supporters of the standard argue that systematizing of documentation can actually lead to a reduction in volume due to the removal of obsolete and surplus documents.
- It is more likely that some reduction will be achieved, which will offset greater volumes in other areas.
- Good existing documentation should be incorporated into any new stem and this is facilitated by the standard not specifying a particular form but merely specifying that documents be fit for their intended purpose.

Clause 4.6: Purchasing

- The purchasing system is designed to ensure that all purchased products and services conform to the requirements and standards of the organization.
- The emphasis is placed on verifying the supplier’s own quality main procedures.
Here a supplier has also obtained external accreditation for their quality management systems, checks may be considerably simplified. As with all procedures, they should be documented.

**Clause 4.7: Purchaser-supplied product**

- CMI Services and products supplied by the customer must be checked for suitability, in the same way as supplies purchased from any other supplier.
- In order to ensure this, procedures should be put in place and documented, so that these services and products may be traced through all processes and storage.

**Clause 4.8: Product identification and traceability**

- Ensure effective process control and to correct any non-conformance, it is necessary to establish procedures to identify and trace materials from input to output.
- This also enables quality problems to be traced to root causes may be that a problem can be traced back to supplied materials, in which case the problem may lie outside the quality system altogether.

**Clause 4.9: Process control**

- Process control requires a detailed knowledge of the process itself. This must be documented, often in graphical form, as a process flow chart or similar.
- Procedures for setting up or calibration must also be recorded.
- Documented instructions should be available to staff to ensure that they have the capability to carry out the task as specified.
- It is staggering how often organizations do not understand their own processes properly.
- The discipline of documenting the actual process precisely and unambiguously for accreditation purposes can be very educational.
Clause 4.10: Inspection and testing

Inspection and testing are required to ensure conformance in three stages:

- Incoming materials or services.
- In process.
- Finished product and/or service.

- All incoming supplies must be checked in some way. The method will vary according to the status of the supplier’s quality management procedures, from full examination to checking evidence supplied with the goods.
- Monitoring in processes is required to ensure that all is going according to plan.
- At the end of the process, any final inspection tests documented in the quality plan must be carried out. Evidence of conformity to quality standards, together with details of any supporting in-process monitoring may be included. In an effective system, however, the final inspection and test should not have to be as extensive as it otherwise would be.
- In addition, it should not reveal many problems, since they should have been eliminated by this stage.

Clause 4.11: Inspection, measuring and testing equipment and maintained.

- Procedures to ensure that calibration and maintenance activities are properly implemented should be documented, identifying the measurements required and the precision associated with each.
- Records must be kept of all activity.
- Checking and calibration activities should become part of regular maintenance.
- Management should ensure that checks are carried out at the prescribed intervals and efficient records kept.
Clause 4.12: Inspection and testing status

All material and services may be classified in one of three categories:
• Awaiting inspection or test.
• Passed inspection.
• Failed inspection.

This status should be clearly identifiable at any stage. It is important that material awaiting inspection is not mistakenly allowed to miss inspection at any stage, as non-conformance may go undetected.

Clause 4.13: Control of non-conforming product

- Standard defines non-conforming product as all products or services falling but side tolerance limits agreed in advance with the custom. Once again it is not prescriptive about performance levels non-conforming products or services should be clearly identified, documented and, if possible, physically separated from the conforming product.
- Procedures should be established to handle non-conforming products by reworking, disposal, re-grading or other acceptable documented courses of action.
- There are circumstances where the standard permits the sale of non-conforming product provided that the customer is clearly aware of the circumstances and is generally offered a concession.
- Representatives of accreditation bodies suggest that this an area where organizations often become lax after a while, relaxing procedures and allowing non-conforming product through.

Clause 4.14: Corrective action

- Corrective action is the key to continual improvement. Such action should be implemented via a systematic programmed which provides guidance and defines the duties of all parties.
- Records should be kept of any action taken so that future audits can investigate its effectiveness.
Clause 4.15: Handling, storage, packaging and delivery

- Handling and associated activities must be designed to protect the quality built into the product.
- Subcontractors employed for transportation should be subject to the same documented procedures as internal employees.
- The scope of this clause is determined by the contract with the customer.
- The clause covers all activities which are the contractual obligation of the supplier.

Clause 4.16: Quality records

- Do not have to conform to a prescribed format, but must be fit for their intended purpose.
- As many will exist before the accredited system is implemented, the aim is to systematize and assimilate existing practice wherever possible, to reduce wasted effort in reproducing previous work in this area.

Clause 4.17: Internal quality audits

- The quality system should be ‘policed’ from within the organization and not dependent upon external inspection.
- Procedures should be established to set up regular internal audits as part of normal management procedure.
- The role of internal audits should be to identify problems early in order to minimize their impact and cost.

Clause 4.18: Training

Training activities should be implemented and documented. In particular, the following procedures are required:

- To establish training needs.
- To carry out training activity.
- To record the training requirements and completed activities for each member of staff.
Clause 4.20: Statistical techniques

- Statistical techniques are required to be used where appropriate.
- The standard does not specify particular techniques or methods but does specify that once again they should be appropriate for the intended purpose.
- Their use may be necessary in order to satisfy other requirements, notably process control.

CAPABILITY MATURITY MODEL

- To determine an organization’s current start of process maturity the SET uses an assessment that results in a 5 point grading scheme.
- The grading scheme determines compliance with a capability maturity model [CMM] that defines key activities required at different levels of process maturity.
- The SET approach provides a measure of the global effectiveness of a company’s S/W engineering practice & establishments 5 process maturity levels that are defined in the following manner:

  LEVEL
  - INITIAL
  - REPETABLE
  - DEFINED
  - MANAGED
  - OPTIMIZED
The set has associated key process areas (KPA’s) with each of the maturity levels. Each KPA is described by identifying the following characteristics:

- GOALS
- COMMITMENTS
- ABILITIES
- ACTIVITIES
- METHODS FOR MONITORING INFORMATION
- METHODS FOR VERIFYING IMPLEMENTATION

18 KPA’s are defined across the maturity model & mapped into different levels of process maturity the following KPA’s should be achieved at each process maturity level

**PROCESS MATURITY LEVEL 2:**
- S/W CONFIGURATION MANAGEMENT
- S/W QUALITY ASSURANCE
- S/W SUBCONTRACT MANAGEMENT
- S/W PROJECT TRACKING & MANAGEMENT
- S/W PROJECT PLANNING
- REQUIREMENTS, MANAGEMENT

**PROCESS MATURITY LEVEL 3:**
- PEER REVIEWS
- INTERGROUP CORDINATION
- S/W PROJECT ENGINEERING
- INTEGRATED S/W MANAGEMENT
- ORGANISATION PROCESS DEFINITION
- ORGANISATION PROCESS FOCUS

**PROCESS MATURITY LEVEL 4:**
- S/W QUALITY MANAGEMENT
- QUANTITATIVE PROCESS MANAGEMENT

**PROCESS MATURITY LEVEL 5:**
- PROCESS CHANGE MANAGEMENT
The Role of CMM

The role of CMM is increasing. This may be attributed to a number of factors:

- The maturity of the model itself
- The increasing general awareness of the need for externally recognized quality standards
- The adoption of model by key software purchases such as national departments of defense.

Advantages of CMM

- It allows for improvement and evolution
- Also used in conjunction with other quality standards
- It highlights the defects as they occur.
- The CMM prioritizes tasks for improvement
- It provides a matrix for strengths and weakness
Schematic view of CMM

The maturity model seeks to measure how well these processes are carried out. There are five stages to six measurement categories in subjectively rating an organization’s quality operation.

The five stages are:

- **Uncertainty**, where management is confused and uncommitted regarding quality management tool
- **Awakening**, where management is beginning to recognize that quality management can help
- **Enlightenment**, where the decision is made to conduct a formal quality improvement programme
- **Wisdom**, where the company has the chance to make changes permanent (things are basically quiet and people wonder why they used to have problems);
- **Certainty**, where quality management is considered an absolutely vital part of company management.
The six measurement categories are:

- management understanding and attitude, characterized as ‘no comprehension of quality as a management tool’ at uncertainty and ‘an essential part of the company system’ at certainty;
- quality organization status, characterized as hidden at uncertainty and a thought leader/main concern at certainty;
- problem handling, which are fought when they occur at uncertainty and prevented at certainty;
- cost of quality as percentage of sales, characterized as 20% at uncertainty and 2.5% at certainty;
- quality improvement actions, characterized as no organized activities at uncertainty and a normal and continued activity at certainty;
- Summation of company quality posture, summarized as ‘we do not know why we have problems with quality’ at uncertainty and ‘we know why we do not have problems with quality’ at certainty.

As a precursor to the maturity model itself, Radice et.al. (1985) working with Humphrey, identified 12 process stages. The process stages were stages in the lifecycle:

1. requirements;
2. product level design;
3. component level design;
4. module level design;
5. code;
6. unit test;
7. functional verification test;
8. product verification test;
9. system verification test;
10. package and release;
11. early support programme;
12. general availability.

The eleven attributes were:

1. process
2. methods
3. adherence to practices
4. tools
5. change control
6. data gathering
7. data communication and use
8. goal setting
9. quality focus
10. customer focus
11. technical awareness.

www.Annauniversityplus.com
## Evolution of the CMM

<table>
<thead>
<tr>
<th>Year</th>
<th>Version published</th>
</tr>
</thead>
<tbody>
<tr>
<td>1987</td>
<td>Software process maturity framework</td>
</tr>
<tr>
<td>1987</td>
<td>Preliminary maturity questionnaire</td>
</tr>
<tr>
<td>1987</td>
<td>Characterizing the software process</td>
</tr>
<tr>
<td>1989</td>
<td>Managing the software process</td>
</tr>
<tr>
<td>1990</td>
<td>Draft version of CMM v0.2</td>
</tr>
<tr>
<td>1991</td>
<td>CMM v0.6 discussion</td>
</tr>
<tr>
<td>1991</td>
<td>CMM v1.0</td>
</tr>
<tr>
<td>1993</td>
<td>CMM v1.1</td>
</tr>
</tbody>
</table>

www.Annauniversityplus.com
1S09000-3

The 1S09000-3 notes for guidance (1991) should not be confused with either IS09003, the standard relating to quality management by final inspection, or 1S09004, which provides general guidance on how to implement a QMS conforming to ISO900n.

1S09000-3 has a target audience of the IT community.

The structure of 1S09000-3 is as follows:

Introductory material — The first three clauses of the standard are concerned with defining the scope of the standard

Section 4: Quality system – framework

This part contains four subsections: management responsibility, quality system, internal quality audit and corrective action.

Section 5: Quality system - lifecycle activities

This section contains nine sections, dealing with activities related to one or more parts of the lifecycle. Many of the corresponding sections in 1S09001 seem unsubstantial in comparison when applied to software.
Section 6: Quality system — supporting activities

This section contains nine items which cover the remaining activities. Some, such as configuration management, are mentioned only briefly in ISO9001. New activities covered include configuration management, measurement, rules, practices and conventions, and tools and techniques. Most of the content makes explicit the implicit requirements of ISO9001.

ISO9000-3 headings are summarized in the following table, which gives all the principal section headings and lists the corresponding clauses in ISO9001, classifying the degree of guidance provided as none, minor, significant or major.
The key areas of guidance provided by ISO9000-3 are requirements definition, lifecycle definition, configuration management and measurements. Software is considered to be different from other applications because:

- it is considered as an intellectual object
- the development process has its own characteristics and importance
- replication always gives an exact copy
- software does not wear
- once a fault is fixed it will not reoccur.

<table>
<thead>
<tr>
<th>Section heading</th>
<th>Sub-section</th>
<th>Subsection title</th>
<th>Related ISO9001 clauses</th>
<th>Addition to ISO9001</th>
</tr>
</thead>
<tbody>
<tr>
<td>4. Quality system – framework</td>
<td>4.1</td>
<td>Management responsibility</td>
<td>4.1</td>
<td>Significant</td>
</tr>
<tr>
<td></td>
<td>4.2</td>
<td>Quality system</td>
<td>4.2</td>
<td>Significant</td>
</tr>
<tr>
<td></td>
<td>4.3</td>
<td>Internal quality audits</td>
<td>4.17</td>
<td>Minor</td>
</tr>
<tr>
<td></td>
<td>4.4</td>
<td>Corrective action</td>
<td>4.14</td>
<td>Minor</td>
</tr>
<tr>
<td>5. Quality system – lifecycle activities</td>
<td>5.1</td>
<td>Contract reviews</td>
<td>4.3</td>
<td>Significant</td>
</tr>
<tr>
<td></td>
<td>5.2</td>
<td>Purchaser’s requirements specification</td>
<td>4.3a, 4.4</td>
<td>Significant</td>
</tr>
<tr>
<td></td>
<td>5.3</td>
<td>Development planning</td>
<td>4.4.2</td>
<td>Significant</td>
</tr>
<tr>
<td></td>
<td>5.4</td>
<td>Quality planning</td>
<td>4.2</td>
<td>Significant</td>
</tr>
<tr>
<td></td>
<td>5.5</td>
<td>Design and implementation</td>
<td>4.4, 4.9</td>
<td>Significant</td>
</tr>
<tr>
<td></td>
<td>5.6</td>
<td>Testing and validation</td>
<td>4.10, 4.13</td>
<td>Significant</td>
</tr>
<tr>
<td></td>
<td>5.7</td>
<td>Acceptance</td>
<td>4.10, 4.13</td>
<td>Significant</td>
</tr>
<tr>
<td></td>
<td>5.8</td>
<td>Replication, delivery and installation</td>
<td>4.15</td>
<td>Significant</td>
</tr>
<tr>
<td></td>
<td>5.9</td>
<td>Maintenance</td>
<td>4.19</td>
<td>Major</td>
</tr>
<tr>
<td>6. Quality system – supporting activities</td>
<td>6.1</td>
<td>Configuration management</td>
<td>4.4, 4.8, 4.5</td>
<td>Major</td>
</tr>
<tr>
<td></td>
<td>6.2</td>
<td>Document control</td>
<td>4.5</td>
<td>Significant</td>
</tr>
<tr>
<td></td>
<td>6.3</td>
<td>Quality records</td>
<td>4.16</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>6.4</td>
<td>Measurements</td>
<td>4.20</td>
<td>Major</td>
</tr>
<tr>
<td></td>
<td>6.5</td>
<td>Rules, practices and conventions</td>
<td>4.9, 4.11</td>
<td>Significant</td>
</tr>
<tr>
<td></td>
<td>6.6</td>
<td>Tools and techniques</td>
<td>4.9, 4.11</td>
<td>Significant</td>
</tr>
<tr>
<td></td>
<td>6.7</td>
<td>Purchasing</td>
<td>4.6</td>
<td>Minor</td>
</tr>
<tr>
<td></td>
<td>6.8</td>
<td>Included software product</td>
<td>4.7</td>
<td>Significant</td>
</tr>
<tr>
<td></td>
<td>6.9</td>
<td>Training</td>
<td>4.18</td>
<td>Minor</td>
</tr>
</tbody>
</table>
SPICE

SPICE is attempting to define a framework for conducting assessments together with guidance on how to use the framework for process improvement and capability. The framework defines on architecture that defines practices and processes for s/w development, operation, maintenance, and support.

SPICE is a major international initiative to develop a standard for Software Process Assessment. This standard covers,

- Process Assessment,
- Improvement and
- Capability

It provides the following benefits to the software suppliers:

- Software suppliers will submit just one process assessment scheme
- Software development organizations will have a tool to initiate and sustain a continuous process improvement programme
- Managers will have a means to ensure that their software is aligned with, and supports, the business needs of the organization

Structure of SPICE

www.Annauniversityplus.com
Capability levels of SPICE

<table>
<thead>
<tr>
<th>Level</th>
<th>Title</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Not performed</td>
<td>General failure to perform base functions</td>
</tr>
<tr>
<td>1</td>
<td>Performed informally</td>
<td>Base functions generally carried out in an ad hoc manner. Identifiable products for the process</td>
</tr>
<tr>
<td>2</td>
<td>Planned and tracked</td>
<td>Base functions generally carried out in a planned manner. Performance verified. Work conforms to standards and requirements</td>
</tr>
<tr>
<td>3</td>
<td>Well defined</td>
<td>Base functions performed according to approved tailored standard documented processes</td>
</tr>
<tr>
<td>4</td>
<td>Quantitatively controlled</td>
<td>Detailed measures of performance collected and analysed. Prediction of performance. Quality of work is quantitatively known</td>
</tr>
<tr>
<td>5</td>
<td>Continuously improving</td>
<td>Quantitative goals based on business goals established. Continuous process improvement enabled by quantitative feedback of performance</td>
</tr>
</tbody>
</table>
SPICE processes are classified into five categories:

- Support process category
- Project process category
- Institute/Organization process category
- Customer – supplier process category
- Engineering process category

Six Sigma concepts

The term Six Sigma represents a stringent level of quality. It is a specific defect rate: 3.4 defective parts per million (ppm). It was made known in the industry by Motorola, Inc., Six Sigma has become an industry standard as an ultimate quality goal.

Sigma (s) is the Greek symbol for standard deviation.

As the following figure indicates, the areas under the curve of normal distribution defined by standard deviations are constants in terms of percentages, regardless of the distribution parameters.

The area under the curve as defined by plus and minus one standard deviation (sigma) from the mean is 68.26%.

The area defined by plus/minus two standard deviations is 95.44%, and so forth. The area defined by plus/minus six sigma is 99.9999998%. The area outside the six sigma area is thus 100% - 99.9999998% = 0.0000002%.
The area within the six sigma limit as the percentage of defect-free parts and the area outside the limit as the percentage of defective parts, it is found that six sigma is equal to 2 defectives per billion parts or 0.002 defective parts per million.

The interpretation of defect rate as it relates to the normal distribution will be clearer if we include the specification.

Given the specification limits (which were derived from customers' requirements), our purpose is to produce parts or products within the limits. Parts or products outside the specification limits do not conform to requirements. If we can reduce the variations in the
production process so that the six sigma (standard deviations) variation of the production process is within the specification limits, then we will have six sigma quality level.

The six sigma value of 0.002 ppm is from the statistical normal distribution. It assumes that each execution of the production process will produce the exact distribution of parts or products centered with regard to the specification limits. In reality, however, process shifts and drifts always result from variations in process execution.

The maximum process shifts as indicated by research is 1.5 sigma. If we account for this 1.5-sigma shift in the production process, we will
get the value of 3.4 ppm. Such shifting is illustrated in the two lower panels of fig. 2.

Given fixed specification limits, the distribution of the production process may shift to the left or to the right. When the shift is 1.5 sigma, the area outside the specification limit on one end is 3.4 ppm, and on the other it is nearly zero.

The slight difference between the centered six sigma and the shifted six sigma may imply something significant. The former is practically equivalent to zero defects, which may invite the debate whether it is feasible to achieve such a goal.

In order to reach six sigma, we have to improve the process. Specifically, we must reduce process variations so that the six sigma variation is still within the specification limits.

The concept and approach of six sigma has been expanded and applied to the improvement of management systems and total quality management.

**Seven Basic Tools for quality control**

Ishikawa's seven basic tools for quality control are:

- checklist (or check sheet),
- Pareto diagram,
- histogram,
- scatter diagram,
- run chart,
- control chart, and
- cause-and-effect diagram.

The following figure shows a simple representation of the tools.
Check sheet

A check sheet is a paper form with printed items to be checked. Its main purposes are to facilitate gathering data and to arrange data while collecting it so the data can be easily used later. Another type of check sheet is the check-up confirmation sheet. It is concerned mainly with the quality characteristics of a process or a product. To distinguish this confirmation check sheet from the ordinary data-gathering check sheet, we use the term checklist. In most software development environments, the data-gathering aspect is automated electronically and goes far beyond the data-gathering check sheet approach, which has been used in manufacturing production.
**Pareto diagram**

A Pareto diagram is a frequency chart of bars in descending order; the frequency bars are usually associated with types of problems. In software development, the X-axis for a Pareto diagram is usually the defect cause and the Y-axis the defect count. By arranging the causes based on defect frequency, a Pareto diagram can identify the few causes that account for the majority of defects. It indicates which problems should be solved first in eliminating defects and improving the operation. Pareto analysis is commonly referred to as the 80–20 principle (20% of the causes account for 80% of the defects), although the cause-defect relationship is not always in an 80–20 distribution.

**Histogram**

The histogram is a graphic representation of frequency counts of a sample or a population. The X-axis lists the unit intervals of a parameter (e.g., severity level of software defects) ranked in ascending order from left to right, and the Y-axis contains the frequency counts. In a histogram, the frequency bars are shown by the order of the X variable, whereas in a Pareto diagram the frequency bars are shown by order of the frequency counts. The purpose of the histogram is to show the distribution characteristics of a parameter such as overall shape, central tendency, dispersion, and skewness. It enhances understanding of the parameter of interest.

**Scatter diagram**

A scatter diagram vividly portrays the relationship of two interval variables. In a cause-effect relationship, the X-axis is for the independent variable and the Y-axis for the dependent variable. Each point in a scatter diagram represents an observation of both the dependent and independent variables. Scatter diagrams aid data-based decision making (e.g., if action is planned on the X variable and some effect is expected on the Y variable).

**Run chart**
A run chart tracks the performance of the parameter of interest over time. The X-axis is time and the Y-axis is the value of the parameter. A run chart is best used for trend analysis, especially if historical data are available for comparisons with the current trend. An example of a run chart in software is the weekly number of open problems in the backlog; it shows the development team's workload of software fixes.

Control chart

A control chart can be regarded as an advanced form of a run chart for situations where the process capability can be defined. It consists of a central line, a pair of control limits (and sometimes a pair of warning limits within the control limits), and values of the parameter of interest plotted on the chart, which represent the state of a process. The X-axis is real time. If all values of the parameter are within the control limits and show no particular tendency, the process is regarded as being in a controlled state. If they fall outside the control limits or indicate a trend, the process is considered out of control. Such cases call for causal analysis and corrective actions are to be taken.

Cause-and-effect diagram

The cause-and-effect diagram, also known as the fishbone diagram, was first used to explain factors that affect the production of steel. It shows the relationship between a quality characteristic and factors that affect that characteristic. Its layout resembles a fishbone, with the quality characteristic of interest labeled at the fish head, and factors affecting the characteristics placed where the bones are located. While the scatter diagram describes a specific bivariate relationship in detail, the cause-and-effect diagram identifies all causal factors of a quality characteristic in one chart.
Part – A

1. Write the purpose of standard
2. What are the benefits of accreditation?
3. What are the advantages of CMM?
4. List out the various quality control tools
5. Mention the various SPICE capability levels

Part – B

6. Compare the ISO 9000 series of quality management standards in detail (15)

(Or)

7. a) Explain the Six sigma concept (8)
   b) Explain in detail the various quality control tools (7)