

ICS 35.080
CCS L 77



**NATIONAL STANDARD OF THE
PEOPLE'S REPUBLIC OF CHINA**

GB/T 38634.2-2020

**Systems and software engineering - Software testing - Part 2:
Test processes**

系统与软件工程 软件测试 第2部分：测试过程

(English Translation)

Issued on: April 28, 2020

Implemented on: November 1, 2020

**Issued by: State Administration for Market Regulation, China National
Standardization Administration**

Table of Contents

Foreword	
Introduction	
1 Scope	
2 Compliance	
2.1 Intended use	
2.2 Full compliance	
2.3 Tailoring compliance	
3 Normative references	
4 Terms and definitions, abbreviations	
4.1 Terms and definitions	
4.2 Abbreviations	
5 Multi-layer test process model	
6 Organizational testing process	
6.1 Overview	
6.2 Organizational testing process	
7 Test management process	
7.1 Overview	
7.2 Test planning process	
7.3 Test monitoring and control process	
7.4 Test completion process	
8 Dynamic test process	
8.1 Overview	
8.2 Test Design and Implementation Process	
8.3 Test environment construction and maintenance process	
8.4 Test execution process	
8.5 Test event reporting process	
Appendix A (informative appendix) Some examples of test design process	
Appendix B (Normative Appendix) Process mapping between this part and ISO /IEC /IEEE12207.2017	
Appendix C (informative appendix) Process mapping between this part and ISO /IEC /IEEE15288.2015	
Appendix D (informative appendix) Process mapping between this part and ISO /IEC 17025:2017	

Appendix E (informative appendix) Process mapping between this part and GB/T 25000.51-2016

Appendix F (informative appendix) Process mapping between this part and GB/T 15532-2008

Appendix G (informative appendix) Process mapping between this part and IEEE1008-2008

Appendix H (informative appendix) Structural changes in this part compared with ISO /IEC /IEEE29119-2.2013

Reference

1 Scope

This part of GB/T 38634 specifies the software testing for governance, management and implementation of any organization, project or smaller-scale testing activities.

The testing process of the test defines the general process of software testing, and gives a supporting information chart describing the process.

This section applies to all tests in the software life cycle model.

This section applies to but not limited to testers, test managers, developers and project managers, especially those responsible for governance, management and implementation of software testing personnel.

2.1 Intended use

The requirements of this section are contained in Chapter 6 ~ Chapter 8: This section provides requirements applicable to many testing processes throughout the software life cycle.

A specific project or organization may not need to use all the processes defined in this section, so the implementation of this section usually involves choosing one Groups apply to the process of an organization or project. Organizations can declare compliance with this section in two ways.

The organization shall declare whether it is completely or tailored in accordance with this section.

2.2 Full compliance

Full compliance is achieved by proving that all requirements of the entire process defined in this section (ie. should be declared) are achieved.

2.3 Tailoring compliance

When this section is used to establish a set of process foundations that do not meet full compliance, record a subset of the process of tailoring compliance. Pass the proof All the requirements of the recorded process subset (ie. should be declared) have been met to achieve tailoring compliance.

When tailoring, as long as the process specified in Chapter 6, Chapter 7, and Chapter 8 of this part is not followed, justifications should be provided (directly or through Reference). All tailoring decisions should document their reasons, including consideration of any applicable risks. Tailoring decisions should be approved by stakeholders.

Example: If the organization follows the information item management process in standards such as ISO 15489 or GB/T 19001 or uses similar internal organizational processes, they can It was decided to use these processes instead of the information item management tasks defined in this section.

3 Normative references

The following documents are indispensable for the application of this document. For dated reference documents, only the dated version applies to this article Pieces. For undated references, the latest version (including all amendments) applies to this document.

GB/T 38634.1 System and Software Engineering Software Testing Part 1: Concepts and Definitions (GB/T 38634.1-2020, ISO /IEC /IEEE29119-1.2013, MOD)

GB/T 38634.4 System and Software Engineering Software Testing Part 4: Testing Technology (GB/T 38634.4-2020, ISO /IEC /IEEE29119-4.2015, MOD)

ISO /IEC /IEEE12207.2017 System and Software Engineering Software Life Cycle Process

4.1 Terms and definitions

The following terms and definitions apply to this document.

4.1.1 Results of testing

The behavior and state set of the test item, or related data, and the state set of the test environment can be obtained as the result of test execution.

Examples. output to hardware, changes to data, generation and display of reports, and sending of communication messages.

4.1.2 Completion criteria

The conditions for the completion of the test activity.

4.1.3 Coverage item

See test coverage item (4.1.33).

4.1.4 Dynamic Testing

Need to run the test of the test item.

4.1.5 Equivalence class

A subset of the range of a variable or variable set. In the test item or its interface, it is expected that the test item will handle all in the subset in the same way. Some values (that is, considered "equivalent").

4.1.6 Equivalence category coverage

The test set covers the proportion of identified equivalence classes in the test items.

Note: In many cases, the equivalence class identification is subjective (especially in the "invalid" equivalence class), so it is not possible to explicitly count the equivalence class of the test item can.

4.1.7 Equivalence class division

A type of test design technique. Use one or more representative members of each equivalence class to design test cases.

4.1.8 Expected results

According to specifications or other sources, the expected behavior of the test item that can be obtained under specific conditions.

4.1.9 Exploratory test

An experience-based test. Testers based on their existing relevant knowledge and preliminary exploration of test items (including previous test results)

As well as heuristic "rules of thumb" about common software behavior and failure types, tests are designed and executed spontaneously.

Note: Exploratory testing looks for hidden attributes (including hidden behaviors). Although the possibility of its own harm is very small, it may interfere with other aspects of the software under test. Other attributes, and therefore the risk of software failure.