INTERNATIONAL STANDARD

ISO/IEC 26551

First edition
2012-12-01

Software and systems engineering — Tools and methods for product line requirements engineering

Ingénierie du logiciel et des systèmes — Outils et méthodes pour l'ingénierie d'exigences pour gammes de produits

Reference number
ISO/IEC 26551:2012(E)

ISO copyright office
Case postale 56 • CH-1211 Geneva 20
Tel. + 41 22 749 01 11
Fax + 41 22 749 09 47
E-mail copyright@iso.org
Web www.iso.org

Published in Switzerland

Contents

Page

Foreword

ISO (the International Organization for Standardization) is a worldwide federation of national standards bodies (ISO member bodies). The work of preparing International Standards is normally carried out through ISO technical committees. Each member body interested in a subject for which a technical committee has been established has the right to be represented on that committee. International organizations, governmental and non-governmental, in liaison with ISO also take part in the work. ISO collaborates closely with the International Electrotechnical Commission (IEC) on all matters of electro-technical standardization.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 3.

Draft International Standards adopted by the technical committees are circulated to the member bodies for voting. Publication as an International Standard requires approval by at least 75 % of the member bodies casting a vote.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO shall not be held responsible for identifying any or all such patent rights.

ISO/IEC 26551 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 7, *Software and systems engineering*.

Introduction

The major purpose of this International Standard is to deal with the capabilities of tools and methods of software and systems product line (SSPL) requirements engineering. This International Standard defines how the tools and methods can support for the software and systems product line-specific requirements engineering processes.

Decision for the initial boundaries of domain is conducted in advance by defining a product line scope before initiating domain requirements engineering processes. Domain requirements engineering will be carried out in a comprehensive manner because common and variable requirements and captured variabilities have consequential impacts on member products in a product line. The outcomes of domain requirements engineering processes are transferred into the requirements of a family of products at the application requirements engineering processes. Therefore requirements engineering tools and methods should consider both engineering processes namely domain requirements engineering and application requirements engineering.

Product line requirements engineering can be differentiated from a single product requirement engineering because of the following aspects:

— There are two core processes in requirements engineering: domain requirements engineering and application requirements engineering. The major aims of the domain requirements engineering processes are to analyze commonality and variability for a family of products, and to prepare necessary variability information for variability modelling. The major aims of the application requirements engineering processes are to define application specific requirements and bind variability defined in domain requirements engineering processes.

— It is essential to analyse the costs and benefits estimation of a product line and thereby an organization can make a go/no-go decision. Moreover, the costs and benefits estimation plays a pivotal role as an indicator for assessing the effectiveness and efficiency of a product line.

This International Standard can be used in the following modes:

— By the users of this International Standard – to benefit people who develop, operate, and manage requirements engineering for software and systems product lines.

— By a product line organization – to provide guidance in the evaluation and selection for tools and methods for product line requirements engineering.

— By providers of tools and methods – to provide guidance in implementing or developing tools and methods by providing a comprehensive set of the capabilities of tools and methods for product line requirements engineering.

ISO/IEC 26550 (ISO/IEC 26550, Software and systems engineering — Reference model for product line engineering and management) addresses both engineering and management processes and covers the key characteristics of product line development. ISO/IEC 26550 provides an overview of the consecutive international standards (i.e., ISO/IEC 26551 through ISO/IEC 26556) as well as the structure of the model:

— Processes and capabilities of methods and tools for product line scoping, domain requirements engineering, and application requirements engineering are provided as ISO/IEC 26551, *Software and systems engineering — Tools and methods for product line requirements engineering*.

— Processes and capabilities of methods and tools for domain design and application design are provided as ISO/IEC 26552, *Software and systems engineering — Tools and methods for product line architecture design*.

- Processes and capabilities of methods and tools for domain realization and application realization are provided as ISO/IEC 26553, *Software and systems engineering — Tools and methods for product line realization*.
- Processes and capabilities of methods and tools for domain verification and validation and application verification and validation are provided as ISO/IEC 26554, *Software and systems engineering — Tools and methods for product line verification and validation*.
- Processes and capabilities of methods and tools for technical management are provided as ISO/IEC 26555, *Software and systems engineering — Tools and methods for product line technical management*.
- Processes and capabilities of methods and tools for organizational management are provided as ISO/IEC *26556, Software and systems engineering — Tools and methods for product line organizational management*.

Software and systems engineering — Tools and methods for product line requirements engineering

1 Scope

This International Standard deals with the tools and methods of requirements engineering for software and systems product line. The scope of this International Standard is as follows:

- provide the terms and definitions specific to requirements engineering for software and systems product lines.
- define process groups and their processes performed during product line requirements engineering. Those processes are described in terms of purpose, inputs, tasks, and outcomes.
- define method capabilities to support the defined tasks of each process.
- define tool capabilities to automate/semi-automate tasks or defined method capabilities.

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

NOTE This International Standard is not suitable for handling physical artifacts. In the Systems arena, the word "Product" must be understood as System-level artefacts, such as requirement documents, architectural data, validation plans, Behavioral Models, etc. In any case, the word "Product" must not be understood as physical items such as electronic boards, mechanical parts or qualified human operators. In the case of the Software components of a Systems, this International Standard can apply twice: once to handle the System-Level Product Line and a second time to handle the Software Part Product Line, if any. The Product Line processes are recursive within the different levels of Products.

2 Normative references

The following referenced documents are indispensable for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO/IEC 26550, *Software and systems engineering — Reference model for product line engineering and management*

3 Terms and definitions

For the purposes of this document, the terms and definitions given in ISO/IEC 26550 and the following apply following terms and definitions apply.

3.1
application assets in requirements
application specific artifacts produced during application requirements engineering such as application requirements specifications and application requirements models

3.2
application requirements elicitation
identifies stakeholders relevant to an application, elicits application specific requirements, and binds the appropriate variants

3.3
application requirements analysis
ensures that all application specific requirements are understood, and scrutinizes incorrect and inconsistent application requirements through modeling. And application requirements that cannot be satisfied through the domain requirements are then analyzed and negotiated

3.4
application requirements specification
documents the application specific requirements and integrates it with the domain requirements specification whose variants are bound

3.5
application requirements verification and validation
confirms that the application specific requirements are consistent and feasible, and ensures that the bound variants satisfy the specific product's requirements

3.6
application requirements management
manages traceability and changes on application requirements

3.7
asset proposal
includes major assets (functional areas and high-level common and variable features of all applications) that will be included in a product line with their quantified costs and benefits, and estimation results

3.8
application specific requirements
requirements specific to an application or requirements not covered in domain requirements

3.9
domain assets in requirements
reusable artifacts produced during domain requirements engineering such as asset proposals, domain requirements specifications and domain requirements models

3.10
domain requirements elicitation
identifies initial requirements from domain stakeholders for a product line

3.11
domain requirements analysis
models domain requirements so as to analyze and scrutinize commonality/variability of a product line in requirements

3.12
domain requirements specification
documents domain requirements for a product line based on domain analysis results

3.13
domain requirements verification and validation
confirms that domain requirements are corrective, consistent, and complete

3.14
domain requirements management
manages traceability and changes with respect to domain requirements and their relevant domain/application artifacts

3.15
functional domain
categorized functions that are generally used together

3.16
production plan
description of how domain assets are to be used to develop member products in a product line

3.17
requirements traceability
covers traceabilities in domain and application requirements respectively, and those between them

3.18
variability in requirements
deals with both external and internal variability in requirements engineering phase. Also variability modeling and traceability with domain requirements artifacts are addressed

4 Reference model for product line requirements engineering

The methods and tools for product line requirements engineering should support systematic management and interaction of the domain and application requirements engineering processes. They also need to be adequately integrated with the subsequent processes of product line engineering lifecycle processes in order to enable traceability between all requirements artifacts and the related design, realization, and testing artifacts. In the rest of this document, product line requirements engineering practices, methods, and tools are described in accordance with a framework focusing on product line requirements engineering (Figure 1).

Product line scoping leads and controls all work on a product line by creating and maintaining the product line scope through ongoing interactions with the domain and application requirements engineering.

Domain requirements engineering serves to:

- decompose features defined in Product Line Scoping into initial requirements and elicit additional requirements and derived requirements from stakeholders and domain experts;
- analyze domain requirements with the variabilities of those;
- model and simulate the static and behavioral constructs of domain requirements;
- document domain requirements specifications that can be bound by specific member products of a product line.

The complexity of variability grows higher in accordance with the complexity of a product line. Separating variability from domain requirements engineering mitigates this problem. Defining variability in requirements independently leads to a clear understanding of the necessary capabilities of tools and methods, and thus helps in selecting tools that support product line requirements engineering.

Application requirements engineering serves to:

- identify gaps between domain features and application specific features
- reuse domain requirements from the asset repository and elicit application specific requirements;
- define application variability model by binding domain variability model and adding application specific variability;
- analyze and document application specific requirements;
- provide feedback to product line scoping and domain requirements engineering for the evolution of a product line.

The reference model for product line requirements engineering in Figure 1 is structured into five *processes, product line scoping, domain requirements engineering, variability management in requirements engineering, asset management in requirements engineering, and application requirements engineering*. Each process is divided into subprocesses to address product line requirements issues, 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 tools and methods are a list of the required support of tools and methods for performing the tasks properly

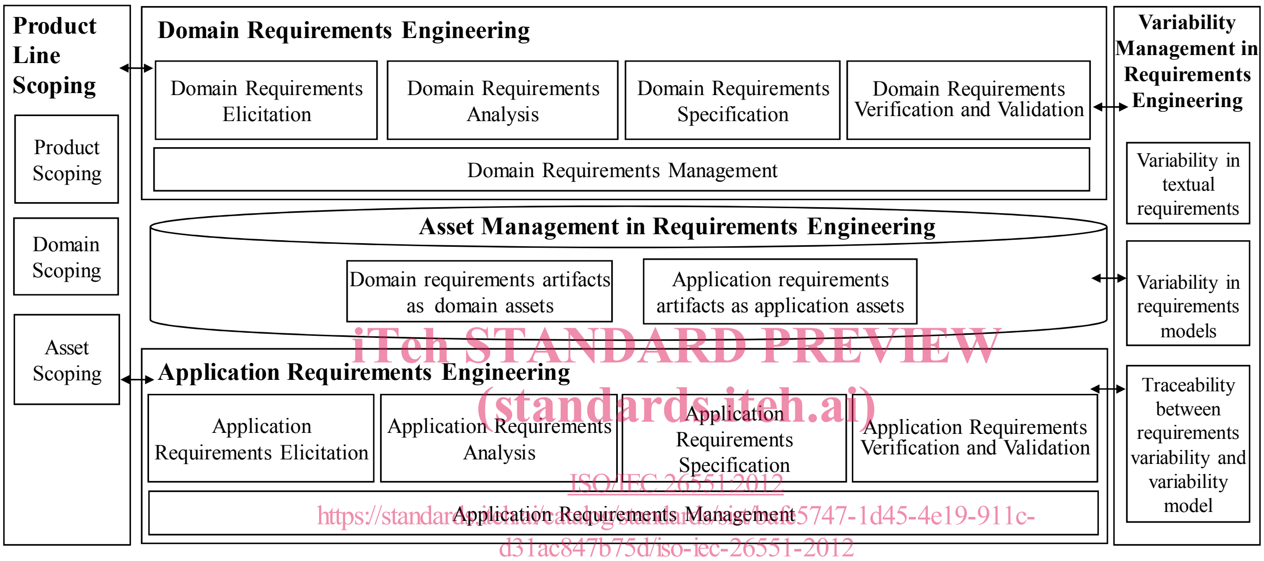
Product Line Scoping
Product Scoping
Domain Scoping
Asset Scoping
Domain Requirements Engineering
Domain Requirements Elicitation
Domain Requirements Analysis
Domain Requirements Specification
Domain Requirements Verification and Validation
Domain Requirements Management
Asset Management in Requirements Engineering
Domain requirements artifacts as domain assets
Application requirements artifacts as application assets
Application Requirements Engineering
Application Requirements Elicitation
Application Requirements Analysis
Application Requirements Specification
Application Requirements Verification and Validation
Application Requirements Management
Variability Management in Requirements Engineering
Variability in textual requirements
Variability in requirements models
Traceability between requirements variability and variability model

Figure 1 — Product line requirements engineering

Product line scoping defines the member products and their major (externally visible) features, analyzes the products from an economic point of view, and controls and schedules producing the product line and its products. The major result of this process is the asset proposals. Asset proposal includes major assets (functional domains and high-level common and variable features) for a product line with their quantified costs and benefits, and ROIs expected from a product line development. More than one asset proposals can be made to find out an optimal set of products and assets. Domain and application requirements engineering start from the features defined in the asset proposals. Product line scoping shall serve to do the following and to define the capabilities of tools and methods for supporting them:

- *Product scoping* determines the product portfolio definition, and provides a roadmap for releasing specific applications to customers or to market.
- *Domain scoping* identifies and bounds the functional domains to provide sufficient reuse potential.
- *Asset scoping* identifies reusable assets, calculates the cost/benefit estimates to justify the product line initiation.

Domain requirements engineering begins with using the outcomes of product line scoping. It comprehensively captures the initial domain requirements for a product line, and constructs an initial-requirements specification including a variability model. It also provides feedback on the changes required in the feature sets and the product roadmap as a whole to Organizational Business Management process group. Domain requirements engineering documents domain requirements specifications for the later use in domain design and in application requirements engineering. Domain requirements engineering shall serve to do the following and to define the capabilities of tools and methods for supporting them:

- *Domain requirements elicitation* captures initial domain requirements and anticipated variations of those.
- *Domain requirements analysis* identifies functional and non-functional requirements with the variabilities of those.
- *Domain requirements specification* documents domain requirements based on analysis results.
- *Domain requirements validation and verification* confirms that the specified domain requirements are consistent and feasible, and ensures that all products' requirements within a product line are well understood.
- *Domain requirements management* provides management services for the dual nature of the requirements engineering, i.e. domain and application requirements engineering.

Variability management in requirements engineering should be conducted in parallel with domain requirements engineering because variability models are clarified and modified gradually together with the domain and application requirements. Variability modeling starts in the domain requirements elicitation phase and continuously evolves throughout the product line life-cycle. This process is responsible for a variability model which documents the external variability explicitly. As domain requirements engineering activities proceed, some additional internal variabilities may be added to the variability model. Variability management in requirements engineering shall serve to do the following and to define the capabilities of tools and methods for supporting them.

- *Variability in textual requirements* expresses and documents variability in requirements using natural language and makes them explicit.
- *Variability in requirements model* expresses and documents variability in requirements using modeling language and makes them explicit.
- *Traceability between requirements variability and a variability model* establishes and maintains links between *textual requirements /requirements models* and a variability model.

During asset management in requirements engineering, requirements artifacts resulting from domain requirements engineering are structured as domain assets. Variability as well as commonality in requirements is managed as domain assets. In addition, application requirements artifacts with high reusability potentials are identified as potential domain assets. Asset management in requirements engineering adds or develops extra elaborations and glues to requirements assets to be used effectively and efficiently. The relationship among requirements domain assets for being reused successfully or managing changes on them are also analyzed in this process area. Processes for configuring domain assets and managing them in asset repository refer to asset management of ISO/IEC 26555. Asset management in requirements engineering shall serve to do the following and to define the capabilities of tools and methods for supporting them.

- *Domain requirements artifacts as domain assets* identify and develop necessary information to help application engineers reuse requirements assets in their application development.
- *Application requirements artifacts as application assets* identify and manage application requirements artifacts as assets to be referred by the application later.

Application requirements engineering identifies specific requirements for each product line member. It starts to assess the reusability of existing common and variable requirements to fully leverage a product line platform. It also can provide feedback to domain requirements engineering so as to make a decision on incorporating application requirements assets into domain assets. Application requirements engineering shall serve to do the following and to define the capabilities of tools and methods for supporting them:

- *Application requirements elicitation* identifies gaps between domain features and application specific features, reuses domain requirements from the asset repository, and elicits application specific requirements.
- *Application requirements analysis* abstracts, organizes, and models application specific requirements. And this subprocess has to ensure that all requirements of the application stakeholders are understood, and has to scrutinize the correctness, completeness, and consistencies of application requirements.
- *Application requirements specification* documents the analyzed application specific requirements with the bound portion of domain requirements specification.

- *Application requirements verification and validation* confirms that the specific product requirements are consistent and feasible, and ensures that the bound variants are relevant to the specific product requirements.
- *Application requirements management* provides management services for subsequent changes of the member product's requirements.

The identification and analysis of the aspects for the product line requirements engineering will enable an organization to understand the requirements engineering processes and to formulate a strategy for the successful implementation of the concept. Requirements engineering processes for product lines should be defined in terms of these aspects, and capabilities of tools and methods for supporting these processes should be identified on the bases of these aspects:

The following table shows the key aspects for each characteristic of product line requirements engineering.

Category	**aspects**
Reuse management	application engineering, domain assets, domain engineering, product management, platform, reusability
Variability management	binding, variability
Complexity management	collaboration, configuration, domain architecture, enabling technology support, texture, traceability
Quality management	measurement & tracking, verification & validation

- *Application engineering:* Domain requirements should be reused and external variability for specific application should be bound. As a result, application requirements are specified.

- *Asset:* Domain requirements engineering provides a common portion of requirements (domain requirements) to application engineering through asset repository. Therefore the domain assets of requirements engineering and management is a distinguished aspect.

- *Binding:* Binding in requirements engineering should consider to reflect the external variability. Thus, binding is a distinct aspect of the product line development.

- *Collaboration:* Since the domain requirements engineering and application requirements engineering can be performed in parallel, collaborations are necessary between engineering teams as well as those among processes such as domain assets, variability management, product management, scoping, etc. This makes collaboration as an important aspect in a product line development environment.

- *Configuration:* Configuration of products and artifacts of product line requirements engineering can be multidimensional, i.e., exist in time and space. Maintaining the integrity of those dimensions is an important aspect.

- *Domain architecture:* Domain requirements engineering provides domain requirements as a major input for establishing reference architecture.

- *Domain engineering:* Domain requirements engineering process does not exist in a single product development. However, this process is necessary for a product line.

- *Enabling technology support:* Technologies that are needed to enable product line requirements engineering are a key success factor of product line implementation.

- *Measurement & tracking:* In product line requirements engineering, data collection, measures, and tracking need to consider domain engineering, application engineering, product management, and domain assets. This means that the measurements for product line are multidimensional and thus the required activities, roles, procedures, tools, and methods should be considered.

- *Platform:* Platform enables to reuse common elements (e.g., artifacts, components, connectors, etc.) among products. Product line requirements engineering artifacts are key elements of a platform.

- *Product management:* Requirements engineering should deal with reusability of the product line from the foreseen product line strategy. Since requirements continuously evolve in accordance with risks and opportunities, product management should monitor and support the evolution of requirements.

- *Reusability:* The reusability of assets from requirements engineering processes is closely related to achieve the overall goal of a product line. Achieving reusability throughout the product line requirements engineering is a differentiating aspect.

- *Texture:* Product line requirements are major inputs for establishing texture. Especially, for reflecting functional and non-functional requirements, corresponding architectural texture shall be established.

- *Traceability:* There exist various kinds of trace links between variabilities and the artifacts from domain/application requirements engineering. It is necessary to develop a traceability scheme to handle tracing.

- *Validation and verification:* In product line development environments, provisioning of objective evidences for validation and verification of requirements from various viewpoints (e.g., domain, application, variability, domain assets, etc.) is an important aspect, so differentiated schemes with single product development should be provided.

- *Variability:* Variability in requirements engineering mainly deals with external variability related to a reusability strategy, which is not a concern of single product development.

5 Product Line Scoping

A product line organization needs to determine with which products and features will consist of a product line, and thereafter the organization determines which features will be implemented by reusing or adapting legacy assets or which of them will be newly developed. Using objective and quantitative endeavours the organization estimates economic benefits expected from a product line and makes a go/no-go decision for product lineinitiation base on the economic benefits. A product line scoping process includes three key sub processes:

- *Product scoping* determines the potential member products and initial common and variable features of those base on market inputs.
- *Domain scoping* decomposes domains into subdomains (or functional domains) and maps the initial features determined in product scoping to the subdomains.
- *Asset scoping* identifies reusable assets, calculates the cost/benefit estimates to justify the product line initiation, and provides a roadmap of a product line.

The above three types of scoping can be iterative.

5.1 Product scoping

Purpose of product scoping

The purpose of product scoping is to determine product portfolio definition:

1) The products that the product line organization should be developing, producing, marketing, and selling;

2) The common and variable features that the products should provide in satisfying customer needs and reaching the long and short term business objectives of the product line organization; and

3) A schedule for introducing the products to markets.