

---

---

**Software and systems engineering —  
Methods and tools for product line  
configuration management**

*Ingénierie du logiciel et des systèmes — Méthodes et outils pour la  
gestion des configurations de gammes de produits*





## **COPYRIGHT PROTECTED DOCUMENT**

© ISO/IEC 2022

All rights reserved. Unless otherwise specified, or required in the context of its implementation, no part of this publication may be reproduced or utilized otherwise in any form or by any means, electronic or mechanical, including photocopying, or posting on the internet or an intranet, without prior written permission. Permission can be requested from either ISO at the address below or ISO's member body in the country of the requester.

ISO copyright office  
CP 401 • Ch. de Blandonnet 8  
CH-1214 Vernier, Geneva  
Phone: +41 22 749 01 11  
Email: [copyright@iso.org](mailto:copyright@iso.org)  
Website: [www.iso.org](http://www.iso.org)

Published in Switzerland

# Contents

Page

<b>Foreword</b>	<b>v</b>
<b>Introduction</b>	<b>vi</b>
<b>1 Scope</b>	<b>1</b>
<b>2 Normative references</b>	<b>1</b>
<b>3 Terms and definitions</b>	<b>1</b>
<b>4 Abbreviated terms</b>	<b>2</b>
<b>5 Reference model for product line configuration management</b>	<b>3</b>
5.1 Overview	3
5.2 Constituents of reference model for product line configuration management	3
<b>6 Product line CM technical management</b>	<b>7</b>
6.1 General	7
6.2 Product line configuration planning	7
6.2.1 Principal constituents	7
6.2.2 Define PL configuration strategy	8
6.2.3 Assign responsibility for PL configuration operationalization	8
6.2.4 Define success measures for PL configuration operationalization	8
6.2.5 Estimate adequate resources needed for PL configuration operationalization	9
6.2.6 Document PL configuration plans	9
6.3 Product line configuration enabling	10
6.3.1 Principal constituents	10
6.3.2 Enable PL configuration environment	10
6.3.3 Provide guidance for PL configuration operationalization	11
6.3.4 Enable measurement environment for quantifying PL configuration operationalization	11
6.4 Product line configuration managing	11
6.4.1 Principal constituents	11
6.4.2 Review PL configuration operationalization status with success measures	12
6.4.3 Control issues on PL configuration operationalization	12
6.4.4 Make corrective actions on PL configuration operationalization	13
<b>7 Product line configuration management operationalization</b>	<b>13</b>
7.1 General	13
7.2 Product line configuration management initiation	13
7.2.1 Principal constituents	13
7.2.2 Initiate PL configuration operation	14
7.2.3 Perform preliminary PL configuration	14
7.3 Product line configuration item identification	14
7.3.1 Principal constituents	14
7.3.2 Select PL configuration items	15
7.3.3 Identify variations of configuration items in time and space	16
7.3.4 Structure configuration information aligned to the structure of a product line	16
7.3.5 Establish unique identifiers facilitating traceability	17
7.4 Product line configuration baselining	17
7.4.1 Principal constituents	17
7.4.2 Capture PL configuration baselines in time and space	18
7.4.3 Create and release PL configuration baselines in time and space	18
7.4.4 Facilitate commonality and variability management in baselines of time and space	19
7.4.5 Maintain baselines readily available by distributed configuration management	19
7.5 Product line configuration status accounting	20

7.5.1	Principal constituents .....	20
7.5.2	Evaluate and coordinate change requests for PL configuration items .....	21
7.5.3	Identify PL configuration delta .....	21
7.5.4	Update baselines for PL configuration delta .....	21
7.5.5	Relate PL configuration delta to relevant PL configuration items .....	22
7.6	Product line configuration reporting .....	22
7.6.1	Principal constituents .....	22
7.6.2	Develop and maintain the PL configuration management status information .....	23
7.6.3	Capture, store, and report PL configuration anomaly .....	23
7.6.4	Capture, store and report PL configuration management data .....	23
<b>8</b>	<b>Product line configuration management support .....</b>	<b>23</b>
8.1	General .....	23
8.2	Quality assurance for product line CM .....	24
8.2.1	Principal constituents .....	24
8.2.2	Objectively evaluate PL configuration activities .....	25
8.2.3	Objectively evaluate PL configuration work products .....	25
8.2.4	Communicate and resolve non-compliance issues .....	25
8.2.5	Establish records of quality assurance activities for PL configuration .....	26
8.3	Product line configuration change control .....	26
8.3.1	Principal constituents .....	26
8.3.2	Identify and record change requests for product line configuration .....	27
8.3.3	Evaluate and coordinate the impacts of change requests on commonality and variability .....	27
8.3.4	Verify conformance among relevant configurations in time and space .....	27
8.3.5	Trace and manage approved changes to baselines .....	28
8.4	Traceability management for product line configuration .....	28
8.4.1	Principal constituents .....	28
8.4.2	Establish bi-directional traceability among PL configurations .....	29
8.4.3	Track and control PL configuration with commonality and variability .....	29
8.4.4	Facilitate integrated traceability management for PL configuration and PL configuration delta .....	30
	<b>Annex A (informative) Roles of configuration management in each SSPL lifecycle process .....</b>	<b>31</b>
	<b>Bibliography .....</b>	<b>33</b>

## 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.

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](http://www.iso.org/directives) or [www.iec.ch/members\\_experts/refdocs](http://www.iec.ch/members_experts/refdocs)).

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](http://www.iso.org/patents)) or the IEC list of patent declarations received (see <https://patents.iec.ch>).

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](http://www.iso.org/iso/foreword.html). In the IEC, see [www.iec.ch/understanding-standards](http://www.iec.ch/understanding-standards).

This document was prepared by Joint 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](http://www.iso.org/members.html) and [www.iec.ch/national-committees](http://www.iec.ch/national-committees).

## Introduction

Software and systems product line (SSPL) engineering and management creates, exploits, and manages a common platform to develop a family of products (e.g. software products, systems architectures) at a lower cost, with reduced time to market and better quality. As a result, it has gained increasing global attention since the 1990s.

SSPL engineering and management maintain changes of commonality and variability of a product line as well as multiple different products derived from the product line. The impacts of the changes in commonality and variability are on each product of a product line, and they are different for each product. Thus, the complexity of configuration management in SSPL is high, and configuration management for a single system that is not aware of these aspects cannot manage configurations of a product line. This document deals with configuration management methods and tools that are aware of these aspects.

This document can be used in the following modes:

- to provide guidance on how to identify, control, report, and evaluate configurations of a product line by organizations that want to adopt SSPL for producing their products;
- to provide guidance on the evaluation and selection for methods and tools for product line configuration management by a product line organization;
- to provide guidance on implementing or developing methods and/or tools by specifying a comprehensive set of methods and tools capabilities for supporting product line configuration management by either providers of methods or tools, or both.

The ISO/IEC 26550 family of standards addresses both engineering and management processes and capabilities of methods and tools in terms of the critical characteristics of product line development. This document provides processes and capabilities of methods and tools for configuration management in product lines.

Other standards in the ISO/IEC 26550 family are as follows: ISO/IEC 26550, ISO/IEC 26551, ISO/IEC 26552, ISO/IEC 26553, ISO/IEC 26554, ISO/IEC 26555, ISO/IEC 26556, ISO/IEC 26557, ISO/IEC 26558, ISO/IEC 26559, ISO/IEC 26560, ISO/IEC 26561, ISO/IEC 26562 and ISO/IEC 26564.

- 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 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 product line transition management are provided in ISO/IEC 26562.
- Processes and capabilities of methods and tools for product line measurement are provided in ISO/IEC 26564.





# Software and systems engineering — Methods and tools for product line configuration management

## 1 Scope

This document, within the context of methods and tools that support the configuration management (CM) capability of software and systems product line engineering:

- specifies processes for product line CM (the processes are described in terms of purpose, inputs, tasks, and outcomes);
- specifies method capabilities to support the defined tasks of each process;
- specifies tool capabilities that automate or semi-automate tasks and methods.

This document does not concern the 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 terminology databases for use in standardization at the following addresses:

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

### 3.1

#### **application asset**

output of a specific application engineering process (e.g. application realization) that may be exploited in other lifecycle processes of application engineering and may be adapted as a *domain asset* (3.3) based on a product management decision

[SOURCE: ISO/IEC 26550:2015, 3.2, modified — Notes to entry have been removed.]

### 3.2

#### **common configuration item**

entity within a configuration that is common to all products

### 3.3

#### **domain asset**

output of domain engineering life cycle processes and can be reused in producing products during application engineering

[SOURCE: ISO/IEC 26550:2015, 3.11, modified — The alternative preferred term "core asset" and notes to entry have been removed.]

**3.4**

**product line configuration**

snapshot of the product line that contains a collection of revisions of every *domain asset* (3.3) and member product in the product line

**3.5**

**product line configuration delta**

difference between the two versions of *product line configuration* (3.4)

**3.6**

**product line configuration item**

resulting artefacts that make up a product line

Note 1 to entry: Product line configuration item is a generic term for configuration items of domain engineering, application engineering, and managerial support.

**3.7**

**product line configuration management**

coordinated activities to direct and control *product line configuration* (3.4)

Note 1 to entry: Product line configuration management includes configuration management activities for domain engineering, application engineering, and organization and technical management.

[SOURCE: ISO/IEC TR 18018:2010, 3.7, modified — "product line" has been added in the term and in the definition; the abbreviated term has been removed; note 1 to entry has been added.]

**3.8**

**variation in time**

existence of different versions of a common or variable artefact at different times

Note 1 to entry: The variation in time dimension is synchronous with software evolution[24].

**3.9**

**variable configuration item**

entity within a configuration that is variable among products

**3.10**

**variation in space**

existence of different versions of a variable artefact in different shapes and used by different products

**3.11**

**variability model**

explicit definition for product line variability

[SOURCE: ISO/IEC 26550:2015, 3.27, modified — Note 1 to entry has been removed.]

## **4 Abbreviated terms**

CM	configuration management
ConOps	concept of operations
PL	product line
SSPL	software and systems product line

## 5 Reference model for product line configuration management

### 5.1 Overview

CM is an essential prerequisite for being able to cope with the complexity of products in different versions. SSPL consists of many common and variable parts in different versions, so the complexity of CM is very high. Therefore, sophisticated CM is essential to succeeding in SSPL engineering and management.

In SSPL, CM is performed for domain engineering and application engineering, unlike CM for single system development. At any moment, some products use different versions of reusable domain assets. Furthermore, adaptations, additions, or deletions of commonality and variability make CM more complex.

Like CM for single system development, product line CM should support the management of configuration items, versions, branches, baselines, and branched baselines. It also should support change control and release control. However, product line CM should support:

- a) management of snapshots of common and variable domain assets uniquely identified at a given reference point so that member products reuse them consistently called variants in time;
- b) management of artefacts for a variation point and its variants, namely different snapshots of artefacts that can replace the variation point called variants in space;
- c) management of snapshots of a member product, including reused domain assets and product-specific artefacts.

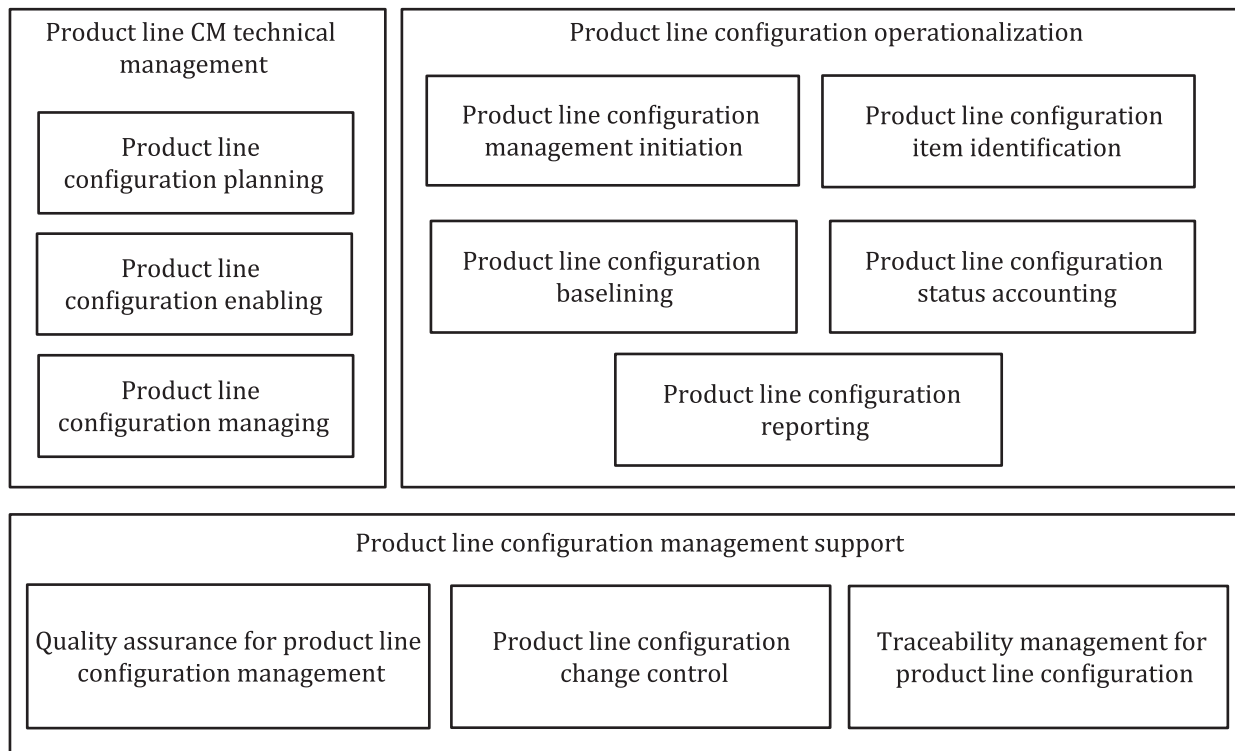
The product line CM process and its supporting methods and tools should have capabilities to maintain the integrity of the configuration items of software and systems produced during product line engineering and management.

### 5.2 Constituents of reference model for product line configuration management

The reference model specifies the structure of supporting processes and subprocesses for a product line CM. [Figure 1](#) shows that a product line CM is structured into three processes: product line CM, PL configuration operationalization, and product line CM support. In the rest of this document, tasks, methods, and tools are described for processes and subprocesses defined in the reference model.

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.



**Figure 1 — Reference model for the product line configuration management**

The product line CM technical management process provides managerial supports for planning PL configuration (e.g. CM strategy, resource estimation, responsibility allocation, success measures), supports for providing necessary resources, tools, and infrastructures for realizing CM plans and supports for analysing the plan versus the actual status of CM. The product line CM technical management includes the following:

- product line configuration planning establishes plans for initiating, operationalizing and supporting product line CM; PL configuration plan contains PL configuration strategy, CM responsibilities and authorities, criteria for identifying configuration items, applicable policies, PL-specific CM process, rules for change and release control, success measures to judge the success of CM, schedule, tools, and resources;
- product line configuration enabling defines, maintains, and assures the availability of environments, guidance, and measurement necessary to performing product line CM; product line CM environments should support coordination and cooperation among participants of domain engineering and application engineering processes so that participants of domain engineering can do their roles and responsibilities without interfering with the progress of application engineering and vice versa; the environments should clarify whether member products can be allowed to change domain configuration items they are reusing; guidance for stabilizing and making changes on configuration items of domain engineering is essential to preventing risks due to the reuse of inconsistent or destabilized versions of domain configuration items;
- product line configuration managing provides integrated management for product line configuration operationalization; this subprocess reviews the product line configuration operationalization's actual status against plans, controls issues, and takes corrective actions if necessary.

The PL configuration operationalization process performs operations for establishing and maintaining configurations of product line engineering and management. This process deals with the identification of PL configuration items, baselining, change and release control of PL configuration items and

baselines, and managing for commonality and variability to apply and undo PL configuration changes. The PL configuration operationalization includes the following:

- product line CM initialization identifies and mobilizes PL configuration participants to initiate the integrated PL configuration operation; Initialization includes the selection of mechanisms and tools for the handling configurations of all PL engineering artefacts;
- product line configuration item identification identifies the software and system items managed under the product line CM; configuration items include common, variable, and product-specific items of software and systems;
- product line configuration baselining creates baselines of product line configurations with configuration descriptions; product line configuration includes configuration of domain assets, including commonality and variability and configuration of an individual product;
- product line configuration status accounting manages differences of one version of a product line from another; deltas between PL configurations are complex because changes and evolutions occur at multiple dimensions such as domain asset changes and evolutions in time, domain asset variations in space, and changes and evolutions in member products; methods and tools should support correct and complete propagation of configuration deltas of domain assets across all member products;
- product line configuration reporting records and reports the status of product line configuration items and change requests. This subprocess reviews PL configuration activities for validating the correctness and completeness of status of PL configurations and maintaining consistency among the common and variable domain assets by ensuring that domain assets are in an appropriate state throughout the entire product line lifecycle.

The product line CM support process provides supports required for coping with the complexity of product line CM in change and traceability management and for assuring the quality of complex product line CM technical management activities. To achieve these, the product line CM support includes the following:

- quality assurance for product line CM objectively evaluates the activities of product line CM that they adhere to the guidance, release and change control rules, and defined CM process; and this subprocess assures that PL configuration items, versions, baselines, and releases as the results of product line CM activities are correct, complete, and consistent;
- product line configuration change control evaluates and manages change requests of common and variable domain assets in time and space and changes performed on each member product specifically;
- traceability management for product line configuration establishes and manages traces among different but relevant configurations of a product line to ensure the integrity of configurations of a product line; traceability of the product line configuration should be of a degree that can support recoverability.

Identifying and analysing the key differentiators between single-system engineering and management and product line engineering and management can help the organizations understand the product line and formulate a strategy for the successful implementation of product line engineering and management. The key aspects are defined in ISO/IEC 26550; and [Table 1](#) shows the category of the key aspects.

**Table 1 — Key aspects for identifying product line CM tasks**

Category	Aspects
Reuse management	application engineering, domain assets, domain engineering, product management, platform, reusability
Variability management	binding, variability

**Table 1** (continued)

Category	Aspects
Complexity management	collaboration, configuration, enabling technology support, reference architecture, texture, traceability
Quality management	measurement and tracking, cross-functional verification and validation

The following is the description for each aspect concerning product line CM. The product line CM processes and tasks shall be identified based on these aspects. The product line CM's concerns are to enable the organization to understand CM processes, subprocesses, tasks, and methods' and tools' capabilities.

- Application engineering: Application engineering produces configuration items of individual products; but unlike single system engineering, several planned products are controlled in the product line CM.
- Binding: Binding times of variabilities can affect the complexity of product line CM.
- Collaboration: Product line organization units responsible for producing configuration items are diverse, so product line CM should collaborate with multiple different organization units such as domain architects, core asset developers, and developers of individual products.
- Configuration: Configurations of a product line include a common platform, variability model, and individual products of a product line.
- Domain asset: Common platform, which is one of the major configurations of a product line, consists of a set of domain assets and variability models. Domain assets are major configuration items of a product line.
- Domain engineering: Domain engineering processes produce domain assets, which are major configuration items of a product line.
- Enabling technology support: Product line CM-enabling supports should provide methods and tools for CM of variabilities and manage traceabilities among common platform, variability model, and individual products.
- Measurement and tracking: The efficiency of product line CM should be measured and improved by tracking product line CM practices' status.
- Platform: Product line CM should manage a common platform and carefully consider the change effects on it.
- Product management: Work products of product management can be configuration items of a product line.
- Reference architecture: Reference architecture is one of the configuration items; and it is used as a basis to identify configuration items and configurations of a product line.
- Reusability: Product line CM enables potential reusability of a common platform and variability model of a product line.
- Texture: The texture of a product line architecture is used to identify configuration items and configurations of a product line.
- Traceability: Many-to-many long-term relationships among domain assets, application assets, and products should be maintained and traced.
- Cross-functional validation and verification: Different aspects of a Product line CM should be validated and verified.
- Variability: Product line CM should support variabilities and their possible individual products.

## 6 Product line CM technical management

### 6.1 General

The product line CM technical management supports the following:

- a) product line configuration planning;
- b) product line configuration enabling;
- c) product line configuration managing.

### 6.2 Product line configuration planning

#### 6.2.1 Principal constituents

##### 6.2.1.1 Purpose

The purpose of this subprocess is to produce and coordinate practical and workable PL configuration plans to manage and control product line elements and configurations over the whole product line lifecycle processes.

NOTE [Annex A](#) discusses the roles of configuration management in product line lifecycle processes.

Product line configuration manages consistency between a product line, its associated product line level configuration definition, and its products associated with each product level configuration definition.

Major CM subjects include variability, a reusable common platform, and individual member products of a product line with their traceability for change management and version control. Every branch and propagation of changes to that reusable common platform and variabilities to all affected branches should be managed to assure their consistency.

##### 6.2.1.2 Inputs

The following inputs should be available to perform the process:

- a) PL business objectives;
- b) PL transition plan (from ISO/IEC 26562);
- c) organizational standard PL processes.

##### 6.2.1.3 Outcomes

The following outcome shall be available as a result of the successful implementation of the product line configuration planning process: Product line configuration management, including the scope, goals, and strategies of product line CM, outcomes of major CM activities, and required resources, is planned.

##### 6.2.1.4 Tasks

The organization shall implement the following tasks with respect to the process:

- a) Define product line configuration strategy: Establish an organizational strategy that guides CM operations in domain engineering, application engineering, and PL management supports. This task defines/tailors processes for managing the configuration of a product line to be developed.
- b) Assign responsibility for PL configuration operationalization: Defines roles, responsibilities, general principles, and management expectations for performing PL configuration management.



- c) Define success measures for PL configuration operationalization: Define measures including quality criteria used for quality evaluation of product line CM processes and their outcomes.
- d) Estimate adequate resources needed for PL configuration operationalization: Identify resources, efforts, schedule, supporting technologies, and costs for managing the configuration of a product line to be developed.
- e) Document PL configuration plans: Consolidate, review, approve, and document the results of PL configuration planning and acquire formal commitment for the required resources and supporting technologies. Plans include CM responsibilities and authorities, criteria for identifying configuration items, applicable policies, PL-specific CM process, schedule, tools, and resources.

#### **6.2.2 Define PL configuration strategy**

- a) The method should support the task with the following capabilities:
  - 1) understanding product line context, PL objectives, and PL transition strategies;
  - 2) analysing the organizational capability for CM in single system development and CM operations of a previous product line;
  - 3) establishing a PL configuration strategy, including success factors of a product line concerning CM and strategy evaluation measures.
- b) A tool should support the task by allowing the user to:
  - 1) access product line context, PL objectives, and PL transition strategies;
  - 2) communicate about alternative strategies with participants for selecting crucial components of a PL configuration strategy;
  - 3) edit PL configuration strategy and integrate them with the PL configuration plan.

#### **6.2.3 Assign responsibility for PL configuration operationalization**

- a) The method should support the task with the following capabilities:
  - 1) analysing critical roles, responsibilities and general principles, and management expectations for performing PL configuration management;
  - 2) collating required roles, responsibilities and general principles, and management expectations with their candidate roles and responsibilities of the PL organization structure;
  - 3) analysing trade-offs among candidate roles and responsibilities in a PL.
- b) A tool should support the task by allowing the user to:
  - 1) access the PL organization structure;
  - 2) distribute defined roles, responsibilities, and general principles to relevant participants of a PL.

#### **6.2.4 Define success measures for PL configuration operationalization**

- a) The method should support the task with the following capabilities:
  - 1) defining success measures that can tie PL configuration operations to success factors;
  - 2) defining formulas of success measures covering associated measures of multiple CM roles;
  - 3) identifying actionable product line management processes and tying them to success measures as well as domain and application engineering outcomes closely related to success measures;



- 4) establishing tracking mechanisms to decide the values of success measures;
- 5) refining indicators, measures, and data elements required based on success measures for monitoring and controlling PL configuration operations;
- 6) defining relations among indicators and measures for reinforcing them as those agreed on PL configuration operations' success measures.

EXAMPLE PL-specific measures that indicate improper configuration and change control to measure the success of PL configuration management:

- number of changes for common domain assets that are not adequately propagated to member products;
- number of changes for variability in time that are not adequately propagated to member products;
- number of changes for variability in space that are not adequately propagated to member products;
- percentage of change requests for domain assets that were completed on time without unexpected incidents or problems.

- b) A tool should support the task by allowing the user to:
  - 1) access the overall objectives of a product line, strategies, and processes;
  - 2) decide the values of success measures through established tracking mechanisms of overall PL.

#### 6.2.5 Estimate adequate resources needed for PL configuration operationalization

- a) The method should support the task with the following capabilities:
  - 1) providing factors that should be considered for performing the estimation of a PL configuration management;
  - 2) analysing relations among factors using historical data of PL configuration management or of products built before adopting PL.
- b) A tool should support the task by allowing the user to:
  - 1) access historical data for managing configuration of the previous PL and appropriate single products;
  - 2) (semi-)automatically calculate estimation results;
  - 3) store estimation results for the further improvement of estimation for the next product line.

#### 6.2.6 Document PL configuration plans

- a) The method should support the task with the following capabilities:
  - 1) defining the documentation standard for a PL configuration plan (the content of a plan includes primary stakeholders of a PL configuration, a PL configuration strategy, PL configuration processes to be applied, and estimates, staffing including required training);
  - 2) providing a template and examples of PL configuration plan that guides the development of a PL-specific configuration strategy, processes.
- b) A tool should support the task by allowing the user to:
  - 1) edit/fill out a PL configuration plan using an editable and changeable template;
  - 2) maintain the version control of the PL configuration plan baseline;

- 3) share the PL configuration plan with stakeholders.

### **6.3 Product line configuration enabling**

#### **6.3.1 Principal constituents**

##### **6.3.1.1 Purpose**

The purpose of this subprocess is to ensure the product line organization's capability to accomplish product line CM goals by providing the required environments and resources necessary to support and operationalize PL configuration management.

This subprocess ensures that certification and accreditation of PL supporting systems incorporate an evaluation and testing of PL configuration strategy and plans.

##### **6.3.1.2 Inputs**

The following inputs should be available to perform the process:

- a) PL configuration plan;
- b) enabled PL development systems (for analysis, specification, modelling, design, implementation, and testing).

##### **6.3.1.3 Outcomes**

The following outcomes shall be available as a result of the successful implementation of the process:

- a) product line configuration environments and resources are enabled;
- b) guidance for PL configuration operationalization is defined.

##### **6.3.1.4 Tasks**

The organization shall implement the following tasks with respect to the process:

- a) Enable PL configuration environment: Provide infrastructure and human resources necessary to execute the PL configuration plan.
- b) Provide guidance for PL configuration operationalization: Define guidance for addressing CM operations at both domain and application engineering processes.
- c) Enable measurement environment for quantifying PL configuration operationalization: Ensure a product line organization's capability to perform CM operationalization.

#### **6.3.2 Enable PL configuration environment**

- a) The method should support the task with the following capabilities:
  - 1) estimating required infrastructures and human resources with essential capabilities based on PL configuration plans. Estimation can be done using historical data, product line community benchmark, or divide and conquer;

- 2) mobilizing estimated required infrastructures and human resources. Acquisition for PL configuration operationalization (e.g. integrated CM for domain assets, application assets, variability model, or CM for one of them) and capability development performed if necessary.
- b) A tool should support the task by allowing the user to:
  - 1) access information necessary for estimating required infrastructures and human resources with essential capabilities based on PL configuration plans;
  - 2) integrate acquired resources with those in-houses.

### 6.3.3 Provide guidance for PL configuration operationalization

- a) The method should support the task with the following capabilities:
  - 1) defining what, why, when, and who for product line CM in organizational management, domain and application engineering, asset management, and technical management aspects;
  - 2) developing exemplary PL configuration operations in each aspect, including the way how different aspects are linked.
- b) A tool should support the task by allowing the user to:
  - 1) access a product line context, CM knowledge and experiences in both product line engineering and single product development;
  - 2) edit guidance and share it for further PL configuration operationalization.

### 6.3.4 Enable measurement environment for quantifying PL configuration operationalization

- a) The method should support the task with the following capabilities:
  - 1) refining actionable measurement tasks for quantifying PL configuration operations;
  - 2) establishing measurement mechanisms for quantifying PL configuration operations.
- b) A tool should support the task by allowing the user to:
  - 1) access best practices or benchmarks of quantifying CM operations of a product line;
  - 2) accrue measured data from PL configuration operations;
  - 3) connect a stable measurement environment with communication channels to share CM issues concerning quantification and resolve conflicts.

## 6.4 Product line configuration managing

### 6.4.1 Principal constituents

#### 6.4.1.1 Purpose

The purpose of this subprocess is to monitor, control, and improve PL configuration planning, enabling, operationalization, and support.

#### 6.4.1.2 Inputs

The following inputs should be available to perform the process:

- a) PL configuration plans;
- b) enabled product line CM environments;

- c) data elements collected from product line CM responsibilities and authorities.

#### **6.4.1.3 Outcomes**

The following outcome shall be available as a result of the successful implementation of the process: Action items for appropriate corrective actions and improvements of PL configuration activities are identified and performed.

#### **6.4.1.4 Tasks**

The organization shall implement the following tasks with respect to the process:

- a) Review PL configuration operationalization status with success measures: Integrate planned data elements from CM activities to determine success measures' value.
- b) Control issues on PL configuration operationalization: Identify root causes of successful CM operations obstacles and determine appropriate corrective actions to achieve product line CM quality goals and their consequent achievement of product line objectives.
- c) Make corrective actions on PL configuration operationalization: Enable corrective or improvement actions on PL configuration operations, track the results of corrective actions, and record lessons learned.

#### **6.4.2 Review PL configuration operationalization status with success measures**

- a) The method should support the task with the following capabilities:
  - 1) defining data elements and their collection points (source of data elements required) of multiple CM roles;
  - 2) ensuring that data elements and collection points are well-harmonized across multiple CM roles;
  - 3) defining indicators of success measures (e.g. effort variance when a success measure is cost reduction);
  - 4) estimating baselines of the defined indicators;
  - 5) representing the status of PL configuration operations with the indicators based on baselines versus actual values.
- b) A tool should support the task by allowing the user to:
  - 1) collect required data from multiple CM roles and calculate formulas of success measures (semi-) automatically;
  - 2) generate and express degrees of success of PL configuration operations graphically;
  - 3) share degrees of success of PL configuration operations with the relevant product line, domain, and application roles.

#### **6.4.3 Control issues on PL configuration operationalization**

- a) The method should support the task with the following capability: identifying CM roles of a product line that should be controlled after finding control issues on PL configuration operationalization based on degrees of success.
- b) A tool should support the task by allowing the user to notify un-harmonized CM operations of multiple CM roles of a product line to take corrective actions.

#### 6.4.4 Make corrective actions on PL configuration operationalization

- a) The method should support the task with the following capabilities:
  - 1) analysing root causes of the gaps between plan versus actual;
  - 2) deriving improvement needs and action items for PL configuration operations;
  - 3) decomposing and assigning refined roles and responsibilities (i.e. product line engineers, domain engineers, application engineers, or configurators) to the relevant product line product management, domain engineering, and application engineering participants of CM operations.
- b) A tool should support the task by allowing the user to:
  - 1) find roles related to action items for the assignment;
  - 2) let relevant participants know their roles and responsibilities for delivering corrective actions;
  - 3) share relations of action items with related roles.

## 7 Product line configuration management operationalization

### 7.1 General

The PL configuration operationalization supports the following:

- a) product line configuration management initiation;
- b) product line configuration identification;
- c) product line configuration baselining;
- d) product line configuration status accounting;
- e) product line configuration reporting.

### 7.2 Product line configuration management initiation

#### 7.2.1 Principal constituents

##### 7.2.1.1 Purpose

The purpose of this subprocess is to execute and check PL configuration setting by considering PL configuration strategy and plan. This subprocess ensures that the enabled PL configuration settings are adequate for a specific PL to be developed.

##### 7.2.1.2 Inputs

The following inputs should be available to perform the process:

- a) PL configuration plans;
- b) guidance for PL configuration operationalization;
- c) enabled PL configuration environments and resources.

### 7.2.1.3 Outcomes

The following outcome shall be available as a result of the successful implementation of the process: Revised PL configuration settings are initiated.

NOTE PL configuration settings include guidance, PL configuration environments, and PL configuration resources.

### 7.2.1.4 Tasks

The organization shall implement the following tasks with respect to the process:

- a) Initiate PL configuration operation: Integrate and implement for starting enabled PL configuration settings for recording, tracking, and reporting of configuration management operations within and across PLs. A product line consists of domain assets and member products reusing the domain assets. PLs include one product line and its evolution versions.
- b) Perform preliminary PL configuration: Start PL configuration procedures with initiated PL configuration settings. As a result, this task improves PL configuration enablers inadequate for a specific PL to be developed.

## 7.2.2 Initiate PL configuration operation

- a) The method should support the task with the following capability: Integrating PL configuration settings for recording, tracking, and reporting of configuration management operations within and across PLs.
- b) A tool should support the task by allowing the user to:
  - 1) interconnect support tools of enabled PL configuration settings for PL configuration management;
  - 2) check and change the interconnected configuration settings for PL configuration operation.

## 7.2.3 Perform preliminary PL configuration

- a) The method should support the task with the following capabilities:
  - 1) ensuring that configuration settings for a PL cover the PL configuration strategy and plan;
  - 2) finding inadequate PL configuration settings and recommending actions for improving PL configuration enablers.
- b) A tool should support the task by allowing the user to record and share PL configuration settings that require improvements.

## 7.3 Product line configuration item identification

### 7.3.1 Principal constituents

#### 7.3.1.1 Purpose

The purpose of this subprocess is to determine PL configuration items that are subject to configuration control.

PL configuration items are PL element contents: domain and application requirements, domain and application designs, domain and application code, domain and application test cases, traceabilities between elements of domain, application and both of them, PL process specifications, and PL operational concepts (ConOps). PL configuration items are common or variable elements among member products.

Therefore, PL configuration items, which are common elements reused by all member products and planned variable elements shared by parts of member products, are identified to be managed.

PL configuration items vary in time and space. Configuration items that vary in time are common or variable artefacts changed over time, whereas configuration items in space are variable artefacts in different shapes used simultaneously by different products. Managing configurations of variations in space are essential in SSPL engineering to build similar products but differ within a defined scope of a product line.

#### **7.3.1.2 Inputs**

The following inputs should be available to perform the process:

- a) work products of organizational management processes;
- b) work products of technical management processes;
- c) work products of domain engineering processes (including domain variability model);
- d) work products of application engineering processes (including application variability model).

#### **7.3.1.3 Outcomes**

The following outcomes shall be available as a result of the successful implementation of the process:

- a) PL configuration items in time and space assigned with unique ID are selected.
- b) Information on PL configuration items is structured.

#### **7.3.1.4 Tasks**

The organization shall implement the following tasks with respect to the process:

- a) Select PL configuration items: Determine configuration items that member products of a product line reuse, so their propagation of products' changes should be traced and managed (select domain asset elements to be controlled under configuration management as configuration items). Configuration items of a member product are selected with the same way of single system development.
- b) Identify variations of configuration items in time and space: Discriminate variant configuration items of common and variable configuration items. Common configuration items vary in time, whereas, variable configuration items differ in space as well as in time.
- c) Structure configuration information aligned to the structure of a product line: Specify configuration information and establish the structure and hierarchy of configuration information based on selected configuration items, including their definition and operational data based on variability model, product line architecture, and individual product's architecture.
- d) Establish unique identifiers facilitating traceability: Give unique identifiers distinguishing common and variable configuration items to trace configuration items.

### **7.3.2 Select PL configuration items**

- a) The method should support the task with the following capabilities:
  - 1) capturing PL elements from organization and technical management, domain engineering, and application engineering managed as configuration items;



NOTE In particular, domain assets among PL elements are captured by differentiating their variation types (i.e. variation in time and variation in space); application assets are captured separately for each member product; elements of organization and technical management involve documents such as PL transition or evolution plans and PL ConOps.

- 2) capturing configuration items from subcontractors, some of them are common to all member products and some others are variable or application-specific;
  - 3) capturing PL elements' operation data (e.g. history of change in time and space, version number, number of change requests);
  - 4) identifying environment configuration, some of them are common to all member products, and some others are variable or application-specific.
- b) A tool should support the task by allowing the user to:
- 1) describe (document) configuration information (using templates or forms) of each configuration item;
- EXAMPLE Configuration information includes configuration identification number, link to other configuration items, link to software structure, link to baseline, link to version hierarchy, link to the storage, owner, creation date, version number, and configuration status (ISO/IEC TR 18018).
- 2) track a configuration item through its complete lifecycle (from a requirement to eventual deployment as part of a release);
  - 3) link change requests associated with the configuration item;
  - 4) specify status (e.g. checked-in, checked-out, the status of a change) of configuration items.

### 7.3.3 Identify variations of configuration items in time and space

- a) The method should support the task with the following capabilities:
- 1) differentiating variations of configuration items in space from others;
  - 2) differentiating variations of configuration items in time from others;
  - 3) capturing different versions of application asset elements;
  - 4) capturing operation data of application asset elements (e.g. history of change, version number, number of change requests).
- b) A tool should support the task by allowing the user to describe each configuration item.

### 7.3.4 Structure configuration information aligned to the structure of a product line

- a) The method should support the task with the following capabilities:
- 1) capturing essential configuration item relationships that should be aligned to the structure of a product line;
  - 2) establishing structural relationships among the common and variable configuration items of domain engineering and domain variability model;
  - 3) establishing structural relationships among the configuration items of application engineering and application variability model;

NOTE Application variability model of a member product expresses the variabilities selected, meaning the application variability model describes which member product is affected by changes of a variable configuration item.



- 4) aligning the established structures of configuration information to the structure of a product line.
- b) A tool should support the task by allowing the user to:
  - 1) track to and from the common and variable configuration items of domain engineering and domain variability model;
  - 2) track to and from the configuration items of application engineering and application variability model;
  - 3) track the configuration item relationship links both upward and downward in the structure of a product line;
  - 4) provide the mapping of the identified PL configuration items to the elements of a product line (e.g. common or variable domain assets such as system, subsystem, component, library, or unit code, such application assets developed member product specifically).

### 7.3.5 Establish unique identifiers facilitating traceability

- a) The method should support the task with the following capabilities:
  - 1) providing ways to maintain consistencies among traces from variability model configuration items to the domain and application-specific configuration items;
  - 2) providing ways to generate immutable unique identifiers that can distinguish common configuration items, configuration items related to variations in time and space of a product line, and application-specific configuration items.
- b) A tool should support the task by allowing the user to:
  - 1) generate immutable unique identifiers (e.g. version ID) to distinguish and track each configuration item;
 

NOTE ID differentiates whether it is a common or variable one and whether it is a variation in time or a variation in space.
  - 2) link change requests associated with the configuration item;
  - 3) specify the status (e.g. checked-in, checked-out, the status of a change) of configuration items.

## 7.4 Product line configuration baselining

### 7.4.1 Principal constituents

#### 7.4.1.1 Purpose

The purpose of this subprocess is to capture and create configuration baselines of a product line and maintain the configurations with appropriate integrity and security.

#### 7.4.1.2 Inputs

The following inputs should be available to perform the process:

- a) PL configuration items;
- b) structured PL configuration information;
- c) product line architecture.

#### 7.4.1.3 Outcomes

The following outcomes shall be available as a result of the successful implementation of the process:

- a) configuration items for common and variable domain assets are baselined;
- b) variability model configuration items are baselined;
- c) integrated PL configuration for reuse is baselined;
- d) member product configurations are independently baselined;
- e) multiple baselines are agreed upon and available.

#### 7.4.1.4 Tasks

The organization shall implement the following tasks with respect to the process:

- a) Capture PL configuration baseline in time and space: Identify PL configuration items that are part of a PL configuration baseline in time and PL configuration items that are part of a PL configuration baseline in space separately.
- b) Create and release PL configuration baselines in time and space: Name the PL configuration items and versions of the identified PL configuration baseline. Naming is performed by distinguishing PL configuration baseline in time and space.
- c) Facilitate commonality and variability management in baselines of time and space: Devise good ways of managing changes and versions of common and variable configuration items in baselines. Changes on common and variable parts of a baseline over time as well as different versions of PL baselines due to variable configuration items, i.e. variability in space, should be managed.
- d) Maintain baselines readily available by distributed configuration management: Place baselines into a repository with the baseline attributes (e.g. creation date, owner, acceptance criteria, and change history) and allow access to the repository for collaboration in large and distributed PL development groups.

#### 7.4.2 Capture PL configuration baselines in time and space

- a) The method should support the task with the following capabilities:
  - 1) capturing common and variable configuration baselines in space dimension;
  - 2) capturing common and variable configuration baselines in time dimension.
- b) A tool should support the task by allowing the user to assign names and versions to the specified common and variable configuration item hierarchy of a PL configuration baseline.

#### 7.4.3 Create and release PL configuration baselines in time and space

- a) The method should support the task with the following capabilities:
  - 1) specifying the baseline attributes of a PL configuration baseline (e.g. creation date, owner, acceptance criteria, change history, relevant member products, and variation points of a baseline with appropriate variability model);
  - 2) tracing across baselines of domain requirements, domain design, domain realization, and domain testing in space dimension;
  - 3) tracing across baselines of domain requirements, domain design, domain realization, and domain testing in time dimension;

- 4) tracing from configuration baselines of domain engineering to configuration baselines of application engineering in space dimension;
  - 5) tracing from configuration baselines of domain engineering to configuration baselines of application engineering in time dimension;
  - 6) designing propagation control channels for distributing configuration control results of the current set of integrated multiple baselines.
- b) A tool should support the task by allowing the user to:
- 1) place a PL configuration baselines into a repository by differentiating baselines of time dimension, baselines of space dimension, and baselines of a member product;
  - 2) edit, search and change the baseline attributes of a PL configuration baseline;
  - 3) trace PL configuration items that comprise each PL configuration baseline;
  - 4) allow (semi-)auto-propagation of changes among a set of related configuration baselines.

#### 7.4.4 Facilitate commonality and variability management in baselines of time and space

NOTE 1 A member product is assembled out of common and variable configuration baselines such as common and variable components. Configuration management supports a variable configuration baseline that evolves in time independently not as a part of a member product because many versions of a component exist, and one of them is assembled into a member product.

NOTE 2 In case of variability, a variation point with its dependent variants (i.e. values of the variation point) is tracked in configuration item versions and baseline versions.

- a) The method should support the task with the following capabilities:
- 1) providing ways to manage versions of commonality and variability in configuration baselines both in time and space dimension for appropriate instantiations of member products;
  - 2) providing ways to manage changes of commonality and variability in configuration baselines in time and space for appropriate propagation of changes into all relevant member products.
- b) A tool should support the task by allowing the user to:
- 1) track sequential versions of commonality and variability in configuration baselines both in time and space dimension;
  - 2) track commonality and variability across baselines of requirements, architecture, detailed design, hardware/software implementation, and test.

#### 7.4.5 Maintain baselines readily available by distributed configuration management

- a) The method should support the task with the following capability: making the current set of integrated multiple baselines of a product line available for use by the distributed roles of a product line (e.g. product line engineer, domain engineers, and product engineers).

NOTE SSPL development entails establishing multiple developmental baselines in both domain engineering and application engineering. Evolving needs occur at both lifecycles, but simultaneous control of versions is allowed separately, i.e. evolving needs for domain assets application-specific assets are independently addressed. Therefore, distributed configuration management responsibilities and access limitations for domain asset configuration and application-specific asset configuration are managed separately. Application-specific asset configuration is managed in a configuration management system operated individually by a member product.

- b) A tool should support the task by allowing the user to:
- 1) allow simultaneous distributed control of domain asset configuration;

- 2) allow simultaneous distributed control of application-specific asset configuration;
- 3) allow access to application-specific asset configuration for change impact analysis.

## 7.5 Product line configuration status accounting

### 7.5.1 Principal constituents

#### 7.5.1.1 Purpose

The purpose of this subprocess is to control and manage configuration changes of a product line as a whole.

Artefacts changed due to evolution in a product line are configuration items in time. Changes in common artefacts are synchronous with single system evolution, but changes in variable artefacts are not. For example, the engineers of a home automation system expect technological progress in identification mechanisms. Therefore, they define a variation point "door lock identification mechanism" with just one variant "magnetic card." Later, when sufficiently reliable fingerprint scanners appear on the market, the engineers replace the variant "magnetic card" with the variant "fingerprint scanner." Since the evolution requires a change pertaining to such a variation point, a variation of a configuration item in time should recognize the changes in the domain assets that include the variation points<sup>[18]</sup>.

Change requests are made to a member product, common configuration items, or variable configuration items. And changes can be made during product line evolution. Changes on a member product are not reflected back into the common and variable configuration items or other member products. A change made to a common or variable configuration item affects all member products, so the change cannot be made to a member product (not allowed backward change propagation). Changes for enhancement or bug fixes are made directly to the common or variable configuration items so that the changes should be immediately applied to all member products. In case of changes for modifications or adding new functions, they should be made after their impacts on existing relevant variation points in a configuration are evaluated, and the issues are coordinated. Method and tool capabilities dealt with in this document are change requests on common and variable configuration items, not on member product specific.

#### 7.5.1.2 Inputs

The following inputs should be available to perform the process:

- a) baseline of a PL configuration;
- b) change requests for a PL configuration.

#### 7.5.1.3 Outcomes

The following outcome shall be available as a result of the successful implementation of the process: Integrity of PL configurations is managed and maintained.

#### 7.5.1.4 Tasks

The organization shall implement the following tasks with respect to the process:

- a) Evaluate and coordinate change requests for PL configuration items: Analyse change requests on PL configuration items over time and space, and approve/disapprove change requests by their impacts on common or variable configuration items, member products, and a product line.
- b) Identify PL configuration delta: Identify differences between PL configurations due to changes or product line evolution.

- c) Update baselines for PL configuration delta: Check integrities of all controlled configuration items, versions, and links in configuration deltas generated by changes or evolutions at the product line level.
- d) Relate PL configuration delta to relevant PL configuration items: Manage and maintain consistencies of a product line configuration and among different product line configurations.

### 7.5.2 Evaluate and coordinate change requests for PL configuration items

- a) The method should support the task with the following capabilities:
  - 1) capturing affected PL configuration items of both domain and application configuration items;
  - 2) analysing change impacts on application-specific configuration items of all member products.
- b) A tool should support the task by allowing the user to:
  - 1) track PL configuration items and related baselines, including member products affected by the proposed changes on common or variable configuration items;
  - 2) track variation points of a configuration affected by the proposed changes;
  - 3) describe the associated information for approved changes.

### 7.5.3 Identify PL configuration delta

- a) The method should support the task with the following capabilities:
  - 1) defining attributes to characterize which product line configurations can be considered as different versions of product line configurations (attributes include types of changes, e.g. a change managed by product line product management role such as addition or deletion of variability at product line product management, addition/deletion of product to/from a product line);
  - 2) managing and specifying interleaved deltas that are spread out different configuration items;
  - 3) enabling undoing of changes made to a PL configuration based on PL configuration delta (forward or backward).
- b) A tool should support the task by allowing the user to:
  - 1) store a sequence of the modification/enhancement/fixes in full to apply the deltas to the stored baselines;
  - 2) allow undoing changes made to a PL configuration.

### 7.5.4 Update baselines for PL configuration delta

- a) The method should support the task with the following capabilities:
  - 1) tracing PL configuration items, versions, and related baselines affected by PL configuration delta;
  - 2) ensuring that status of PL configuration delta are consistent with PL configurations;
  - 3) ensuring that PL configuration changes controlled are properly incorporated into PL configuration items, versions, and baselines, such as those of domain assets, variability models, and application assets;

- 4) allowing a member product to select whether it applies the PL configuration deltas related to domain asset changes for evolution or enhancement.
- b) A tool should support the task by allowing the user to:
  - 1) immediately and directly update member products with changes for bug fixes of common and variable configuration items (forward change propagation, apply the PL configuration deltas immediately for bug fixes);
  - 2) update member products with changes for enhancing or evolving domain assets;
  - 3) trace status of the PL configuration deltas across configuration items, versions, and baselines.

### 7.5.5 Relate PL configuration delta to relevant PL configuration items

- a) The method should support the task with the following capabilities:
  - 1) tracing PL configuration items related to PL configuration delta;
  - 2) tracing member products related to PL configuration delta;
  - 3) ensuring the completeness and integrity of two different PL configuration versions (e.g. revision  $n$  and revision  $n+1$ ).
- b) A tool should support the task by allowing the user to:
  - 1) store PL configuration delta-related PL configuration items;
  - 2) allow (semi-)automatic consistency check for two different PL configuration versions.

## 7.6 Product line configuration reporting

### 7.6.1 Principal constituents

#### 7.6.1.1 Purpose

The purpose of this subprocess is to produce PL configuration management records and status reports that show the status and history of controlled configurations of a product line and individual products.

#### 7.6.1.2 Inputs

The following inputs should be available to perform the process:

- a) disposition report of change requests on PL configuration;
- b) implemented approved changes on PL configuration.

#### 7.6.1.3 Outcomes

The following outcome shall be available as a result of the successful implementation of the process: PL configuration records and status reports are produced.

#### 7.6.1.4 Tasks

The organization shall implement the following tasks with respect to the process:

- a) Develop and maintain the PL configuration management status information: Collect management records and status reports from CM roles for domain assets and member products.

- b) Capture, store and report PL configuration anomaly: Collate the current PL configuration status against the previous PL configuration status to check PL configuration anomaly from information correctness, timeliness, integrity, and security.
- c) Capture, store, and report PL configuration management data: Produce management records and status reports of a product line.

#### **7.6.2 Develop and maintain the PL configuration management status information**

- a) The method should support the task with the following capabilities:
  - 1) tracing links among status reports of multiple CM roles;
  - 2) maintaining consistency of status reports among those of multiple CM roles.
- b) A tool should support the task by allowing the user to:
  - 1) track status reports of multiple CM roles with their relations;
  - 2) check consistency among status reports of multiple CM roles (semi-)automatically.

#### **7.6.3 Capture, store, and report PL configuration anomaly**

- a) The method should support the task with the following capability: consolidating the current PL configuration status and the previous PL configuration status to confirm information correctness, timeliness, integrity, and security.
- b) A tool should support the task by allowing the user to record and retrieve the current PL configuration status and the previous PL configuration status.

#### **7.6.4 Capture, store and report PL configuration management data**

- a) The method should support the task with the following capabilities:
  - 1) integrating captured status reports of multiple CM roles of a product line associated with the same change request;
  - 2) harmonizing reports in various formats consisting of different attributes because each CM role manages different configuration items, versions, and baselines;
  - 3) generating integrated status reports of product line CM performed by multiple CM roles.
- b) A tool should support the task by allowing the user to:
  - 1) search, access, and trace status reports of multiple CM roles of a product line associated with the same change request;
  - 2) edit/fill out a template or document of integrated status reports of product line CM performed by multiple CM roles.

## **8 Product line configuration management support**

### **8.1 General**

The product line CM support supports the following:

- a) quality assurance for product line CM;
- b) product line configuration change control;
- c) traceability management for product line configuration.



## 8.2 Quality assurance for product line CM

### 8.2.1 Principal constituents

#### 8.2.1.1 Purpose

The purpose of this subprocess is to measure the product line CM activities and work products to assure that the product line CM achieves the planned objectives and qualities by adhering to the defined CM process.

#### 8.2.1.2 Inputs

The following inputs should be available to perform the process:

- a) product line CM practices;
- b) established and managed PL configuration items, versions, and baselines;
- c) quantitative and qualitative quality assurance measures for product line CM;
- d) quality assurance practices performed for other PL lifecycle phases.

#### 8.2.1.3 Outcomes

The following outcomes shall be available as a result of the successful implementation of the process:

- a) evidence of the product line CM quality is produced;
- b) non-conformance issues with their status and results of the product line CM are produced and managed;
- c) status reports of corrective actions for the product line CM are produced and managed;
- d) status reports of quality trends for the product line CM are produced and managed;
- e) best practices for product line CM and proposals for improving quality assurance practices of other PL lifecycle phases are documented.

#### 8.2.1.4 Tasks

The organization shall implement the following tasks with respect to the process:

- a) Objectively evaluate PL configuration activities: Assure that performed CM processes adhere to the organization's process descriptions, standards, and procedures.
- b) Objectively evaluate PL configuration work products: Assure that work products associated with CM processes satisfy the stated criteria.
- c) Communicate and resolve non-compliance issues: Objectively trace, communicate, and resolve non-compliance issues found during the evaluation of PL configuration activities and PL configuration work products.
- d) Establish records of quality assurance activities for PL configuration: Identify and record best practices of PL configuration activities and PL configuration work products found during quality assurance activities for PL configuration. Proposals that can improve other PL quality assurance activities are identified and recorded.



### 8.2.2 Objectively evaluate PL configuration activities

- a) The method should support the task with the following capabilities:
  - 1) selecting or sampling product line CM activities from multiple CM roles (e.g. selection criteria or sampling method);
  - 2) tracing CM activities with complicated relations across multiple CM roles;
  - 3) evaluating CM activities (e.g. clearly stated criteria) and identifying non-compliance issues (especially in closely related CM activities of multiple CM roles);
  - 4) identifying quality risks possible in the CM activities of distributed multiple CM roles to cooperate.
- b) A tool should support the task by allowing the user to:
  - 1) access and trace evidence of CM activities from multiple CM roles (especially, CM activities defined as having complicated relations across multiple CM roles) and defined CM processes (e.g. process descriptions, standards, and procedures);
  - 2) trace CM activities with complicated relationships across multiple CM roles;
  - 3) edit/fill out non-compliance issues of CM activities;
  - 4) edit/fill out a quality risk assessment template for CM activities of multiple CM roles.

### 8.2.3 Objectively evaluate PL configuration work products

- a) The method should support the task with the following capabilities:
  - 1) selecting or sampling product line CM work products from multiple CM roles (e.g. selection criteria or sampling method);
  - 2) tracing CM work products with complicated relations across multiple CM roles;
  - 3) evaluating CM work products (e.g. clearly stated criteria) and identifying non-compliance issues (especially work products in closely related CM activities of multiple CM roles);
  - 4) identifying quality risks possible in the CM work products of distributed multiple CM roles to cooperate.
- b) A tool should support the task by allowing the user to:
  - 1) access and trace work products of CM activities from multiple CM roles (especially, work products of CM activities defined as having complicated relations across multiple CM roles);
  - 2) trace CM work products from CM activities with complicated relationships across multiple CM roles;
  - 3) edit/fill out non-compliance issues of CM work products.

### 8.2.4 Communicate and resolve non-compliance issues

- a) The method should support the task with the following capabilities:
  - 1) establishing escalation lines for resolving non-compliance issues when they cannot be resolved with the appropriate product line CM roles;
  - 2) tracking non-compliance issues throughout the established escalation lines;
  - 3) analysing non-compliance issues if there are any product line CM relevant quality trends;

- 4) enabling continuous process improvement of product line CM.
- b) A tool should support the task by allowing the user to:
  - 1) edit/fill out a document for non-compliance issues;
  - 2) allow communication links among staffs and managers within the established escalation lines;
  - 3) share the status of non-compliance issues among staffs and managers within the established escalation lines;
  - 4) analyse statistically quality trends of product line CM process and work products.

### **8.2.5 Establish records of quality assurance activities for PL configuration**

- a) The method should support the task with the following capability: supporting a template for reporting the results of quality assurance activities for PL configuration (the template includes quality assurance activities, the status of corrective actions, and quality trends).
- b) A tool should support the task by allowing the user to document a report for the results of quality assurance activities for PL configuration following the documentation template.

## **8.3 Product line configuration change control**

### **8.3.1 Principal constituents**

#### **8.3.1.1 Purpose**

The purpose of this subprocess is to control each change to a constituent product line configuration item as commonality, variability, the product line, or a product evolves from one baseline to the next.

#### **8.3.1.2 Inputs**

The following input should be available to perform the process: change requests on commonality, variability, the product line, or a product.

#### **8.3.1.3 Outcomes**

The following outcomes shall be available as a result of the successful implementation of the process:

- a) a disposition report of change requests is produced;
- b) approved changes are implemented.

#### **8.3.1.4 Tasks**

The organization shall implement the following tasks with respect to the process:

- a) Identify and record change requests for PL configuration: Identify and record change requests, including owner and configuration of a change request, the rationale of a change request, and scope of changes, i.e. product line level or product level.
- b) Evaluate and coordinate the impacts of change requests on commonality and variability: Analyse propagation of changes to all affected domain assets, variabilities, individual products, and product lines. Changes in the product line affect baselines associated with different stakeholders, participants, developers, or users, and communications among the affected parties are thus formally established, and opinions of the affected parties should be collected and managed.
- c) Verify conformance among relevant configurations in time and space: Check consistencies of changes among configurations managed in time and configurations managed in space. Because the

complexity of configurations is high and change impacts are broad and can have several causes, the reversibility of all affected configuration items and the impacted artefacts should be assured and guaranteed.

- d) Trace and manage approved changes to baselines: Trace and manage the status of approved changes and produce a disposition report, including rationales of accepting or rejecting decisions of change requests, audit trails, and authorization of the modifications.

### **8.3.2 Identify and record change requests for product line configuration**

- a) The method should support the task with the following capabilities:
  - 1) analysing interrelated change requests on configuration items of product line product management, domain assets, variability models, application assets, and member products;
  - 2) defining detailed attributes for interrelated change requests on configuration items of product line product management, domain assets, variability models, application assets, and member products (attributes include requester, date of request, configuration and version identification number, and rationale for a change);
  - 3) tracing the interrelated change requests on configuration items of product line product management, domain assets, variability models, application assets, and member products.
- b) A tool should support the task by allowing the user to:
  - 1) edit/fill out a form or a template interrelated change requests on configuration items of product line product management, domain assets, variability models, application assets, and member products;
  - 2) trace the change requests of domain engineering and application engineering separately.

### **8.3.3 Evaluate and coordinate the impacts of change requests on commonality and variability**

- a) The method should support the task with the following capability: analysing the proposed interrelated changes for the current product line with their impacts on domain assets, including commonality and variability, variability models, application assets, and member products.
- b) A tool should support the task by allowing the user to:
  - 1) trace affected configuration items of the proposed interrelated changes on domain assets, variability models, application assets, and each member product;
  - 2) enable access to the associated information of the proposed interrelated changes on domain assets, variability models, application assets, and each member product;
  - 3) store the change request information and decisions on the change requests for tracking change results on the approved change requests.

### **8.3.4 Verify conformance among relevant configurations in time and space**

- a) The method should support the task with the following capabilities:
  - 1) analysing conformance of a product line configuration at a particular point of time for approved changes;
  - 2) analysing conformance of a product line configuration at a particular point of space (i.e. version) for approved changes;

- 3) analysing conformance between product line configuration delta and before/after product line configuration for approved changes.
- b) A tool should support the task by allowing the user to:
- 1) enable an access authority to all the product line configuration items in time and space;
  - 2) trace relevant configurations in time and space.

### **8.3.5 Trace and manage approved changes to baselines**

- a) The method should support the task with the following capabilities:
- 1) allowing concurrent access to the configuration information of the entire product line;
  - 2) controlling a set of related versions of configuration items within a product line configuration for creating and modifying their new versions, but changes do not require version modification of a product line;
  - 3) controlling the version of a product line configuration for modifying and creating the new version when changes are anticipated which should not affect the existing version of a product line;
  - 4) propagating changes of configuration items of a product line configuration to the associated configuration items without creating new versions of a product line configuration;
  - 5) propagating changes of configuration items of a product line configuration to the associated configuration items with creating new versions of a product line configuration.
- b) A tool should support the task by allowing the user to:
- 1) produce separate branches of product line configuration such as branches for configurations of domain assets, application assets, member product, variability model, and product line product management;
  - 2) place a new version into the version branches separated by multiple configurations;
  - 3) merge the contents of designated versions of a product line.

## **8.4 Traceability management for product line configuration**

### **8.4.1 Principal constituents**

#### **8.4.1.1 Purpose**

The purpose of this subprocess is to facilitate the traceability of configurations, configuration items, and artefacts to control their changes.

#### **8.4.1.2 Inputs**

The following inputs should be available to perform the process:

- a) structured and managed PL configuration traces;
- b) baselined configurations of common and variable domain assets, application assets, and member products.

#### 8.4.1.3 Outcomes

The following outcome shall be available as a result of the successful implementation of the process: restructured PL configuration traces.

#### 8.4.1.4 Tasks

The organization shall implement the following tasks with respect to the process:

- a) Establish bi-directional traceability among PL configurations: Establish traces among configurations, their relevant commonality and variability, and artefacts to manage changes on product line configurations.
- b) Track and control PL configuration with commonality and variability: Compute traces for changes of configurations on commonality and variability definition, domain assets, and variants, then trace and control their relevant artefacts such as platform, implementation of variants, and products.
- c) Facilitate integrated traceability management for PL configuration and PL configuration delta: Trace two different versions of product line configurations with product line configuration delta for undoing changes.

#### 8.4.2 Establish bi-directional traceability among PL configurations

- a) The method should support the task with the following capabilities:
  - 1) bi-directionally tracing across baselines (e.g. the product line product management baseline, domain and application requirements baseline, domain and application design baseline, and member product baseline);
  - 2) bi-directionally tracing product line artefacts, including derived ones;
  - 3) tracing the configuration items and versions by change attributes;
  - 4) tracking the status of the modification.
- b) A tool should support the task by allowing the user to:
  - 1) enable all designated traces;
  - 2) represent the status of changes in a reporting form (e.g. table, graph, or chart).

#### 8.4.3 Track and control PL configuration with commonality and variability

- a) The method should support the task with the following capabilities:
  - 1) bi-directionally tracing across baselines of a product line configuration with its defined commonality and variability;
  - 2) bi-directionally tracing product line artefacts, including derived ones with their relevant commonality and variability definition;
  - 3) tracing the configuration items and versions by change attributes of commonality and variability definition;
  - 4) tracking the status of the modification concerning commonality and variability definition.
- b) A tool should support the task by allowing the user to:
  - 1) enable all designated traces;
  - 2) represent the status of changes in a reporting form (e.g. table, graph, or chart).

**8.4.4 Facilitate integrated traceability management for PL configuration and PL configuration delta**

- a) The method should support the task with the following capabilities:
  - 1) bi-directionally tracing across baselines of two different versions of product line configuration and product line configuration delta;
  - 2) bi-directionally tracing product line artefacts, including derived ones based on product line configuration delta;
  - 3) tracing the configuration items and versions by change attributes of product line configuration delta;
  - 4) tracking the status of the modification.
- b) A tool should support the task by allowing the user to:
  - 1) enable all designated traces;
  - 2) express the status of changes in a reporting form (e.g. table, graph, or chart).

## **Annex A** **(informative)**

### **Roles of configuration management in each SSPL lifecycle process**

#### **A.1 General**

Configurations of a product line are multidimensional in time and space. Configuration items can be domain and application assets, platform releases, and member products. Configurations of each of them have versions. Configuration management serves to manage changes and versions of configurations to cope with the complexity of the number of member applications consisting of many parts in different versions. These configurations shall be verified and validated, as incorrect configurations propagate to all member products and can cause their failures later on. Therefore, all configuration management shall be handled by a single configuration management process. For the integrated configuration management, configuration management in each SSPL lifecycle process shall include the roles specified in [A.2](#) and [A.3](#).

#### **A.2 Configuration management role in domain engineering process**

This role is responsible for managing configurations of platform release and other domain assets, the annotations necessary to reuse domain assets (e.g. glues, processes, and their descriptions that are attached to domain assets and prescribe how to use the assets). Change requests and versions of a domain are managed after they have been baselined. When configuration management is set up in domain engineering process, it considers the variability model from the very beginning. Otherwise, configuration management becomes too complex later on.

Platform release versions depend on domain asset versions. For example, changes in domain assets may impact numerous member products that use domain assets. Among domain assets, configuration management of domain architecture is critical because the domain architecture is bound into numerous configurations during application engineering engagements. Change management shall be carefully orchestrated because the domain architecture has a long-life span and be responsive to long-term evolutions of markets and technologies while remaining as immune as possible to changes in the market conditions of individual members of the product line and to progressions of component-level technologies.

Traceability links between domain requirements assets and the domain architecture, between domain realization assets and the domain architecture, between domain verification and validation assets and the domain architecture, and between the domain architecture and the application architectures derived from it should be maintained at a comprehensible level to enable change management and configuration management.

#### **A.3 Configuration management role in application engineering process**

This role is responsible for the versions of member products depending on asset versions and platform release versions. Each member product can exist in multiple configurations at any given time. The possible configurations of a member product can change over time.

Feedback from application architects related to the domain architecture is also managed here. Configuration management is important because the member products may offer plenty of variability for mass-customization. Each member product can thus exist in numerous configurations that shall be verified and validated. Change management supports and extends the application life cycle by dealing with changes in individual members' market conditions and progressions of component-level technologies.

Configuration management is important because each member product evolves throughout its life cycle and retains variability that different stakeholders, such as member product users, can bind. Each member product can thus exist in multiple configurations during its life cycle. It is essential for maintenance to know the versions of all domain components and interfaces used in the member product. Domain realization may provide member products with new domain asset versions that significantly refresh the member products (especially those having relatively longer life cycles) from technological and other viewpoints.

Traceability links between application requirements assets and the application architecture, between application realization assets and the application architecture, between application verification and validation assets and the application architecture, and between the domain architecture and the application architecture should be maintained at a comprehensible level to enable change management, configuration management, and the provisioning of feedback to product line architects.



## Bibliography

- [1] IEEE 828-2012, *Configuration management in systems and software engineering*.
- [2] ISO/IEC/IEEE 12207, *Systems and software engineering — Software life cycle processes*
- [3] ISO/IEC 14102, *Information technology — Guideline for the evaluation and selection of CASE tools*
- [4] ISO/IEC/IEEE 15288, *Systems and software engineering — System life cycle processes*
- [5] ISO/IEC 15940, *Systems and software engineering — Software Engineering Environment Services*
- [6] ISO/IEC/TR 18018:2010, *Information technology — Systems and software engineering — Guide for configuration management tool capabilities*
- [7] ISO/IEC/TR 19759, *Software Engineering — Guide to the software engineering body of knowledge (SWEBOK)*
- [8] ISO/IEC 19770, *Information technology — IT asset management*
- [9] ISO/IEC/IEEE 21840, *Systems and software engineering — Guidelines for the utilization of ISO/IEC/IEEE 15288 in the context of system of systems (SoS)*
- [10] ISO/IEC 25000, *Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — Guide to SQuaRE*
- [11] ISO/IEC 26550:2015, *Software and systems engineering — Reference model for product line engineering and management*
- [12] ISO/IEC 26551, *Software and systems engineering — Tools and methods for product line requirements engineering*
- [13] ISO/IEC 26552, *Software and systems engineering — Tools and methods for product line architecture design*
- [14] ISO/IEC 26553, *Information technology — Software and systems engineering — Tools and methods for product line realization*
- [15] ISO/IEC 26554, *Information technology — Software and systems engineering — Tools and methods for product line testing*
- [16] ISO/IEC 26555, *Software and systems engineering — Tools and methods for product line technical management*
- [17] ISO/IEC 26556, *Information technology — Software and systems engineering — Tools and methods for product line organizational management*
- [18] ISO/IEC 26557, *Software and systems engineering — Methods and tools for variability mechanisms in software and systems product line*
- [19] ISO/IEC 26558, *Software and systems engineering — Methods and tools for variability modelling in software and systems product line*
- [20] ISO/IEC 26559, *Software and systems engineering — Methods and tools for variability traceability in software and systems product line*
- [21] ISO/IEC 26560, *Software and systems engineering — Tools and methods for product line product management*
- [22] ISO/IEC/IEEE 29148, *Systems and software engineering — Life cycle processes — Requirements engineering*

- [23] CLEMENTS P., NORTHROP L. Software Product Lines: Practices and Patterns. Addison-Wesley, 2002
- [24] POHL K., BÖCKLE G., VAN DER LINDEN F.J. Software Product Line Engineering: Foundations, Principles and Techniques. Springer, 2005
- [25] VAN DER LINDEN F.J., SCHMID K., ROMMES E. Software Product Lines in Action: The Best Industrial Practice in Product Line Engineering. Springer, 2007



