INTERNATIONAL STANDARD

ISO/IEC 26553

First edition
2018-11

Information technology — Software and systems engineering — Tools and methods for product line realization

Technologies de l'information — Ingénierie des systèmes et du logiciel — Outils et méthodes pour la réalisation d'une gamme de produits

Reference number
ISO/IEC 26553:2018(E)

ISO copyright office
CP 401 • Ch. de Blandonnet 8
CH-1214 Vernier, Geneva
Phone: +41 22 749 01 11
Fax: +41 22 749 09 47
Email: copyright@iso.org
Website: www.iso.org

Published in Switzerland

Contents

Page

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).

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 of the voluntary nature of standards, 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 Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 7, *Software and systems engineering*.

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 main purpose of this document is to deal with the capabilities of methods and tools of software and systems product line (SSPL) realization which includes detailed design and implementation. This document defines how the tools and methods can support the software and systems product line-specific realization processes.

Domain realization will be carried out based on domain architecture that provides structures and constraints that govern the subsequent SSPL lifecycle processes. The outcomes of domain realization processes are transferred into the realization of a member product at the application realization processes. Therefore realization support tools and methods should consider both engineering processes, namely domain realization and application realization.

Product line realization can be differentiated from a single product development because of the following aspects:

— The outcomes of domain requirements engineering and domain architecture form the basis for product line realization unlike the case of a single product development. There are two core processes in product line realization: domain realization and application realization. The major aims of the domain realization processes are to conduct detailed design and further implementation based on domain architecture, which includes commonality and variability for a family of products, and to prepare necessary variability information for variability modelling. Whereas, the major aims of the application realization processes are to conduct detailed design and implementation for application realization and to bind variability whose defined binding time is realization stage.

This document can be used in the following modes:

— by the users of this document: to benefit people who conduct detailed design and implementation for software and systems product lines;

— by a product line organization: to provide guidance on the evaluation and selection for methods and tools for product line realization; and

— by providers of methods and tools: to provide guidance on implementing or developing tools and methods by providing a comprehensive set of the capabilities of tools and methods for product line realization.

The ISO/IEC 26550 family of standards addresses both engineering and management processes and capabilities of methods and tools in terms of the key characteristics of product line development. This document provides processes and capabilities of methods and tools for product line realization. Other standards in the ISO/IEC 26550 family are as follows:

ISO/IEC 26550, ISO/IEC 26551, ISO/IEC 26555, ISO/IEC 26557, ISO/IEC 26558 and ISO/IEC 26559 are published. ISO/IEC 26552, ISO/IEC 26554, ISO/IEC 26556, ISO/IEC 26560, ISO/IEC 26561, ISO/IEC 26562 and ISO/IEC 26563 are planned International Standards. The following list provides an overview of the series:

— processes and capabilities of methods and tools for domain requirements engineering and application requirements engineering are provided in ISO/IEC 26551;

— processes and capabilities of methods and tools for domain design and application design are provided in ISO/IEC 26552;

— processes and capabilities of methods and tools for domain testing and application testing are provided in ISO/IEC 26554;

— processes and capabilities of methods and tools for technical management are provided in ISO/IEC 26555;

— processes and capabilities of methods and tools for organizational management are provided in ISO/IEC 26556;

- processes and capabilities of methods and tools for variability mechanisms are provided in ISO/IEC 26557;
- processes and capabilities of methods and tools for variability modeling are provided in ISO/IEC 26558;
- processes and capabilities of methods and tools for variability traceability are provided in ISO/IEC 26559;
- processes and capabilities of methods and tools for product management are provided in ISO/IEC 26560;
- processes and capabilities of methods and tools for technical probe are provided in ISO/IEC 26561;
- processes and capabilities of methods and tools for transition management are provided in ISO/IEC 26562;
- processes and capabilities of methods and tools for configuration management of asset are provided in ISO/IEC 26563; and
- others (ISO/IEC 26564 to ISO/IEC 26599) are to be developed.

Information technology — Software and systems engineering — Tools and methods for product line realization

1 Scope

This document, within the context of tools and methods of detailed design and implementation for software and system product lines:

- provides the terms and definitions specific to realization for software and systems product lines;
- defines processes performed during product line realization (those processes are described in terms of purpose, inputs, tasks and outcomes);
- defines method capabilities to support the defined tasks of each process; and
- defines tool capabilities to automate/semi-automate tasks or defined method capabilities.

This document concerns processes and capabilities of realization tools and methods for a family of products, not for a single system.

2 Normative references

There are no normative references in this document.

3 Terms and definitions

For the purposes of this document, the following terms and definitions apply.

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

- ISO Online browsing platform: available at https://www.iso.org/obp
- IEC Electropedia: available at http://www.electropedia.org/

3.1
application component
component that is selected, reused or newly developed for a *member product* (3.14)

3.2
application configuration
structure of a *member product* (3.14), including *application components* (3.1) and *application interfaces* (3.3)

3.3
application interface
interface that is selected, reused or newly developed by a *member product* (3.14)

3.4
application realization
one of the application engineering processes that includes detailed design and implementation

3.5
application-specific component
component that is developed for a specific *member product* (3.14)

3.6
aspect
special consideration within product line engineering process groups and tasks with which specialized methods and tools can be associated

3.7
binding time of variability
stage when the value of variability is determined

3.8
component implementation
activity of realizing a component, including unit test

3.9
configuration parameter
parameter provided by variable components or interfaces, so that its value is selected when bindings occur

3.10
domain component
reusable component among *member products* (3.14) within a product line

3.11
domain interface
reusable interface among the components of a *member product* (3.14) within a product line

3.12
domain realization
one of the domain engineering processes that include detailed design and implementation

3.13
extractive approach
approach of developing the initial baseline of a product line from one or more existing products

3.14
member product
product belonging to the product line

3.15
proactive approach
approach of developing an innovative product line or product variations based on organizational predictions that anticipate a stated product need

3.16
reactive approach
approach of developing a product line or product variations in response to stated needs or customer requirements

3.17
texture
architectural texture
collection of common development rules and constraints for realising the applications of a product line

3.18
variability implementation
variability development in source codes or executable modules

4 Reference model for product line realization

4.1 Overview

Product line realization supports the detailed design and implementation of a product line. This document provides requirements and guidance for building reusable components and member product-specific components. Product line realization consists of two separated processes: domain realization and application realization. Domain realization develops reusable components and interfaces that will be reused in application realization. COTS (commercial off-the-shelf), open sources and/or licensed third-party platforms can also be a significant part of domain assets together with the organizations' own domain artefacts. Many components are derived from domain components by selecting variants while member product-specific components or interfaces are built in application realization. However, not all planned components and interfaces are built in domain realization. Hence proper strategies for domain and application realization are necessary to enhance the reusability and to raise cost efficiency of the product line. Organization realization management provides subprocesses that deal with these at the organizational level. Annex A describes the scope of realization activities.

Realization of a product line shall be conducted using the common rules produced during the architecture design. Architectural textures provide rules and constraints that realization should adhere to, so that domain and application realization are coordinated well with each other.

NOTE 1 The architectural texture is the collection of common rules for realizing the system, such as coding rules and general mechanisms. Styles and design patterns are examples of the texture. Architectural texture demands the use of a hierarchy of layers, subsystems and components. It can require the use of the façade pattern or the presence of an interface with a prescribed set of functions as each component.

The reference model specifies the structure of supporting processes and subprocesses for product line realization. As shown in Figure 1, product line realization can be structured into seven processes; *organizational realization management, detailed domain design, domain implementation, variability management in realization, asset management in realization, detailed application design and application implementation*. Each process is divided into subprocesses to address product line realization issues, and each subprocess is described in terms of the following attributes:

- the title of the subprocess;
- the purpose of the subprocess;
- the outcomes of the subprocess;
- the inputs to produce the outcomes;
- the tasks to achieve the outcomes; and
- the capabilities of methods and tools required for performing the tasks effectively and efficiently.

NOTE 2 When the process, subprocess, outcomes and tasks are listed or described in a sentence they are italicized in order to increase their visibility.

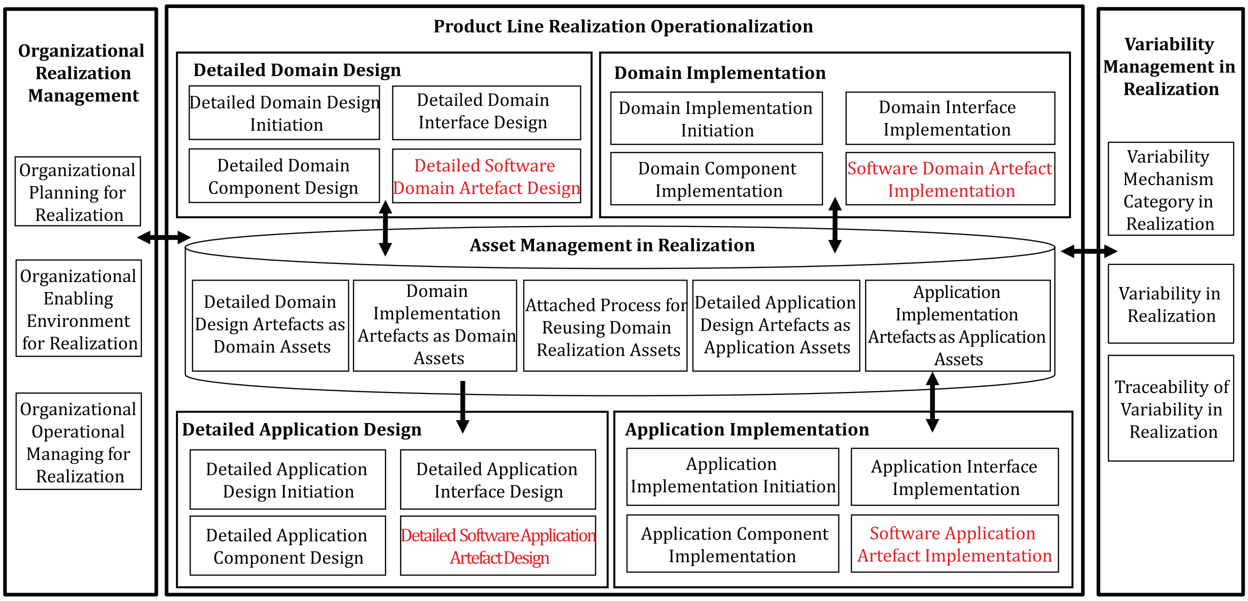
Organizational Realization Management
Organizational Planning for Realization
Organizational Enabling Environment for Realization
Organizational Operational Managing for Realization
Product Line Realization Operationalization
Detailed Domain Design
Detailed Domain Design Initiation
Detailed Domain Interface Design
Detailed Domain Component Design
Detailed Software Domain Artefact Design
Domain Implementation
Domain Implementation Initiation
Domain Interface Implementation
Domain Component Implementation
Software Domain Artefact Implementation
Asset Management in Realization
Detailed Domain Design Artefacts as Domain Assets
Domain Implementation Artefacts as Domain Assets
Attached Process for Reusing Domain Realization Assets
Detailed Application Design Artefacts as Application Assets
Application Implementation Artefacts as Application Assets
Detailed Application Design
Detailed Application Design Initiation
Detailed Application Interface Design
Detailed Application Component Design
Detailed Software Application Artefact Design
Application Implementation
Application Implementation Initiation
Application Interface Implementation
Application Component Implementation
Software Application Artefact Implementation
Variability Management in Realization
Variability Mechanism Category in Realization
Variability in Realization
Traceability of Variability in Realization

Figure 1 — Product line realization reference model

An organization's life-cycle models and strategy guide its realization process in a proactive approach, a reactive approach or an extractive approach. According to the predictability of variations among member products, those approaches can be applied, i.e. when the variations are predictable, a proactive approach is suitable, whereas when they are not, a reactive approach or an extractive approach can be used. Though the variations are not predictable, the reuse of the existing assets (or COTS components), should be considered during components or interfaces realization.

4.2 Organization realization management

The organizational realization management provides realization-specific managerial supports such as plans, enabling environments and operations. Organizational realization management shall serve to do the following and to define the capabilities of tools and methods for supporting them:

- *Organizational planning for realization* establishes an organization level plan to proceed to product line realization from product line architecture design based on targeted product line requirements.
- *Organizational enabling environment for realization* establishes environments required for domain realization (detailed domain design and implementation), for deriving the realization of a specific member product from domain realization, and for the additional realization of product-specific requirements. The enabling environment includes organizational structure, technologies and tools that support product line realization.
- *Organizational operational managing for realization* monitors and controls the status of product line realization for harmonizing works conducted in domain engineering and application engineering processes.

4.3 Domain realization

The domain realization encompasses the detailed design and implementation of domain components and interfaces. Detailed domain design shall serve to do the following and to define the capabilities of tools and methods for related support:

- *Detailed domain design initiation* starts the detailed domain design by following domain architecture design specification.

- *Detailed domain interface design* elaborates on the detailed design of the interfaces of common and variable components to expose required and/or provided functionality of the components.
- *Detailed domain component design* elaborates the details of components such as classes and objects to be decomposed into data structures and algorithms based on the reference architecture and texture.
- *Detailed software domain artefacts design* conducts detailed design for software domain artefacts' functionalities that will be commonly used by member products of a product line.

4.4 Domain implementation

The domain implementation builds components and interfaces to be commonly used or to be selectively used by member products. Domain implementation includes building and buying components and supporting infrastructure. The planned variability should be realized with adequate variability mechanisms, and core components and interfaces are validated based on the domain architecture. Domain implementation shall serve to do the following and to define the capabilities of tools and methods for related support:

- *Domain implementation initiation* starts the domain implementation by following detailed domain design results.
- *Domain interface implementation* realizes domain interfaces based on the detailed interface design.
- *Domain component implementation* realizes domain components (e.g., coding, compiling, linking and loading in the case of software systems) based on the detailed design of the component.
- *Software domain artefacts implementation* produces software files for software domain artefacts' functionalities that will be commonly used by member products of a product line.

The difference between domain realization and a single system realization is in that the reusable components of domain realization are loosely coupled, configurable, and may not be executable because of variability.

4.5 Asset management in realization

Major domain assets in realization are detailed design artefacts, implementation artefacts and executable components. Application assets produced from application realization should be managed for application. In addition, application assets may be a part of core assets after being reviewed and reengineered by domain engineers. The domain asset management in realization shall serve to do the following and to define the capabilities of tools and methods for related support.

- *Detailed domain design artefacts as domain assets* make detailed domain design models and specifications conform to the domain asset structure that includes attributes to describe domain assets and index/annotation to retrieve or trace them.
- *Domain implementation artefacts as domain assets* make domain components and domain interfaces, which are declared in programming language files, conformant with the domain asset structure.
- *Attached process for reusing domain realization assets* defines processes that should be adhered to when a member product reuses domain realization assets.
- *Detailed application design artefacts as application assets* establish and maintain detailed application design models and specifications for each member product to be referred to for later maintenance and evolution.
- *Application implementation artefacts as application assets* establish and maintain member product-specific components and interfaces including executable artefacts for each member product to be referred to for later maintenance and evolution.

4.6 Detailed application application design

Application realization encompasses the detailed design and implementation of application components and interfaces. The detailed application design shall serve to do the following and to define the capabilities of tools and methods for related support:

- *Detailed application design initiation* initiates detailed application design that follows application architecture.
- *Detailed application interface design* elaborates on the detailed design of the interfaces of application components to expose required and/or provided functionality of the components.
- *Detailed application component design* elaborates the details of application components such as classes and objects to be decomposed into data structures and algorithms.
- *Detailed software application artefacts design* conducts detailed design for application-specific software application artefacts' functionalities.

4.7 Application implementation

The application implementation builds member product-specific components and interfaces. The application implementation shall serve to do the following and to define the capabilities of tools and methods for related support:

- *Application implementation initiation* srarts application implementation by following detailed application design. In order to start application implementation, implementation configurations are derived from domain implementations and detailed application design should be ready for guiding application implementation.
- *Application interface implementation* implements application interfaces in accordance with the detailed design specification.
- *Application component implementation* implements application components in accordance with the detailed design specification.
- *Software application artefacts implementation* produces software files for application-specific software application artefacts' functionalities.

4.8 Variability management in realization

Variability in domain design is distributed over domain components and interfaces in detailed design. Variability can appear in manifold manners in domain assets, so they make the variability management complicated, particularly if they show widespread impacts. Variability mechanisms in detailed design and variability mechanisms in implementation are quite different, so there are two different variability managements in realization, i.e. variability management in detailed design and variability management in implementation. Variability can be handled in a late stage of the member product lifecycle (e.g. building time or run-time) as this offers high flexibility. The variability management in realization shall serve to do the following and to define the capabilities of tools and methods for related support:

- *Variability mechanism category in realization* identifies and classifies variability realization mechanisms used in realization stage.
- *Variability in realization* deals with the capabilities to locate and implement variants in interfaces and variants over components. This subprocess deals with variability model and variability mechanisms in domain realization.
- *Traceability of variability in realization* establishes and maintains trace links between variability and realization artefacts (e.g. detailed design artefacts, codes and object files) including trace links between different abstraction levels of variability.

The identification and analysis of the key differentiators between single-system engineering and management and product line engineering and management can help organizations to understand the product line and to formulate a strategy for successful implementation of product line engineering and management. The key aspects have been defined in ISO/IEC 26550 and Table 1 shows the category of the key aspects.

Table 1 — Key aspects for identifying product line-specific realization tasks

Category	Aspects
Reuse management	application engineering, domain assets, domain engineering, product management, platform, reusability
Variability management	binding, variability
Complexity management	collaboration, configuration, enabling technology support, reference architecture, texture, traceability
Quality management	measurement and tracking, cross-functional verification and validation

For product line realization, relevant processes and tasks shall be identified on the basis of these aspects. The concerns specific to product line realization will enable an organization to understand the product line realization relevant processes, subprocesses, tasks, methods and tools' capabilities. The following describes each aspect concerning product line realization.

— **Application engineering**. Application realization is the task of application engineering where domain components and interfaces are reused and member product-specific components and interfaces are implemented.

— **Binding**. Binding time of variability is important for the flexibility of a product line. Variability can be introduced in realization and their binding times should be determined by considering trade-offs. During domain realization, variability bindings may occur at the defined binding times, and decisions for the binding mechanisms of internal variability may be made.

— **Collaboration**. Because it is very important that domain architecture and architectural texture are adhered to and large numbers of designers and developers are involved in domain realization, proper methods for coordinating and aligning designers and developers should be devised.

— **Configuration**. As bindings occur and member product-specific components and interfaces are added during realization, multiple versions of components and interfaces are generated. Their configurations should be managed considering the entire product line.

— **Domain asset**. Domain components and interfaces are the major assets of domain realization. Domain components and interfaces, including variability models, should be properly managed and an asset base should be used to support their reuse.

— **Domain engineering**. Domain realization is the task in domain engineering where domain components and interfaces are designed and coded and, if possible, compiled and built.

— **Enabling technology support**. Technologies for managing the complexity of detailed domain designs, coding and building should be supported.

— **Measurement and tracking**. Domain components and interfaces should be measured and monitored from the viewpoint of the overall product line objectives for on time corrective actions.

— **Platform**. Components and interfaces that constitute the platform of a product line are developed during domain realization.

— **Product management**. Since member products within a product line evolve continuously in accordance with the changes of markets and business opportunities, product management should monitor and support the evolution of domain realization.