**HBCS22E04 - SOFTWARE TESTING AND QUALITY ASSURANCE**

**UNIT I**

**4Marks**

**1. Define Software Quality.**

Ans: Software Quality shows how good and reliable a product is. To convey an associate degree example, think about functionally correct software. It performs all functions as laid out in the SRS document.

1. Is achieved through a disciplined approach - called software engineering SE

2. Can be defined, described, and measured

3. Can be assessed before any code has been written

4 Cannot be tested into a product.

**2. List all the components in SQA architecture.**

SQA system components can be classified into six classes:

1. Pre-project quality components.
2. Project life cycle quality components.
3. Infrastructure error preventive and improvement components.
4. Software quality management components.
5. Standardization, certification and SQA assessment components.

**3.What is mean by software quality assurance plan?**

A Quality Assurance Plan (QAP) is a document or set of documents that outlines the systematic processes, procedures, and standards for ensuring the quality of a product or service. It is a key component of quality management and is used in various industries to establish and maintain a consistent level of quality in deliverables.SQA plan will be used in conjunction with the typical development, prototyping, design, production, and release cycle.

**4.Outline McCall’s factor model.**

This model classifies all software requirements into 11 software quality factors. The 11 factors are grouped into three categories – product operation, product revision, and product transition factors.

* **Product operation factors** − Correctness, Reliability, Efficiency, Integrity, Usability.
* **Product revision factors** − Maintainability, Flexibility, Testability.
* **Product transition factors** − Portability, Reusability, Interoperability.

**5.List out the importance of Software Quality Assurance.**

* **Time:**Software testing can take a long time to complete. It can take days or weeks depending on the complexity of the application and the number of tests that need to be performed.
* **Scope:**Software testing is limited in scope. It can only test the software that is being tested, not the entire system. This can lead to problems if other parts of the system are not tested properly.
* **Assumptions:**Software testing is based on assumptions. If the assumptions are incorrect, the software may not work correctly.
* **Complexity:**Software testing can be complex and time-consuming. This can lead to delays in the development process and can be costly.

**6marks**

1. **Compare Quality Challenge and Quality Assurance.**

| **S.No.** | **Quality Control** | **Quality Assurance** |
| --- | --- | --- |
| 1. | QC is the set of activities & techniques used to monitor & verify that a product or service meets predefined standards, identifying & correcting defects during or after the manufacturing | QA is the systematic process of ensuring that a product or service meets specified requirements & is consistent with planned activities, focusing on preventing defects before they occur. |
| 2. | Reactive approach - Detection of Defects | Proactive approach - Prevention of defects |
| 3. | It focuses on the Product. | It focuses on the Process / System. |
| 4. | Corrective actions are the part of Quality Control. | Preventive actions are the part of Quality Assurance. |
| 5. | It is Narrow process. | It is Border level process. |
| 6. | Validation like performance tests can be a part of QC. | Systematic or Process level verification can be a part of QA. |
| 7. | Ex: First part approvals, Product testing and Inspections. | Ex: Auditing, Proactive Reviews. |

## 2. Extend the objectives of quality factors based on quality category.

## Below are the factors of Product Quality, that are discussed in detail.

## Product Operation

## Product Revision

## Product Transition

## Product Operation

## Product Operation includes five software quality factors, which are related to the requirements that directly affect the operation of the software such as operational performance, convenience, ease of usage, and correctness. These factors help in providing a better user experience.

## Correctness

## Efficiency

## Integrity

## Reliability.

## Usability

## Product Revision

## Product Revision includes three software quality factors, which are required for testing and maintenance of the software. They provide ease of maintenance, flexibility, and testing efforts to support the software to be functional according to the needs and requirements of the user in the future.

## Maintainability

## Flexibility

## Testability

## Product Transition

## Product Transition includes three software quality factors, that allow the software to adapt to the change of environments in the new platform or technology from the previous.

## Portability

## Re-usability

## Interoperability

**3. Outline the major components of SQA and explain in detail.**

An SQA system always combines a wide range of SQA components. These components can be classified into the following six classes −

* **Pre-project components**

This assures that the project commitments have been clearly defined considering the resources required, the schedule and budget; and the development and quality plans have been correctly determined.

* **Components of project life cycle activities assessment**

The project life cycle is composed of two stages: the development life cycle stage and the operation–maintenance stage.

* **The development life cycle stage components detect design and programming errors.**

Its components are divided into the following sub-classes: Reviews, Expert opinions, and Software testing.

The SQA components used during the operation–maintenance phase include specialized maintenance components as well as development life cycle components, which are applied mainly for functionality to improve the maintenance tasks.

* **Components of infrastructure error prevention and improvement**

The main objective of these components, which is applied throughout the entire organization, is to eliminate or at least reduce the rate of errors, based on the organization’s accumulated SQA experience.

* **Components of software quality management**

This class of components deal with several goals, such as the control of development and maintenance activities, and the introduction of early managerial support actions that mainly prevent or minimize schedule and budget failures and their outcomes.

* **Components of standardization, certification, and SQA system assessment**

These components implement international professional and managerial standards within the organization. The main objectives of this class are utilization of international professional knowledge, improvement of coordination of the organizational quality systems with other organizations, and assessment of the achievements of quality systems according to a common scale. The various standards may be classified into two main groups: quality management standards and project process standards.

**4. Discover the advantages of Quality Assurance**.

* **Improves Reliability:**Quality assurance helps to improve the reliability of the software by ensuring that the software meets the requirements of the user. This helps to ensure that the software is reliable and performs as expected.
* **Increases Customer Satisfaction:**Quality assurance helps to improve customer satisfaction by ensuring that the software meets the customer’s expectations. This helps to ensure that the customer is satisfied with the software and will continue to use it.
* **Enhances Usability:**Quality assurance helps to improve the usability of the software by ensuring that the user interface is intuitive and easy to use.
* **Increased Productivity:**Quality assurance helps to ensure that the product is developed in an efficient manner. This helps to improve the productivity of the development process and ensures that the product meets the quality requirements.
* **Improved Customer Satisfaction:**Quality assurance helps to ensure that the product meets the customer’s requirements and works as expected. This helps to increase customer satisfaction.

**5.Categorize the application of Quality Assurance.**

* **Design and Code Reviews:**Design and code reviews are a type of quality assurance that verifies the design and code of a system. It is done to ensure that the design and code of the system meet the specified requirements and are of high quality.
* **Process Auditing:**Process auditing is a type of quality assurance that verifies the processes of a system. It is done to ensure that the processes of the system are efficient and effective.
* **Automated Testing:**Automated testing is a type of quality assurance that verifies the functionality of a system using automated tools. It is done to ensure that the system is functioning according to the specified requirements.
* **Defect Tracking:**Defect tracking is a type of quality assurance that tracks and manages the defects in a system. It is done to ensure that all the defects in the system are identified and resolved.
* **Configuration Management:**Configuration management is a type of quality assurance that verifies and manages the configuration of a system. It is done to ensure that the system is configured according to the specified requirements.
* **Risk Management:**Risk management is a type of quality assurance that verifies and manages the risks associated with a system. It is done to ensure that the system is secure and protected against any potential risks.

**10 Marks**

**1. What are the distinct roles of Software Testing and Quality Assurance? Explain in detail.**

| **Factor** | **Software Testing** | **Quality Assurance** |
| --- | --- | --- |
| **Responsibilities** | Software Testing involves finding and documenting the defects in the system. | Quality Assurance involves ensuring that the developed product meets the customer’s requirements and is of the desired quality. |
| **Techniques** | Software Testing involves techniques like unit testing, functional testing, integration testing, system testing, etc. | Quality Assurance involves techniques like root cause analysis, process audits, defect management, etc. |
| **Output** | The output of Software Testing is the list of defects identified in the system. | The output of Quality Assurance is the assurance that the developed product meets the customer’s requirements and is of the desired quality. |
| **Cost** | Software Testing generally costs more than Quality Assurance. | Quality Assurance generally costs less than Software Testing. |
| **Time** | Software Testing generally takes more time than Quality Assurance. | Quality Assurance generally takes less time than Software Testing. |
| **Effort** | Software Testing requires more effort than Quality Assurance. | Quality Assurance requires less effort than Software Testing. |
| **Testing** | Software Testing is the process of executing a program or system with the intent of finding errors in it. | Quality Assurance is not a testing process. |
| **Risk** | Software Testing involves risk associated with finding and documenting the defects in the system. | Quality Assurance involves risk associated with ensuring that the developed product meets the customer’s requirements and is of the desired quality. |
| **Documentation** | Software Testing involves documenting the defects identified in the system. | Quality Assurance involves documenting the process of ensuring that the developed product meets the customer’s requirements and is of the desired quality. |
| **Cost Effectiveness** | Software Testing is not always cost-effective. | Quality Assurance is usually cost-effective. |

**2.Explain in detail about Software Quality Assurance.**

**Quality Assurance**

Quality Assurance (QA) is the process of ensuring that the software product meets the specified quality standards. It is a continuous process that involves planning, designing, developing, and testing the software. QA focuses on the quality of the software product and is done by software engineers and developers. Its primary goal is to ensure that the software meets the customer’s requirements and is of high quality. QA is a process-oriented approach that involves implementing best practices and process improvements to ensure the quality of the software. It includes activities such as requirements gathering, design review, code review, unit testing, system testing, integration testing, and regression testing.

**Characteristics of Quality Assurance**

* **Purpose:**The purpose of quality assurance is to ensure that the software meets all the requirements and is free from any defects.
* **Scope**: Quality assurance is focused on the implementation of processes, procedures, and standards to ensure the quality of the software.
* **Involvement:** Quality assurance professionals are involved in monitoring the development process, reviewing documentation, and auditing the product.
* **Techniques:** Quality assurance typically involves the use of automation tools and manual inspections.
* **Responsibilities:**Quality assurance professionals are responsible for ensuring that the development process is compliant with industry standards and regulations.
* **Deliverables:** The deliverable of quality assurance is typically a report that includes a list of non-conformances, a summary of the audit, and recommendations for improvement.

**Application of Quality Assurance**

* **Design and Code Reviews:**Design and code reviews are a type of quality assurance that verifies the design and code of a system. It is done to ensure that the design and code of the system meet the specified requirements and are of high quality.
* **Process Auditing:**Process auditing is a type of quality assurance that verifies the processes of a system. It is done to ensure that the processes of the system are efficient and effective.
* **Automated Testing:**Automated testing is a type of quality assurance that verifies the functionality of a system using automated tools. It is done to ensure that the system is functioning according to the specified requirements.
* **Defect Tracking:**Defect tracking is a type of quality assurance that tracks and manages the defects in a system. It is done to ensure that all the defects in the system are identified and resolved.
* **Configuration Management:**Configuration management is a type of quality assurance that verifies and manages the configuration of a system. It is done to ensure that the system is configured according to the specified requirements.
* **Risk Management:**Risk management is a type of quality assurance that verifies and manages the risks associated with a system. It is done to ensure that the system is secure and protected against any potential risks.

**3.Outline the stages of the software development life cycle and illustrate each stage with a neat diagram.**

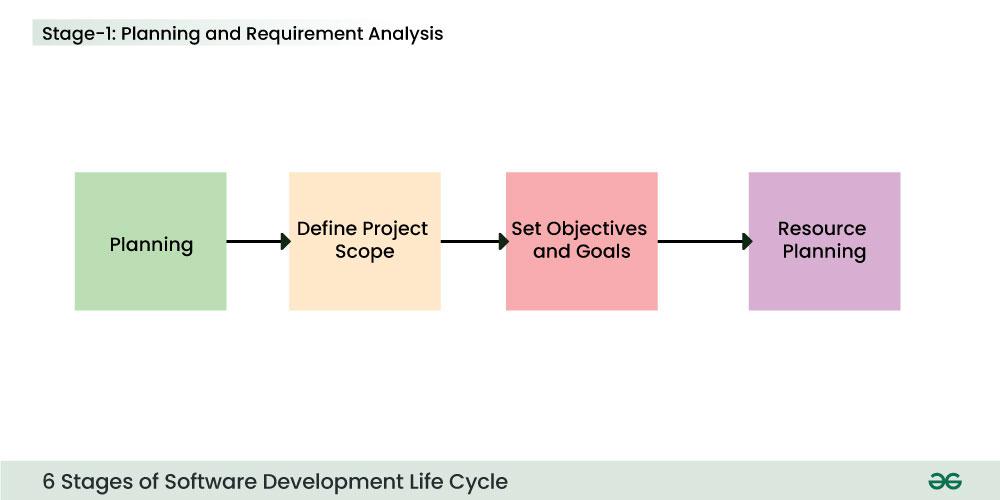
**Software development life cycle (SDLC) is a structured process that is used to design, develop, and test good-quality software.** SDLC, or software development life cycle, is a methodology that defines the entire procedure of software development step-by-step.



The goal of the SDLC life cycle model is to deliver high-quality, maintainable software that meets the user’s requirements. SDLC in software engineering models outlines the plan for each stage so that each stage of the software development model can perform its task efficiently to deliver the software at a low cost within a given time frame that meets users’ requirements.

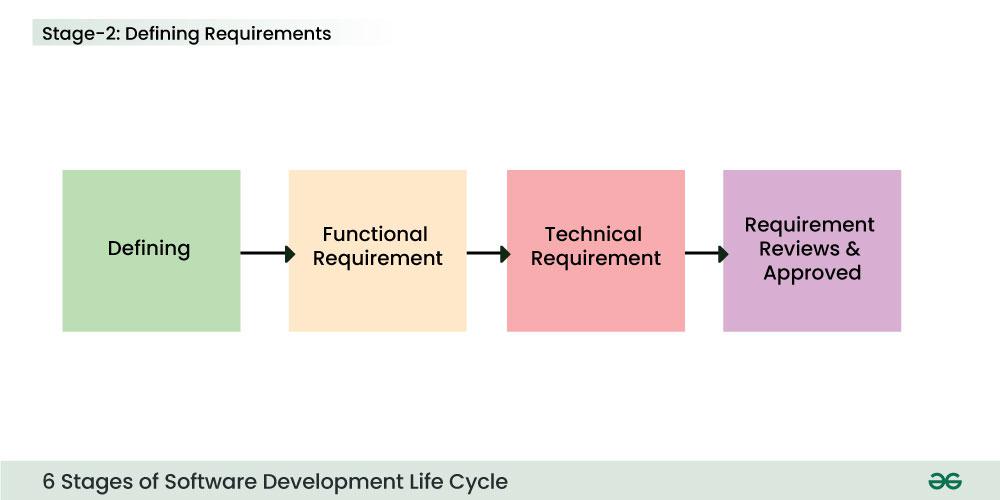
### Stage-1: Planning and Requirement Analysis

Planning is a crucial step in everything, just as in software development. In this same stage, requirement analysis is also performed by the developers of the organization. This is attained from customer inputs, and sales department/market surveys.



### Stage-2: Defining Requirements

In this stage, all the requirements for the target software are specified. These requirements get approval from customers, market analysts, and stakeholders.   
This is fulfilled by utilizing SRS (Software Requirement Specification). This is a sort of document that specifies all those things that need to be defined and created during the entire project cycle.

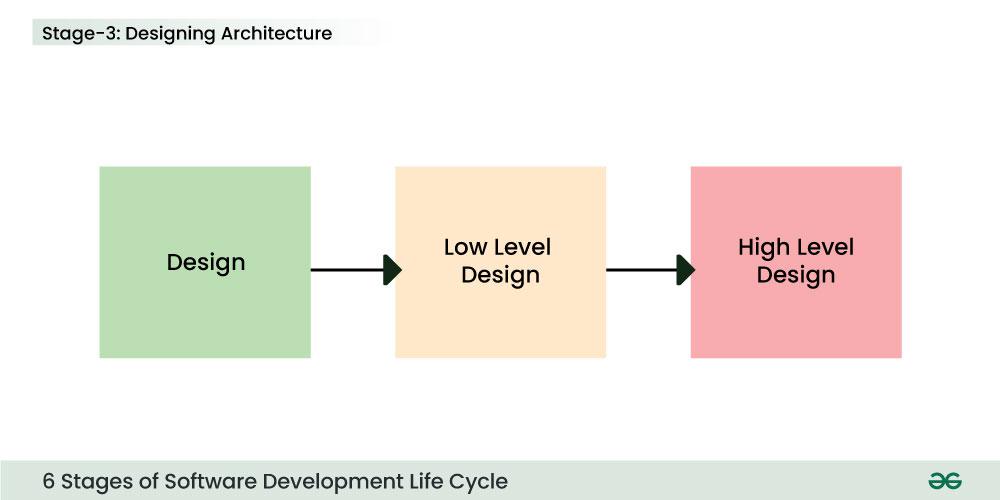


*:*

Stage-3: Designing Architecture

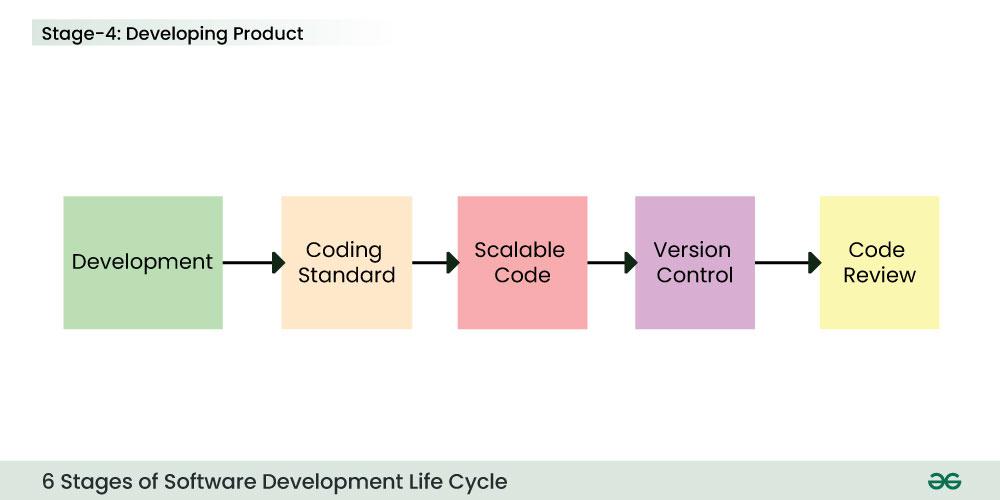
[SRS](https://www.geeksforgeeks.org/software-requirement-specification-srs-format) is a reference for software designers to come up with the best architecture for the software. Hence, with the requirements defined in SRS, multiple designs for the product architecture are present in the Design Document Specification (DDS).

This DDS is assessed by market analysts and stakeholders. After evaluating all the possible factors, the most practical and logical design is chosen for development.



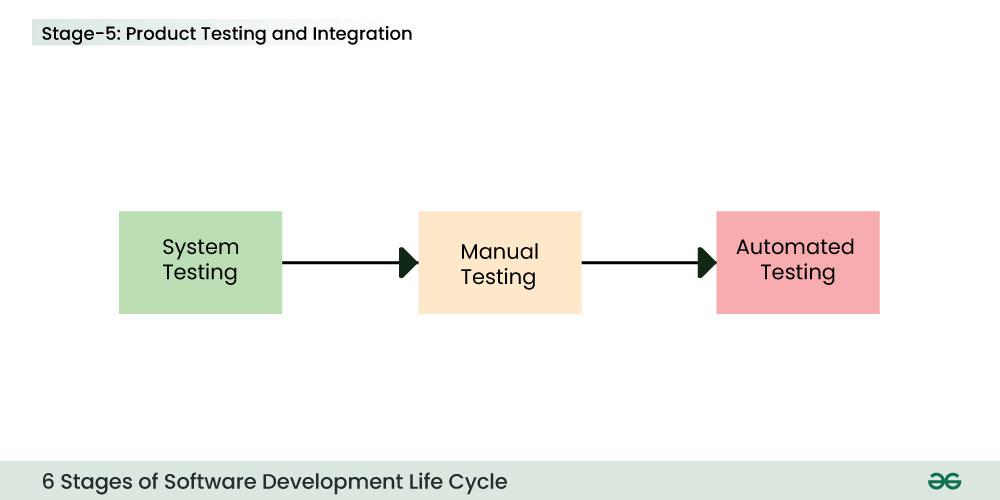
### Stage-4: Developing Product

At this stage, the fundamental development of the product starts. For this, developers use a specific programming code as per the design in the DDS. Hence, it is important for the coders to follow the protocols set by the association. Conventional programming tools like compilers, interpreters, debuggers, etc



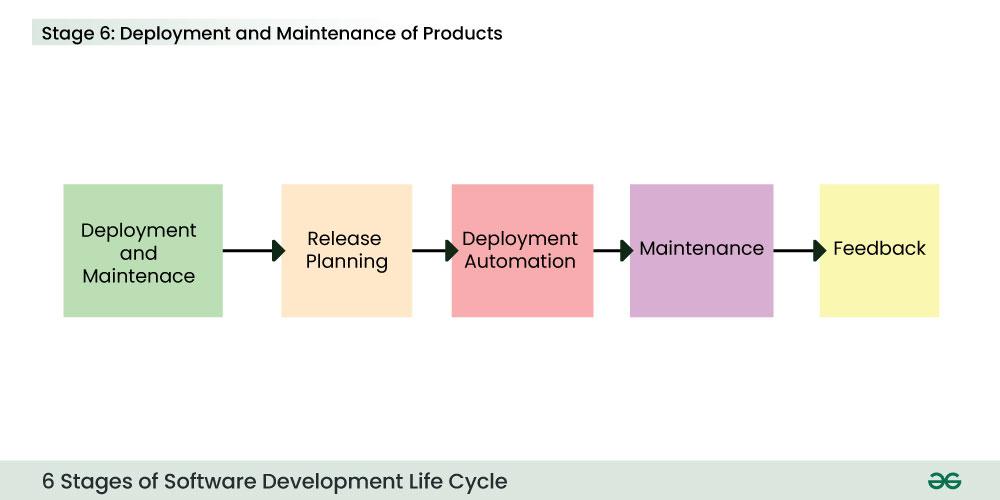
### Stage-5: Product Testing and Integration

After the development of the product, testing of the software is necessary to ensure its smooth execution. Although, minimal testing is conducted at every stage of SDLC. Therefore, at this stage, all the probable flaws are tracked, fixed, and retested. This ensures that the product confronts the quality requirements of SRS.



### Stage-6: Deployment and Maintenance of Products

After detailed testing, the conclusive product is released in phases as per the organization’s strategy. Then it is tested in a real industrial environment. It is important to ensure its smooth performance. If it performs well, the organization sends out the product as a whole. After retrieving beneficial feedback, the company releases it as it is or with auxiliary improvements to make it further helpful for the customers..



**4.Extend about Software Quality Assurance plan and Evaluate the Significance of Software Quality.**

## What is a Software Quality Assurance Plan?

A Quality Assurance Plan (QAP) is a document or set of documents that outlines the systematic processes, procedures, and standards for ensuring the quality of a product or service. It is a key component of quality management and is used in various industries to establish and maintain a consistent level of quality in deliverables. For a software product or service, an SQA plan will be used in conjunction with the typical development, prototyping, design, production, and release cycle. An SQA plan will include several components, such as purpose, references, configuration and management, tools, code controls, testing methodology, problem reporting and remedial measures, and more, for easy documentation and referencing.



## Importance of Software Quality Assurance Plan

* **Quality Standards and Guidelines:** The SQA Plan lays out the requirements and guidelines to make sure the programme satisfies predetermined standards for quality.
* **Risk management:** It is the process of recognizing, evaluating and controlling risks in order to reduce the possibility of errors and other problems with quality.
* **Standardization and Consistency:** The strategy guarantees consistent methods, processes, and procedures, fostering a unified and well-structured approach to quality assurance.
* **Customer Satisfaction:** The SQA Plan helps to ensure that the finished product satisfies customer needs, which in turn increases overall customer satisfaction.
* **Resource optimization:** It is the process of defining roles, responsibilities, and procedures in order to maximize resource utilization and minimize needless rework.
* **Early Issue Detection:** SQA Plans help identify problems early on, which lowers the expense and work involved in fixing them.

**5.Demonstrate McCall’s model in detail.**

## McCall’s Factor Model

This model classifies all software requirements into 11 software quality factors. The 11 factors are grouped into three categories – product operation, product revision, and product transition factors.

* **Product operation factors** − Correctness, Reliability, Efficiency, Integrity, Usability.
* **Product revision factors** − Maintainability, Flexibility, Testability.
* **Product transition factors** − Portability, Reusability, Interoperability.

### **Correctness**

* The required accuracy of output that can be negatively affected by inaccurate data or inaccurate calculations.
* The completeness of the output information, which can be affected by incomplete data.
* The availability of the information.
* The standards for coding and documenting the software system.

### **Reliability**

Reliability requirements deal with service failure. They determine the maximum allowed failure rate of the software system, and can refer to the entire system or to one or more of its separate functions.

### **Efficiency**

It deals with the hardware resources needed to perform the different functions of the software system. It includes processing capabilities (given in MHz), its storage capacity (given in MB or GB) and the data communication capability (given in MBPS or GBPS).

### **Integrity**

This factor deals with the software system security, that is, to prevent access to unauthorized persons, also to distinguish between the group of people to be given read as well as write permit.

### **Usability**

Usability requirements deal with the staff resources needed to train a new employee and to operate the software system.

### **Maintainability**

This factor considers the efforts that will be needed by users and maintenance personnel to identify the reasons for software failures, to correct the failures, and to verify the success of the corrections.

### **Flexibility**

This factor deals with the capabilities and efforts required to support adaptive maintenance activities of the software. These include adapting the current software to additional circumstances and customers without changing the software

### **Testability**

Testing of the software system as well as with its operation. It includes predefined intermediate results, log files, and also the automatic diagnostics performed by the software system prior to starting the system, to find out whether all components of the system are in working order and to obtain a report about the detected faults.

### **Portability**

Portability requirements tend to the adaptation of a software system to other environments consisting of different hardware, different operating systems, and so forth.

### **Reusability**

The reuse of software is expected to save development resources, shorten the development period, and provide higher quality modules.

### **Interoperability**

Interoperability requirements focus on creating interfaces with other software systems or with other equipment firmware