DIGITALIZATION – IMPLEMENTATION

BORUT JEREB
University of Maribor, Faculty of Logistics, Celje, Slovenia
borut.jereb@um.si

Software is intangible, and its creation is still associated with individuals in the digitization of processes. As software has become highly complex, organizations developing and using it face new opportunities and challenges throughout its entire life cycle. ISO/IEC 12207:2011 is a standard that provides a framework for describing the life cycle of software systems based on an interdisciplinary approach. The software life cycle encompasses processes from conception to development or acquisition, through testing, deployment, maintenance, and eventual retirement. Understanding the life cycle helps all stakeholders establish systematic and organized management of software-related work, which is crucial for improving quality and reducing issues in the digitization of business processes. Software quality, crucial for user satisfaction, is reflected in functionality, reliability, performance, usability, maintainability, portability, security, and scalability. ISO/IEC 25010:2017 is a standard defining a comprehensive quality model, enabling the assessment and improvement of software.

DOI
https://doi.org/10.18690/um.fl.2.2026.2

ISBN
978-961-299-074-9

Keywords:
digitalization,
software lifecycle,
software quality,
ISO/IEC 12207,
ISO/IEC 25010

University of Maribor Press

1 Introduction

Software consists of a set of instructions used to perform specific tasks on computers. Software is a key component of modern computer systems. It is intangible, meaning that it does not have a physical presence like hardware. In the vast majority of cases, its creation is still associated with an individual (human) or individuals working on a joint project to digitize a process.

Software improves our daily lives and drives technological progress. This can be for work, entertainment, communication or information retrieval. Software plays a central role in the usability of computers and digital devices, enabling the automation of various processes, increasing efficiency and reducing manual labor.

The complexity of software has reached a very high level. As a result, while organizations that create and use software systems are gaining new opportunities on the one hand, they are also on the other hand facing greater challenges. These challenges exist throughout the entire system life cycle and at all levels. ISO/IEC 12207 is a standard that provides a common process framework for describing the life cycle of systems based on an interdisciplinary approach.

The software development life cycle consists of various processes, such as: development, testing, deployment, and decommissioning. The cycle can be iterative, meaning that the phases are not always sequential. For example, developers can return to the planning phase if they encounter problems during implementation. It is important to be aware of and familiar with the software life cycle because this knowledge helps us ensure that software is developed in a systematic and organized manner. This is a prerequisite for improving software quality and reducing the number of problems in all processes of its life cycle.

Software quality is a key aspect of any software development process. It refers to the level of excellence or fitness for purpose of a software product. Software quality is crucial because it directly impacts user satisfaction, reliability, and the overall success of the process being digitized. By focusing on functionality, reliability, performance, usability, maintainability, portability, security, and scalability, developers can deliver reliable software products that meet user expectations and business requirements. Regular testing, user feedback, and quality metrics help ensure continuous improvement and maintain high standards of software quality. Quality software has high value.

Software quality is therefore crucial in the implementation of all (business) processes, as it ensures that software products meet user expectations and thus provides new value through the digitalization of business. Organizations rely on established standards and frameworks to effectively achieve and assess software quality.

The ISO/IEC 25000 family of standards, also known as the SQuaRE (Software Quality Requirements and Evaluation) series, provides a comprehensive and internationally recognized set of guidelines for software quality management. The ISO/IEC 25000 series includes standards and technical reports that address various aspects of software quality characteristics, including quality models, evaluation procedures, and measurement methods.

ISO/IEC 25010 serves as the foundational standard in a series that defines a comprehensive quality model and a set of quality characteristics. Quality characteristics include functionality, reliability, usability, efficiency, maintainability, and portability. By tracking these characteristics, organizations can assess, measure, and improve the quality of their software products. This framework also provides a systematic approach that enables organizations to define quality requirements and evaluation criteria and select appropriate evaluation techniques.

Because such standards promote consistency and comparability in software quality assessment, they enable effective communication between stakeholders. They help improve decision-making processes, enable effective risk management (including costs), enhance customer satisfaction, and encourage continuous improvement. They enable organizations to achieve greater transparency, reliability, and interoperability in their digitized business processes.

Quality is thus integrated into all software lifecycles in phases such as (among other things):

- **Planning**: Quality aspects should be included in project planning to ensure that quality objectives are met.

- **Testing:** Thorough testing at various stages of the lifecycle ensures that quality standards are maintained.
- **Feedback loops:** Regular feedback from users and stakeholders helps to identify and address quality issues.
- **Continuous improvement:** The software lifecycle includes continuous improvement measures that ensure that quality standards evolve in line with changing needs and technologies.

Both, software life cycle and software quality, are related fields where quality assurance practices are embedded throughout the entire life cycle to ensure reliability, functionality, and high quality of software.

2 Software Lifecycle and ISO/IEC 12207:2017

ISO/IEC 12207:2017 is a standard that is used for the entire life cycle of software systems, products and services, including conceptualization, development, production, use, support and retirement, and for their delivery, either within or outside an organization (*ISO/IEC/IEEE 12207*, 2017). The life cycle processes in this document can be applied concurrently, iteratively and recursively to its component parts.

There is a wide diversity of software in terms of purpose, scope, complexity, size, innovation, adaptability, quantity, locations, life span and development. The standard supports this diversity and does not limit it. It applies to single software systems that are written for a single and unique solution, to software systems for wide commercial or public distribution, and to flexible software systems. It also applies to completely stand-alone software systems and to software systems that are embedded and integrated into other, larger, more complex and comprehensive systems.

Software systems, as covered by this standard, are human-made, and created and used to provide products or services in specific environments for the benefit of users and other stakeholders. Software systems may include the following system elements: hardware, software, data, people, processes (e.g., procedures for providing services to users), procedures (e.g., instructions for operators), objects, services, materials, and naturally occurring entities. Depending on the user, the software systems considered are products or services.

The perception and definition of a particular software system, its architecture, and its system elements depend on the interests and responsibilities of the stakeholders. A system of interest to one stakeholder may be viewed as a system element within a system of interest to another stakeholder. Furthermore, a system of interest may be viewed as part of the environment for a system of interest to another stakeholder.

The ISO/IEC 12207:2017 standard is used by many organizations and industries worldwide that are involved in the development, maintenance and management of software. The main groups that use this standard are:

- **Informatics and Information Technology**: This standard is most used in the IT industry, where organizations develop software solutions, applications, systems, and services. This includes both large companies that develop software solutions for the public and specialized IT companies.
- **Telecommunications:** The telecommunications industry often uses ISO/IEC 12207 to develop software used in networks and communication devices.
- **Automotive:** The automotive industry uses this standard to develop embedded software in cars, including engine control, safety systems, and infotainment solutions.
- **Healthcare:** In the healthcare industry, the standard is used to develop software for medical devices, electronic health records, and other information systems.
- **Military:** Military organizations use this standard to develop and maintain software for military systems, including intelligence, communication, and control systems.
- **Financial industry:** In the financial industry, this standard is used to develop software for financial transactions, banking systems, and asset management.
- **Aerospace:** The aerospace industry uses this standard to develop software for aircraft, satellites, and space missions.
- **Energy and other industrial sectors:** This standard can also be used in industrial sectors such as: manufacturing, energy, and others; for the development of software for process control and automation.

ISO/IEC 12207:2017 is used in all areas where software development is crucial to the functioning of organizations and where it is necessary to ensure quality management of the entire software life cycle from concept to termination. Thus, it includes, among others (*ISO/IEC/IEEE 12207*, 2017):

- **Development**: addresses the processes and activities involved in software development from initial concept and requirements definition to design, coding, testing, and implementation.
- **Maintenance:** addresses the processes and activities associated with maintaining and improving existing software systems.
- **Delivery:** includes processes for acquiring software from external suppliers or providing software to external customers.
- **Quality Assurance:** defines processes for ensuring software quality throughout its life cycle.
- **Process Improvement:** provides a basis for evaluating and improving processes that help organizations improve their software development and maintenance processes throughout their life cycle.
- **Negotiation Guide:** provides guidelines for forming agreements between customers and suppliers regarding software processes and activities.
- **Different Life Cycle Models:** supports different software life cycle models, including iterative, incremental, and classical approaches.

Also, a standard can be used in one or more ways. If the criterion is a scope, it can be used on the page (*ISO/IEC/IEEE 12207*, 2017):

- **Organizations** - to help establish the desired process environment.
- **Projects** - to help select, structure and use elements of the established environment to deliver products and services.
- **Clients and** suppliers - to help develop agreements on processes and activities.
- **Process auditors** - as a reference process model for use in conducting process audits that can be used to support organizational process improvements.

Its purpose is to provide a standardized framework for software life cycle processes and as such serves several key purposes (*ISO/IEC/IEEE 12207*, 2017):

- **Process standardization**: ISO/IEC 12207 aims to standardize software development processes and to make them consistent and repeatable across organizations and projects. This standard helps establish a common language and set of practices for software engineering.
- **Quality assurance:** Promotes the quality of software and services by defining processes and activities that ensure that software meets specified requirements and standards. By following the guidelines in ISO/IEC 12207, organizations can improve the quality of their software.
- **Risk reduction:** The standard helps reduce the risks associated with software development and maintenance by providing a structured approach. It helps identify and mitigate potential problems early in the software lifecycle.
- **Lifecycle management:** ISO/IEC 12207 covers the entire software lifecycle from concept and requirements through to design, development, testing, implementation and maintenance to retirement. It provides a comprehensive approach to managing software throughout its life.
- **Interoperability:** By providing a common framework, the standard facilitates interoperability between different software components and systems developed by different organizations. It ensures that software can work together smoothly.
- **Alignment with stakeholder requirements:** ISO/IEC 12207 emphasizes the importance of understanding and aligning software development with stakeholder requirements. This helps ensure that software products meet the expectations and requirements of users and other relevant parties.
- **Process improvement:** The standard can be used as a basis for process review and improvement. Organizations can use it to identify areas where they can improve their software development processes.
- **International compatibility:** ISO/IEC 12207 is an international standard, making it applicable and recognized worldwide. This global recognition can be particularly beneficial for organizations that participate in international collaborations or offer software to a global market.

2.1 Processes of the ISO/IEC 12207:2017 standard

Each process in this document is described in terms of the following characteristics:

- **the process name** represents the scope of the entire process,

- **the purpose** describes the goals of the process,
- **the outcomes** express the tangible results expected from the successful implementation of the process,
- **the activities** are groups of related tasks in the process,
- **the tasks** are requirements, recommendations, or permitted actions intended to support the achievement of outcomes.

The standard groups the processes that can be performed during the software system life cycle into four process groups. Each life cycle process in these groups is described in terms of its purpose and desired outcomes, with a set of related activities and tasks that can be performed to achieve these outcomes. The four process groups and the processes included in each group are shown in Figure 2.1.

The process groups are as follows (Figure 2.1) (*ISO/IEC/IEEE 12207*, 2017):

- **Negotiation/Contracting Processes:** Organizations are producers and users of software systems. One organization (as the client) may engage another organization (as the supplier) for products or services. This is accomplished with agreements (or contracts). The latter enables both clients and suppliers to realize value and support their organizations' business strategies. Negotiation processes are organizational processes that are used beyond the scope of the project life cycle, as well as during the project life cycle. Typically, organizations operate simultaneously or sequentially as both clients and suppliers of software systems. Negotiation processes can be used less formally when the client and supplier are within the same organization. They can also be used within an organization to agree on the responsibilities of the organization, the project, and technical functions.
- **Project organization processes:** These processes are concerned with providing resources that ensure that a project meets the needs and expectations of the organization's stakeholders. They are usually focused on the strategic level of managing and improving the organization's business or operations. They do this by providing and allocating resources and by managing risks in competitive or uncertain situations. They are usually used outside the scope of the project life cycle but can also be used during the project life cycle. They establish the environment in which projects are implemented. The organization: defines the processes and life cycle models used by projects; establishes, redirects or cancels

projects; provides the necessary resources, including people and financial resources; and establishes and monitors quality criteria for software systems and other products that are the result of development for internal and external customers. Project organization processes create a business image for many organizations and imply commercial and profit motives. They are equally important for non-profit organizations, as they are also accountable. They are accountable for resources and face risks in their activities.

- **Technical process management:** In this management field, we are concerned with managing the resources and means for the implementation of technical processes that are available to us and are usually assigned by the management layer of the organization. By using them, we achieve the fulfillment of agreements that the organization or organizations have entered. Technical management processes relate to the technical implementation of projects, in particular, planning in terms of costs, time frames and objectives, checking actions to ensure compliance with plans and performance criteria, and identifying and selecting actions to eliminate deficiencies or delays. These processes are used to establish and implement technical plans for the project, manage information within a team of technical personnel, assess technical progress against plans for a software system, products or services, control technical tasks until completion, and assist in decision-making. Typically, several projects will coexist in any organization. Technical management processes can be used and implemented at the corporate level to meet requirements and needs at the level of the entire organization.
- **Technical processes:** These processes transform stakeholder needs into a product or service. By using the product or managing the service, technical processes provide the capabilities needed whenever and wherever to meet the requirements (and ensure satisfaction) of stakeholders. Technical processes are used to create and use a software system either in the form of a model or as a product that is operationally useful. Technical processes are used at any level in the hierarchy of the software system structure and at any stage in the life cycle.

In addition to the intended processes, additional processes can be defined that prove necessary and useful for an organization. The order of the subchapters in which the processes are defined in the standard does not determine the order in which the processes are implemented during the system life cycle or any of its phases.

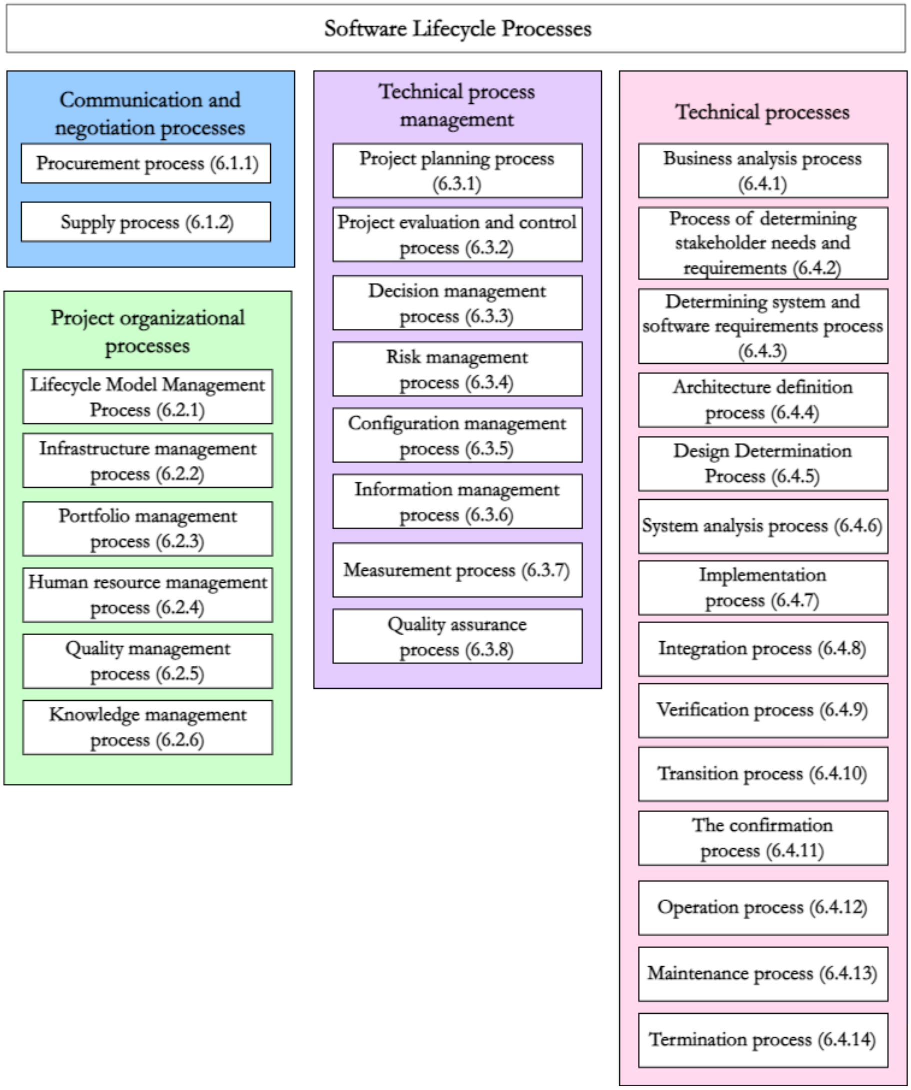

Software Lifecycle Processes
Communication and negotiation processes
Procurement process (6.1.1)
Supply process (6.1.2)
Project organizational processes
Lifecycle Model Management Process (6.2.1)
Infrastructure management process (6.2.2)
Portfolio management process (6.2.3)
Human resource management process (6.2.4)
Quality management process (6.2.5)
Knowledge management process (6.2.6)
Technical process management
Project planning process (6.3.1)
Project evaluation and control process (6.3.2)
Decision management process (6.3.3)
Risk management process (6.3.4)
Configuration management process (6.3.5)
Information management process (6.3.6)
Measurement process (6.3.7)
Quality assurance process (6.3.8)
Technical processes
Business analysis process (6.4.1)
Process of determining stakeholder needs and requirements (6.4.2)
Determining system and software requirements process (6.4.3)
Architecture definition process (6.4.4)
Design Determination Process (6.4.5)
System analysis process (6.4.6)
Implementation process (6.4.7)
Integration process (6.4.8)
Verification process (6.4.9)
Transition process (6.4.10)
The confirmation process (6.4.11)
Operation process (6.4.12)
Maintenance process (6.4.13)
Termination process (6.4.14)

Figure 2.1: Processes and process groups according to ISO/IEC 12207:2017

Source: Summarized by (ISO/IEC/IEEE 12207, 2017).

3 Model for software quality requirements and assessment ISO/IEC 25010:2011

ISO/IEC 25010 is part of the ISO/IEC 250xx family of standards, also known as SQuaRE (Software Quality Requirements and Evaluation) (*ISO/IEC 25010*, 2011). ISO/IEC 25000 is the foundational standard that provides a comprehensive framework for software quality assessment and management. SQuaRE was developed by the International Organization for Standardization (ISO) and the International Electrotechnical Commission (IEC) to provide a structured and consistent approach to software quality assessment. This approach helps organizations understand, define, and assess the quality of their software products and make decisions about improvements and optimizations. The entire family of standards establishes quality models, defines quality attributes, and defines quality requirements for software products. These models and attributes serve as tools for assessing and measuring various aspects of software quality. It is used by software development organizations and other stakeholders to assess, communicate, and improve the quality of software products. By following the guidelines and principles set forth in these standards, organizations can improve the quality of their software products, ultimately leading to increased customer satisfaction and successful software implementations. The basic SQuaRE reference model is shown in Figure 2.2.

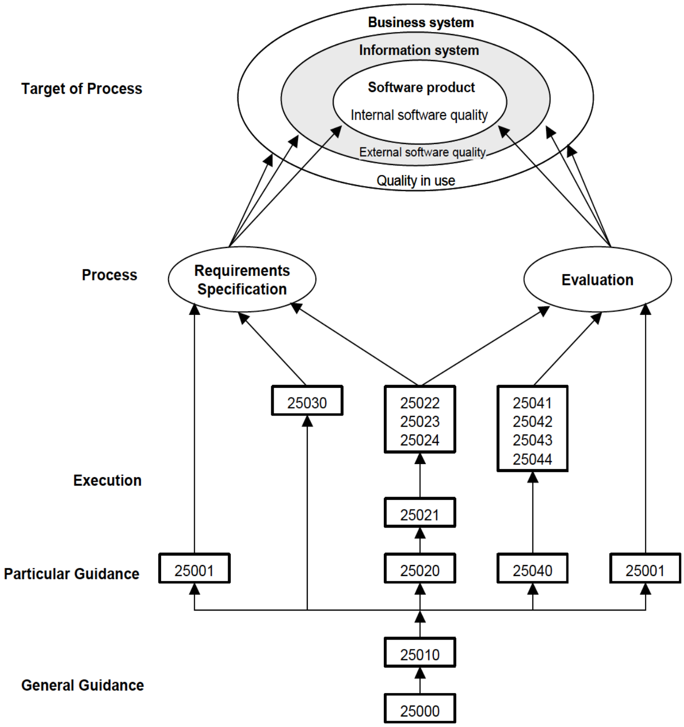


Figure 2.2: SQuaRE Software Quality Standards Family Reference Model

Source: (*ISO/IEC 25010*, 2011).

ISO/IEC 25010 provides a detailed quality model with specific characteristics and sub-characteristics for assessing the quality of a software product. ISO/IEC 25000 and ISO/IEC 25010 are related standards but they perform different tasks within the broader SQuaRE series. Thus, ISO/IEC 25000 is an informative standard that introduces broader concepts and structures for assessing software quality. At the same time, ISO/IEC 25010 is a normative standard that clearly defines the quality model and criteria for assessing the quality of a software product. The key differences between ISO/IEC 25000 and ISO/IEC 25010 are presented in Table 2.1 (*ISO/IEC 25010*, 2011).

Table 2.1: Key differences between ISO/IEC 25000 and ISO/IEC 25010

	ISO/IEC 25000	ISO/IEC 25010
Scope and purpose	the core standard in the SQuaRE series, provides a framework for the requirements and evaluation of software product quality, defines general concepts, terms and principles related to software quality management, introduces quality models and metrics used to assess software quality and guides users to other specific standards in the series, including ISO/IEC 25010.	a specific standard within the SQuaRE series that defines a comprehensive quality model, highlights quality characteristics and sub-characteristics that can be used to assess and measure the quality of a software product, delves into various aspects of software quality, providing specific criteria for evaluating software.
Content	is an informative standard that introduces the software product quality framework, the overall structure of quality characteristics and sub-characteristics, acts as a guiding document for understanding software quality assessment within the SQuaRE series.	a normative standard with specific requirements, intended for direct use in assessing software quality, defines eight basic quality characteristics and their sub-characteristics (accuracy, consistency, efficiency).
Use	is primarily used to provide an overview of software quality management and to direct users to other relevant standards in the SQuaRE series, sets the context and terminology for quality models and metrics for assessing software quality.	is used directly for assessing the quality of a software product, serves as a reference for practitioners who wish to assess and measure the quality characteristics of a software product, guides the selection of relevant quality characteristics and sub-characteristics for assessment according to the specific needs and objectives of the assessment

Source: (ISO/IEC 25010, 2011)

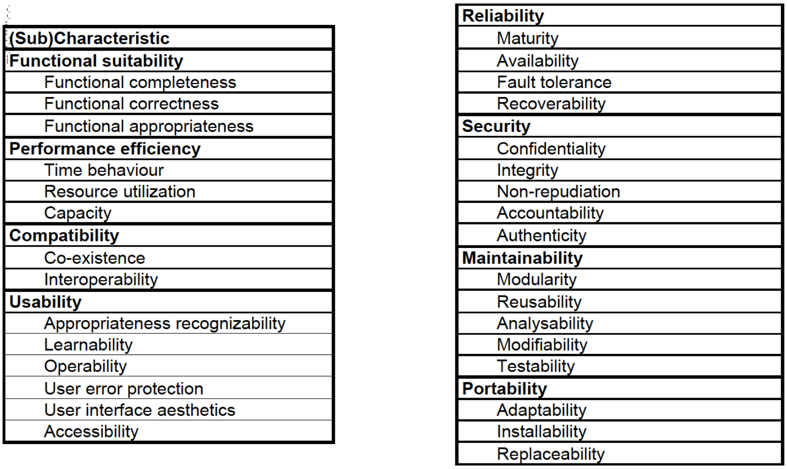
The model described in ISO 25010:2011 assumes (*ISO/IEC 25010*, 2011):

- **a quality model** that encompasses eight main quality characteristics, each representing a key aspect of software quality:

  - Functional suitability: all the capabilities in which the software provides the necessary functions to meet specified needs.
  - Operating efficiency: the ability of the software to perform within expected time frames and with expected use of other IT resources, response times, and data transfer.
  - Compatibility: ability of software to work with other systems, software, or hardware.
  - Usability: ease of use of the software and the user experience.
  - Reliability: ability of software to maintain its level of performance under specified conditions over a specified period of time.
  - Security: ability of software to protect data and functionality from unauthorized access and damage.
  - Durability: effort required to implement changes, correct errors, or adapt the software to changes.
  - Portability: ease with which software can be transferred from one environment to another.
- **Sub-characteristics:** each of the characteristics listed above is broken down into specific sub-characteristics, which makes it easier to assess and focus on specific areas of quality assessment.
- **Quality requirements:** includes a set of quality requirements that can specify the desired level of each quality characteristic and sub-characteristic for a specific software product.
- **Quality in use:** emphasizes the importance of quality assessment on the part of end users who ascertain the quality of the software during actual use.

Figures 2.3 and 2.4 show the set of characteristics and sub-characteristics of software quality as defined by the ISO/IEC 25010:2011 quality assessment model.

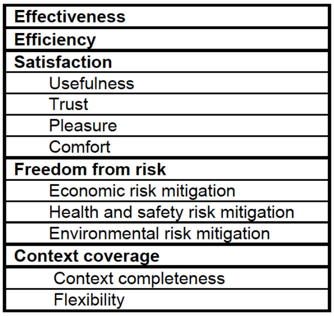
The quality of a software product can be assessed by measuring internal properties (usually static measurements of intermediate products), by measuring external properties (usually measuring the behavior of the code during execution), or by measuring quality properties in use (when the product is in actual or simulated use). See Figure 2.1 and Figure 2.5.



(Sub)Characteristic
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

Figure 2.3: Software quality characteristics and sub-characteristics according to ISO/IEC 25010:2011

Source: (ISO/IEC 25010, 2011).



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

Figure 2.4: Characteristics and sub-characteristics of quality in the use of software according to ISO/IEC

Source: (ISO/IEC 25010, 2011).

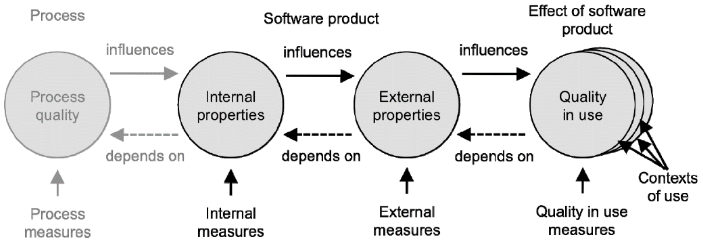


Figure 2.5: Software Quality Lifecycle

Source: (ISO/IEC 25010, 2011).

4 Comment on the applicability of both standards

The ISO/IEC 12207 and ISO/IEC 25010 standards play a key role in the digitalization of logistics processes, as they provide a structure and guidelines for the effective management of software development and its quality assurance. The following is a brief critical evaluation of their importance in the context of the digitalization of logistics processes.

ISO/IEC 12207 defines standard processes for software lifecycle management. This standard is important for the digitalization of logistics processes for the following reasons:

- **Structured approach to software development:** ISO/IEC 12207 provides a structured framework for planning, developing, maintaining and managing software. In the context of digital logistics, this means that companies can effectively plan and implement digital solutions that are reliable and compliant with best practices.
- **Managing complexity:** Logistics processes are complex and involve many actors and technologies. ISO/IEC 12207 helps manage this complexity by clearly defining processes, responsibilities and activities throughout the software lifecycle.
- **Reducing risks:** By providing clear guidelines for software development and maintenance, the standard helps reduce the risks associated with software errors, which is particularly important in logistics, where errors can cause significant financial losses and supply chain disruptions.
- **Increasing compliance and standardization:** Using ISO/IEC 12207 ensures that software development processes are consistent and standardized. This is crucial for the interoperability of different systems and technologies used in digitalized logistics processes.

ISO/IEC 25010 defines a software quality model that includes various quality characteristics such as: functionality, efficiency, reliability, usability and other indicators. Its importance in the digitalization of logistics processes is mainly as follows:

- **Quality measurement:** ISO/IEC 25010 enables the precise measurement and evaluation of software quality. In logistics, where accuracy and reliability are crucial, companies can use this standard to ensure that digital solutions meet high quality standards.
- **User experience improvement:** Software quality has a significant impact on the user experience. ISO/IEC 25010 helps design software that is easy to use, efficient and tailored to the needs of users in logistics processes.
- **Reliability and maintainability:** Reliability and maintainability are key quality characteristics in logistics, where system failures or outages can cause serious disruptions. ISO/IEC 25010 helps ensure that software solutions are reliable and easy to maintain.
- **Decision support:** The standard provides clear criteria for assessing software quality, which can help management decide whether to implement new digital solutions or improve existing systems.

One criticism of both standards is that they impose additional costs and resource requirements. Small and medium-sized logistics companies may not have the resources to fully implement these standards. In addition, they may be too complex and difficult to understand and implement for some environments, which can be a barrier for companies that do not have specialized knowledge and experience in systems engineering and software quality.

The ISO/IEC 12207 and ISO/IEC 25010 standards are important aids in the digitalization of logistics processes, as they provide a structured approach to software development and quality assurance. Their use can improve the efficiency, reliability and quality of digital solutions in logistics. However, companies must be mindful of the costs, resources and flexibility associated with implementing these standards and be aware of the potential challenges and limitations they may bring.

References

ISO/IEC 25010:2011–Systems and software engineering–Systems and software Quality Requirements and Evaluation (SQuaRE) –System and software quality models (Version 1). (2011). https://www.iso.org/standard/35733.html

ISO/IEC/IEEE 12207:2017–Systems and software engineering–Software life cycle processes (Version 1). (2017). https://www.iso.org/standard/63712.html