

6.4.11.3.2 Software disposal execution. This activity consists of the following tasks:

6.4.11.3.2.1 The software disposal plan shall be executed.

6.4.11.3.2.2 Users shall be given notification of the plans and activities for the retirement of software products and services. Notifications shall include the following:

- a) Description of any replacement or upgrade with its date of availability.
- b) Statement of why the software product is no longer to be supported.
- c) Description of other support options available, once support has been removed.

6.4.11.3.2.3 Parallel operations of the retiring and any new software product should be conducted for smooth transition to the new system. During this period, user training shall be provided as specified in the contract.

6.4.11.3.2.4 When the scheduled retirement arrives, notification shall be sent to all concerned. All associated development documentation, logs, and code should be placed in archives, when appropriate.

6.4.11.3.2.5 Data used by, or associated with, the retired software product shall be accessible in accordance with the contract requirements for data protection and audit applicable to the data.

7 Software Specific Processes

7.1 Software Implementation Processes

7.1.1 Software Implementation Process

NOTE The Software Implementation Process is a conforming instance of the Implementation Process of ISO/IEC 15288, specialized to the particular needs of implementing a software product or service.

7.1.1.1 Purpose

The purpose of the Software Implementation Process is to produce a specified system element implemented as a software product or service.

This process transforms specified behaviour, interfaces and implementation constraints into actions that create a system element implemented as a software product or service, otherwise known as a "software item." This process results in a software item that satisfies architectural design requirements through verification and stakeholder requirements through validation.

7.1.1.2 Outcomes

As a result of the successful implementation of the Software Implementation Process:

- a) an implementation strategy is defined;
- b) implementation technology constraints on the design are identified;
- c) a software item is realized; and
- d) a software item is packaged and stored in accordance with an agreement for its supply.

In addition to its activities, the Software Implementation Process has the following lower-level processes:

— Software Requirements Analysis Process*

- Software Architectural Design Process*
- Software Detailed Design Process
- Software Construction Process
- Software Integration Process*
- Software Qualification Testing Process*

NOTE Users of ISO/IEC 15288 may decide that the processes marked by an asterisk (*) in the list above are to be provided by recursive application of ISO/IEC 15288 even for software elements of the system.

7.1.1.3 Activities and tasks

The project shall implement the following activities in accordance with applicable organization policies and procedures with respect to the Software Implementation Process.

7.1.1.3.1 Software implementation strategy. This activity consists of the following tasks:

7.1.1.3.1.1 If not stipulated in the contract, the developer shall define or select a life cycle model appropriate to the scope, magnitude, and complexity of the project. The life cycle model shall be comprised of stages and the purpose and outcomes of each stage. The activities and tasks of the Software Implementation Process shall be selected and mapped onto the life cycle model.

NOTE These activities and tasks may overlap or interact and may be performed iteratively or recursively.

NOTE Ideally, this is performed by using an organizationally-defined life cycle model.

7.1.1.3.1.2 The implementer shall:

- a) Document the outputs in accordance with the Software Documentation Management Process (subclause 7.2.1).
- b) Place the outputs under the Software Configuration Management Process (subclause 7.2.2) and perform change control in accordance with it.
- c) Document and resolve problems and non-conformances found in the software products and tasks in accordance with the Software Problem Resolution Process (subclause 7.2.8).
- d) Perform supporting processes as specified in the contract.
- e) Establish baselines and incorporate configuration items at appropriate times, as determined by the acquirer and the supplier.

7.1.1.3.1.3 The implementer shall select, tailor, and use those standards, methods, tools, and computer programming languages (if not stipulated in the contract) that are documented, appropriate, and established by the organization for performing the activities of the Software Implementation Process and supporting processes.

NOTE Implementation technology constraints on the design should be identified as part of the software implementation strategy.

7.1.1.3.1.4 The implementer shall develop plans for conducting the activities of the Software Implementation process. The plans should include specific standards, methods, tools, actions, and responsibility associated with the development and qualification of all requirements including safety and security. If necessary, separate plans may be developed. These plans shall be documented and executed.

7.1.1.3.1.5 Non-deliverable items may be employed in the development or maintenance of the software product. However, it shall be ensured that the operation and maintenance of the deliverable software product after its delivery to the acquirer are independent of such items; otherwise, those items should be considered as deliverable.

7.1.2 Software Requirements Analysis Process

NOTE The Software Requirements Analysis Process in this International Standard is a lower-level process of the Software Implementation Process. Users of ISO/IEC 15288 may decide that this process is to be provided by the Requirements Analysis Process of ISO/IEC 15288 in a recursive application of that standard.

7.1.2.1 Purpose

The purpose of Software Requirements Analysis Process is to establish the requirements of the software elements of the system.

7.1.2.2 Outcomes

As a result of successful implementation of the Software Requirements Analysis Process:

- a) the requirements allocated to the software elements of the system and their interfaces are defined;
- b) software requirements are analyzed for correctness and testability;
- c) the impact of software requirements on the operating environment are understood;
- d) consistency and traceability are established between the software requirements and system requirements;
- e) prioritization for implementing the software requirements is defined;
- f) the software requirements are approved and updated as needed;
- g) changes to the software requirements are evaluated for cost, schedule and technical impact; and
- h) the software requirements are baselined and communicated to all affected parties.

7.1.2.3 Activities and tasks

The project shall implement the following activities and tasks in accordance with applicable organization policies and procedures with respect to the Software Requirements Analysis Process.

7.1.2.3.1 Software requirements analysis. For each software item (or configuration item, if identified) this activity consists of the following tasks:

7.1.2.3.1.1 The implementer shall establish and document software requirements (including the quality characteristics specifications) described below.

- a) Functional and capability specifications, including performance, physical characteristics, and environmental conditions under which the software item is to perform.
- b) Interfaces external to the software item.
- c) Qualification requirements.
- d) Safety specifications, including those related to methods of operation and maintenance, environmental influences, and personnel injury.
- e) Security specifications, including those related to compromise of sensitive information.

- f) Human-factors engineering (ergonomics) specifications, including those related to manual operations, human-equipment interactions, constraints on personnel, and areas needing concentrated human attention, that are sensitive to human errors and training.
- g) Data definition and database requirements.
- h) Installation and acceptance requirements of the delivered software product at the operation and maintenance site(s).
- i) User documentation requirements.
- j) User operation and execution requirements.
- k) User maintenance requirements.

NOTE 1 Guidance for specifying quality characteristics may be found in ISO/IEC 9126-1.

NOTE 2 Implementation priority of the software requirements should be determined.

NOTE 3 If usability is an important requirement, recommendations for obtaining a desired level of usability can be found in ISO TR 18529, *Ergonomics—Ergonomics of human-system interaction—Human-centred lifecycle process descriptions*. Annex E contains a process view which is focused on usability.

7.1.2.3.1.2 The implementer shall evaluate the software requirements considering the criteria listed below. The results of the evaluations shall be documented.

- a) Traceability to system requirements and system design.
- b) External consistency with system requirements.
- c) Internal consistency.
- d) Testability.
- e) Feasibility of software design.
- f) Feasibility of operation and maintenance.

7.1.2.3.1.3 The implementer shall conduct review(s) in accordance with subclause 7.2.6.

NOTE Following a successful evaluation and review, the software requirements should be approved, baselined and communicated to all affected parties. Subsequent changes to the software requirements baseline should be evaluated for cost, schedule and technical impact.

7.1.3 Software Architectural Design Process

NOTE The Software Architectural Design Process in this International Standard is a lower-level process of the Software Implementation Process. Users of ISO/IEC 15288 may decide that this process is to be provided by the Architectural Design Process of ISO/IEC 15288 in a recursive application of that standard.

7.1.3.1 Purpose

The purpose of the Software Architectural Design Process is to provide a design for the software that implements and can be verified against the requirements.

7.1.3.2 Outcomes

As a result of successful implementation of the Software Architectural Design Process:

- a) a software architectural design is developed and baselined that describes the software items that will implement the software requirements;
- b) internal and external interfaces of each software item are defined; and
- c) consistency and traceability are established between software requirements and software design.

7.1.3.3 Activities and tasks

The project shall implement the following activities in accordance with applicable organization policies and procedures with respect to the Software Architectural Design Process.

NOTE This activity is implemented for each software item, consistent with a system architectural design.

7.1.3.3.1 Software architectural design. For each software item (or configuration item, if identified) this activity consists of the following tasks:

7.1.3.3.1.1 The implementer shall transform the requirements for the software item into an architecture that describes its top-level structure and identifies the software components. It shall be ensured that all the requirements for the software item are allocated to its software components and further refined to facilitate detailed design. The architecture of the software item shall be documented.

NOTE The software architectural design also provides a basis for verifying the software items, integration of software items with each other, and integration of software items with the rest of the system items.

7.1.3.3.1.2 The implementer shall develop and document a top-level design for the interfaces external to the software item and between the software components of the software item.

7.1.3.3.1.3 The implementer shall develop and document a top-level design for the database.

7.1.3.3.1.4 The implementer should develop and document preliminary versions of user documentation.

7.1.3.3.1.5 The implementer shall define and document preliminary test requirements and the schedule for Software Integration.

7.1.3.3.1.6 The implementer shall evaluate the architecture of the software item and the interface and database designs considering the criteria listed below. The results of the evaluations shall be documented.

- a) Traceability to the requirements of the software item.
- b) External consistency with the requirements of the software item.
- c) Internal consistency between the software components.
- d) Appropriateness of design methods and standards used.
- e) Feasibility of detailed design.
- f) Feasibility of operation and maintenance.

7.1.3.3.1.7 The implementer shall conduct review(s) in accordance with subclause 7.2.6

7.1.4 Software Detailed Design Process

NOTE The Software Detail Design Process in this International Standard is a lower-level process of the Software Implementation Process.

7.1.4.1 Purpose

The purpose of the Software Detailed Design Process is to provide a design for the software that implements and can be verified against the requirements and the software architecture and is sufficiently detailed to permit coding and testing.

7.1.4.2 Outcomes

As a result of successful implementation of the Software Detailed Design Process:

- a) a detailed design of each software component, describing the software units to be built, is developed;
- b) external interfaces of each software unit are defined; and
- c) consistency and traceability are established between the detailed design and the requirements and architectural design.

7.1.4.3 Activities and tasks

The project shall implement the following activities in accordance with applicable organization policies and procedures with respect to the Software Detailed Design Process.

7.1.4.3.1 Software detailed design. For each software item (or configuration item, if identified) this activity consists of the following tasks:

7.1.4.3.1.1 The implementer shall develop a detailed design for each software component of the software item. The software components shall be refined into lower levels containing software units that can be coded, compiled, and tested. It shall be ensured that all the software requirements are allocated from the software components to software units. The detailed design shall be documented.

7.1.4.3.1.2 The implementer shall develop and document a detailed design for the interfaces external to the software item, between the software components, and between the software units. The detailed design of the interfaces shall permit coding without the need for further information.

7.1.4.3.1.3 The implementer shall develop and document a detailed design for the database.

7.1.4.3.1.4 The implementer shall update user documentation as necessary.

7.1.4.3.1.5 The implementer shall define and document test requirements and the schedule for testing software units. The test requirements should include stressing the software unit at the limits of its requirements.

7.1.4.3.1.6 The implementer shall update the test requirements and the schedule for Software Integration.

7.1.4.3.1.7 The implementer shall evaluate the software detailed design and test requirements considering the criteria listed below. The results of the evaluations shall be documented.

- a) Traceability to the requirements of the software item;
- b) External consistency with architectural design;
- c) Internal consistency between software components and software units;
- d) Appropriateness of design methods and standards used;
- e) Feasibility of testing;
- f) Feasibility of operation and maintenance.

7.1.4.3.1.8 The implementer shall conduct review(s) in accordance with subclause 7.2.6.

7.1.5 Software Construction Process

NOTE The Software Construction Process in this International Standard is a lower-level process of the Software Implementation Process.

7.1.5.1 Purpose

The purpose of the Software Construction Process is to produce executable software units that properly reflect the software design.

7.1.5.2 Outcomes

As a result of successful implementation of Software Construction Process:

- a) verification criteria are defined for all software units against their requirements;
- b) software units defined by the design are produced;
- c) consistency and traceability are established between software units and requirements and design; and
- d) verification of the software units against the requirements and the design is accomplished.

7.1.5.3 Activities and tasks

The project shall implement the following activities and tasks in accordance with applicable organization policies and procedures with respect to the Software Construction Process.

7.1.5.3.1 Software construction. For each software item (or configuration item, if identified) this activity consists of the following tasks:

7.1.5.3.1.1 The implementer shall develop and document the following:

- a) Each software unit and database.
- b) Test procedures and data for testing each software unit and database.

7.1.5.3.1.2 The implementer shall test each software unit and database ensuring that it satisfies its requirements. The test results shall be documented.

7.1.5.3.1.3 The implementer shall update the user documentation as necessary.

7.1.5.3.1.4 The implementer shall update the test requirements and the schedule for Software Integration.

7.1.5.3.1.5 The implementer shall evaluate software code and test results considering the criteria listed below. The results of the evaluations shall be documented.

- a) Traceability to the requirements and design of the software item.
- b) External consistency with the requirements and design of the software item.
- c) Internal consistency between unit requirements.
- d) Test coverage of units.
- e) Appropriateness of coding methods and standards used.

- f) Feasibility of software integration and testing.
- g) Feasibility of operation and maintenance.

7.1.6 Software Integration Process

NOTE The Software Integration Process in this International Standard is a lower-level process of the Software Implementation Process. Users of ISO/IEC 15288 may decide that this process is to be provided by the Integration Process of ISO/IEC 15288 in a recursive application of that standard.

7.1.6.1 Purpose

The purpose of the Software Integration Process is to combine the software units and software components, producing integrated software items, consistent with the software design, that demonstrate that the functional and non-functional software requirements are satisfied on an equivalent or complete operational platform.

7.1.6.2 Outcomes

As a result of successful implementation of the Software Integration Process:

- a) an integration strategy is developed for software units consistent with the software design and the prioritized software requirements;
- b) verification criteria for software items are developed that ensure compliance with the software requirements allocated to the items;
- c) software items are verified using the defined criteria;
- d) software items defined by the integration strategy are produced;
- e) results of integration testing are recorded;
- f) consistency and traceability are established between software design and software items; and
- g) a regression strategy is developed and applied for re-verifying software items when a change in software units (including associated requirements, design and code) occur.

7.1.6.3 Activities and tasks

The project shall implement the following activities and tasks in accordance with applicable organization policies and procedures with respect to the Software Integration Process.

7.1.6.3.1 Software integration. For each software item (or configuration item, if identified) this activity consists of the following tasks:

7.1.6.3.1.1 The implementer shall develop an integration plan to integrate the software units and software components into the software item. The plan shall include test requirements, procedures, data, responsibilities, and schedule. The plan shall be documented.

7.1.6.3.1.2 The implementer shall integrate the software units and software components and test as the aggregates are developed in accordance with the integration plan. It shall be ensured that each aggregate satisfies the requirements of the software item and that the software item is integrated at the conclusion of the integration activity. The integration and test results shall be documented.

NOTE A regression strategy should be developed to be applied for re-verifying the software items when a change is made to software units (including associated requirements, design and code).

7.1.6.3.1.3 The implementer shall update the user documentation as necessary.

7.1.6.3.1.4 The implementer shall develop and document for each qualification requirement of the software item a set of tests, test cases (inputs, outputs, test criteria), and test procedures for conducting Software Qualification Testing. The developer shall ensure that the integrated software item is ready for Software Qualification Testing.

7.1.6.3.1.5 The implementer shall evaluate the integration plan, design, code, tests, test results, and user documentation considering the criteria listed below. The results of the evaluations shall be documented.

- a) Traceability to the system requirements.
- b) External consistency with the system requirements.
- c) Internal consistency.
- d) Test coverage of the requirements of the software item.
- e) Appropriateness of test standards and methods used.
- f) Conformance to expected results.
- g) Feasibility of software qualification testing.
- h) Feasibility of operation and maintenance.

NOTE Evaluation criteria should include consistency and traceability between the software design and the software items.

7.1.6.3.1.6 The implementer shall conduct review(s) in accordance with subclause 7.2.6.

7.1.7 Software Qualification Testing Process

NOTE The Software Qualification Testing Process in this International Standard is a lower-level process of the Software Implementation Process. Users of ISO/IEC 15288 may decide that this process is to be provided by the Verification Process of ISO/IEC 15288 in a recursive application of that standard.

7.1.7.1 Purpose

The purpose of the Software Qualification Testing Process is to confirm that the integrated software product meets its defined requirements.

7.1.7.2 Outcomes

As a result of successful implementation of the Software Qualification Testing Process:

- a) criteria for the integrated software is developed that demonstrates compliance with the software requirements;
- b) integrated software is verified using the defined criteria;
- c) test results are recorded; and
- d) a regression strategy is developed and applied for re-testing the integrated software when a change in software items is made.

NOTE A regression strategy should be developed, to be applied for re-testing the integrated software when a change is made to software items.

7.1.7.3 Activities and tasks

The project shall implement the following activities in accordance with applicable organization policies and procedures with respect to the Software Qualification Testing Process.

7.1.7.3.1 Software qualification testing. For each software item (or configuration item, if identified) this activity consists of the following tasks:

7.1.7.3.1.1 The implementer shall conduct qualification testing in accordance with the qualification requirements for the software item. It shall be ensured that the implementation of each software requirement is tested for compliance. The qualification testing results shall be documented.

7.1.7.3.1.2 The implementer shall update the user documentation as necessary.

7.1.7.3.1.3 The implementer shall evaluate the design, code, tests, test results, and user documentation considering the criteria listed below. The results of the evaluations shall be documented.

- a) Test coverage of the requirements of the software item.
- b) Conformance to expected results.
- c) Feasibility of system integration and testing, if conducted.
- d) Feasibility of operation and maintenance.

7.1.7.3.1.4 The implementer shall support audit(s) in accordance with subclause 7.2.7. The results of the audits shall be documented. If both hardware and software are under development or integration, the audits may be postponed until the System Qualification Testing.

7.1.7.3.1.5 Upon successful completion of the audits, if conducted, the implementer shall update and prepare the deliverable software product for System Integration, System Qualification Testing, Software Installation, or Software Acceptance Support as applicable.

NOTE The Software Qualification Testing Process may be used in the Software Verification Process (subclause 7.2.4) or the Software Validation Process (subclause 7.2.5).

7.2 Software Support Processes

NOTE The support processes listed under this subclause are specific to software and are labelled Software Support Processes. Although they play an integral role in assisting the Software Implementation Process, the Software Support Processes may also provide services to other processes, e.g., the Agreement Processes, Systems Qualification Testing, Software Acceptance Support, Software Operation, and the Software Maintenance Process.

7.2.1 Software Documentation Management Process

NOTE The Software Documentation Management Process is a specialization of the Information Management Process from the Project Process Group in this International Standard.

7.2.1.1 Purpose

The purpose of the Software Documentation Management Process is to develop and maintain the recorded software information produced by a process.

NOTE ISO/IEC 15289 provides more detailed content for life cycle process information items (documentation).

7.2.1.2 Outcomes

As a result of successful implementation of the Software Documentation Management Process:

- a) a strategy identifying the documentation to be produced during the life cycle of the software product or service is developed;
- b) the standards to be applied for the development of the software documentation are identified;
- c) documentation to be produced by the process or project is identified;
- d) the content and purpose of all documentation is specified, reviewed and approved;
- e) documentation is developed and made available in accordance with identified standards; and
- f) documentation is maintained in accordance with defined criteria.

7.2.1.3 Activities and tasks

The project shall implement the following activities in accordance with applicable organization policies and procedures with respect to the Software Documentation Management Process.

7.2.1.3.1 Process implementation. This activity consists of the following task:

7.2.1.3.1.1 A plan, identifying the documents to be produced during the life cycle of the software product shall be developed, documented, and implemented. For identified documentation, the following shall be addressed:

- a) Title or name.
- b) Purpose and content.
- c) Intended audience.
- d) Procedures and responsibilities for inputs, development, review, modification, approval, production, storage, distribution, maintenance, and configuration management.
- e) Schedule for intermediate and final versions.

7.2.1.3.2 Design and development. This activity consists of the following tasks:

7.2.1.3.2.1 Each identified document shall be designed in accordance with applicable documentation standards for medium, format, content description, page numbering, figure/table placement, proprietary/security marking, packaging, and other presentation items.

NOTE The documentation may originate and may terminate in any form (e.g., verbal, textual, graphical, numerical) and may be stored, processed, replicated and transmitted using any medium (e.g., electronic, printed, magnetic, optical).

7.2.1.3.2.2 The source and appropriateness of input data for the documents shall be confirmed. Automated documentation support tools may be used.

7.2.1.3.2.3 The prepared documents shall be reviewed and edited for format, technical content, and presentation style against their documentation standards. They shall be approved for adequacy by authorized personnel prior to issue.

7.2.1.3.3 Production. This activity consists of the following tasks:

7.2.1.3.3.1 The documents shall be produced and provided in accordance with the plan. Production and distribution of documents may use paper, electronic, or other media. Master materials shall be stored in accordance with requirements for record retention, security, maintenance, and backup.