INTERNATIONAL STANDARD

ISO/IEC 26552

First edition
2019-05

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

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

Reference number
ISO/IEC 26552:2019(E)

© ISO/IEC 2019



COPYRIGHT PROTECTED DOCUMENT

© ISO/IEC 2019

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
Fax: +41 22 749 09 47
Email: copyright@iso.org
Website: www.iso.org

Published in Switzerland

Contents

Page

Foreword ..... **vi**
Introduction ..... **vii**
1 Scope **1**
2 Normative references **1**
3 Terms and Definitions **1**
4 Reference model for product line architecture design **2**
4.1 Overview 2
4.2 Architecture management 3
4.3 Domain design 3
4.4 Asset management 4
4.5 Variability management in design 4
4.6 Application design 5
5 Architecture management **6**
5.1 General 6
5.2 Architecture design planning 7
5.2.1 Principal constituents 7
5.2.2 Establish architecture design goals 7
5.2.3 Define key procedures for architecture design 8
5.2.4 Define schedules and required resources for architecture design 8
5.2.5 Specify how to monitor, measure and control the effectiveness of architecture design 9
5.2.6 Document the architecture design plan 9
5.3 Architecture design enabling 10
5.3.1 Principal constituents 10
5.3.2 Prepare for the architecture enablement 10
5.3.3 Develop and establish enabling capabilities and resources 11
5.3.4 Deploy capabilities and resources for architecture enablement 11
5.3.5 Improve architecture enablement capabilities and resources 12
5.4 Architecture design managing 12
5.4.1 Principal constituents 12
5.4.2 Prepare for architecture management execution 13
5.4.3 Implement the architecture management plans 14
5.4.4 Close and prepare for the architecture management plan change 14
6 Domain design **14**
6.1 General 14
6.2 Conceptual architecture design 15
6.2.1 Principal constituents 15
6.2.2 Analyse problem space of the domain architecture 16
6.2.3 Synthesize potential solution alternatives 16
6.2.4 Formulate potential domain architecture(s) 17
6.2.5 Capture domain architecture concepts and properties 17
6.2.6 Hand off conceptualized domain architecture to users and other stakeholders 18
6.3 Domain architectural structure design 18
6.3.1 Principal constituents 18
6.3.2 Develop architecture viewpoints for the product line 19
6.3.3 Develop models and views of the domain architecture 19
6.3.4 Relate the domain architecture to requirements 20
6.3.5 Relate the domain architecture to detailed design 20
6.4 Architectural texture design 21
6.4.1 Principal constituents 21
6.4.2 Analyse common rules guiding realization 21
6.4.3 Define common ways to deal with variability at domain realization 22

© ISO/IEC 2019 – All rights reserved

6.4.4 Define common ways to deal with variability at application design and realization ... 22
6.4.5 Formulate architectural texture ... 23
6.5 Domain architecture documentation ... 23
6.5.1 Principal constituents ... 23
6.5.2 Assess the domain architecture documentation for structure and texture ... 24
6.5.3 Hand off architecture documentation to downstream users ... 24
6.6 Domain architecture evaluation ... 25
6.6.1 Principal constituents ... 25
6.6.2 Determine evaluation criteria for domain architecture ... 26
6.6.3 Establish measurement techniques for domain architecture ... 26
6.6.4 Review evaluation-related information for domain architecture ... 27
6.6.5 Analyse domain architecture and assess stakeholder satisfaction ... 27
6.6.6 Formulate findings and recommendations for domain architecture ... 28
6.6.7 Communicate evaluation results ... 28

7 Variability management in design ... 28
7.1 General ... 28
7.2 Internal variability in domain architecture ... 29
7.2.1 Principal constituents ... 29
7.2.2 Identify newly added internal variability ... 29
7.2.3 Refine external variability into internal variability ... 30
7.2.4 Relate internal variability with variability in requirements ... 30
7.3 Variability model in architecture ... 31
7.3.1 Principal constituents ... 31
7.3.2 Model variability in views of architecture(s) ... 31
7.3.3 Maintain variability model in architecture ... 32
7.3.4 Document variability in architecture ... 32
7.4 Variability mechanism in architecture ... 33
7.4.1 Principal constituents ... 33
7.4.2 Identify variability mechanisms in architecture by category ... 33
7.4.3 Guide the use of variability mechanism category in architecture ... 34
7.4.4 Trace the usage status of variability mechanism category in architecture ... 34
7.4.5 Update variability mechanism category in architecture ... 35
7.5 Variability traceability in architecture ... 35
7.5.1 Principal constituents ... 35
7.5.2 Define trace links among variability in different architectural artefacts ... 36
7.5.3 Define trace links between architectural artefacts and variability model ... 36

8 Asset management in design ... 36
8.1 General ... 36
8.2 Managing domain design artefacts as domain assets ... 37
8.2.1 Principal constituents ... 37
8.2.2 Identify architectural artefacts managed as domain assets ... 37
8.2.3 Define configuration and annotation for domain architecture assets ... 38
8.3 Managing application design artefacts as application assets ... 38
8.3.1 Principal constituents ... 38
8.3.2 Identify architectural artefacts managed as application assets ... 39
8.3.3 Define configuration and annotation for application architecture assets ... 39

9 Application design ... 40
9.1 General ... 40
9.2 Binding in architecture ... 40
9.2.1 Principal constituents ... 40
9.2.2 Decide values of variabilities in architecture ... 41
9.2.3 Conduct bindings in architecture ... 41
9.2.4 Validate consistencies with bindings in requirements ... 42
9.2.5 Validate whether binding decisions adhere to the architectural texture ... 42
9.3 Application specific architectural structure design ... 42
9.3.1 Principal constituents ... 42

 © ISO/IEC 2019 – All rights reserved

9.3.2 Develop models of the application specific architecture 43
9.3.3 Validate whether the application specific architecture adheres to the architectural texture 44
9.3.4 Relate the application specific architecture to requirements 44
9.3.5 Relate the application specific architecture to detailed design 44
9.4 Application architecture documentation 45
9.4.1 Principal constituents 45
9.4.2 Assess the application specific architecture documentation 46
9.4.3 Hand off application specific architecture documentation to downstream users 46
9.5 Application architecture evaluation 47
9.5.1 Principal constituents 47
9.5.2 Determine application specific evaluation criteria 48
9.5.3 Establish application specific measurement techniques 48
9.5.4 Review evaluation-related information for application architecture 48
9.5.5 Analyse application architecture and assess stakeholder satisfaction 49
9.5.6 Formulate findings and recommendations for application architecture 49
9.5.7 Communicate evaluation results with application specific stakeholders 50

Annex A (informative) **Cross-reference with ISO/IEC/IEEE 42020, ISO/IEC/IEEE 42010 and ISO/IEC/IEEE 15288** **51**

Annex B (informative) **Variability specification elements in ADL** **58**

Annex C (informative) **Architecture structure and texture example** **59**

Bibliography **60**

© ISO/IEC 2019 – All rights reserved

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work. In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular, the different approval criteria needed for the different types of document should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see www.iso.org/directives).

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see www.iso.org/patents).

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation of the voluntary nature of standards, the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT) see www.iso.org/iso/foreword.html.

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

Any feedback or questions on this document should be directed to the user's national standards body. A complete listing of these bodies can be found at www.iso.org/members.html.

© ISO/IEC 2019 – All rights reserved

Introduction

The main purpose of this document is to deal with the capabilities of methods and tools of architecture design for software and systems product line (SSPL). This document defines how the tools and methods can support for the software and systems product line-specific architecture processes.

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

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

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

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

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

This document can be used in the following modes:

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

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

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

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

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

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

— Processes and capabilities of methods and tools for domain realization and application realization are provided in ISO/IEC 26553;

© ISO/IEC 2019 – All rights reserved

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

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

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

— Processes and capabilities of methods and tools for variability mechanisms are provided in ISO/IEC 26557;

— Processes and capabilities of methods and tools for variability modelling are provided in ISO/IEC 26558;

— Processes and capabilities of methods and tools for variability traceability are provided in ISO/IEC 26559;

— Processes and capabilities of methods and tools for product management are provided in ISO/IEC 26560;

— Processes and capabilities of methods and tools for technical probe are provided in ISO/IEC 26561;

— Processes and capabilities of methods and tools for transition management are provided in ISO/IEC 26562;

— Processes and capabilities of methods and tools for configuration management of asset are provided in ISO/IEC 26563;

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

© ISO/IEC 2019 – All rights reserved

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

1 Scope

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

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

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

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

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

2 Normative references

There are no normative references in this document.

3 Terms and Definitions

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

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

— ISO Online browsing platform: available at http://www.iso.org/obp

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

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

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

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

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

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

© ISO/IEC 2019 – All rights reserved

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

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

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

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

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

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

4 Reference model for product line architecture design

4.1 Overview

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

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

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

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

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

— the title of the subprocess;

— the purpose of the subprocess;

— the inputs to produce the outcomes;

© ISO/IEC 2019 – All rights reserved