Software quality is essential as it safeguards security and usability. A high-quality software brings in predictability. It guarantees the functionality and credibility of the enterprise. It is further critical for customer satisfaction and holding their trust in your services.

Defining Software Quality:

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. 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:

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.

ISO 9126:

ISO/IEC 9126 is an international standard proposed to make sure 'quality of all softwareintensive products' which includes a system like safety-critical where in case of failure of software lives will be in jeopardy. ISO i.e. International Organization for standardization and IEC i.e. International Electrotechnical Commission have developed ISO/IEC 9126 standards for software engineering -> Product Quality to provide an all-inclusive specification and evaluation model for the quality of the software product. The standard is divided into 4 parts as depicted in the following figure :



Part-1 Software Engineering – Product Quality "Quality model" :

It describes quality model framework which explains relationships between different approaches to quality as well as identifying quality characteristics and sub-characteristics of software products.

Part-2 Software Engineering – Product Quality "External Metrices" :

It's use is to describes external metrices that are used to measure characteristics and subcharacteristics which are identifies in part 1.

Part-3 Software Engineering – Product Quality "Internal Metrices" :

It's use is to describes internal metrices that are used to measure characteristics and subcharacteristics which are identifies in part 1.

Part-3 Software Engineering – Product Quality "Quality in use metrices" :

It's use is to identify metrices which are used to measure effects of combined quality characteristics for user.

As from above discussion, it is concluded that first three parts are concerned with describing and measuring quality of software product and fourth part concerned about quality of software product from user point of view.

Furthermore, first part i.e. Quality model is concerned classified into two categories as depicted in the following figure :



Internal External Quality Part : It determines the quality of a software product through six characteristics which are Functionality, Reliability, Usability, Efficiency, Maintainability and Portability. Each characteristics is subdivided into related sub-characteristics which are also depicted in the above example.

- 1. Functionality: The functions are those that will satisfy implied needs.
 - Suitability
 - Accuracy
 - Interoperability
 - Security
 - Functionality Compliance
- 2. **Reliability:** A set of attributes that will bear on the capability of software to maintain the level of performance.
 - Maturity
 - Fault Tolerance
 - Recoverability
 - Reliability Compliance
- 3. Usability: A set of attributes that bear on the effort needed for use by a implied set of users.
 - Understandability
 - Learn ability
 - Operability
 - Attractiveness
 - Usability Compliance
- 4. Efficiency: A set of attributes that bear on the relationship between the level of performance of the software under stated conditions.
 - Time Behavior
 - Resource Utilization
 - Efficiency Compliance
- 5. **Maintainability:** A set of attributes that bear on the effort needed to make specified modifications.
 - Analyzability

- Changeability
- Stability
- Testability
- Maintainability Compliance
- 6. **Portability:** A set of attributes that bear on the ability of software to be transferred from one environment to another.
 - Adaptability
 - Installability
 - Co-existence
 - Replace ability
 - Portability Compliance

Quality in use Model : It identifies the four quality characteristics i.e. Effectiveness, Productivity, Safety, Satisfaction.

Practicle Software Qualtiy Measures:

In <u>Software Engineering</u>, Software Measurement is done based on some <u>Software</u> <u>Metrics</u> where these software metrics are referred to as the measure of various characteristics of a <u>Software</u>.

In Software engineering <u>Software Quality Assurance (SAQ)</u> assures the quality of the software. Set of activities in SAQ are continuously applied throughout the software process. <u>Software Quality</u> is measured based on some software quality metrics.

There is a number of metrics available based on which software quality is measured. But among them, there are few most useful metrics which are most essential in software quality measurement. They are –

- 1. Code Quality
- 2. Reliability
- 3. Performance
- 4. Usability
- 5. Correctness
- 6. Maintainability
- 7. Integrity
- 8. Security

Now let's understand each quality metric in detail -

1. Code Quality – Code quality metrics measure the quality of code used for the software project development. Maintaining the software code quality by writing Bug-free and semantically correct code is very important for good software project development. In code quality both Quantitative metrics like the number of lines, complexity, functions, rate of bugs generation, etc, and Qualitative metrics like readability, code clarity, efficiency, maintainability, etc are measured.

2. Reliability – Reliability metrics express the reliability of software in different conditions. The software is able to provide exact service at the right time or not is checked. Reliability can be checked using Mean Time Between Failure (MTBF) and Mean Time To Repair (MTTR).

3. Performance – Performance metrics are used to measure the performance of the software. Each software has been developed for some specific purposes. Performance metrics measure the performance of the software by determining whether the software is

fulfilling the user requirements or not, by analyzing how much time and resource it is utilizing for providing the service.

4. Usability – Usability metrics check whether the program is user-friendly or not. Each software is used by the end-user. So it is important to measure that the end-user is happy or not by using this software.

5. Correctness – Correctness is one of the important software quality metrics as this checks whether the system or software is working correctly without any error by satisfying the user. Correctness gives the degree of service each function provides as per developed.

6. Maintainability – Each software product requires maintenance and up-gradation. Maintenance is an expensive and time-consuming process. So if the software product provides easy maintainability then we can say software quality is up to mark. Maintainability metrics include time requires to adapt to new features/functionality, Mean

Time to Change (MTTC), performance in changing environments, etc.

7. Integrity – Software integrity is important in terms of how much it is easy to integrate with other required software's which increases software functionality and what is the control on integration from unauthorized software's which increases the chances of cyberattacks.

8. Security – Security metrics measure how much secure the software is? In the age of cyber terrorism, security is the most essential part of every software. Security assures that there are no unauthorized changes, no fear of cyber attacks, etc when the software product is in use by the end-user.

Product Versus Process Quality Management:

Product:

In the context of software engineering, Product includes any software manufactured based on the customer's request. This can be a problem solving software or computer based system. It can also be said that this is the result of a project.

Process:

Process is a set of sequence steps that have to be followed to create a project. The main purpose of a process is to improve the quality of the project. The process serves as a template that can be used through the creation of its examples and is used to direct the project.

The main difference between a process and a product is that the process is a set of steps that guide the project to achieve a convenient product. while on the other hand, the product is the result of a project that is manufactured by a wide variety of people.



While the process is a set of sequence steps that have to be followed to create a project.

Whereas the process is focused on completing each step being developed.

In contrast, the process consistently follows guidelines.

Whereas the process tends to be long-term.

While the purpose of the process is to make the quality of the project better.

A process serves as a model for producing various goods in a similar way.

When resources with similar processes or functions are grouped together, it is referred to as a process layout.

1.

2.

7.

In the case of products, the firm guidelines
are followed.

A product focuses on the final result.

Product is the final production of the project.

4. A product tends to be short-term.

The main goal of the product is to completethe work successfully.

6. Product is created based on the needs andexpectations of the customers.

A product layout is a style of layout design in which the materials required to make the product are placed in a single line depending on the order of operations.