ISO/IEC 26554:2018 (E)

Information technology — Software and systems engineering — Tools and methods for product line testing

Contents

Foreword

Introduction

1 Scope

2 Normative references

3 Terms and definitions

4 Abbreviated terms

5 Reference model for product line testing

5.1 Overview
5.2 Product line test management
5.3 Domain testing
5.4 Asset management in testing
5.5 Variability management in testing
5.6 Application testing

6 Product line test management

6.1 General
6.2 Product line test strategy
6.2.1 Principal constituents
6.2.1.1 Purpose
6.2.1.2 Inputs
6.2.1.3 Outcomes
6.2.1.4 Tasks
6.2.2 Define product line test goals
6.2.3 Identify and analyse risks in product line test
6.2.4 Establish product line test strategies
6.3 Product line test process
6.3.1 Principal constituents
6.3.1.1 Purpose
6.3.1.2 Inputs
6.3.1.3 Outcomes
6.3.1.4 Tasks
6.3.2 Select and tailor domain test process
6.3.3 Select and tailor application-specific test process
6.4 Product line test planning
6.4.1 Principal constituents
6.4.1.1 Purpose
6.4.1.2 Inputs
6.4.1.3 Outcomes
6.4.1.4 Tasks
6.4.2 Develop organizational test plan
6.4.3 Gain consensus on organizational test plan
6.4.4 Document and share organizational test plan
6.5 Product line test monitoring and control
6.5.1 Principal constituents
6.5.1.1 Purpose
6.5.1.2 Inputs
6.5.1.3 Outcomes

6.5.1.4 Tasks
6.5.2 Initiate monitoring and controlling product line test progress
6.5.3 Monitor product line test progress
6.5.4 Control product line test progress
6.5.5 Report product line test progress

7 Domain testing

7.1 General
7.2 Domain test initiation and design
7.2.1 Principal constituents
7.2.1.1 Purpose
7.2.1.2 Inputs
7.2.1.3 Outcomes
7.2.1.4 Activities
7.2.2 Domain test initiation
7.2.2.1 Tasks
7.2.2.2 Initiate domain testing
7.2.2.3 Define domain test requirements and conditions
7.2.3 Domain test design for unit testing
7.2.3.1 General
7.2.3.2 Tasks
7.2.3.3 Create domain test cases for unit testing
7.2.3.4 Derive domain test procedures for unit testing
7.2.3.5 Develop auxiliary test implementation for absent variants-related unit testing
7.2.4 Domain test design for integration testing
7.2.4.1 General
7.2.4.2 Tasks
7.2.4.3 Create domain test cases for integration testing
7.2.4.4 Derive domain test procedures for integration testing
7.2.4.5 Develop auxiliary test implementation for absent variants-related integration testing
7.2.5 Domain test design for system testing
7.2.5.1 General
7.2.5.2 Tasks
7.2.5.3 Create domain test cases for system testing
7.2.5.4 Derive domain test scenarios for system testing
7.3 Domain test environment set-up and maintenance
7.3.1 Principal constituents
7.3.1.1 Purpose
7.3.1.2 Inputs
7.3.1.3 Outcomes
7.3.1.4 Tasks
7.3.2 Set up domain test environments
7.3.3 Enable interoperability with other domain engineering environments
7.3.4 Maintain domain test environments
7.4 Domain test execution
7.4.1 Principal constituents
7.4.1.1 Purpose
7.4.1.2 Inputs
7.4.1.3 Outcomes
7.4.1.4 Activities
7.4.2 Domain static testing
7.4.2.1 General
7.4.2.2 Tasks
7.4.2.3 Prepare review, inspection or static analysis for domain
7.4.2.4 Execute domain static test
7.4.2.5 Record static test results
7.4.3 Domain dynamic test execution
7.4.3.1 General
7.4.3.2 Tasks
7.4.3.3 Execute domain test procedures
7.4.3.4 Compare domain test results
7.4.3.5 Record variability relevant test
7.5 Domain test reporting
7.5.1 Principal constituents

7.5.1.1 Purpose
7.5.1.2 Inputs
7.5.1.3 Outcomes
7.5.1.4 Tasks
7.5.2 Analyse domain test results
7.5.3 Create/update domain test reports

8 Asset management in testing

8.1 General
8.2 Domain test artefacts as domain assets
8.2.1 Principal constituents
8.2.1.1 Purpose
8.2.1.2 Inputs
8.2.1.3 Outcomes
8.2.1.4 Tasks
8.2.2 Identify domain test artefacts managed as domain assets
8.2.3 Structure configuration and annotation for domain test assets
8.3 Application test artefacts as application assets
8.3.1 Principal constituents
8.3.1.1 Purpose
8.3.1.2 Inputs
8.3.1.3 Outcomes
8.3.1.4 Tasks
8.3.2 Identify application test artefacts managed as application assets
8.3.3 Structure configuration and annotation for application test assets

9 Variability management in testing

9.1 General
9.2 Variability mechanism category in testing
9.2.1 Principal constituents
9.2.1.1 Purpose
9.2.1.2 Inputs
9.2.1.3 Outcomes
9.2.1.4 Tasks
9.2.2 Identify variability mechanisms in testing by category
9.2.3 Guide the use of variability mechanism category by PL test strategy
9.2.4 Guide the use of variability mechanism category by test levels
9.2.5 Trace the usage status of variability mechanism category in testing
9.2.6 Update variability mechanism category in testing
9.3 Variability in test artefacts
9.3.1 Principal constituents
9.3.1.1 Purpose
9.3.1.2 Inputs
9.3.1.3 Outcomes
9.3.1.4 Tasks
9.3.2 Define variability type in test artefacts
9.3.3 Define variability representation in test artefacts
9.4 Traceability of variability in test
9.4.1 Principal constituents
9.4.1.1 Purpose
9.4.1.2 Inputs
9.4.1.3 Outcomes
9.4.1.4 Tasks
9.4.2 Define explicit links between variability in test assets and variability model
9.4.3 Define explicit links between application test assets and application variability model

10 Application testing

10.1 General
10.2 Application test initiation and design
10.2.1 Principal constituents
10.2.1.1 Purpose
10.2.1.2 Inputs
10.2.1.3 Outcomes
10.2.1.4 Activities

10.2.2 Application test initiation
10.2.2.1 Tasks
10.2.2.2 Initiate application testing
10.2.2.3 Define application test requirements and conditions
10.2.3 Application-specific test design for unit testing
10.2.3.1 General
10.2.3.2 Tasks
10.2.3.3 Create application-specific test cases for unit testing
10.2.3.4 Derive application-specific test procedures for unit testing
10.2.3.5 Develop auxiliary test implementation for application-specific unit testing
10.2.4 Application test design for integration testing
10.2.4.1 General
10.2.4.2 Tasks
10.2.4.3 Create application test cases for integration testing
10.2.4.4 Derive application test procedures for integration testing
10.2.4.5 Develop auxiliary test implementation for application integrated testing
10.2.5 Application test design for system testing
10.2.5.1 General
10.2.5.2 Tasks
10.2.5.3 Create application test cases for system testing
10.2.5.4 Derive application test scenarios for system testing
10.3 Application test environment set-up and maintenance
10.3.1 Principal constituents
10.3.1.1 Purpose
10.3.1.2 Inputs
10.3.1.3 Outcomes
10.3.1.4 Tasks
10.3.2 Set up application test environments
10.3.3 Enable interoperability with other application engineering environments
10.3.4 Maintain application test environments
10.4 Application test execution
10.4.1 Principal constituents
10.4.1.1 Purpose
10.4.1.2 Inputs
10.4.1.3 Outcomes
10.4.1.4 Activities
10.4.2 Application static testing
10.4.2.1 General
10.4.2.2 Tasks
10.4.2.3 Prepare review, inspection or static analysis for application
10.4.2.4 Execute application static test
10.4.2.5 Record static test results
10.4.3 Application dynamic test execution
10.4.3.1 General
10.4.3.2 Tasks
10.4.3.3 Execute application test procedures
10.4.3.4 Compare application test results
10.4.3.5 Record absent variants-relevant test execution
10.5 Application test reporting
10.5.1 Principal constituents
10.5.1.1 Purpose
10.5.1.2 Inputs
10.5.1.3 Outcomes
10.5.1.4 Tasks
10.5.2 Analyse application test results
10.5.3 Create/update application test reports

Annex A (informative) Exemplar product line test strategy

A.1 Overview of testing strategy
A.2 Exemplar organizational test strategies
A.2.1 General
A.2.2 SAS
A.2.3 CRS
A.2.4 Combined SAS and CRS

Annex B (informative) Execution of SSPL testing

Annex C (informative) Mapping of ISO/IEC/IEEE 29119-2 and ISO/IEC/IEEE 15288

Page count: 55