INTERNATIONAL STANDARD

ISO/IEC 26552

First edition
2019-05

Software and systems engineering — Tools and methods for product line architecture design

Ingénierie du logiciel et des systèmes — Outils et méthodes pour la conception architecturale des gammes de produits

Reference number
ISO/IEC 26552:2019(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 architecture design for software and systems product line (SSPL). This document defines how the tools and methods can support for the software and systems product line-specific architecture processes.

Domain architecture provides structures and constraints that govern all the subsequent SSPL lifecycle processes as well as being transferred into the architecture design of a member product at the application design processes. Therefore, SSPL architecture design should be defined in detail, considering constraints, so that other processes have a consistent foundation. Supporting tools and methods of architecture design should consider those engineering processes that use and are affected by architecture design.

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

— There are two core processes in architecture design: domain and application architecture design. The major aims of the domain architecture design processes are to design architectural structure and texture based on domain requirements which includes commonality and variability for a family of products, and to prepare necessary variability information for variability modelling. On the other hand, the major aims of the application architecture design processes are to derive application architecture through binding and add application-specific architectural structure.

— The outcomes of domain requirements engineering form the basis for product line architecture design and application-specific requirements might compel to add new components or tailor the structure unlike in the case of a single product development.

— The architectural texture, one of the major outcomes of product line architecture design defines common ways to deal with variability in domain realisation as well as in application design and application realisation. Domain realization should adhere to the rules defined in the architectural texture, and application architecture should comply with the rules defined in the architectural texture.

This document can be used in the following modes:

— by the users of this document — to benefit people who conduct domain and application architecture design for software and systems product lines;

— by a product line organization — to provide guidance in the evaluation and selection for methods and tools for domain and application architecture design;

— by providers of methods and tools — to provide guidance in implementing or developing tools and methods by providing a comprehensive set of the capabilities of tools and methods for domain and application architecture design.

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 domain design and application design. Other standards in the ISO/IEC 26550 family are as follows:

ISO/IEC 26550, ISO/IEC 26551, ISO/IEC 26552, ISO/IEC 26554, ISO/IEC 26555, ISO/IEC 26556, ISO/IEC 26557, ISO/IEC 26558 and ISO/IEC 26559 are published. ISO/IEC 26560, ISO/IEC 26561 and ISO/IEC 26562 are to be published. ISO/IEC 26563 is a planned International Standard.

— 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 realization and application realization are provided in ISO/IEC 26553;

— 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 modelling 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;

— Others (ISO/IEC 26564 to ISO/IEC 26599): To be developed.

Software and systems engineering — Tools and methods for product line architecture design

1 Scope

This document, within the context of methods and tools for architecture design for software and systems product lines:

— defines processes and their subprocesses performed during domain and application architecture design. Those processes are described in terms of purpose, inputs, tasks and outcomes;

— defines method capabilities to support the defined tasks of each process;

— defines tool capabilities to automate/semi-automate tasks or defined method capabilities.

This document does not concern processes and capabilities of tools and methods for a single system but rather deals with those for a family of products.

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 http://www.iso.org/obp

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

3.1
application architecture
architecture concept, including the architectural structure and rules (e.g. common rules and constraints), and architecture artefacts (such as descriptions) that constrains a specific member product within a product line

3.2
architectural texture
texture
collection of common development rules, guidelines and constraints that deals with common and variable aspect of the *product line architecture* (3.8)

3.3
architecture structure
physical or logical layout of the components of a system design and their internal and external connections

Note 1 to entry: Annex C provides an example of architecture structure and *texture* (3.2).

3.4
aspect
special consideration within product line engineering process groups and tasks to which one can associate specialized methods and tools

3.5
domain architecture
common architecture for a product line that can embrace *variability* (3.10) of member products

3.6
external variability
variability (3.10) that is visible to customers

3.7
internal variability
variability (3.10) that is hidden from customers

3.8
product line architecture
architecture, including both *domain architecture* (3.5) and *application architecture* (3.1)

3.9
reference architecture
core architecture that captures the high-level architecture concept of *domain architecture* (3.5) and *application architecture* (3.1)

3.10
variability
characteristics that can differ among member products of a product line

4 Reference model for product line architecture design

4.1 Overview

Product line architecture design consists of two development life cycle processes, domain design and application design. Domain design develops the domain architecture, including architectural structure and texture that enables domain realization and all member products within a product line. The domain architecture captures the high-level design of a product line, including the external and internal variabilities defined in domain requirements and architecture design as well as commonalities. Most of internal variabilities due to the chosen technical solutions are defined during architecture design, and external variabilities are refined into internal variabilities if necessary. The domain architecture should be valid for those variabilities.

Application design derives the application architecture from the domain architecture in accordance with application requirements and decisions for technical solutions. Member-product specific adaptations shall be done to satisfy member product specific requirements. The adaptations should adhere to the texture of the domain architecture.

NOTE 1 Annex A describes the cross-reference with ISO/IEC/IEEE 42010, ISO/IEC/IEEE 42020 and ISO/IEC/IEEE 15288.

The reference model specifies the structure of supporting processes and subprocesses for product line architecture design. The reference model for product line architecture design in Figure 1 is structured into five processes, architecture management, domain design, variability management in design, asset management in design and application design.

Each process is divided into subprocesses and each subprocess is described in terms of the following attributes:

— the title of the subprocess;

— the purpose of the subprocess;

— the inputs to produce the outcomes;

— the tasks to achieve the outcomes;

— the outcomes of the subprocess;

— 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 get them noticed.

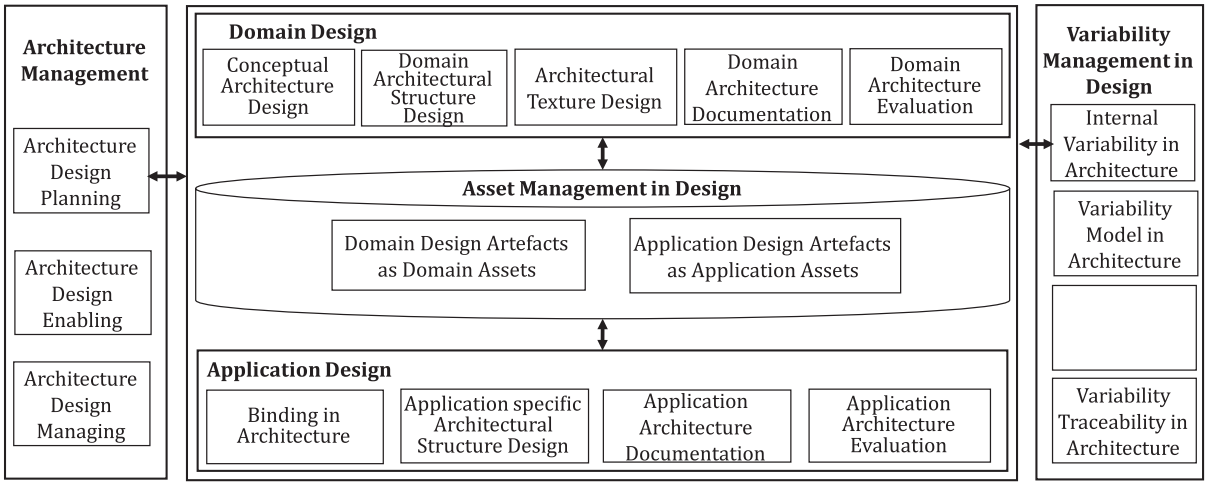

Figure 1 — Product line architecture design reference model

4.2 Architecture management

The architecture management shall serve to do the following and to define capabilities of tools and methods for supporting them:

— *Architecture design planning* makes and maintains overall plans, including guidance for making well-aligned domain and application architecture design with product line objectives. This subprocess also defines schedules and required resources for architecture design;

— *Architecture design enabling* establishes environments required for domain architecture design, for deriving architectures for member products from the domain architecture that complies with the architectural texture, and for adding new components satisfying member product specific requirements. Enabling environment includes organizational structure, technologies and tools that support product line architecture design;

— *Architecture design managing* monitors and controls the status of product line architecture design for harmonizing works conducted in other domain engineering and application engineering processes. It aims at managing domain architecture and maintenance for the relevant artefacts undergoing evolutions.

4.3 Domain design

The domain design develops a domain architecture that enables the realization of the planned commonality and variability within a product line. The aim of the domain design is to produce a domain architecture including architectural textures. The domain architecture reflects additional or refined internal variability introduced by the technical solution. The domain design is divided into five subprocesses and shall serve to do the following and to define capabilities of tools and methods for supporting them:

— *Conceptual architecture design* sets up high-level domain architecture design taking into account the commonality and variability in requirements without going into implementation details; it confirms the readiness for performing domain architecture design;

— *Domain architectural structure design* produces a domain architecture that decomposes a product line into its key components and their relationships. Analysis and modelling should be conducted to assure that the domain architecture covers all commonality and variability in requirements and that added or refined internal variability is covered;

— *Architectural texture design* defines constraints and rules called the architectural texture that can guide architecture designers and domain engineers in evolving a product line over time without destroying its key architectural concepts. The architectural texture deals with specific situations that can occur during architecture design, realization and testing;

— *Domain architecture documentation* describes domain architectural decisions, structure and textures so that domain realization, domain testing and application architecture design use the documentation as the basis for performing their tasks;

— *Domain architecture evaluation* reviews the domain architectural structure and texture for assuring that the domain architecture meets functional and non-functional (quality) requirements.

4.4 Asset management

The asset management in design shall serve to do the following and to define capabilities of methods and tools for supporting them:

— *Domain design artefacts as domain assets* supports member products of a product line to develop their architectures, including architectural specifications using the reusable architecture design artifacts by processing them as domain assets. Domain architecture design artefacts that will be reused by the product line members includes architecture specification or architecture descriptions (described using architecture description languages). Reusable domain artefacts such as components, interfaces and test cases (for integration testing) are also selected and managed as domain assets;

— *Application design artefacts as application assets* supports member products to manage their reusable application specific architecture design artefacts for further reuse. Application specific components, interfaces and test cases (for integration testing) are selected and managed along with the application specific architecture.

4.5 Variability management in design

Some external variabilities are refined into several internal variabilities, and many internal variabilities are newly introduced as well during architecture design. Thus, variability modelling, tracing and supporting mechanisms should be carefully managed in architecture design stage. The variability management in design shall serve to do the following and to define capabilities of methods and tools for supporting them:

— *Internal variability in domain architecture* refines external variability into internal variability at the architecture level and defines technical issues as internal variability. Most internal variabilities of a product line are refined and newly introduced at the architecture level;

— *Variability model in architectur*e integrates internal variability introduced in the domain architecture with the variability model in requirements and explicitly represent and document the results, including the detailed binding information;

— *Variability mechanism in architecture* deals with the capabilities to implement variants in architecture;

— *Variability traceability in architecture* establishes and maintains trace links between variability and architecture including trace links between different abstraction levels of variability.

4.6 Application design

The application design derives the application architecture from the domain architecture based on the application requirements. Product-specific adaptations are then built as far as the texture allows. The application architecture should be consistent with the domain architecture so as to enable the reuse of domain artefacts. Application design shall serve to do the following and to define capabilities of methods and tools for supporting them:

— *Binding in architecture* binds the variants for the variation points of the domain architecture according to the architectural texture and binding decisions. Through binding, variation points are substituted with the selected variants and domain components of the domain architecture will be selected or unselected; it confirms the readiness for performing the application specific architecture design;

— *Application specific architectural structure design* is conducted for covering application specific requirements to parts of the architecture. A large part of the application architecture is derived from the domain architecture, but application specific requirements have to be designed by the application architect;

— *Application architecture documentation* describes application specific architectural decisions and textures so that application realization and testing use the documentation as the basis for performing their tasks;

— *Application architecture evaluation* is conducted only for the application specific architecture. However, the pre-evaluated parts of the architecture during domain design should be assessed in order to confirm their integrity with the bound variants.

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 architecture design 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

The following are the descriptions for each aspect of Table 1 concerning domain and application architecture design. The domain and application architecture design relevant processes and tasks shall be identified on the basis of these aspects. The concerns specific to domain and application architecture design enable an organization to understand domain and application architecture design relevant processes, subprocesses, tasks, methods and tools' capabilities:

— Application engineering: Application architecture design is the stage of application engineering in which the domain architecture and selected reusable components and interfaces are reused. Application-specific architectural element should be developed and integrated. The provision of systematic reusability is a key aspect peculiar to product line development.

— Binding: Application architects bind the variation points and variants in domain assets (i.e. domain architecture, selected reusable components and interfaces) according to the application variability models, so processes, methods and tools of architecture design for product lines should contain capabilities for binding.

— Collaboration: Domain requirements engineering and domain architecture design should be performed in a closely related and iterative way. Hence, requirements engineers and architects should collaborate for determining right technical solutions and for producing a well-structured domain architecture.

— Configuration: Since selected components and interfaces of the domain architecture should be reused in application architecture and they should be realized by the subsequent phases in accordance with the defined architectural texture, configurations of domain assets should be consistently managed.

— Domain asset: Domain design assets such as domain architecture, selected components and selected interfaces that will be reused in the application design and realization are the major domain assets of product line architecture, which distinguishes architecture design of the product line development from that of single product development.

— Domain engineering: Domain engineering processes, which include domain design, are product line-specific aspects that do not exist in single product development.

— Enabling technology support: Enabling technologies for supporting efficient reuse and management of variability and assets distinguish architecture design for a product line from single product development.

— Measurement and tracking: Reusability of components and interfaces selected during domain and application design should be properly measured and tracked.

— Platform: Domain architecture design enables the development of a platform and the development of application architecture based on the platform.

— Product management: The domain architecture should evolve in accordance with the evolution and changes of market situations and/or technology trends.

— Reference architecture: The reference architecture is the major outcome from the domain design process.

— Reusability: The domain architecture should be developed using available or generic artefacts as the basis for application architecture. Achieving desired level of reusability in a domain architecture is a key aspect peculiar to product line development.

— Texture: The architectural texture should be defined in necessary detail so that domain realization, domain testing and application design activities follow the constraints and rules defined in the texture.

— Traceability: Traceability among requirements, architecture and realization for domain and application should be maintained and managed.

— Cross functional validation and verification: Detailed processes for validation and verification for domain and application architecture as well as textures should be performed.

— Variability: Internal variability as well as external variability should be modelled and managed. Variability modelling and management are distinguished aspects of product line development.

5 Architecture management

5.1 General

Architecture management supports the following subprocesses:

— *Architecture design planning*;

— *Architecture design enabling*;

— *Architecture design managing*.

