INTERNATIONAL STANDARD

ISO/IEC/ IEEE 24748-3

First edition
2020-10

Systems and software engineering — Life cycle management —

Part 3:
Guidelines for the application of ISO/IEC/IEEE 12207 (software life cycle processes)

Ingénierie des systèmes et du logiciel — Gestion du cycle de vie —
Partie 3: Lignes directrices pour l'application de l'ISO/IEC/IEEE 12207 (processus du cycle de vie du logiciel)

© ISO/IEC 2020
© IEEE 2020

iTeh STANDARD PREVIEW
(standards.iteh.ai)

COPYRIGHT PROTECTED DOCUMENT

Contents

iTeh STANDARD PREVIEW
(standards.iteh.ai)

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work. In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular the different approval criteria needed for the different types of document should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see www.iso.org/directives).

IEEE Standards documents are developed within the IEEE Societies and the Standards Coordinating Committees of the IEEE Standards Association (IEEE-SA) Standards Board. The IEEE develops its standards through a consensus development process, approved by the American National Standards Institute, which brings together volunteers representing varied viewpoints and interests to achieve the final product. Volunteers are not necessarily members of the Institute and serve without compensation. While the IEEE administers the process and establishes rules to promote fairness in the consensus development process, the IEEE does not independently evaluate, test, or verify the accuracy of any of the information contained in its standards.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see www.iso.org/patents).

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation on the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT) see www.iso.org/iso/foreword.html.

This document was prepared by ISO/IEC JTC 1, *Information technology*, SC 7, *Systems and software engineering*, in cooperation with the Systems and Software Engineering Standards Committee of the IEEE Computer Society, under the Partner Standards Development Organization cooperation agreement between ISO and IEEE.

This document cancels and replaces ISO/IEC TR 24748-3:2011, which has been technically revised.

The main changes compared to ISO/IEC TR 24748-3:2011 are as follows:

— revised presentation of concepts, consistent with ISO/IEC/IEEE 12207:2017;

— completely updated presentation of guidance for each life cycle process, including aspects of purpose; outcomes and outputs; activities, tasks, and approaches;

— identified closely related processes;

— identified related international standards for each process, which offer more detailed requirements and guidance.

A list of all parts in the ISO/IEC/IEEE 24748 series can be found on the ISO website.

Any feedback or questions on this document should be directed to the user's national standards body. A complete listing of these bodies can be found at www.iso.org/members.html.

Introduction

The purpose of this document is to provide guidance on the application of the software life cycle processes standard, ISO/IEC/IEEE 12207:2017. Taken together, the parts of the ISO/IEC/IEEE 24748 series are intended to facilitate the joint usage of the process content of the two high-level life cycle process standards (ISO/IEC/IEEE 12207:2017 and ISO/IEC/IEEE 15288:2015), which in turn may be used together with various more specialized lower-level process standards. In this way, ISO/IEC/IEEE 24748 (all parts) provides unified and consolidated guidance on the life cycle management of systems and software engineering. Its purpose is to help ensure consistency in system concepts and life cycle concepts, models, stages, processes, process application, key points of view, adaptation and use in various domains as the two standards (and others) are used in combination. It should help an organization to design, develop, and sustain software systems using a life cycle model.

ISO/IEC/IEEE 24748-1 provides guidance for the concepts of life cycle management applicable to both systems and software engineering. It covers fundamental concepts such as system-of-interest, stages, processes, projects, and organizations. This document focuses on and expands the coverage of those aspects and processes most relevant to software systems. A companion guidance document, ISO/IEC/IEEE 24748-2, provides similar guidance for the application of ISO/IEC/IEEE 15288:2015.

In conjunction with ISO/IEC/IEEE 24748-1, this document aids in identifying and planning the use of the life cycle processes described in ISO/IEC/IEEE 12207:2017. Since in many respects the Organizational Project Enabling processes and the Technical Management processes are quite similar for software systems to those used for any type of system, this document concentrates on specific guidance for the Technical processes and how they can be effectively used during the software life cycle. ISO/IEC/IEEE 24748-5 focuses on the Technical Management processes, especially Project Planning and Project Assessment and Control, as applied to software projects. The proper use of these processes can contribute to a project being completed successfully, meeting its objectives and requirements for each stage and for the overall project.

This document elaborates on factors, 'best-practice' or typical approaches and methods that should be considered when applying ISO/IEC/IEEE 12207:2017. It does this in the context of the various ways in which ISO/IEC/IEEE 12207:2017 can be applied. It is intended to be useful in a variety of software life cycle situations, including the use of agile methods, which are the most widely used on all types and sizes of projects.

Systems and software engineering — Life cycle management —

Part 3:
Guidelines for the application of ISO/IEC/IEEE 12207 (software life cycle processes)

1 Scope

This document is a guideline for the application of ISO/IEC/IEEE 12207:2017. This document establishes guidance to implement a common framework for software life cycle processes, with well-defined terminology, that can be referenced by the software industry. This document provides guidance on defining, controlling, and improving software life cycle processes within an organization or a project. This document recommends methods and approaches suitable for a variety of life cycle models. The guidance emphasizes the importance of establishing a strategy, planning, and the involvement of stakeholders, with the ultimate goal of achieving customer satisfaction.

This document applies to the acquisition, supply, design and development, transition, operation, maintenance, and disposal (whether performed internally or externally to an organization) of software systems, products, and services (including software as a service (SaaS)), and the software portion of any system. Software includes the software portion of firmware. The guidance on processes, activities, and tasks in this document can also be applied during the acquisition of a system that contains software. The guidance in this document can also be used as a basis for selecting, establishing, and improving organizational environments, e.g., methods, procedures, techniques, tools, and trained personnel.

In the context of this document, there is a continuum of human-made systems from those that use little or no software to those in which software is the primary interest. It is rare to encounter a complex system without software, and all software systems require physical system components (hardware) to operate, either as part of the software system-of-interest (SoI) or as an enabling system or infrastructure. Thus, the choice of whether to apply this document for guidance to the software life cycle processes, or ISO/IEC/IEEE 24748-2, depends on the SoI. At a high level, both documents have the same life cycle process framework, but differ in guidance for activities and tasks to perform software engineering or systems engineering, respectively.

The processes and process groups in this document are identical in their purpose and outcomes with those in ISO/IEC/IEEE 12207:2017 and in ISO/IEC/IEEE 15288:2015, with one exception: the System/ Software Requirements Definition process of ISO/IEC/IEEE 12207:2017 and this document has a different name from the System Requirements Definition process of ISO/IEC/IEEE 15288:2015.

Use of the guidance in this document is appropriate regardless of software system size or complexity or organizational size. The process outcomes from the ISO/IEC/IEEE 12207:2017 life cycle processes are meant to be generic and applicable to the engineering of any software system in any size organization.

This document does not prescribe nor detail a specific software life cycle model, development methodology, method, modelling approach, or technique and method. The variety of ways for implementing software (e.g., development of new code, integration of existing open source components and commercial products, or modifications to existing software, including transition to new platforms) make it impossible to detail specific procedures.

This document does not establish a management system or provide guidance on the use of any management system standard. However, it is intended to be compatible with the quality management system specified by ISO 9001, the service management system specified by ISO/IEC 20000-1, the

IT asset management system specified by ISO/IEC 19770 (all parts), and the information security management system specified by ISO/IEC 27000.

Clause 6 provides guidance on aspects of purposes, outcomes, activities, and tasks in ISO/IEC/IEEE 12207:2017. However, this document does not repeat the detailed requirements and recommendations for purposes, outcomes, activities, and tasks for each life cycle process found in ISO/IEC/IEEE 12207:2017. Clause 6 also provides references to specialized standards that provide more detailed requirements and guidance for the various processes and information products (information items). This document does not detail information items (process inputs and outputs) in terms of name, format, explicit content and recording media.

NOTE   ISO/IEC/IEEE 15289 addresses the content for life cycle process information items (documentation).

2   Normative references

There are no normative references in this document.

3   Terms, definitions, and abbreviated terms

3.1   Terms and definitions

No terms and definitions are listed in this document.

ISO, IEC, and IEEE maintain terminological databases for use in standardization at the following addresses:

— ISO Online browsing platform: available at https://www.iso.org/

— IEC Electropedia: available at http://www.electropedia.org/

— IEEE Standards Dictionary Online: available at: http://dictionary.ieee.org

NOTE   For additional terms and definitions in the field of systems and software engineering, see ISO/IEC/IEEE 24765, which is published periodically as a "snapshot" of the SEVOCAB (Systems and software Engineering Vocabulary) database and is publicly accessible at www.computer.org/sevocab.

3.2   Abbreviated terms

API   application program interface

CM   configuration management

COTS   commercial-off-the-shelf

FCA   functional configuration audit

IDEF   Integration DEFinition

MOE   measure of effectiveness

MOP   measure of performance

NDI   non-developmental item

PCA   physical configuration audit

PII   personally identifiable information

PRM   process reference model

QA   quality assurance

QM   quality management

SaaS   software as a service

SME   subject matter expert

SoI   system-of-interest

SoS   system of systems

TPM   technical performance measure

V&V   validation and verification

VSE   very small entity

WBS   work breakdown structure

4 Concepts for software and software systems

4.1 General

This clause is included to help explain essential concepts as applicable to software and software systems. While understanding concepts does not give the ability to immediately apply them without further thought and work, it is the foundation for their practical use in different project, organizational and life cycle environments.

4.2 Software system concepts

The application of ISO/IEC/IEEE 12207:2017 presupposes an understanding of system concepts. A system is a combination of interacting elements organized to achieve one or more stated purposes. Software is the subsystems or elements of a system consisting of computer programs, related procedures, associated documentation, and data pertaining to the operation of the subsystem or element. Software occurs in most systems, even if it is not the predominant element of interest as it is in a software system.

For the purposes of this document, software systems are considered as created by humans and utilized to provide services in defined environments for the benefit of users and other stakeholders. These systems may be configured with one or more of the following: hardware, software, services, humans, processes (e.g. review process), procedures (e.g. operator instructions), facilities, and naturally occurring entities (e.g. water, organisms, minerals). A system may be considered as a product or as the services it provides. A system element is a member of a set of elements that constitutes a system. A system element is a discrete part of a system that can be implemented to fulfil specified requirements.

NOTE 1   Additional discussion regarding systems, systems of systems, and system structure, is provided in ISO/IEC/IEEE 24748-1. ISO/IEC/IEEE 24748-2 provides more information on concepts related to system life cycle management.

System concepts are directly applicable to software systems. An underlying principle of ISO/IEC/IEEE 12207:2017 is that software engineering applies similar processes to systems engineering, but that software is the leading method for system requirements realization. Consequently, processes are aligned and adapted for methods and approaches relevant for software.

NOTE 2   As applicable to all systems and projects, guidance to concepts is found in ISO/IEC/IEEE 24748-1. ISO/IEC/IEEE 24748-2 includes guidance more specifically applicable to systems where hardware or other non-software elements are the primary concern.

Typically, the system engineering approach to develop a system design is described as a hierarchical, top-down process of systematic decomposition of the system into its subsystems and elements (components). This top-down approach has been traditionally applied to the architecture of software systems as well, from the top-level system-of-interest down through the lowest system element level of the software system structure, such as a line of code. However, the malleable, easily refactored nature of software leads to a different way of considering the software structure. The common practice of producing a minimum of new code and integrating a software system from available components (open source modules, application program interfaces (APIs), software services, support software such as a database management system and a web browser) leads to a more holistic view of its structure. Software systems include a number of components for processing information that can be directly or indirectly related to the software functions and requirements. In compartmentalized software, often none of the sub-systems or components can be considered "top-level" in a hierarchy. Traceability of a high-level stakeholder requirement to each specific subsystem can be difficult.

Characteristic properties at the boundary of a SoI arise from the interactions between subordinate systems. Whatever the boundaries chosen to define the software system, the concepts and models in this document are generic and permit a practitioner to correlate or adapt individual instances of life cycles to its software concepts and principles.

Enabling systems are required for each life cycle process and are typically integrated as an infrastructure for concurrent performance of multiple development, test, and operational processes. Enabling systems are deployed throughout the software life cycle to provide the SoI with support as needed. Each life cycle stage can require one or more enabling systems. An enabling system has its own life cycle; when ISO/IEC/IEEE 12207:2017 (or ISO/IEC/IEEE 15288:2015, if applicable) is applied to it, it then becomes a SoI.

Selection of methods and tools considers software size and complexity, project duration and the number of contributing organizations. Selection of tools should be based on connectivity to other tools that provide inputs or use its output. Ease of use, need for training before successfully using the tool, and availability of administrative support and enabling systems are also factors in tool selection and use. Use of a method or tool does not replace the standard process to be followed but should support the set of activities of a selected process.

4.3 Organizational concepts

An organization is a person or a group or people and facilities with an arrangement of responsibilities, authorities, and relationships. An identified part of an organization (even as small as a single individual) or an identified group of organizations can be regarded as an organization, if it has responsibilities, authorities, and relationships. When an organization, as a whole or a part, enters into a contract, it is a party. Organizations are separate bodies, but the parties may be from the same organization or from separate organizations.

ISO/IEC/IEEE 12207:2017 is intended to be applicable to organizations of all sizes and governance structures, from very small entities (VSE) of fewer than 25 people working on a single project, to large-scale organizations engaged in system of systems (SoS) sustainment for years or decades. The requirements of ISO/IEC/IEEE 12207:2017 are usually stated in terms of "the organization" or "the project," rather than by individual titles or roles. The extent to which tasks and roles can be differentiated depends on the resources available to the organization. Responsibilities that involve checking, validating, or verifying should be assigned to separate persons where the size of the organization permits. In some cases, only one person may be assigned to cover all tasks and take most of the roles and responsibilities.

Modern organizations strive to develop a robust set of life cycle processes that are applied repeatedly to the projects of the organization. To accommodate that need, ISO/IEC/IEEE 12207:2017 is intended to be useful at either the organization level or at the project level. An organization can adopt the standard and supplement it with appropriate policies, procedures, and tools.

A project of the organization typically conforms to the organization's processes rather than conforming directly to the standard. In some cases, projects may be executed by an organization that does not have

an appropriate set of processes applied consistently at an organizational level. Such a project may apply the provisions of the standard directly to a project.

An organization may perform one or more processes as part of its services. A process may be performed by one organization or more than one organization, with one of the organizations being identified as the responsible party. A single project may involve multiple organizations working together as partners. Such a project should use ISO/IEC/IEEE 12207:2017 to establish common terminology, as well as information flows and interfaces among the organizations to enhance communication.

In this document, as in ISO/IEC/IEEE 12207:2017, an organization (or a party) derives its name from the process it is currently performing, for example, it is called an acquirer when it performs the Acquisition process. In ISO/IEC/IEEE 12207:2017, the agreeing parties are called the acquirer and the supplier.

The application of ISO/IEC/IEEE 12207:2017 does not require a specific organizational structure for projects. There are numerous organizational models which can be successfully used to perform the processes of ISO/IEC/IEEE 12207:2017—as long as it is clear who is accountable for providing sufficient resources and for producing each outcome. Organizations can use hierarchical management structures or team-focused, agile work groups. The teams may include representatives specializing in each stage of the life cycle or each life cycle process.

Adaptation and application of ISO/IEC/IEEE 12207:2017 on a project, in which many persons may be legitimately involved, depends on responsibility and accountability. The teams or groups need to be given the appropriate responsibility and authority for doing the work required to meet the project requirements, for example the activities and tasks of a process. A party is identified with overall responsibility and accountability for that entire process, even though the execution of individual tasks may be by different people or groups.

Whether the existing basis for processes is an older version of ISO/IEC 12207 or some other reference point, the fundamental starting point is to identify all the changes needed to go from that basis to ISO/IEC/IEEE 12207:2017. If the existing process basis is an older standard, the amount of changes can be noticeably less than if a different process basis is in use.

NOTE   ISO/IEC/IEEE 12207:2017, Annex I provides a mapping from the processes of ISO/IEC 12207:2008 to the processes in ISO/IEC/IEEE 12207:2017. See also ISO/IEC/IEEE 12207-2 for more detailed mappings of outcomes, activities, and tasks.

Whatever the reason for an organization's application of ISO/IEC/IEEE 12207:2017 is, a suggested implementation and transition strategy consists of the following:

a)   Identify the desired life cycle model and goals for the organization and the project. Implementing a single process, without considering how it relates to other explicit or de facto processes, is less likely to be beneficial. The Life Cycle Management and Business and Mission Analysis processes can be useful for this effort. If no obvious link is established between this project and the organization's business focus, then lasting commitment to achieve the project goals will be difficult if not impossible to maintain.

b)   Plan the implementation. The Transition and Project Planning process can be useful. Identify roles and responsibilities of the project team/organization, assigning a single point of responsibility for each process. In many cases, one individual or organization may be responsible for more than one process, particularly in small projects or organizations.

c)   Tailor ISO/IEC/IEEE 12207:2017, if applicable.

d)   Conduct pilot project(s).

e)   Formalize the approach. A project management plan or change management plan can be useful.

f)   Institutionalize the approach, so that the process is supported by top management commitment, organizational policies and procedures, and is used consistently and automatically throughout the project or organization. This also involves measuring performance and implementing process improvement again as necessary.

This strategy is typical of a Change Management approach to introduce changes into an organization or project. The strategy described above may be repeated several times within a project or across an organization as additional processes are implemented or improved.

As with any program which results in changes to work practices, it is essential that the top management within the affected organization is visibly committed to implementing and supporting the changes. In a two-party situation, this can be initiated by a contract and then, as for general organizational use, policies are established with support of top management of both parties.

Bringing all the stakeholders together in this effort is critical: even one area left out that should have been in the planning can materially disrupt applying the new basis. One way of proceeding is for a small group to develop a checklist of transition activities, such as the following:

a)   documentation changes, including flow and nomenclature of policies and procedures;

b)   staff training needs;

c)   responsibility changes, including need for new agreements;

d)   impacts on tools and databases; and

e)   changes in the inputs required by and outputs from each process.

The initial checklist should then be used by an immediately following, larger, group of all stakeholders to work through what other items need to be added and what the specific changes are for each item on the checklist. Repeated reviews of checklist drafts should be held to find the final few surprises.

Once there is a detailed listing of the changes derived in this, or equivalent, manner, the time and cost impact of each needs to be assessed. Then further analysis of the sequence of implementing the changes is necessary. The group should explore phasing in changes in a way that minimizes cost, project disruption and the potential for adverse human reactions. Readiness criteria should be developed for starting each step of the transition, as well as checks for successful completion after each step. Quantitative metrics should be developed and used.

Throughout, a core group should be maintained to oversee the change from one basis to another, with periodic meetings of the entire group of stakeholders.

When a project or organization is already in a steady state, i.e. where the processes have been established and institutionalized, then the implementation strategy can be shortened, considering the risks and opportunities of the project.

4.4   Project concepts

A project is an endeavour with defined scope and start and finish dates, undertaken to work on a product or service in accordance with specified resources and requirements. Typically, a project exists to satisfy an agreement by providing the desired deliverables to the quality expected. For this purpose, ISO/IEC/IEEE 12207:2017 provides a set of Organizational Project-Enabling processes.

Any project is assumed to be conducted within the context of an organization. This is important because a software project is dependent upon various outcomes produced by the business processes of the organization, e.g., employees to staff the project and facilities to house the project.

To perform needed operations and transformations upon software systems during their life cycles, the organization creates and monitors projects. The organization both constrains and supports a project. Examples of such organizational constraints and supports are the following:

—   sets standards, policies, and procedures by which projects are carried out within the organization;

—   initiates, redirects or terminates projects according to business opportunities and strategies;

—   provides requested resources including physical and human within availability and financial constraints;

— provides infrastructure support;

— manages the overall quality of software systems produced by a project for internal or external customers.

Projects have defined scope, resources (including time) and focus. The organization can establish projects that are contiguous with stages in the software life cycle (for example, a project to develop software concepts and requirements, followed by another project to integrate, validate, and verify the software system). Projects can also align with the various organizations responsible for different software components. The scope can involve managing all of the stages of the life cycle, a subset of the stages, one or more defined processes, or one or more process activities. Although any software system should be sustained over its full life cycle, it is common for projects to span only portions of that life cycle.

The time scale of projects can vary, for example, one day or several years. The focus of the project is related to the software and its elements in some form of system structure or stage partitioning. The project places certain demands on the organization and the organization places demands on the project. The project requires physical infrastructure, financial and human resource support to carry out project work.

Relationships can exist between a project and other projects, and subprojects. A subproject is a set of resources and tasks organized to undertake a portion of a project. A subproject may be considered a project by those assigned the work.

Many software "projects" involve continuing delivery of capabilities to sustain a software system. Ongoing sustainment efforts, responsive to changes in the environment of the SoI or to new functional and non-functional requirements, handle a continuous stream of work packages, with the timespan limited only by established budgetary or financial reporting restrictions.

Figure 1 illustrates the concept of a software system and its relationship to an organization and its projects.
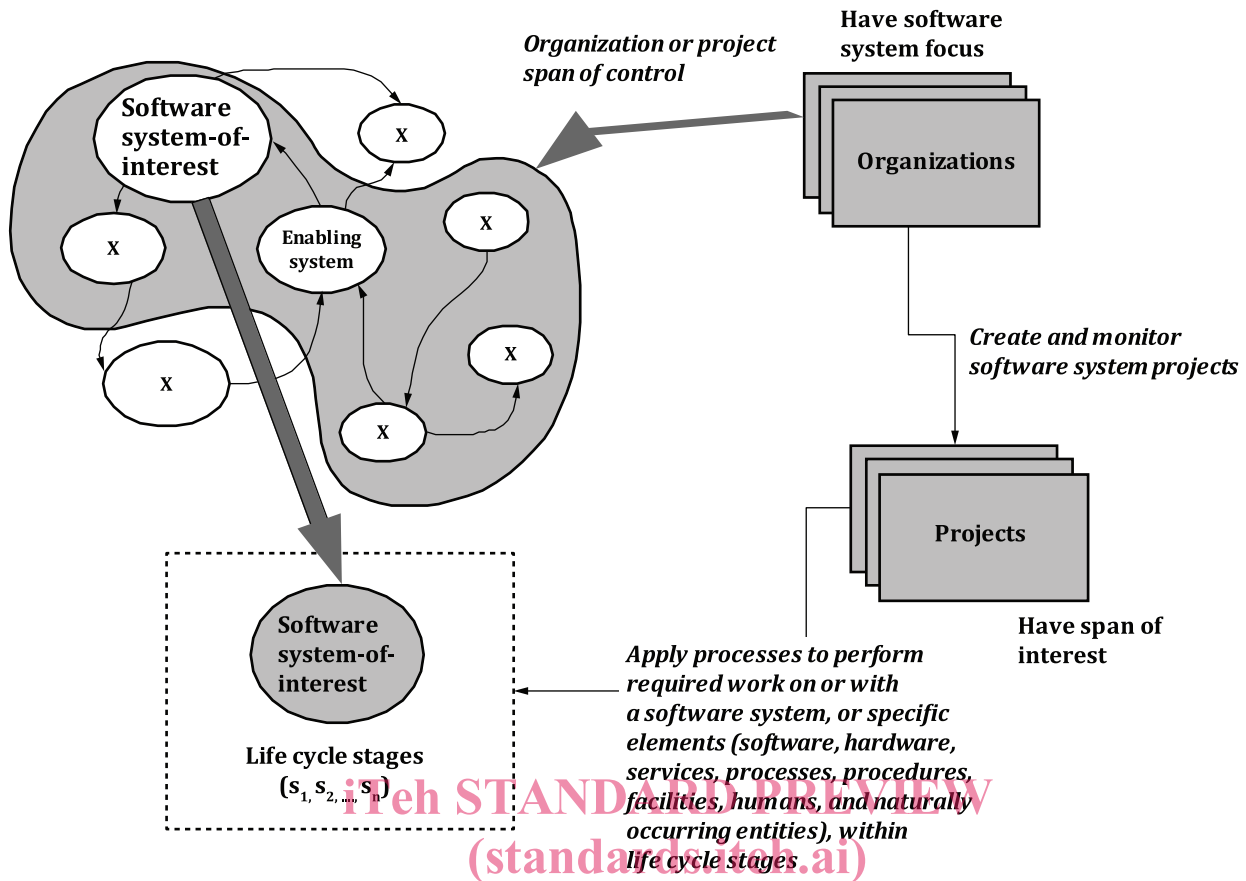
Figure 1 — Organizations, projects, and software systems

Some enabling systems are under direct control of the project. The software and those enabling systems make up the project span of control.

5 Process and life cycle concepts

5.1 Process concepts

A process is an integrated set of activities that transform inputs (for example, a set of data such as requirements) into desired outputs (for example, a set of data describing a desired solution). The software engineering processes in ISO/IEC/IEEE 12207:2017 are defined so that each process has a distinct purpose and outcomes (results). The purpose statement provides the overall rationale for the use of the process. The outcomes are the expected observable results from carrying out the activities of a process. The outcomes provided for each process in ISO/IEC/IEEE 12207:2017 provide a benefit that motivates selection and execution of that process. Attainment of the results fulfills the purpose for performing the process. However, outcomes are not identical with process outputs, which are tangible artifacts and services that can be evaluated. Processes are managed through controls and performed using enabling mechanisms (Figure 2).
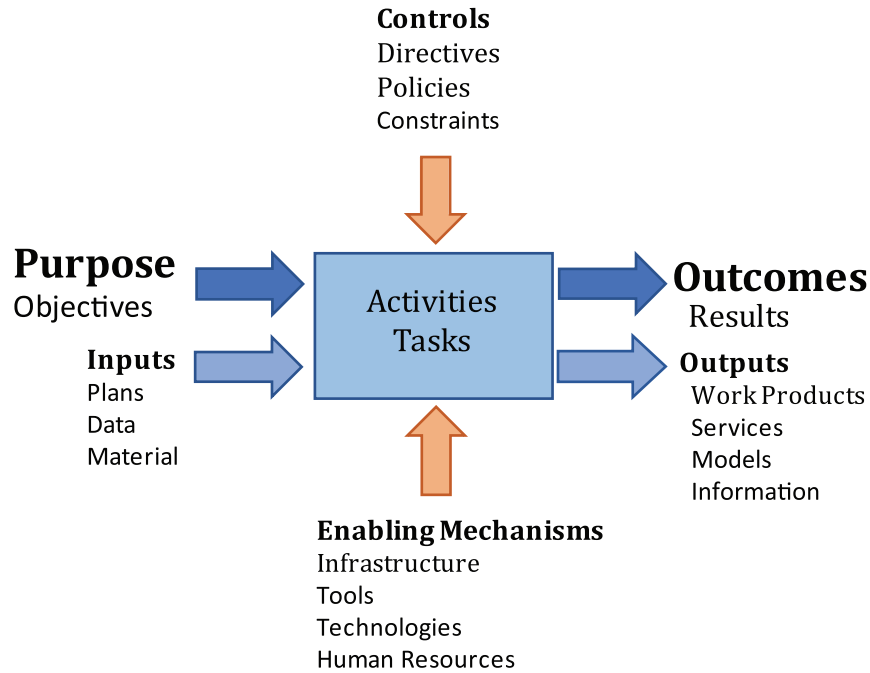
Figure 2 — Process elements

NOTE 1   Additional detail regarding this form of process description can be found in ISO/IEC/IEEE 24774.

Inputs can come from outside an organization or project, or from other processes that precede or accompany the process being examined. Outputs can go to other processes or back to the same process (recursive processing) inside the organization, project (or both), or they can go outside the project or organization, or both. Examples of inputs to, and outputs from, a process include the following:

a)   information, such as requirements, interface or architecture definitions;

b)   data, such as measurements and test reports;

c)   material that either ends up in the output or is consumed in producing the output;

d)   services that are part of a chain of services, such as setting up a computer prior to, or coincident with establishing an account.

The source of process controls and constraints include organizational or organization management directives and constraints; governmental regulations and laws; project agreements; interfaces with processes used on other systems for which the project is responsible; and internal standard policies and procedures of the organization. Government regulations and laws do not require any specific software design or implementation. They can be viewed as constraining the software system requirements. For consistent application, controls are typically automated through the enabling systems, or applied using methods, procedures, and techniques by the workforce.

ISO/IEC/IEEE 12207:2017 describes the set of processes that are applied to the life cycle of any software system. Therefore, ISO/IEC/IEEE 12207:2017 is designed so that its processes can be applied for a software project of any type, size and complexity, whether focused on tangible products, services, or a mix of both. The processes are designed to be used whether the software is considered as a stand-alone element, or a part of the total software system.

While the processes identified in ISO/IEC/IEEE 12207:2017 are intended to model all the work of software engineering, they do not form a closed loop model. The outputs of one process are not directly identified as inputs for the other processes; the inputs of one process are not always outputs of the other processes. Controls, directives, governance, policy, and organizational strategy and vision drive each process.