Appendix 3.1 ISO/IEC 25010 Quality Model & Attributes

(Summarized from https://www.iso.org/obp/ui/#iso:std:iso-iec:25010:ed-1:v1:en)

4.1 Quality in use model

Quality in use is the degree to which a system can be used by specific users to meet their needs to achieve specific goals with effectiveness, efficiency, freedom from risk and satisfaction in specific contexts of use.

The properties of quality in use are categorized into five characteristics: effectiveness, efficiency, satisfaction, freedom from risk and context coverage (Table 1).

Table 1 — Quality in use characteristics and subcharacteristics

Characteristic	Subcharacteristics
Effectiveness	-
Efficiency	-
Satisfaction	Usefulness
	Trust
	Pleasure
	Comfort
Freedom from risk	Economic risk mitigation
	Health and safety risk mitigation
	Environmental risk mitigation
Context coverage	Context completeness
	Flexibility

NOTE Usability (4.2.4) is defined as a subset of quality in use consisting of effectiveness, efficiency and satisfaction, for consistency with its established meaning.

4.1.1 effectiveness: accuracy and completeness with which users achieve specified goals

4.1.2 efficiency: resources expended in relation to the accuracy and completeness with which users achieve goals.
Note: Relevant resources can include time to complete the task (human resources), materials, or the financial cost of usage.

4.1.3 satisfaction: degree to which user needs are satisfied when a system is used in a specified context of use.
Note[1]: For a user who does not directly interact with the system, only purpose accomplishment and trust are relevant.
Note[2]: Satisfaction is the user's response to interaction with the system, and includes attitudes towards use of the product.

4.1.3.1 usefulness: degree to which a user is satisfied with their perceived achievement of pragmatic goals, including the results of use and the consequences of use

4.1.3.2 trust: degree to which a user or other stakeholder has confidence that a system will behave as intended

4.1.3.3 pleasure: degree to which a user obtains pleasure from fulfilling their personal needs
Note: Personal needs can include needs to acquire new knowledge and skills, to communicate personal identity and to provoke pleasant memories.

4.1.3.4 comfort: degree to which the user is satisfied with physical comfort

4.1.4 freedom from risk: degree to which a system mitigates the potential risk to economic status, human life, health, or the environment
Note: Risk is a function of the probability of occurrence of a given threat and the potential adverse consequences of that threat's occurrence.

4.1.4.1 economic risk mitigation: degree to which a system mitigates the potential risk to financial status, efficient operation, commercial property, reputation or other resources in the intended contexts of use

4.1.4.2 health and safety risk mitigation: degree to which a system mitigates the potential risk to people in the intended contexts of use

4.1.4.3 environmental risk mitigation: degree to which a system mitigates the potential risk to property or the environment in the intended contexts of use

4.1.5 context coverage: degree to which a system can be used with effectiveness, efficiency, freedom from risk and satisfaction in both specified contexts of use and in contexts beyond those initially explicitly identified (Note: Context of use is relevant to both quality in use and some product quality (sub)characteristics (where it is referred to as “specified conditions”).)

4.1.5.1 context completeness: degree to which a system can be used with effectiveness, efficiency, freedom from risk and satisfaction in all the specified contexts of use
Note: Context completeness can be specified or measured either as the degree to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency, freedom from risk and satisfaction in all the intended contexts of use, or by the presence of product properties that support use in all the intended contexts of use.
EXAMPLE: The extent to which software is usable using a small screen, with low network bandwidth, by a non-expert user; and in a fault-tolerant mode (e.g. no network connectivity).

4.1.5.2 flexibility: degree to which a system can be used with effectiveness, efficiency, freedom from risk and satisfaction in contexts beyond those initially specified in the requirements.
Note[1]: Flexibility can be achieved by adapting a product (see 4.2.8.1) for additional user groups, tasks and cultures.
Note[2]: Flexibility enables products to take account of circumstances, opportunities and individual preferences that had not been anticipated in advance.
Note[3]: If a product is not designed for flexibility, it might not be safe to use the product in unintended contexts.
Note[4]: Flexibility can be measured either as the extent to which a product can be used by additional types of users to achieve additional types of goals with effectiveness, efficiency, freedom from risk and satisfaction in additional types of contexts of use, or by a capability to be modified to support adaptation for new types of users, tasks and environments, and suitability for individualization.

4.2 Product quality model

The product quality model categorizes product quality properties into eight characteristics (functional suitability, reliability, performance efficiency, usability, security, compatibility, maintainability and portability). Each characteristic is composed of a set of related subcharacteristics (Table 2).

Table 2 — Product Quality characteristics and subcharacteristics

Functional suitability
Functional completeness
Functional correctness
Functional appropriateness
Performance efficiency
Time behaviour
Resource utilization
Capacity
Compatibility
Co-existence
Interoperability
Usability
Appropriateness recognizability
Learnability
Operability
User error protection
User interface aesthetics
Accessibility

Reliability
Maturity
Availability
Fault tolerance
Recoverability
Security
Confidentiality
Integrity
Non-repudiation
Accountability
Authenticity
Maintainability
Modularity
Reusability
Analysability
Modifiability
Testability
Portability
Adaptability
Installability
Replaceability

4.2.1 functional suitability: degree to which a system provides functions that meet stated and implied needs when used under specified conditions. Note: Functional suitability is only concerned with whether the functions meet stated and implied needs, not the functional specification.

4.2.1.1 functional completeness: degree to which the set of functions covers all the specified tasks and user objectives

4.2.1.2 functional correctness: degree to which a system provides the correct results with the needed degree of precision

4.2.1.3 functional appropriateness: degree to which the functions facilitate the accomplishment of specified tasks and objectives. EXAMPLE: A user is only presented with the necessary steps to complete a task, excluding any unnecessary steps.

4.2.2 performance efficiency: performance relative to the amount of resources used under stated conditions.

Note: Resources can include other software products, the software and hardware configuration of the system, and materials (e.g. print paper, storage media).

4.2.2.1 time behaviour: degree to which the response and processing times and throughput rates of a system, when performing its functions, meet requirements

4.2.2.2 resource utilization: degree to which the amounts and types of resources used by a system, when performing its functions, meet requirements

Note: Human resources are included as part of **efficiency** (4.1.2).

4.2.2.3 capacity: degree to which the maximum limits of a system parameter meet requirements. Note: Parameters can include the number of items that can be stored, number of concurrent users, communication bandwidth, throughput of transactions, and size of database.)

4.2.3 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 hardware or software environment

4.2.3.1 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

4.2.3.2 interoperability: degree to which two or more systems, products or components can exchange information and use the information that has been exchanged

4.2.4 usability: degree to which a system can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use.
Note: Usability can either be specified or measured as a product quality characteristic in terms of its subcharacteristics, or specified or measured directly by measures that are a subset of quality in use.)

4.2.4.1 appropriateness recognizability: degree to which users can recognize whether a system is appropriate for their needs (SEE: 4.2.1.3)
Note[1]: Appropriateness recognizability will depend on the ability to recognize the appropriateness of the system's functions from initial impressions of the system and/or any associated documentation.
Note: The information provided by the system can include demonstrations, tutorials, documentation or, for a web site, the information on the home page.

4.2.4.2 learnability: degree to which a system can be used by specified users to achieve specified goals of learning to use the system with effectiveness, efficiency, freedom from risk and satisfaction in a specified context of use
Note: Can be specified or measured either as the extent to which a system can be used by specified users to achieve specified goals of learning to use the system with effectiveness, efficiency, freedom from risk and satisfaction in a specified context of use, or by product properties corresponding to suitability for learning.

4.2.4.3 operability: degree to which a system has attributes that make it easy to operate and control
Note: Operability corresponds to controllability, (operator) error tolerance and conformity with user expectations.

4.2.4.4 user error protection: degree to which a system protects users against making errors

4.2.4.5 user interface aesthetics: degree to which a user interface enables pleasing and satisfying interaction for the user
Note: This refers to properties of the system that increase the pleasure and satisfaction of the user, such as the use of colour and the nature of the graphical design.

4.2.4.6 accessibility: degree to which a system can be used by people with the widest range of characteristics and capabilities to achieve a specified goal in a specified context of use
Note[1]: The range of capabilities includes disabilities associated with age.
Note[2]: Accessibility for people with disabilities can be specified or measured either as the extent to which a system can be used by users with specified disabilities to achieve specified goals with effectiveness, efficiency, freedom from risk and satisfaction in a specified context of use, or by the presence of product properties that support accessibility.

4.2.5 reliability: degree to which a system, component performs specified functions under specified conditions for a specified period of time
Note[1]: Wear does not occur in software. Limitations in reliability are due to faults in requirements, design and implementation, or due to contextual changes.
Note[2]: Dependability characteristics include availability and its inherent or external influencing factors, such as availability, reliability (including fault tolerance and recoverability), security (including confidentiality and integrity), maintainability, durability, and maintenance support.

4.2.5 reliability: (cont'd)

4.2.5.1 maturity: degree to which a system, component meets needs for reliability under normal operation
Note: The concept of maturity can also be applied to other quality characteristics to indicate the degree to which they meet required needs under normal operation.

4.2.5.2 availability: degree to which a system, component is operational and accessible when required for use
Note: Externally, availability can be assessed by the proportion of total time during which the system, component is in an up state. Availability is therefore a combination of maturity (which governs the frequency of failure), fault tolerance and recoverability (which governs the length of down time following each failure.

4.2.5.3 fault tolerance: degree to which a system, component operates as intended despite the presence of hardware or software faults

4.2.5.4 recoverability: degree to which, in the event of an interruption or a failure, a system can recover the data directly affected and re-establish the desired state of the system
Note: Following a failure, a computer system will sometimes be down for a period of time, the length of which is determined by its recoverability.

4.2.6: security: degree to which a system 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
Note[1]: As well as data stored in or by a system, security also applies to data in transmission.
Note[2]: Survivability (the degree to which a system continues to fulfil its mission by providing essential services in a timely manner in spite of the presence of attacks) is covered by **recoverability** (4.2.5.4).
Note[3]: Immunity (the degree to which a system is resistant to attack) is covered by **integrity** (4.2.6.2).
Note[4]: Security contributes to **trust** (4.1.3.2).

4.2.6.1 confidentiality:
degree to which a system ensures that data are accessible only to those authorized to have access

4.2.6.2 integrity: degree to which a system, component prevents unauthorized access to, or modification of, computer programs or data

4.2.6.3 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

4.2.6.4 accountability: degree to which the actions of an entity can be traced uniquely to the entity

4.2.6.5 authenticity: degree to which the identity of a subject or resource can be proved to be the one claimed

4.2.7 maintainability degree of effectiveness and efficiency with which a system can be modified by the intended maintainers
Note[1]: Modifications can include corrections, improvements or adaptation of the software to changes in environment, and in requirements and functional specifications. Modifications include those carried out by specialized support staff, and those carried out by business or operational staff, or end users.
Note[2]: Maintainability includes installation of updates and upgrades.
Note[3]: Maintainability can be interpreted as either an inherent capability of the system to facilitate maintenance activities, or the quality in use experienced by the maintainers for the goal of maintaining the system.

4.2.7.1 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

4.2.7.2 reusability: degree to which an asset can be used in more than one system, or in building other assets.

4.2.7 maintainability (cont'd)

4.2.7.3 analysability: degree of effectiveness and efficiency with which it is possible to assess the impact on a system of an intended change to one or more of its parts, or to diagnose a product for deficiencies or causes of failures, or to identify parts to be modified
Note: Implementation can include providing mechanisms for the system to analyse its own faults and provide reports prior to a failure or other event.

4.2.7.4 modifiability: degree to which a system can be effectively and efficiently modified without introducing defects or degrading existing product quality
Note[1]: Implementation includes coding, designing, documenting and verifying changes.
Note[2]: **Modularity** (4.2.7.1) and **analysability** (4.2.7.3) can influence modifiability.
Note[3]: Modifiability is a combination of changeability and stability.

4.2.7.5 testability: degree of effectiveness and efficiency with which test criteria can be established for a system, component and tests can be performed to determine whether those criteria have been met

4.2.8 portability: degree of effectiveness and efficiency with which a system, component can be transferred from one hardware, software or other operational or usage environment to another
Note: Portability can be interpreted as either an inherent capability of the system to facilitate porting activities, or the quality in use experienced for the goal of porting the system.

4.2.8.1 adaptability: degree to which a system can effectively and efficiently be adapted for different or evolving hardware, software or other operational or usage environments
Note[1]: Adaptability includes the scalability of internal capacity (e.g. screen fields, tables, transaction volumes, report formats, etc.
Note[2]: Adaptations include those carried out by specialized support staff, and those carried out by business or operational staff, or end users.
Note[3]: If the system is to be adapted by the end user, adaptability corresponds to suitability for individualization.

4.2.8.2 installability: degree of effectiveness and efficiency with which a system can be successfully installed and/or uninstalled in a specified environment
Note: If the system is to be installed by an end user, installability can affect the resulting functional appropriateness and operability.

4.2.8.3 replaceability: degree to which a product can replace another specified software product for the same purpose in the same environment
Note[1]: Replaceability of a new version of a software product is important to the user when upgrading. Note[2]: Replaceability can include attributes of both installability and adaptability. The concept has been introduced as a subcharacteristic of its own because of its importance.
Note[3]: Replaceability will reduce lock-in risk: so that other software products can be used in place of the present one, for example by the use of standardized file formats.

4.3 General

4.3.1 asset: anything that has value to a person or organization
Note: In this International Standard assets are work products such as requirements documents, source code modules, measurement definitions, etc.

4.3.2 benchmark: standard against which results can be measured or assessed

4.3.3 component: entity with discrete structure, such as an assembly or software module, within a system considered at a particular level of analysis

4.3.4 direct user: person who interacts with the product
Note: This includes primary and secondary users.

4.3.5 external measure of software quality: measure of the degree to which a software product enables the behaviour of a system to satisfy stated and implied needs for the system including the software to be used under specified conditions
Note: Attributes of the behaviour can be verified and/or validated by executing the software product during testing and operation. EXAMPLE: The number of failures found during testing is an external measure of software quality related to the number of faults present in the computer system. The two measures are not necessarily identical since testing might not find all faults, and a fault can give rise to apparently different failures in different circumstances.

4.3.6 indirect user: person who receives output from a system, but does not interact with the system

4.3.7 internal measure of software quality: measure of the degree to which a set of static attributes of a software product satisfies stated and implied needs for the software product to be used under specified conditions
Note[1]: Static attributes include those that relate to the software architecture, structure and its components.
Note[2]: Static attributes can be verified by review, inspection, simulation and/or automated tools.
EXAMPLE: Complexity measures and the number, severity, and failure frequency of faults found in a walk through are internal software quality measures made on the product itself.

4.3.8 quality in use: degree to which a system can be used by specific users to meet their needs to achieve specific goals with effectiveness, efficiency, freedom from risk and satisfaction in specific contexts of use

4.3.9 quality property: measurable component of quality

4.3.10 quality measure: measure that is defined as a measurement function of two or more values of quality measure elements.

4.3.11 quality measure element: measure defined in terms of an attribute and the measurement method for quantifying it, including optionally the transformation by a mathematical function

4.3.12 risk: function of the probability of occurrence of a given threat and the potential adverse consequences of that threat's occurrence

4.3.13 software quality: degree to which a software product satisfies stated and implied needs when used under specified conditions.

4.3.14 software quality requirement: requirement that a software quality attribute be present in software

4.3.15 stakeholder: individual or organization having a right, share, claim or interest in a system or in its possession of characteristics that meet their needs and expectations

4.3.16 user: individual or group that interacts with a system or benefits from a system during its utilization
Note: Primary and secondary users interact with a system, and primary and indirect users can benefit from a system (see 3.6).

4.4 Terms and definitions from ISO/IEC 25000

4.4.1 attribute: inherent property or characteristic of an entity that can be distinguished quantitatively or qualitatively by human or automated means.

4.4.2 context of use: users, tasks, equipment (hardware, software and materials), and the physical and social environments in which a product is used

4.4.3 end user: individual person who ultimately benefits from the outcomes of the system (Note: The end user can be a regular operator of the software a casual user such as a member of the public.)

4.4.4 implied needs: needs that may not have been stated but are actual needs (Note: Some implied needs only become evident when the software product is used in particular conditions. EXAMPLE: Implied needs include needs not stated but implied by other stated needs and needs not stated because they are considered to be evident or obvious.)

4.4.5 measure, noun: variable to which a value is assigned as the result of measurement (Note[1]: The term “measures” is used to refer collectively to base measures, derived measures, and indicators.

4.4.6 measure, verb: make a measurement

4.4.7 measurement: set of operations having the object of determining a value of a measure Note: Measurement can include assigning a qualitative category such as the language of a source program (ADA, C, COBOL, etc.

4.4.8 quality model: defined set of characteristics, and of relationships between them, which provides a framework for specifying quality requirements and evaluating quality

4.4.9 software product: set of computer programs, procedures, and possibly associated documentation and data
Note[1]: Products include intermediate products, and products intended for users such as developers and maintainers.
Note[2]: In SQuaRE standards, software quality has the same meaning as software product quality.

4.4.10 software quality characteristic: category of software quality attributes that bears on software quality
Note: Software quality characteristics can be refined into multiple levels of subcharacteristics and finally into software quality attributes.

4.4.11 system: combination of interacting elements organized to achieve one or more stated purposes
Note[1]: A system may be considered as a as the services it provides. Note[2]: In practice, the interpretation of its meaning is frequently clarified by the use of an associative noun, e.g. aircraft system. Alternatively, the word system may be substituted simply by a context-dependent synonym, e.g. aircraft, though this may then obscure a system principles perspective.

4.4.12 user: individual or group that benefits from a system during its utilization

4.4.13 validation: confirmation, through the provision of objective evidence, that the requirements for a specific intended use or application have been fulfilled
Note[1]: “Validated” is used to designate the corresponding status. Note[2]: In design and development, validation concerns the process of examining a product to determine conformity with user needs. Note[3]: Validation is normally performed on the final product under defined operating conditions. It can be necessary in earlier stages. Note[4]: Multiple validations might be necessary if there are different intended uses.

4.4.14 verification: confirmation, through the provision of objective evidence, that specified requirements have been fulfilled
Note[1]: “Verified” is used to designate the corresponding status.
Note[2]: In design and development, verification concerns the process of examining the result of a given activity to determine conformity with the stated requirement for that activity.