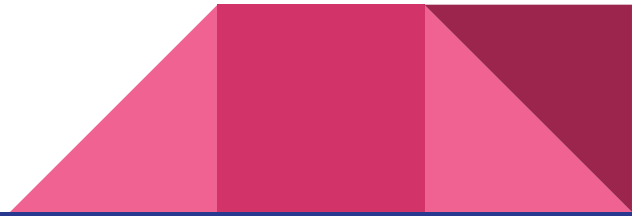
Topic 4 : Software Quality Models

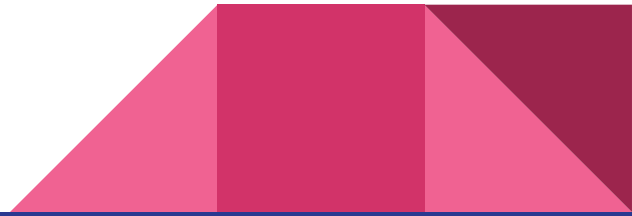
Software Quality Models

- Software Quality Models provide a systematic framework to assess and improve the quality of software products.
- They help us evaluate various dimensions of software quality, identify areas for improvement, and implement best practices.
- It defines the characteristics and attributes of software quality and how they are related.



Software Quality Models

- Different models may have different perspectives, goals, and criteria for evaluating software quality.
- For example,
  - some models may focus on the internal aspects of software, such as code quality, complexity, and modularity, while others may emphasize the external aspects, such as user satisfaction, functionality, and performance.



McCall’s Model

McCall’s Model

- This model was developed in the 1977s by James McCall and his colleagues at the US Air Force.
- McCall’s Quality Model aims to cover the gap between **users and developers** by highlighting several kinds of software quality factors that reflect both the views of users and developers’ interests.
- This model is incorporated with many attributes, termed software factors, which influence software.

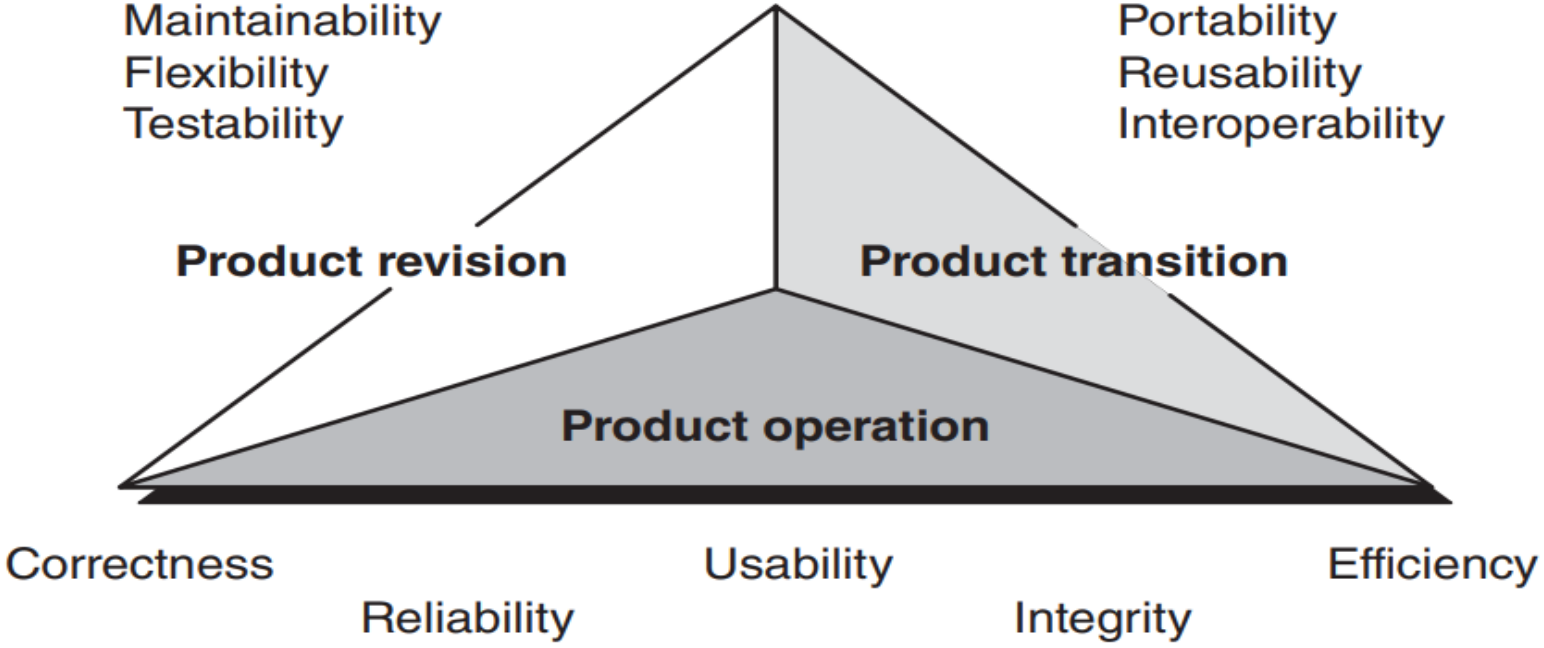
McCall’s Model

- The model distinguishes between two levels of quality attributes:
  - Quality Factors
  - Quality Criteria
- **Quality Factors:** The higher-level quality attributes that can be accessed directly are called quality factors. These attributes are external. The attributes at this level are given more importance by the users and managers.
- **Quality Criteria:** The lower or second-level quality attributes that can be accessed either subjectively or objectively are called Quality Criteria. These attributes are internal. Each quality factor has many second-level quality attributes or quality criteria.

McCall’s Model

- This model classifies all software requirements into 11 software quality factors.
- The 11 factors are organized into three product quality factors:
  - **Product Operation**,
  - **Product Revision**
  - **Product Transition**.

McCall’s Model - Factors of Product Quality



Factors of Product Quality - Product Operation

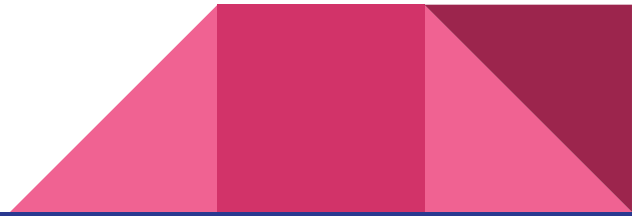
- Its operation characteristics
- Product Operation includes five software quality factors, which are related to the requirements that directly affect the operation of the software such as operational performance, convenience, ease of usage, and correctness.
- These factors help in providing a better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

Factors of Product Quality - Product Operation

- Correctness: The extent to which software meets its requirements specification.
- Efficiency: The number of hardware resources and code the software, needs to perform a function.
- Integrity: The extent to which the software can control an unauthorized person from accessing the data or software.
- Reliability: The extent to which software performs its intended functions without failure.
- Usability: The extent of effort required to learn, operate, and understand the functions of the software.

Factors of Product Quality - Product Revision

- Ability to undergo changes
- 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.



Factors of Product Quality - Product Revision

- Maintainability: The effort required to detect and correct an error during maintenance.
- Flexibility: The effort needed to improve an operational software program.
- Testability: The effort required to verify software to ensure that it meets the specified requirements..

Factors of Product Quality - Product Transition

Adaptability to new environments

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.

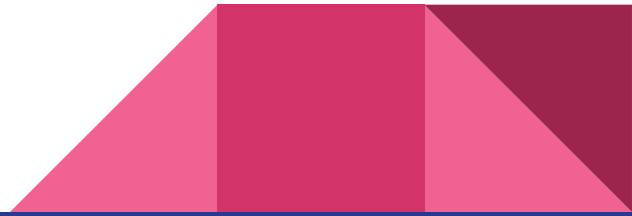
- Portability: The effort required to transfer a program from one platform to another.
- Re-usability: The extent to which the program’s code can be reused in other applications.
- Interoperability: The effort required to integrate two systems.

Quality Categories	Quality Factors	Broad Objectives
Product operation	Correctness	Does it do what the customer wants?
	Reliability	Does it do it accurately all of the time?
	Efficiency	Does it quickly solve the intended problem?
	Integrity	Is it secure?
	Usability	Can I run it?
Product revision	Maintainability	Can it be fixed?
	Testability	Can it be tested?
	Flexibility	Can it be changed?
Product transition	Portability	Can it be used on another machine?
	Reusability	Can parts of it be reused?
	Interoperability	Can it interface with another system?

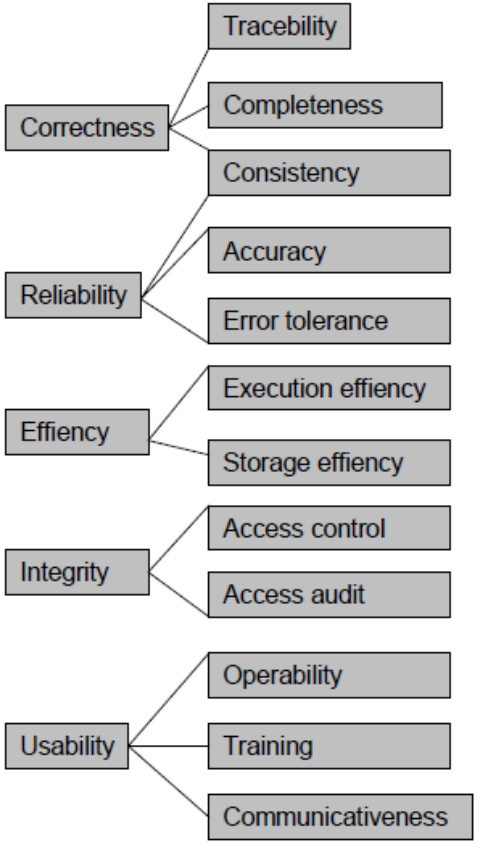
McCall’s Model – Quality Criteria

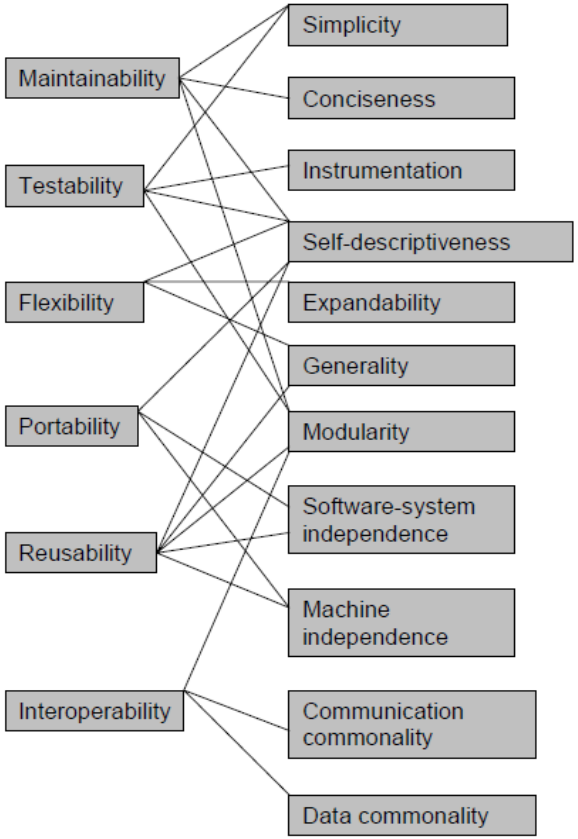
The model furthermore details the three types of quality characteristics (major perspectives) in a hierarchy of factors, criteria and metrics:

- 11 Factors (To specify): They describe the external view of the software, as viewed by the users.
- 23 quality criteria (To build): They describe the internal view of the software, as seen by the developer.
- Metrics (To control): They are defined and used to provide a scale and method for measurement.



McCall's Model – Quality Criteria





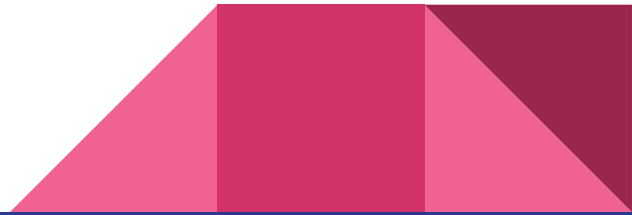
Boehm’s Model

Boehm’s Model

- In 1978, B.W. Boehm introduced his software quality model, which defines software quality through a hierarchical structure of attributes and metrics.
- This model is similar to the McCall Quality Model but encompasses a wider range of characteristics, including hardware performance-related ones.
- Boehm’s model categorizes quality attributes into three levels: primary uses (high-level characteristics), intermediate constructs (mid-level characteristics), and primitive constructs (basic characteristics).

Boehm’s Model – High Level Characteristics

- As-Is Utility:: It defines the way a utility signifies the as-is utility. It creates a question of how easily, reliably and efficiently an as can be utilized.
- Maintainability:: This aspect decides how convenient it is to understand, change or re-evaluate a process.
- Portability: This aspect helps in deciding an effective way to change an environment.

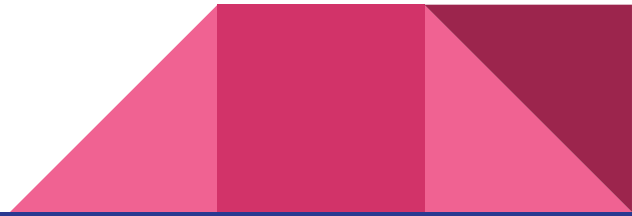


Boehm’s Model – Intermediate Level Characteristics

- Flexibility: It is very easy to amend the software as per the requirement. Parameters of the software should be so flexible that they can react on numerous situations.
- Reliability: Software performance should be reliable with zero defects. Result should be accurate.
- Portability: Software can run on different computer' program example DOS, windows.
- Efficiency: Practical & efficient use of resources or data collected. Optimum utilization of resources should be made.

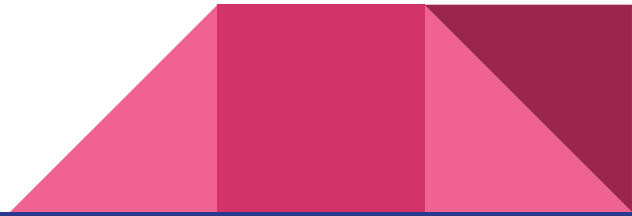
Boehm’s Model – Intermediate Level Characteristics

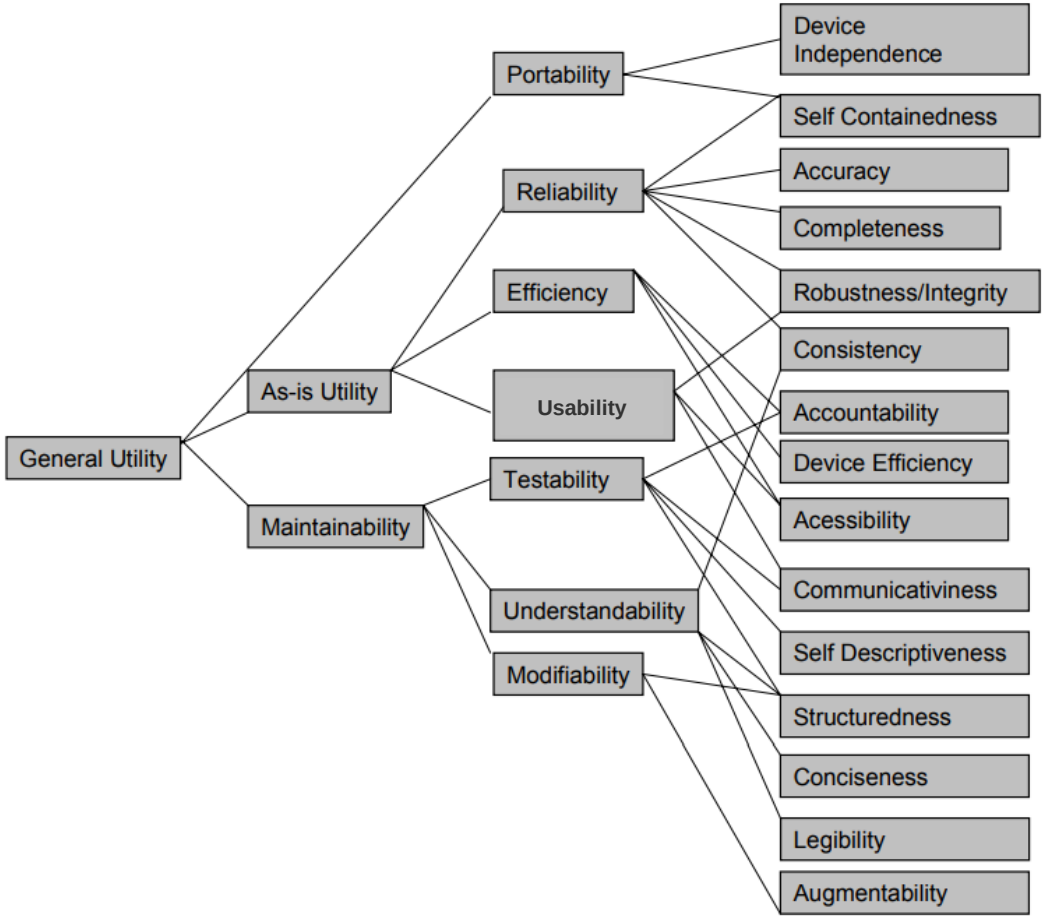
- Testability: Software should be tested easily and as a result users can easily check that the results are correct, so that they can rely on result blindly.
- Understandability: Software should be simple to understand for users so that they can use it properly and efficiently.
- Usability: Users can apply it easily and comfortably.



Boehm’s Model – Primitive Level Characteristics

- Device Independence
- Accuracy
- Completeness
- Consistency
- Device Efficiency
- Accessibility
- Communicativeness
- Self-descriptiveness
- Legibility
- Structuredness
- Conciseness
- Augment-ability



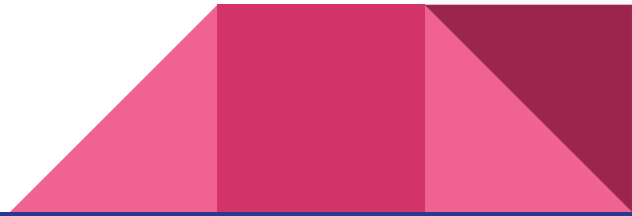

General Utility
As-is Utility
Maintainability
Portability
Reliability
Efficiency
Usability
Testability
Understandability
Modifiability
Device Independence
Self Containedness
Accuracy
Completeness
Robustness/Integrity
Consistency
Accountability
Device Efficiency
Acessibility
Communicativiness
Self Descriptiveness
Structuredness
Conciseness
Legibility
Augmentability

Boehm’s Model – Primitive Level Characteristics

- The lowest level structure of the characteristics hierarchy in Boehm’s model is the primitive characteristics metrics hierarchy.
- The primitive characteristics provide the foundation for defining qualities metrics.
- which was one of the goals when Boehm constructed his quality model.
- Consequently, the model presents one ore more metrics supposedly measuring a given primitive characteristic.

Activity 01

- Compare and contrast McCall's and Boehm’s Models.



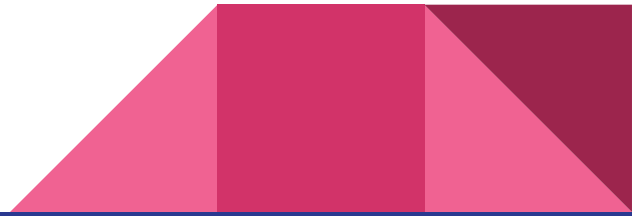
FURPS/FURPS+ Model

FURPS

- A later, and perhaps somewhat less renown, model that is structured in basically the same manner as the previous two quality models.
- FURPS stands for:
  - Functionality – which may include feature sets, capabilities and security
  - Usability - which may include human factors, aesthetics, consistency in the user interface, online and context-sensitive help, wizards and agents, user documentation, and training materials
  - Reliability - which may include frequency and severity of failure, recoverability, predictability, accuracy, and mean time between failure (MTBF)

FURPS

- Performance - imposes conditions on functional requirements such as speed, efficiency, availability, accuracy, throughput, response time, recovery time, and resource usage
- Supportability - which may include testability, extensibility, adaptability, maintainability, compatibility, configurability, serviceability, installability, localizability (internationalization)

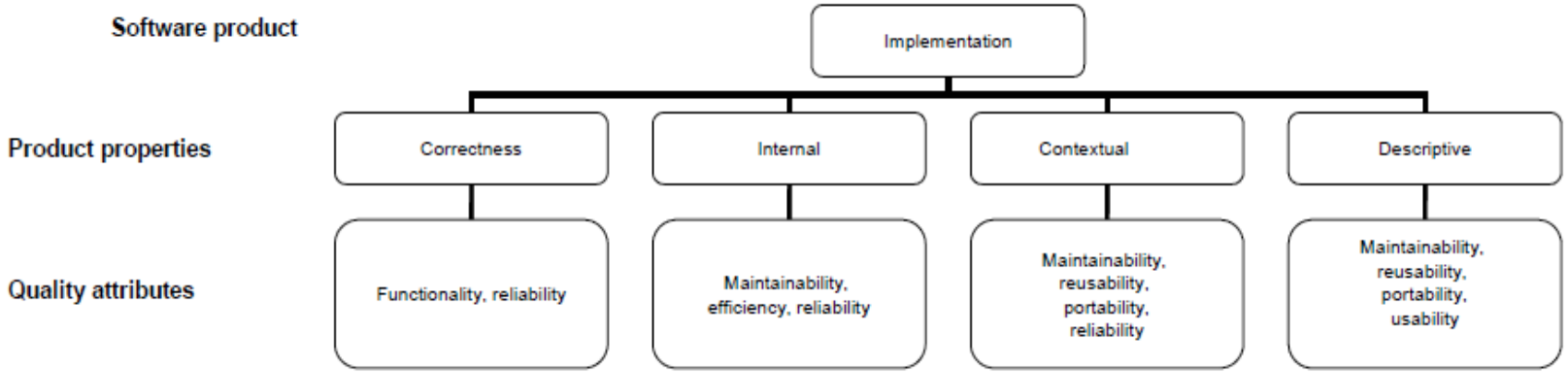


Dromey's Quality Model

Dromey's Quality Model

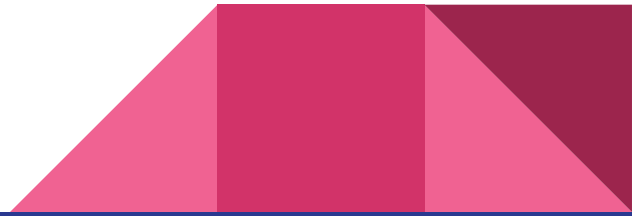
- An even more recent model similar to the McCall's, Boehm's and the FURPS(+) quality model, is the quality model presented by R. Geoff Dromey .
- Dromey proposes a product based quality model that recognizes that quality evaluation differs for each product and that a more dynamic idea for modeling the process is needed to be wide enough to apply for different systems.
- Dromey is focusing on the relationship between the quality attributes and the sub-attributes, as well as attempting to connect software product properties with software quality attributes.

Dromey's Quality Model



Dromey's Quality Model

- Correctness : Evaluates if some basic principles are violated.
- Internal : Measure how well a component has been deployed according to its intended use.
- Contextual : Deals with the external influences by and on the use of a component.
- Descriptive : Measure the descriptiveness of a component



Dromey's Quality Model

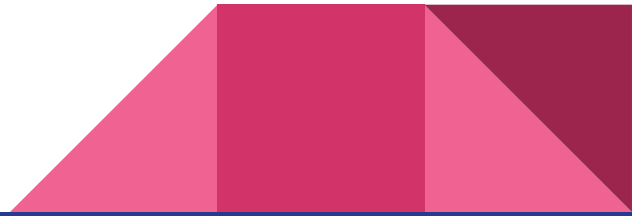
- Dromey's Quality Model is further structured around a 5 step process:

1) Chose a set of high-level quality attributes necessary for the evaluation.
2) List components/modules in your system.
3) Identify quality-carrying properties for the components/modules (qualities of the component that have the most impact on the product properties from the list above).
4) Determine how each property effects the quality attributes.
5) Evaluate the model and identify weaknesses.

ISO 25010 Model

ISO 25010 Model

- ISO 25010, titled “Systems and software engineering – Systems and software Quality Requirements and Evaluation (SQuaRE) – System and software quality models”, is a software quality standard.
- It describes the models, consisting of characteristics and sub-characteristics in use together with practical guidance on the use of the quality models.



ISO 25010 Model

SOFTWARE PRODUCT QUALITY								
FUNCTIONAL SUITABILITY	PERFORMANCE EFFICIENCY	COMPATIBILITY	INTERACTION CAPABILITY	RELIABILITY	SECURITY	MAINTAINABILITY	FLEXIBILITY	SAFETY
FUNCTIONAL COMPLETENESS FUNCTIONAL CORRECTNESS FUNCTIONAL APPROPRIATENESS iso25000.com	TIME BEHAVIOUR RESOURCE UTILIZATION CAPACITY	CO-EXISTENCE INTEROPERABILITY	APPROPRIATENESS RECOGNIZABILITY LEARNABILITY OPERABILITY USER ERROR PROTECTION USER ENGAGEMENT INCLUSIVITY USER ASSISTANCE SELF-DESCRIPTIVENESS	FAULTLESSNESS AVAILABILITY FAULT TOLERANCE RECOVERABILITY	CONFIDENTIALITY INTEGRITY NON-REPUDIATION ACCOUNTABILITY AUTHENTICITY RESISTANCE	MODULARITY REUSABILITY ANALYSABILITY MODIFIABILITY TESTABILITY	ADAPTABILITY SCALABILITY INSTALLABILITY REPLACEABILITY	OPERATIONAL CONSTRAINT RISK IDENTIFICATION FAIL SAFE HAZARD WARNING SAFE INTEGRATION

ISO 25010 - Functional Suitability

- This characteristic represents the degree to which a product or system provides functions that meet stated and implied needs when used under specified conditions.
  This characteristic is composed of the following sub-characteristics:
  - Functional completeness - Degree to which the set of functions covers all the specified tasks and intended users' objectives.
  - Functional correctness - Degree to which a product or system provides accurate results when used by intended users.
  - Functional appropriateness - Degree to which the functions facilitate the accomplishment of specified tasks and objectives.

ISO 25010 - Performance Efficiency

- This characteristic represents the degree to which a product performs its functions within specified time and throughput parameters and is efficient in the use of resources (such as CPU, memory, storage, network devices, energy, materials...) under specified conditions. This characteristic is composed of the following sub-characteristics:
  - Time behaviour - Degree to which the response time and throughput rates of a product or system, when performing its functions, meet requirements.
  - Resource utilization - Degree to which the amounts and types of resources used by a product or system, when performing its functions, meet requirements.
  - Capacity - Degree to which the maximum limits of a product or system parameter meet requirements.

ISO 25010 - Compatibility

- Degree to which a product, system or component can exchange information with other products, systems or components, and/or perform its required functions while sharing the same common environment and resources. This characteristic is composed of the following sub-characteristics:
  - Co-existence - Degree to which a product can perform its required functions efficiently while sharing a common environment and resources with other products, without detrimental impact on any other product.
  - Interoperability - Degree to which a system, product or component can exchange information with other products and mutually use the information that has been exchanged.

ISO 25010 - Interaction Capability

- Degree to which a product or system can be interacted with by specified users to exchange information to the user interface to complete specific tasks in a variety of contexts of use. This characteristic is composed of the following sub-characteristics:
  - Appropriateness recognizability - Degree to which users can recognize whether a product or system is appropriate for their needs.
  - Learnability - Degree to which the functions of a product or system can be learnt to be used by specified users within a specified amount of time.
  - Operability - Degree to which a product or system has attributes that make it easy to operate and control.

ISO 25010 - Interaction Capability

- User error protection. Degree to which a system prevents users against operation errors.
- User engagement - Degree to which a user interface presents functions and information in an inviting and motivating manner encouraging continued interaction.
- Inclusivity - Degree to which a product or system can be used by people of various backgrounds (such as people of various ages, abilities, cultures, ethnicities, languages, genders, economic situations, etc.).
- User assistance - Degree to which a product can be used by people with the widest range of characteristics and capabilities to achieve specified goals in a specified context of use.
- Self-descriptiveness - Degree to which a product presents appropriate information, where needed by the user, to make its capabilities and use immediately obvious to the user without excessive interactions with a product or other resources (such as user documentation, help desks or other users).

ISO 25010 - Reliability

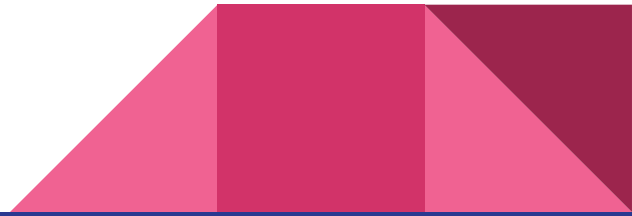
- Degree to which a system, product or component performs specified functions under specified conditions for a specified period of time. This characteristic is composed of the following sub-characteristics:
  - Faultlessness - Degree to which a system, product or component performs specified functions without fault under normal operation.
  - Availability - Degree to which a system, product or component is operational and accessible when required for use.
  - Fault tolerance - Degree to which a system, product or component operates as intended despite the presence of hardware or software faults.
  - Recoverability - Degree to which, in the event of an interruption or a failure, a product or system can recover the data directly affected and re-establish the desired state of the system.

ISO 25010 - Security

- Degree to which a product or system defends against attack patterns by malicious acts and protects information and data so that persons or other products or systems have the degree of data access appropriate to their types and levels of authorization. This characteristic is composed of the following sub-characteristics:
  - Confidentiality - Degree to which a product or system ensures that data are accessible only to those authorized to have access.
  - Integrity - Degree to which a system, product or component ensures that the state of its system and data are protected from unauthorized modification or deletion either by malicious action or computer error.

ISO 25010 - Security

- Non-repudiation - Degree to which actions or events can be proven to have taken place so that the events or actions cannot be repudiated later.
- Accountability - Degree to which the actions of an entity can be traced uniquely to the entity.
- Authenticity - Degree to which the identity of a subject or resource can be proved to be the one claimed.
- Resistance - Degree to which the product or system sustains operations while under attack from a malicious actor.

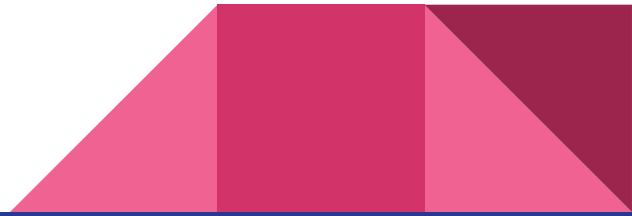


ISO 25010 - Maintainability

- This characteristic represents the degree of effectiveness and efficiency with which a product or system can be modified to improve it, correct it or adapt it to changes in environment, and in requirements. This characteristic is composed of the following sub-characteristics:
  - Modularity - Degree to which a system or computer program is composed of discrete components such that a change to one component has minimal impact on other components.
  - Reusability - Degree to which a product can be used as an asset in more than one system, or in building other assets.

ISO 25010 - Maintainability

- Analysability - Degree of effectiveness and efficiency with which it is possible to assess the impact on a product or system of an intended change to one or more of its parts, to diagnose a product for deficiencies or causes of failures, or to identify parts to be modified.
- Modifiability - Degree to which a product or system can be effectively and efficiently modified without introducing defects or degrading existing product quality.
- Testability - Degree of effectiveness and efficiency with which test criteria can be established for a system, product or component and tests can be performed to determine whether those criteria have been met



ISO 25010 - Flexibility

- Degree to which a product can be adapted to changes in its requirements, contexts of use or system environment. This characteristic is composed of the following sub-characteristics:
  - Adaptability - Degree to which a product or system can effectively and efficiently be adapted for or transferred to different hardware, software or other operational or usage environments.
  - Scalability - Degree to which a product can handle growing or shrinking workloads or to adapt its capacity to handle variability.
  - Installability - Degree of effectiveness and efficiency with which a product or system can be successfully installed and/or uninstalled in a specified environment.
  - Replaceability - Degree to which a product can replace another specified software product for the same purpose in the same environment.

ISO 25010 - Safety

- This characteristic represents the degree to which a product under defined conditions to avoid a state in which human life, health, property, or the environment is endangered. This characteristic is composed of the following sub-characteristics:
  - Operational constraint - Degree to which a product or system constrains its operation to within safe parameters or states when encountering operational hazard.
  - Risk identification - Degree to which a product can identify a course of events or operations that can expose life, property or environment to unacceptable risk.

ISO 25010 - Safety

- Fail safe - Degree to which a product can automatically place itself in a safe operating mode, or to revert to a safe condition in the event of a failure.
- Hazard warning - Degree to which a product or system provides warnings of unacceptable risks to operations or internal controls so that they can react in sufficient time to sustain safe operations.
- Safe integration - Degree to which a product can maintain safety during and after integration with one or more components.