INTERNATIONAL STANDARD

ISO/IEC 26551

Second edition
2016-05-01

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 26551:2016(E)

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

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 on the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the WTO principles in the Technical Barriers to Trade (TBT) see the following URL: Foreword - Supplementary information

The committee responsible for this document is ISO/IEC JTC 1, *Information technology*, Subcommittee SC 7, *Software and systems engineering*.

This second edition of ISO/IEC 26551 cancels and replaces the first edition (ISO/IEC 26551:2012), which has been technically revised.

Introduction

The main purpose of this International Standard is to establish a baseline for 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 the software and systems product line specific requirements engineering processes.

A decision for the initial boundaries of domain is made to define a product line scope before initiating domain requirements engineering processes. Domain requirements engineering is 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 are to be considered (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 reasons:

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

A detailed comparison showing the differences in requirements engineeing tasks between single product and product line is described in Annex A.

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.

The ISO/IEC 26550 family of standards addresses both engineering and management processes and covers the key characteristics of product line development. The ISO/IEC 26550 family of standards provides an overview of the consecutive International Standards (i.e. this International Standard through ISO/IEC 26599), as well as the structure of the model:

ISO/IEC 26550, ISO/IEC 26551 and ISO/IEC 26555 are published. ISO/IEC 26557, ISO/IEC 26558 and ISO/IEC 26559 are to be published. ISO/IEC 26552, ISO/IEC 26553, ISO/IEC 26554, ISO/IEC 26556, ISO/IEC 26560, ISO/IEC 26561, ISO/IEC 26562, ISO/IEC 26563 are planned International Standards.

— Processes and capabilities of methods and tools for domain requirements engineering and application requirements engineering are provided in this International Standard;

— Processes and capabilities of methods and tools for domain design and application design are provided in ISO/IEC 26552 (International Standard under development);

- — Processes and capabilities of methods and tools for domain realization and application realization are provided in ISO/IEC 26553 (International Standard under development);
- — Processes and capabilities of methods and tools for domain testing and application testing are provided in ISO/IEC 26554 (International Standard under development);
- — 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 (International Standard under development);
- — 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 (International Standard under development);
- — Processes and capabilities of methods and tools for technical probe are provided in ISO/IEC 26561 (International Standard under development);
- — Processes and capabilities of methods and tools for transition management are provided in ISO/IEC 26562 (International Standard under development);
- — Processes and capabilities of methods and tools for configuration management of asset are provided in ISO/IEC 26563 (International Standard under development);
- — Others (ISO/IEC 26564 to ISO/IEC 26599): To be developed.

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

1 Scope

This International Standard, within the context of tools and methods of requirements engineering for software and systems product lines:

— provides the terms and definitions specific to requirements engineering for software and systems product lines and associated member products;

— defines process groups and their processes performed during product line requirements engineering (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 International Standard concerns processes and capabilities of requirements tools and methods for a family of products, not for a single system.

This International Standard is not applicable to physical artefacts. Instead, system-level artefacts and software lifecycle artefacts such as requirements documents, architectural data, validation plans, behavioural models, etc. are produced using methods and tools in this International Standard. In the case of the software components of a system, this International Standard can apply twice: once to handle the system elements of the product line and a second time to handle the software elements of the product line, if any. The product line processes are recursive within the different levels of products.

NOTE The requirements in this International Standard apply to the family of systems, software or services.

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.

3.1
application assets in requirements
application specific artefacts produced during application requirements engineering such as *application requirements specifications* (3.4) and application requirements models

3.2
application requirements elicitation
subprocess for identifying stakeholders relevant to an application, eliciting application specific requirements, and binding the appropriate variants

3.3
application requirements analysis
subprocess that understands all *application specific requirements* (3.8), scrutinizes incorrect and inconsistent application requirements through modelling, and then analyses and negotiates application requirements that cannot be satisfied through the domain requirements

3.4
application requirements specification
subprocess that documents the *application specific requirements* (3.8) and integrates it with the *domain requirements specification* (3.12) whose variants are bound

3.5
application requirements verification and validation
subprocess that confirms that the *application specific requirements* (3.8) are consistent and feasible and ensures that the bound variants satisfy the specific product's requirements

3.6
application requirements management
subprocess that manages traceability and changes on application requirements

3.7
aspect
special consideration within product line engineering process groups and tasks to which we can associate specialized methods and tools

3.8
asset proposal
artefact that includes major assets (functional areas and high-level common and variable features of all applications) that can be included in a product line with their quantified costs and benefits, and estimate results

3.9
application specific requirements
requirements specific to an application or requirements not covered in domain requirements

3.10
domain assets in requirements
reusable artefacts produced during domain requirements engineering such as *asset proposals* (3.7), *domain requirements specifications* (3.12), and domain requirements models

3.11
domain requirements elicitation
subprocess that identifies initial requirements from domain stakeholders for a product line

3.12
domain requirements analysis
subprocess that models domain requirements so as to analyse and scrutinize commonality/variability of a product line in requirements

3.13
domain requirements specification
subprocess that documents domain requirements for a product line based on domain analysis results

3.14
domain requirements verification and validation
subprocess that confirms that domain requirements are correct, consistent, and complete

3.15
domain requirements management
subprocess that manages traceability and changes with respect to domain requirements and their relevant domain/application artefacts

3.16
functional domain
categorized functions that are generally used together

3.17
production plan
description of how domain assets are to be used to develop member products in a product line

3.18
requirements traceability
traceabilities in domain and application requirements respectively and those between them

3.19
texture
architectural texture
collection of common development rules and constraints for realising the applications of a product line

3.20
variability in requirements
external and internal variability in requirements engineering

Note 1 to entry: Variability modelling and traceability with domain requirements artefacts are also 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 other product line engineering lifecycle processes in order to enable traceability between all requirements artefacts and the related design, realization, and testing artefacts. In the rest of this International Standard, 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;

— analyse domain requirements with variabilities in requirements;

— model and simulate the static and behavioural 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 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;

— analyse 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 technical management 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.

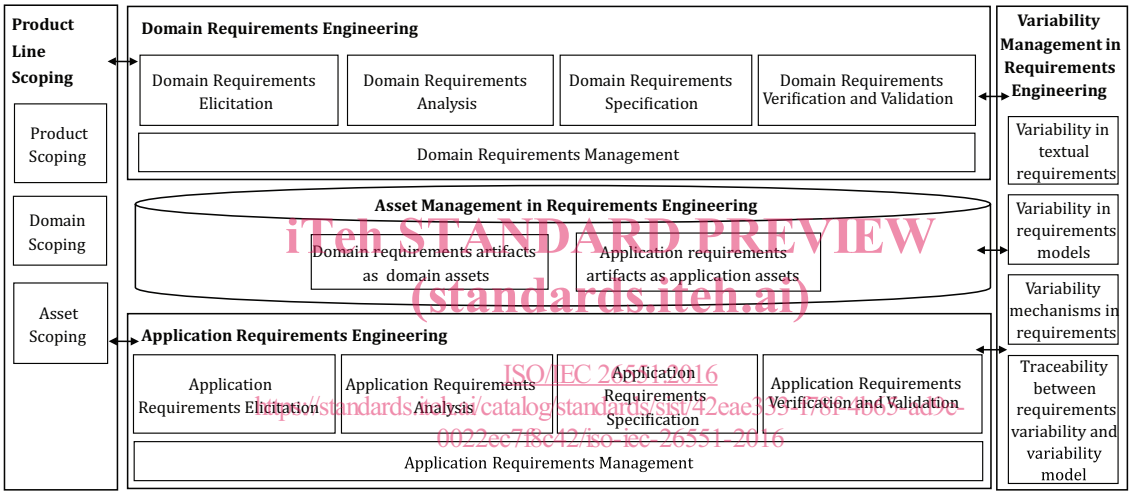

Figure 1 — Product line requirements engineering

Product line scoping defines the member products and their major (externally visible) features, analyses 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 proposal 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, estimates the cost/benefit 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 requirements;

— *Domain requirements analysis* identifies functional and non-functional requirements with the variabilities of those requirements;

— *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 modelling starts as part of domain requirements elicitation 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 modelling language and makes them explicit;

— *Variability mechanisms in requirements* categorize variability mechanisms that are to be used as part of requirements;

— *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 artefacts 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 artefacts 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 is also analysed 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 artefacts* as domain assets identify and develop necessary information to help application engineers reuse requirements assets in their application development;

— *Application requirements artefacts* as application assets identify and manage application requirements artefacts 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 can also 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. 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 analysed 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.

NOTE 1 Processes of product line requirements engineering are compatible with ISO/IEC 12207 and ISO/IEC/IEEE 15288 as depicted in process mapping with ISO/IEC 12207, ISO/IEC/IEEE 15288, and ISO/IEC 29148 (Annex B).

The identification and analysis of the aspects for the product line requirements engineering 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:

NOTE 2 The relationship between sub processes, key aspects, and associated tool and method capabilities is depicted in Annex C.

Table 1 shows the key aspects for each characteristic of product line requirements engineering.

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

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 and tracking, cross functional verification and 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 reflecting 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 an important aspect in a product line development environment;

- *Configuration:* Configuration of products and artefacts 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 and 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. artefacts, components, connectors, etc.) among products. Product line requirements engineering artefacts 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. Particularly for developing rules and specific detail of functional and non-functional requirements, corresponding architectural texture have to be established;
- *Traceability:* There exist various kinds of trace links between variabilities and the artefacts 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 constitute a product line, and thereafter the organization determines which features are implemented by reusing or adapting legacy assets or which of them are 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 line initiation 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;