**MC call**

**Product operation**

**Correctness**

Correctness is a measure of how well a software product meets its requirements.

It is important because it ensures that the product works as intended.

Correctness can be improved by following good software development practices.

**Realiability:**

Reliability is a measure of how likely a software product is to perform its intended functions correctly over time. It is important because it ensures that the product is available when users need it. Reliability can be improved by following good software development practices, such as testing and fault tolerance.

**Efficient:**

Efficiency is a measure of how well a software product uses system resources. It is important because it ensures that the product can perform its intended functions without causing performance issues. Efficiency can be improved by following good software development practices, such as optimizing code and using efficient algorithms.

**Usability:**

Usability is a measure of how easy it is for users to learn, use, and understand a software product. It is important because it ensures that users can effectively use the product to achieve their goals. Usability can be improved by following good software development practices, such as user-centered design and usability testing.

Product revision Factor

**Maintability**

Maintainability is how easy it is to make changes to a software product.

It is important for fixing bugs, adding features, and adapting to changing requirements.

Maintainability can be improved by using modular design and well-documented code.

Flexability

Flexibility is the ability of a software product to be changed or adapted to new requirements.

It is important for maintaining a product over time and for meeting the needs of changing users.

Flexibility can be improved by using modular design and well-documented code.

**Testability:** Testability is the ease with which a software product can be tested. It is important for ensuring that a product meets its requirements and for finding bugs early in the development process. Testability can be improved by using modular design, well-documented code, and automated testing.

**Interoperability:**

Interoperability is the ability of different software products to work together. It is important for ensuring that data can be exchanged between products and for providing a consistent user experience. Interoperability can be improved by using standard data formats and by following common development practices.

**Portability**

Portability is the ability of a software product to be moved from one environment to another. It is important for ensuring that a product can be used in different organizations and for reducing the cost of maintenance. Portability can be improved by using standard software components and by following common development practices.