



ETSI Guide

**Methods for Testing and Specification (MTS);
Standards engineering process;
A Handbook of validation methods**

ReferenceREG/MTS-00122 ValidHandB

Keywords

interoperability, methodology, procedure, testing,
validation**ETSI**

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - NAF 742 C
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° 7803/88

Important notice

Individual copies of the present document can be downloaded from:

<http://www.etsi.org>

The present document may be made available in more than one electronic version or in print. In any case of existing or perceived difference in contents between such versions, the reference version is the Portable Document Format (PDF). In case of dispute, the reference shall be the printing on ETSI printers of the PDF version kept on a specific network drive within ETSI Secretariat.

Users of the present document should be aware that the document may be subject to revision or change of status. Information on the current status of this and other ETSI documents is available at

<http://portal.etsi.org/tb/status/status.asp>

If you find errors in the present document, please send your comment to one of the following services:

http://portal.etsi.org/chaicor/ETSI_support.asp

Copyright Notification

No part may be reproduced except as authorized by written permission.
The copyright and the foregoing restriction extend to reproduction in all media.

© European Telecommunications Standards Institute 2012.
All rights reserved.

DECT™, **PLUGTESTS™**, **UMTS™** and the ETSI logo are Trade Marks of ETSI registered for the benefit of its Members.
3GPP™ and **LTE™** are Trade Marks of ETSI registered for the benefit of its Members and
of the 3GPP Organizational Partners.
GSM® and the GSM logo are Trade Marks registered and owned by the GSM Association.

Contents

Intellectual Property Rights	5
Foreword.....	5
Introduction	5
1 Scope	6
2 References	6
2.1 Normative references	6
2.2 Informative references.....	6
3 Definitions and abbreviations.....	8
3.1 Definitions.....	8
3.2 Abbreviations	8
4 Validation overview	9
5 Reasons for validating standards.....	9
5.1 Problems commonly found by validating standards.....	9
5.1.1 Editorial issues.....	10
5.1.2 Technical issues	10
5.2 Avoiding problems normally found during validation	11
5.3 Using validation methods to detect problems.....	12
6 Methods for validating standards	12
6.1 Different types of standard	12
6.2 Explicit validation	12
6.2.1 Review methods.....	12
6.2.1.1 Overview of reviewing methods	12
6.2.1.2 Using reviews to validate standards	13
6.2.1.3 Basic review process.....	13
6.2.1.3.1 Phase 1: Planning the review	14
6.2.1.3.1.1 Assign a leader for the review process.....	15
6.2.1.3.1.2 Select and invite the reviewers.....	15
6.2.1.3.1.3 Schedule a date for the review	15
6.2.1.3.2 Phase 2: Preparing for the review.....	16
6.2.1.3.3 Phase 3: Reviewing the specification	16
6.2.1.3.3.1 Presentation of the overview from the document author(s)	16
6.2.1.3.3.2 Inspection of the document.....	17
6.2.1.3.3.3 Preparation and agreement of any required change proposals	17
6.2.1.3.4 Phase 4: Processing change proposals.....	17
6.2.1.3.4.1 Implement changes in a document revision	17
6.2.1.3.4.2 Review actual changes.....	17
6.2.2 Model-based validation methods	17
6.2.2.1 Modelling and simulation	18
6.2.2.2 Model quality assessment	19
6.2.3 Test suite validation	19
6.2.4 Product-based validation (proof-by-use)	19
6.2.4.1 ETSI Plugtests™ events.....	19
6.2.4.1.1 Benefits of using a testing events for validating standards.....	19
6.2.4.1.2 Criteria for selecting a testing event for validation.....	20
6.2.4.1.3 Overall process	20
6.2.4.1.3.1 Responsibilities.....	20
6.2.4.1.3.2 Factors to consider in making a Plugtests™ event successful	21
6.2.4.2 Prototyping.....	23
6.3 Implicit validation	23
6.3.1 Requirements cataloguing.....	23
6.3.2 Test specification development.....	24
6.3.2.1 ICS and IFS.....	24

6.3.2.2	Test Suite Structure	25
6.3.2.3	Test Purposes	25
6.3.2.4	Test Suite	25
6.3.2.5	IXIT.....	25
6.3.3	Conformance and interoperability testing	25
6.3.4	Development of descriptive and guidance documentation	26
6.3.5	Product development	26
6.4	Validation best practice	26
6.4.1	Identify the most appropriate validation level	26
6.4.1.1	Defined validation levels.....	26
6.4.1.2	Select the required method(s).....	28
6.4.1.3	Plan validation into the standards development process	28
6.4.1.4	Resolve problems detected during validation.....	29
Annex A:	Sample review forms.....	30
A.1	Introduction	30
A.1.1	A form for recording reviewers comments.....	30
Annex B:	Test Suite Validation Guidelines	31
B.1	Introduction	31
B.2	Validation levels.....	31
B.2.1	Level 1 validation (Basic)	31
B.2.1.1	Description.....	31
B.2.2	Level 2 validation (Strong).....	31
B.2.2.1	Description.....	31
B.2.3	Level 3 validation (Rigorous).....	32
B.2.3.1	Description.....	32
B.3	Completing the Terms of Reference.....	32
B.3.1	The extent of validation.....	32
B.3.2	Responsibilities	33
B.3.3	Costs associated with validation.....	33
B.4	The validation report	34
History	36

Intellectual Property Rights

IPRs essential or potentially essential to the present document may have been declared to ETSI. The information pertaining to these essential IPRs, if any, is publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: "*Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards*", which is available from the ETSI Secretariat. Latest updates are available on the ETSI Web server (<http://ipr.etsi.org>).

Pursuant to the ETSI IPR Policy, no investigation, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Foreword

This ETSI Guide (EG) has been produced by ETSI Technical Committee Methods for Testing and Specification (MTS).

Introduction

ETSI produces technical standards for a wide range of communication applications. All these standards need to be thoroughly validated and many different methods are available for this purpose. The present document collects and presents those validation methods which are practical and have been successfully applied either within ETSI or in other standardization bodies. Each method has its advantages and disadvantages and they are probably most effective if used in combination and at all stages of the standards development process. A three level approach for the basic, strong and rigorous validation of standards is presented to guide the selection and application of standard validation methods.

1 Scope

The present document identifies and describes a number of practical methods that can be used to validate all types of standards produced by ETSI. The methods described are as follows:

- review methods;
- model-based methods;
- product-based methods; and
- testing.

It also gives guidance on how and when each of these methods can be used most effectively within ETSI's standards engineering process.

2 References

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the referenced document (including any amendments) applies.

Referenced documents which are not found to be publicly available in the expected location might be found at <http://docbox.etsi.org/Reference>.

NOTE: While any hyperlinks included in this clause were valid at the time of publication ETSI cannot guarantee their long term validity.

2.1 Normative references

The following referenced documents are necessary for the application of the present document.

Not applicable.

2.2 Informative references

The following referenced documents are not necessary for the application of the present document but they assist the user with regard to a particular subject area.

- [i.1] ETSI EG 201 383: "Methods for Testing and Specification (MTS); Use of SDL in ETSI deliverables; Guidelines for facilitating validation and the development of conformance tests".
- [i.2] ETSI EG 201 872: "Methods for Testing and Specification (MTS); Methodological approach to the use of object-orientation in the standards making process".
- [i.3] ETSI EG 202 107: "Methods for Testing and Specification (MTS); Planning for validation and testing in the standards-making process".
- [i.4] ETSI EG 202 237: "Methods for Testing and Specification (MTS); Internet Protocol Testing (IPT); Generic approach to interoperability testing".
- [i.5] ETSI EG 202 568: "Methods for Testing and Specification (MTS); Internet Protocol Testing (IPT); Testing: Methodology and Framework".
- [i.6] ETSI ES 201 873-1: "Methods for Testing and Specification (MTS); The Testing and Test Control Notation version 3; Part 1: TTCN-3 Core Language".

- [i.7] ETSI ES 202 553: "Methods for Testing and Specification (MTS); TPLan: A notation for expressing Test Purposes".
- [i.8] ETSI TS 102 385-2: "Broadband Radio Access Networks (BRAN); HiperMAN; Conformance Testing for WiMAX/HiperMAN 1.2.1; Part 2: Test Suite Structure and Test Purposes (TSS&TP)".
- [i.9] ETSI TS 102 558: "Methods for Testing and Specification (MTS); Internet Protocol Testing (IPT); IPv6 Security; Requirements Catalogue".
- [i.10] ETSI TS 102 593: "Methods for Testing and Specification (MTS); Internet Protocol Testing (IPT); IPv6 Security; Conformance Test Suite Structure and Test Purposes (TSS&TP)".
- [i.11] ETSI TS 102 795: "Electromagnetic compatibility and Radio spectrum Matters (ERM); digital Private Mobile Radio (dPMR); Part 3: Requirements catalogue".
- [i.12] ETSI TS 102 870-2: "Intelligent Transport Systems (ITS); Testing; Conformance test specifications for GeoNetworking Basic Transport Protocol (BTP); Part 2: Test Suite Structure and Test Purposes (TSS&TP)".
- [i.13] ETSI ETR 184: "Methods for Testing and Specification (MTS); Overview of validation techniques for European Telecommunication Standards (ETs) containing SDL".
- [i.14] ETSI ETR 212: "Methods for Testing and Specification (MTS); Implementation Conformance Statement (ICS) proforma style guide".
- [i.15] ETSI ETS 300 406: "Methods for Testing and Specification (MTS); Protocol and profile conformance testing specifications; Standardization methodology".
- [i.16] ETSI Directives, Version 27 (2010).
- [i.17] ISO/IEC 9646-1: "Information technology; Open Systems Interconnection; Conformance testing methodology and framework; Part 1: General concepts".
- [i.18] ITU-T Recommendation Z.100: "Specification and Description Language (SDL)".
- [i.19] ITU-T Recommendation Z.120: "Message Sequence Chart (MSC)".
- [i.20] ITU-T Recommendation X.680: "Information technology - Abstract Syntax Notation One (ASN.1): Specification of basic notation".
- [i.21] OMG: "Unified Modelling Language™ (UML®)".
- [i.22] OMG: "UML Testing Profile (UTP)".
- [i.23] IEEE Guide to the Software Engineering Body of Knowledge (SWEBOK).
- [i.24] IEEE 1028-2008: "IEEE Standard for Software Reviews and Audits".
- [i.25] "The Elements of UML 2.0 Style", Scott W. Ambler; Cambridge University Press; ISBN: 0-521-61678-6.
- [i.26] "Seven process modeling guidelines (7PMG)", J. Mendling, H. A. Reijers, and W. M. P. van der Aalst; Information and Software Technology Volume 52, Issue 2, February 2010.
- [i.27] "Design and Code inspections to reduce errors in program development", M.E., Fagan; IBM Systems Journal Volume 15, Issue 3, 1976.
- [i.28] "An approach to quality engineering of TTCN-3 test specifications", Helmut Neukirchen, Benjamin Zeiss, Jens Grabowski; International Journal on Software Tools for Technology Transfer (STTT) Volume 10 Issue 4, 2008.
- [i.29] "A Flexible Framework for Quality Assurance of Software Artefacts with Applications to Java, UML, and TTCN-3 Test Specifications", Jens Nodler, Helmut Neukirchen, Jens Grabowski; Proceedings of the 2nd International Conference on Software Testing, Verification and Validation (ICST), 2009; IEEE Computer Society, ISBN 978-0-7695-3601-9.

- [i.30] "On the Application of Software Metrics to UML Models", Jacqueline A. McQuillan and James F. Power; Lecture Notes in Computer Science, 2007; Volume 4364, Models in Software Engineering; Springer.
- [i.31] "Come, Let's Play: Scenario-Based Programming Using LSCs and the Play-Engine", David Harel, Rami Marelly, Springer 2005.
- [i.32] "A Visual Approach to Validating System Level Designs", Jochen Klose, Thomas Kropf, Jürgen Ruf, ISSS'02, October 2-4, 2002, Kyoto, Japan.

3 Definitions and abbreviations

3.1 Definitions

For the purposes of the present document, the following terms and definitions apply:

validation: the process, with associated methods, procedures and tools, by which an evaluation is made that a standard can be fully implemented, conforms to rules for standards, satisfies the purpose expressed in the record of requirements on which the standard is based and that an implementation that conforms to the standard will have the functionality expressed in the record of requirements on which the standard is based

3.2 Abbreviations

For the purposes of the present document, the following abbreviations apply:

7PMG	Seven Process Modelling Guidelines
ASN.1	Abstract Syntax Notation number 1 [i.20]
ATS	Abstract Test Suite
CR	Change Request
CTI	ETSI Centre for Testing and Interoperability
HiperMAN	High Performance Radio Metropolitan Access Network
ICS	Implementation Conformance Statement
IFS	Implementable Functions Statement
IPv6	Internet Protocol version 6
ITS	Intelligent Transport System
IXIT	Implementation eXtra Information for Testing
LSC	Life Sequence Charts
MSC	Message Sequence Chart [i.19]
MV	Member Vote
SDL	Specification and Design Language [i.18]
STF	Specialist Task Force
SUT	System Under Test
SWEBOK	Software Body of Engineering Knowledge
T3Q	TTCN-3 Quality evaluation software tool
TB	Technical Body
TC	Test Case
TC/WG	Technical Committee/Working Group
TD	Test Descriptions
ToR	Terms of Reference
TP	Test Purpose
TPLan	Notation for expressing Test Purposes [i.7]
TSS	Test Suite Structure
TTCN-3	Testing and Test Control Notation version 3 [i.6]
UML	Unified Modelling Language [i.21]
UTP	UML Testing Profile [i.22]
WG	Working Group
WiMax	Worldwide Interoperability for Microwave ACCESS

4 Validation overview

Any standard is a collection of requirements which, if implemented correctly, achieve certain objectives in a product. These requirements could specify physical characteristics such as length, voltage and accessibility or behaviour such as the interchange of messages and the responses to particular stimuli. The purpose of validation is to ensure as far as possible that the requirements expressed in the standard do, in fact, meet the objectives.

The objectives against which a standard may be validated may come from a variety of sources and cover a number of different aspects within the standard. These can include:

- technical objectives related to the technology standardised:
 - the extent to which the requirements can be realized by products;
- editorial objectives:
 - the ease with which the requirements can be read and understood;
 - the extent to which the requirements are expressed unambiguously;
 - the accuracy of references to other standards.

The objectives against which a standard is measured can be validated either directly by, for example, reviewing the specification itself, or indirectly by, for example, evaluating the performance of an implementation of the standard.

5 Reasons for validating standards

5.1 Problems commonly found by validating standards

The purpose of a standard is to specify requirements that are to be met by systems implementing the standard. These requirements need to be unambiguous, precise and consistent.

Standards should follow established principles of standards writing, as follows:

- a well-defined and precise terminology should be used consistently in the standard;
- requirements should be clearly identified as mandatory or optional;
- different configurations and their associated parameters should be well-defined;
- the means of conforming to the standard should be clearly identified; and
- definitions and requirements should not conflict with each other.

Even a simple validation of a standard will identify specification errors that can be easily rectified and that, when corrected, will improve the overall quality of the document. Such improvements make the standard easier to understand and to implement in a product, thus reducing the number of potential interoperability problems that might occur.

Problems found during validation fall into two distinct, though often related, categories which are:

- editorial; and
- technical.

Inconsistencies and other problems in a standard can result from:

- distributed, cooperative authorship of a standard;
- the separation of a standard into multiple parts;
- the evolution of a standard through different revisions;

- the use of multiple specification techniques within the standard.

The resultant synchronization and consistency problems may potentially lead to different interpretations of the standard.

5.1.1 Editorial issues

Editorial problems in a standard may seem to be trivial issues but, if it is poorly expressed or presented, an otherwise excellent requirement can cause confusion to implementers. The result of this is likely to be products which are supposedly standardized but which are unable to interoperate with other similar products.

Problems associated with the editorial correctness of a standard include the following:

- Missing or out-of-date references to other standards
 - The documents identified as normative references in clause 2 of a standard are considered to be part of the standard itself. If the correct standards are not included in this clause or the reference information is out-of-date then associated requirement(s) can be inaccurate or misleading.
- Insufficient, conflicting or incorrect abbreviations and definitions of terms
 - Clear definitions of terms are important in ensuring that the standard can be understood in the same way by all readers. Incorrectly specified or missing abbreviations can cause confusion, at best, and misinterpretation at worst.
- Requirements poorly expressed in English so that they are difficult to understand
 - Although the guidelines for the use of modal verbs ("shall", "should", "may") in standards are very clearly specified [i.16] and understood, they do not prevent requirements from being constructed in a way that is difficult to interpret unambiguously.
- Incorrect syntax used in specification languages (graphic or textual)
 - Languages and notations such as ASN.1 and MSC can make the requirements in a standard completely unambiguous. However, if the syntax of such specifications is not correct (i.e. not checked) then they lose their precision and are open to varying interpretations.

5.1.2 Technical issues

Although likely to be more damaging than editorial problems, inaccuracies in the technical specification are generally less common in standards. It is, nevertheless, still important to find and correct such problems.

Examples of the specification problems include:

- Requirements that are incomplete or omitted from the specification

EXAMPLE 1: *When an implementation receives a REQUEST message containing a value of less than 10 in the COUNT field it shall respond with a VALID_REQUEST message to the sender. If the value of the COUNT field is greater than 10, it shall respond to the sender with an INVALID_REQUEST message.*

The wording of the requirement means that the processing of COUNT values less than 10 and values greater than 10 are specified but there is no specification for how a value of exactly 10 is to be handled.

- Requirements that are implied rather than expressly specified
 - if there is more than one requirement tied together in a single statement, this can lead to confusion and misinterpretation.

EXAMPLE 2: *When an implementation sends a CALL_PROCEEDING message to the calling party it shall also forward the SETUP message to the destination.*

Although this requirement is not wrong, it does not tell the full story as the initial stimulus is implied (the SETUP is forwarded so must have been received) but not clearly specified. A more complete formulation would be:

When an implementation receives a SETUP message from the calling party, it shall respond with a CALL_PROCEEDING message back to the calling party and also forward the SETUP to the destination.

- Requirements that conflict with other requirements in the same or a referenced standard
 - a statement that requires an implementation to take a particular course of action at one point in a standard clearly conflicts with a requirement elsewhere not to take the same course of action. However, a requirement that mandates an action using a "shall" is equally in conflict with a requirement elsewhere that allows that same action to be optional by using "may".
- Incorrect semantics used in a specification language
 - a specification in a language such as ASN.1 or UML may be syntactically correct (and, thus, pass through a compiler) while expressing incorrect semantics. Most language compilers will detect errors in the static semantics of a specification but simulation or live operation may be required to detect errors in the dynamic semantics.
- Requirements that cannot be achieved practically
 - requirements that demand behaviour that cannot be met within the practical constraints of cost and physics will result in implementations that take a best effort approach and, consequently, are unlikely to comply with conformance tests or interoperate with other implementations.
- Requirements that specify unnecessary restrictions on the implementation of the standard (over-engineering)
 - standards should limit their requirements to the specification of observable behaviour at external interfaces. They should specify what is to be observed but not how that should be achieved.
- Large numbers of implementation options which may result in incompatible implementations
 - implementations of standards that specify a large number of optional requirements are likely to experience significant problems in interoperating. Ideally, a standard should specify no options but in most cases, this is impractical. Options should, therefore be kept to a minimum.

5.2 Avoiding problems normally found during validation

Many of the problems found during validation can be avoided by following simple procedures during the standards development process. Activities which can help to avoid common problems in standards include the use of:

- guidelines for structuring and writing a standard [i.16];
- guidelines for using modelling techniques for the technical parts of the standard [i.1], [i.2];
- the use of profile standards to constrain the number of options in a base standard when used in a specific application environment;
- checklists for analysing the correct writing of a standard;
- continuous reviews and feedback during standards writing; and
- validation throughout the development of a standard.

One of the simplest methods for ensuring that each requirement in a standard is, at least, valid is to ask the question "Can the requirement be tested?" If the answer to this question is "No" then it is unlikely that the requirement should be included in the standard.

5.3 Using validation methods to detect problems

The use of even the simplest validation techniques during standards development will help to identify inconsistencies, inaccuracies and other defects which can be resolved before publication and, thus, improve the quality of the output document. There are a range of techniques described in the present document and these will be most effective if used regularly at all stages of the development process.

6 Methods for validating standards

6.1 Different types of standard

Different types of standard require different forms of validation depending on the content of the standard and the methods used to define it. Standards can be classified as follows:

- collections of definitions of terms such as taxonomies and ontology's;
- methodological standards containing mainly text about methods and processes;
- technical specifications typically containing a combination of text and partial models; and
- test specifications defined by text and (partial) test suites for test systems.

The present document differentiates between explicit and implicit validation methods. Explicit validation methods encompass all those in which the primary objective is validation in itself, i.e. reviews, model-based validation, prototyping, and system testing [i.23]. Implicit validation methods encompass those methods where the primary objective is not validating the standard but where validation is a by-product of the activity. Implicit validation methods include requirements cataloguing, test development, product development and documentation.

The selection of validation methods should be guided by the specification technique used within a standard. Text is mainly informal and open for all review-based techniques. Model-based validation is applicable to technical standards and test standards that contain (partial) models and/or test suites only. Furthermore, all standards can be validated by the different implicit validation methods.

6.2 Explicit validation

6.2.1 Review methods

6.2.1.1 Overview of reviewing methods

Systematic reviews are methods of reviewing specifications which, if used correctly, can identify many of the inaccuracies and inconsistencies present in the specifications. They are particularly effective when used to validate the following:

- text and other informally defined specifications;
- sets of related standards where correlation and consistency between documents is essential.

IEEE 1028 [i.24] identifies a range of systematic review types which can be used in the validation of engineering design (specifically, software) specifications. These strictly controlled reviews work well in the large engineering project environment for which they are intended but could be considered as too strict for the standards engineering process. Nevertheless, the fundamental approach to reviewing specifications can be an effective tool in the validation of standards. The types of review identified in IEEE 1028 [i.24] include:

- management review
 - a management study into a project's status and allocation of resources;

- technical review
 - a form of peer review in which a team of qualified personnel examines the suitability of the specification for its intended use and identifies discrepancies from its base requirements and other standards;
- inspection
 - a formal type of peer review where the reviewers follow a well-defined process to find defects [i.27];
- walk-throughs
 - a form of peer review where the author leads members of the editing team and other interested parties through a specification and the participants ask questions and make comments about defect; and
- audits
 - a type of review conducted by personnel external to the specification development project to evaluate compliance with requirements, standards, contractual agreements or other criteria.

Although each review method has its own particular use within an engineering design life-cycle, they all have fundamental similarities in the way they are constructed and managed:

- a group of impartial experts reviews a specification in detail in order to identify any defects that need to be rectified prior to publication
 - these experts may have an interest in the document under review (as members of the responsible TC, for example) but should not have taken any direct role in its development;
- one of the expert reviewers is appointed to chair and manage the review process;
- any required changes are undertaken by the authors outside the review;
- the process is iterative such that the revised document is resubmitted for further review until no further changes are required.

6.2.1.2 Using reviews to validate standards

Review methods are unlikely to yield the same depth of validation that an automated, model-based technique can offer (see clause 6.2.2). However, reviews have the benefit that they can be used to validate any type of standard and do not depend on there being behavioural content within the specification.

It is certain that the strict, systematic methods described in IEEE 1028 [i.24] will identify a significant proportion of the defects in a design specification and this approach can be applied to the standardization process very effectively. Despite this advantage, it is possible that the overheads in terms of effort and timescale will be unacceptable within the development schedules of some standards. In those cases, reviews can be used less formally although this is likely to be less effective in identifying defects.

The process of approving standards using the ETSI Remote Consensus procedure and the scrutiny of editorial consistency offered by the ETSI EditHelp service are both useful and valuable adjuncts to the more formal peer review process but they cannot be considered to be substitutes for it.

6.2.1.3 Basic review process

The process of organising and carrying out the review of a standard involves the same common set of steps regardless of the formality intended. This process, shown in Figure 1, comprises 4 distinct phases, as follows:

- 1) Planning.
- 2) Preparation for the review.
- 3) Document review meeting.
- 4) Processing change proposals.

NOTE: The processing of change proposals (Phase 4) takes place outside of the review process itself but it is included here as an essential part of the iteration loop.

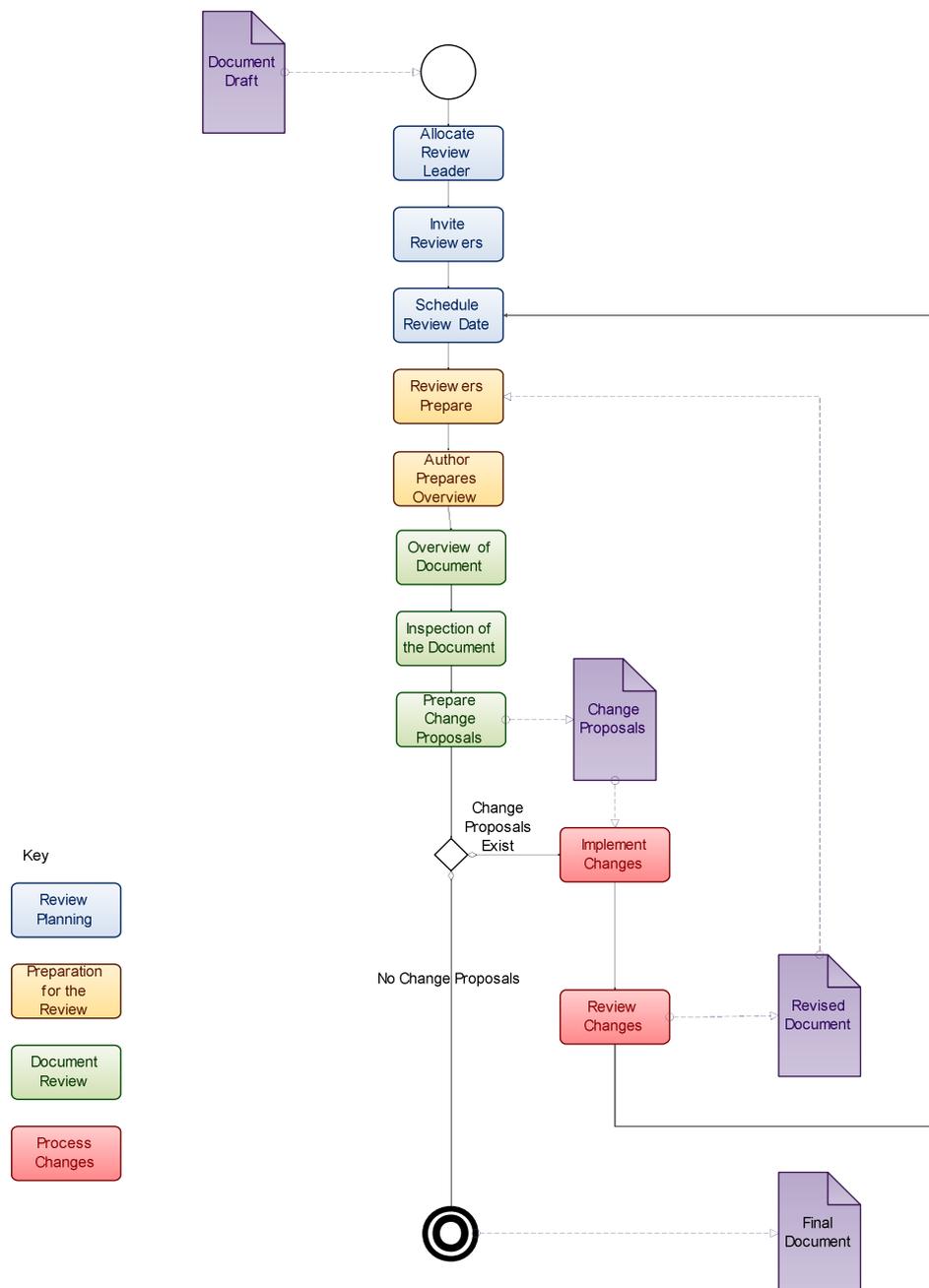


Figure 1: Document review process

6.2.1.3.1 Phase 1: Planning the review

The planning phase of a review can start as soon as it is known when a draft of the document to be reviewed will be available. It is not necessary to wait until the draft, itself, has been completed. This phase of the process involves the following three steps:

- Assign a leader for the review process.
- Select and invite the reviewers.
- Schedule a date for the review.

6.2.1.3.1.1 Assign a leader for the review process

When selecting an individual to lead a review group it is not necessary to choose an expert in the subject under review, although this can be useful. It is much more important to find a leader who is:

- impartial
 - no commercial, political or technical bias with respect to the document under consideration and, thus, unlikely to influence the outcome of the review unduly;
- authoritative
 - able to ensure that the review is successful by encouraging active participation by the reviewers, maintaining focus on the review subject and fostering consensus;
- a good organizer

In most cases, the review leader could be one of the following:

- the Chair of the associated ETSI Technical Body (TB) or Working Group (WG);
- the ETSI Technical Officer supporting the associated TB or WG; or
- the document rapporteur if that person is not one of the contributing authors. This could be the case if, for example, the document is a deliverable from a Specialist Task Force (STF).

6.2.1.3.1.2 Select and invite the reviewers

Each reviewer of a document should be chosen on the basis of particular knowledge or expertise that they possess. It is unlikely that a single reviewer will have all the necessary skills but, between them, the reviewers should be conversant in all aspects of the technical content of the document under review and in the process for producing the document. The selected reviewers should, therefore, be:

- independent
 - not one of the contributing authors;
- knowledgeable
 - capable in (at least part of) the subject area of the document or in the associated standards development process;
- assigned specific responsibilities within the context of the review
 - each should know why they have been selected as a reviewer and which clauses that they should pay particular attention to.

There is no easy formula for determining how many reviewers should be selected but a smaller group is likely to be more effective than a large one. The size of the reviewers group will depend on a number of factors including:

- the quantity of material to be reviewed;
- the breadth and depth of the technical content;
- the availability of suitable reviewers.

In many cases the membership of the responsible WG should be sufficient but, if this is a large group with a frequently changing membership then it may be necessary to appoint a sub-group for the reviewing task.

6.2.1.3.1.3 Schedule a date for the review

Establishing a date and time for the review to take place should not be a time-consuming task but the following criteria are important:

- the date should be convenient for the leader, the reviewers and the author(s) of the document;

- a sufficient period should be planned between the availability of the document and the review, itself
 - this period will depend on the size of the document and the other commitments of the reviewers but it should be a minimum of two weeks but not so long that it affects progress in the development of the standard;
- enough time is scheduled to complete the review
 - this, too, will depend on the size of the document to be reviewed and also on its maturity but it is unlikely to take less than half a day and, for a large and immature document, could be two or more days. A previously unreviewed draft document is likely to require more time than one that is in its final stages before publication. Frequent, incremental reviews throughout the development process are, therefore, a more efficient use of reviewers time and a more effective method of reviewing a draft standard.

6.2.1.3.2 Phase 2: Preparing for the review

Both the authors of a document and its reviewers have responsibilities immediately prior to the review. Once the draft is available for the reviewers to study, they should:

- ensure that they understand the requirements and the overall objectives of the document and its related project;
- read the document (or parts of the document) to be reviewed;
- prepare constructive comments and questions to be put to the authors during the review.

NOTE: There is no requirement for reviewers to record their comments in any specific format. The content of the comment is more important than its presentation form. However, the simple form shown in Table A.1 in Annex A can be used to capture the essential information associated with a reviewers comment.

At the same time, the authors should prepare a short presentation summarizing:

- the purpose of the document under review;
- the relationship of the document with other published or planned documents;
- the status of the draft;
- any new material added since its previous review;
- any modifications since its previous review;
- any special methods used during development;
- objectives of the impending review.

6.2.1.3.3 Phase 3: Reviewing the specification

A document review meeting will always be most effective if all the participants meet together in a common location. However, little effectiveness is lost if some or all of the participants join using remote meeting facilities.

There are three distinct activities involved in the actual review of the draft document. These are:

- presentation of the overview from the document author(s);
- inspection of the document;
- preparation and agreement of any required change proposals.

6.2.1.3.3.1 Presentation of the overview from the document author(s)

At the start of the review meeting one of the document authors should present the overview previously prepared during Phase 2. Purely as a guideline, this presentation will probably lack sufficient detail if it lasts less than 15 minutes and will contain too much information if it goes on beyond 30 minutes. At the end of the presentation the authors should be prepared to answer questions for clarification from the review leader and the group of reviewers.

6.2.1.3.3.2 Inspection of the document

The purpose of inspecting a document is to identify possible defects where either technical or editorial requirements have not been met. The most effective inspection method is to consider comments from the reviewers on a clause-by-clause basis but some flexibility should be allowed.

In order to ensure efficient progress through the inspection, reviewers comments should be:

- fully justified
 - enough information is provided to make it clear that the comment is valid;
- impersonal
 - comments should relate only to technical or editorial defects and should avoid personal comments regarding the authors;
- supported by a proposed change
 - comments should identify a possible defect in the reviewed document without offering an alternative solution for the authors to consider.

6.2.1.3.3.3 Preparation and agreement of any required change proposals

Each change proposal should be discussed and either accepted or rejected by the review group. If accepted, the change should be assigned to one (or more) of the document authors and scheduled for inclusion in the next or a future revision of the draft.

It is particularly important that all accepted change proposals are recorded so that the resultant modifications to the document can be considered in a subsequent review. The Reviewers Change Proposal form specified as Table A.1 in Annex A can be used for this purpose. Alternatively, a revision of the reviewed draft document with change marks and comments would be a valid record of all accepted change proposals.

If there are no change proposals accepted at the end of the review then the document can be considered to have completed its validation by peer-review.

6.2.1.3.4 Phase 4: Processing change proposals

6.2.1.3.4.1 Implement changes in a document revision

If there are accepted change proposals at the end of the document review meeting, the authors should consider each of these and make the necessary modifications to the document.

The task of implementing the accepted change proposals is outside the basic review process and should follow whatever method is normally used for revising draft documents within the associated TB.

6.2.1.3.4.2 Review actual changes

Before the revised document is resubmitted for review, the author(s) should discuss each of the implemented changes with the review leader to gain an independent opinion of whether the accepted change proposals have been implemented correctly. At this stage the review leader should expect the authors to justify any deviations from the accepted change proposals from the previous review.

It may also be valuable at this stage to present the revisions to the WG or TB responsible for the document.

6.2.2 Model-based validation methods

It is possible to validate standards that specify behaviour (protocol and service standards, for example) more thoroughly by using model-based methods. These require at least part of the specified system to be defined in a formalized way using a fully-fledged general modelling technology (for example, SDL or UML) or a specific, partial modelling technique for selected aspects or views of the system (for example, ASN.1 or MSC).

Model-based validation techniques such as simulation and state-space exploration offer the most rigorous and thorough validation of a protocol or service specification. However, both techniques require specific expertise and extensive resources to develop the requisite models and their application is further hindered by the non-existence of powerful tools able to cope with the modelling concepts required for (industrial) standards.

Any model-based validation method can be effective only if the following criteria can be met:

- parts of the standard are (or can be) modelled;
- the proposed modelling techniques are standardized;
- automated specification and analysis tools exist to support the proposed modelling techniques.

If these criteria cannot be met then the review methods (see clause 6.2.1) and product-based methods (see clause 6.2.4) are likely to be the most effective means of validation.

There are few modelling methods that are both standardized and supported by automated analysis tools. Apart from the more established methods such as SDL, MSC and UML, there are new methods available for the validation of a standard using significantly fewer resources. Practical model-based validation methods include modelling and simulation using languages which include:

- Specification and Design Language (SDL) [i.18];
- Unified Modelling Language (UML) [i.21];
- Message Sequence Charts (MSC) [i.19];
- Life Sequence Charts (LSC) [i.31], [i.32]:
 - a dialect of the MSC language which is already commonly used as a specification method within protocol and service standards.

6.2.2.1 Modelling and simulation

Scenario-based specification of system behaviour is often used in the development of technical standards where informal flow graphs and comparable techniques are used.

The Specification and Design Language (SDL) [i.18] is one of the few modelling techniques that is standardized and its use in validation is described in ETR 184 [i.13]. However, although SDL design and simulation tools are available, there are no state-space exploration tools currently supported. In general, model simulation implies validation by inspection of the output of the simulation which is most often presented as MSCs [i.19]. This is a useful method of validation but does not have the rigor of the methods described in ETR 184 [i.13].

Message Sequence Charts was one of the first formal specification techniques for visual scenario specification to be standardized in ITU-T Recommendation Z.120 [i.19]. UML sequence diagrams are based on MSC and have similar syntax and semantics. The Life Sequence Chart methodology is a development of MSC which supports the visual specification and simulation of reactive systems. They go beyond MSCs by defining behavioural properties, i.e. "things that must occur". The additional LSC constructs are:

- scenarios:
 - universal scenarios:
 - describe system behaviour that applies in all instances;
 - specify action-reaction steps of the specified system;
 - existential scenario:
 - monitored and observed during simulation;
- pre-charts:
 - define preconditions for the main charts.

LSC specifications help to define and analyse potential behavioural scenarios for implementing technical requirements within a standard.

6.2.2.2 Model quality assessment

A number of guidelines exist which define best practises for modelling system, component, service, protocol, process and other behaviour-oriented specifications. Some are defined for a specific modelling notation while others are more general and relate to the entity to be modelled. Useful examples of such guidelines are those published by Scott W. Ambler [i.25] and J. Mendling et al [i.26].

These guidelines can be followed manually during reviews (see clause 6.2.1) or by static analysis tools which provide a thorough, complete and automated method of guideline checking.

The quality of a model can also be validated by checking for the presences of modelling patterns that represent modelling practices to be followed and to check for the absence of modelling anti-patterns reflecting bad modelling styles. Modelling patterns and anti-patterns can be rephrased as modelling guidelines; however the reverse is typically not true.

Useful examples for modelling patterns and anti-patterns are defined by Helmut Neukirchen et al for test suites in TTCN-3 [i.28] and a flexible overall approach to model quality [i.29] is described by Jens Nodler et al.

Furthermore, model quality can be assessed with metrics that represent the complexity, readability, maintainability, coherence, and similar properties of modelling artefacts. Metrics are typically derived automatically by tools as a manual calculation would be cumbersome and prone to errors [i.30]. Such metrics are useful typically if large portions of a standard are specified or supported by models.

6.2.3 Test suite validation

A standardized conformance or interoperability test specification should be validated using similar methods to those for other standards. Such a specification typically comprises [i.4], [i.17]:

- a set of test purposes such as those specified in the TPLan notation [i.7];
- the definition of a test suite structure and of the implementation conformance/interoperability statement and the extra information for testing (often defined textually);
- an Abstract Test Suite (ATS) which is most often specified in TTCN-3 [i.6].

There are three levels of validation defined for TTCN-3 test specifications (see Annex B), as follows:

- 1) The test suite successfully analyses/compiles on one or more TTCN-3 test tools. Optionally, a design review of the test suite is performed.
- 2) The tests are executed on one test platform against at least one System Under Test (SUT). Where possible, tests will be run to completion. Optionally, a back-to-back validation (mirror test cases) and/or a data-driven codec validation are performed.
- 3) The tests are executed on several test platforms against several SUTs. Tests will be run to completion and trace analysis will be done.

Further details on the validation processes for the three levels are given in Annex B of the present document.

6.2.4 Product-based validation (proof-by-use)

6.2.4.1 ETSI Plugtests™ events

6.2.4.1.1 Benefits of using a testing events for validating standards

An ETSI's Plugtests™ event is an organized event at which prototype or production implementations of a standard (or standards) are interconnected and tested for both interoperability and possibly conformance. Such an event provides validation of both the base standard(s) and the implementations of the standard(s).

A Plugtests™ event follows a pre-defined sequence of tests which is specified to provide either a comprehensive validation of the standard(s) or a more focussed one on a smaller part of the standard. This flexibility makes testing events very effective methods of validation even at an early stage of development.

There are a number of benefits to be gained from a validation exercise based upon ETSI Plugtests™ events. These are:

- they provide validation of the standard(s) as well as a measure of interoperability capabilities and general behaviour of the implementations;
- simultaneous validation of a set of standards (or multiple related standards) can be achieved more rapidly and with greater accuracy than with "desk-top" methods such as peer reviews;
- the sequences of tests can be arranged to target validation of specific functionality within the standard(s);
- problems associated with implementation (for example, ambiguity of requirements, lack of clarity in the specification and over-complexity) are easier to locate because product must, necessarily, have implemented the standard(s) before the event can take place;
- a single event can achieve in one week of cooperative testing what might otherwise take many months.

6.2.4.1.2 Criteria for selecting a testing event for validation

While testing events are very effective standards validation tools, they are not appropriate for all types of standards and the following criteria should be considered when determining whether to instigate validation based upon testing events:

- the standard(s) in question should specify some form of communications functionality
 - Plugtests™ events are usually based on the testing of intercommunication and interoperability and so they are particularly suited to the validation of protocol, service and transmission standards;
- implementations of the standard are available for testing
 - product development by two or more organizations should be proceeding either in parallel with the standards or even in advance;
 - product developers should be willing to take prototypes and development products to a testing event;
- the complexity of the standardized specification should be sufficient that it would be difficult and time-consuming to validate it purely by inspection and review
 - testing events are particularly well suited to the validation of sets of related standards or groups of related standards that do not necessarily depend upon each other (for example, groups of supplementary services or applications).

If all of these criteria can be met then a testing event should be considered as a validation method.

6.2.4.1.3 Overall process

6.2.4.1.3.1 Responsibilities

Once a decision has been made by a TB to use an ETSI Plugtests™ event as part of its validation process, there is a partnership that needs to be established between the various groups that will be involved in organizing, managing and participating in the event. Each of the partners has specific responsibilities, as follows:

- ETSI Technical Body:
 - defining the scope of testing with respect to validation of the standard(s);
 - specifying the capabilities required from the Systems Under Test (SUT);
 - providing knowledge of the standard(s);

- specifying the interoperability and, if appropriate, conformance tests to be undertaken during the event:
 - automated test suites are capable of testing a greater proportion of the standardised requirements in a shorter time than equivalent manual testing. They are also able to produce more information on any failures that may occur;
- Industry:
 - defining the scope of testing with respect to the evaluation of the products implementing the standard(s);
 - providing SUTs that offer the capabilities required for testing;
 - providing expertise in the standardized technology and the SUTs;
- ETSI Centre for Testing and Interoperability (CTI):
 - providing expertise on testing methods and specifications;
 - providing technical and IT support before and during the event:
 - general management;
 - project coordination;
 - providing the testing and communications infrastructure;
 - providing expertise in the interconnection and operation of products within the communications infrastructure;
 - providing logistics support in organizing and managing the event;
 - maintaining an operational link to the TB.

6.2.4.1.3.2 Factors to consider in making a Plugtests™ event successful

A Plugtests™ event can only be a successful validation exercise if all the partners contribute to the preparation and execution of the event. There is no simple formula for determining how long it will take to prepare for a Plugtests event as this period will vary considerably depending upon:

- the maturity of the technology being standardized;
- the complexity of the standard specifications;
- the availability of a range of suitable SUTs;
- the availability of infrastructure resources;
- the availability of logistical resources such as an organiser within ETSI CTI and a suitable venue.

The chart in Figure 2 shows the relationships between the activities involved and the responsibilities for each task.

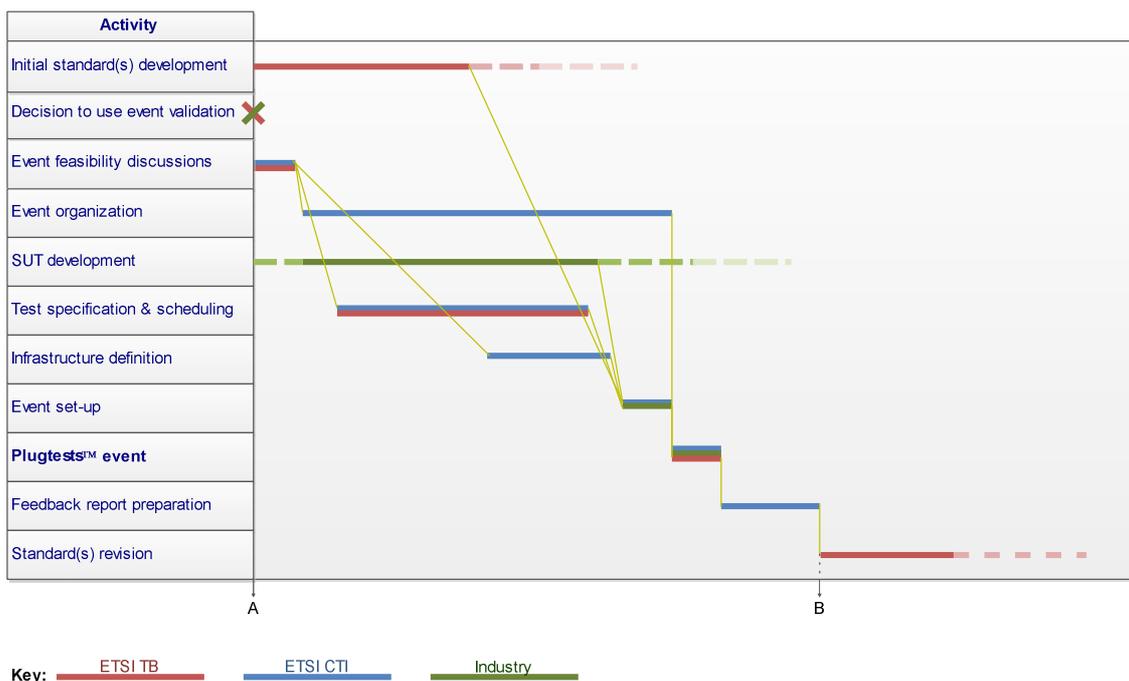


Figure 2: Generic schedule for a Plugtests™ event

In some cases it will be possible to save time and money by having the SUTs communicating remotely over a public or other wide-area network provided by ETSI CTI. Such an arrangement can make the synchronization and management of test activities more difficult but this can be outweighed by the fact that equipment does not need to be transported to a common location in order to carry out the testing. However, it is usually considered essential to have representatives from each of the participating organizations present together in order to coordinate the testing and discuss the results.

Most testing events focus on the interoperability of the SUTs because it is easier to demonstrate the general behaviour of a product in respect of the standard(s) it has implemented by taking this approach. Nevertheless, the effectiveness of the event for validation purposes will almost certainly be improved if some, or all, of the SUTs have successfully undergone some conformance testing before the event.

Information will be collected during the testing event and then compiled into a report for presentation to the "client" TB afterwards. Care is taken to ensure that this report is anonymous in that it does not identify the participating organizations nor any defects discovered in their products during the event. The primary purpose of the report is to identify and describe any defects found in the standard(s) during the event so that they can be rectified by the TB in a subsequent revision of the document(s).

The time between the decision to use an interoperability event to validate the standard(s) and the completion of the feedback report (points A and B on the plan shown in Figure 2) can vary considerably depending upon a number of factors. As examples, the time between points A and B can be as short as 3 months if:

- the underlying technology is mature;
- the standards, themselves, are mature and have already been published in earlier versions;
- products implementing the standard(s) already exist; and
- interoperability events have already been used to validate previous versions of the standard(s).

However, it may take as long as 2 years to realize an event if:

- the technology is novel and little experience of it is available;
- the first edition of the standard(s) is under development within the TB and unpublished; and
- industry is developing product implementations but do not have them available for testing.

6.2.4.2 Prototyping

While the use of testing events is an extremely good way of validating protocol and service standards if product implementations are available, when the subject matter of a standard concerns physical characteristics rather than behaviour, it is a less effective method. In such cases, the only validation method which should be considered as an alternative to walk through is prototyping. This involves the implementation of a design specification in a small number of representative, rather than commercial, pieces of equipment for the purpose of demonstrating the feasibility of a concept or the correctness of the specification.

The fabrication of prototype models usually requires highly skilled technicians working in a laboratory environment. As a result, the cost of prototypes can be high and, in some cases, prohibitive. These costs normally make it impossible for ETSI to fund such implementations of its standards for validation purposes and it is, therefore, a method that can only be used if a member organization (or group of organizations) is willing to undertake the work.

The use of a prototype model to validate a standard or set of standards, involves the following steps:

- a clear agreement is reached on which parts of the standard are to be modelled in the prototype;
- a full schedule of tests and observations to be performed and the expected results is prepared and agreed before prototype validation begins;
- prototype models are built using parts and methods that ensure that the models are completed as quickly and as accurately as possible within the constraints of the available resources;
- testing is performed according to the agreed schedule;
- a clear report is produced giving details of:
 - problems found in the specification during manufacture and system testing;
 - the actual results of validation testing;
 - the impact of the results on the standard;
- the results are reviewed by the responsible TB and actions assigned to ensure that any necessary changes to the standard are completed.

This is likely to be an iterative process where the modifications to the standard as a result of the validation exercise are incorporated into the prototype model(s) and re-tested. When planning to use prototyping as a means of validating a standard, the following aspects should be taken into consideration:

- prototypes can be used to model partial specifications as well as complete specifications;
- existing commercial products can be used as platforms for testing prototype functions;
- it is essential that the organization(s) carrying out the building and testing of prototype models share the results of the project with the other members of the responsible ETSI Technical Committee.

6.3 Implicit validation

6.3.1 Requirements cataloguing

The specification of a broad technology often takes place across multiple standardization bodies and so the set of requirements to be implemented comes from a range of documents produced in disparate styles. Building a coherent set of test specifications from such disperse sources can be simplified by gathering the requirements together into a single requirements catalogue [i.5]. This lists all implementation requirements from the various sources and organizes them into an appropriate structure. It is often created as a tree structure based upon standardized functionality with each node of the tree representing a specified function.

Although the primary purpose of a catalogue is to collect together the base requirements prior to the development of conformance and interoperability test specifications, the process of collating it can have considerable benefits in validating the catalogued standards. The development of a catalogue involves close analysis of the text in the base standards in order to find:

- explicit implementation requirements;
- implicit implementation requirements;
- duplicated requirements;
- contradictory requirements; and
- incomplete or otherwise incorrect requirements.

The nature of the cataloguing process in recording all requirements means that very little additional effort is required to benefit from the coincidental validation. All that is required is to report back to the originating committee on the implicit, duplicate, contradictory and incorrect requirements so that they can be improved in a later release of the standard(s).

Examples of published requirements catalogues can be found in TS 102 558 [i.9] (IPv6 security) and in TS 102 795 [i.11] (digital private mobile radio).

6.3.2 Test specification development

The development of test specifications can only commence when there is a collection of testable requirements (see also clause 6.3.1) available. A test specification developed under the methodologies in use at ETSI for conformance testing [i.17] and interoperability testing [i.4] should include the following documentation:

- an Implementation Conformance Statement (ICS) or an Interoperable Functions Statement (IFS);
- a Test Suite Structure (TSS);
- a set of Test Purposes;
- an Abstract Test Suite (ATS) comprising either (or both):
 - Test Descriptions (TD);
 - Test Cases (TC);
- an Implementation eXtra Information for Testing (IXIT) statement.

The process of producing either interoperability or conformance test specifications requires close inspection of the standard(s) to be tested. Each of the requirements needs to be located and analysed individually and in combination before tests can be written for them. This will result in a level of validation similar to that offered by a rigorous peer review (see clause 6.2.1).

6.3.2.1 ICS and IFS

The ICS and the IFS are similar questionnaires on the optional and mandatory features specified for an implementation of a given standard. They are used to determine whether all mandatory features are implemented and which optional ones. The development of the ICS or IFS helps to check that all mandatory and optional features are clearly stated and well described. The questions in the implementation statement should be linked to the requirements in the base standard and, if it exists, the requirements catalogue.

Guidelines on the specification of an ICS and an IFS are described in ETR 212 [i.14] and EG 202 237 [i.4] respectively.

6.3.2.2 Test Suite Structure

The TSS defines the grouping of tests in the test suite to be developed. It is usually specified as a tree of nested groups of tests and its structure will depend on the nature and properties of the standard to be tested. For example, the structuring into valid and invalid test cases on a "per feature" basis could be used for protocol testing [i.15]. The development of the TSS requires close inspection of the functional grouping of requirements (such as normal and exceptional behaviour) specified in the base standard and can help in checking that they are well-defined.

6.3.2.3 Test Purposes

The TPs describe the purpose of each potential test case in determining whether the requirements in the base standard or the requirements catalogue have been met. The development of TPs involves the identification and understanding of each of the requirements in the base standard even if there is no requirements catalogue available to simplify the task. This process indirectly reviews the completeness, accuracy and comprehensibility of the base requirements.

Examples of published TPs can be found in TS 102 385-2 (WiMax/HiperMAN) [i.8], TS 102 593 (IPv6 security) [i.10] and TS 102 870-2 (ITS basic transport protocol) [i.12].

6.3.2.4 Test Suite

The test suite itself defines the functional details of the test which are outlined in the test suite structure and test purposes. It is the basis for automated conformance and interoperability testing (see clause 6.3.3), but can also serve for manual testing where needed. The development of test cases necessarily involves detailed analysis of the individual requirements in the base standards and the requirements catalogue. In this way it is possible to determine whether the requirements have been expressed precisely enough in stating how an implementation should behave. Automated test cases are usually specified in the TTCN-3 language [i.6] and this formalized approach can locate other inconsistencies, ambiguities and open issues in the base standard.

6.3.2.5 IXIT

The IXIT poses questions related to the implementation which are necessary for test execution. It is a means of determining that all configuration and parameterization options of an implementation are well-defined and correctly captured in the test specification as well as in the standard.

6.3.3 Conformance and interoperability testing

Conformance testing is the process of establishing the extent to which a System under Test (SUT) satisfies conformance requirements as defined in the Implementation Conformance Statement (ICS). It ensures that a product correctly implements the standard and is able to exchange directives and information with another implementation using a known protocol or set of protocols. Testing the conformance of a product to a standard is considered to be essential in ensuring that the product is able to inter-operate with other products that implement the same standard or a complementary one.

ETS 300 406 [i.15] describes a methodology which should be followed when producing conformance test specifications. A conformance test consists of two parts [i.17]:

- the static conformance review checking whether the choices between the implementation options that the manufacturer claims to have implemented is a combination permitted by the standard;
- the dynamic conformance test using test case execution to determine whether the product has implemented the standard correctly.

Interoperability is the linking of systems, networks or services so that they can work together successfully. Interoperability testing is achieved by connecting devices from different vendors and operating them, either manually or automatically, according to scenarios based on a protocol standard. Interoperability testing is often performed at multi-vendor events in order to obtain valuable information about the capability of inter-operation of similar products. This is an approach often employed in an ETSI Plugtests™ event at various stages during the development cycle (see clause 6.2.4.1).

Conformance testing and interoperability testing are complimentary to each other. Each has its strengths and weaknesses but together they can provide the best assurance that products will interoperate.

Each test case in a conformance or interoperability test suite is related to one or more requirement of the base standard. The execution of a standardized test suite against an implementation of the base standard can help to validate the test suite itself as well as providing a final validation of the standard by exercising and observing the functionality of an implementation. In this way it is possible to identify requirements that have been specified accurately and unambiguously but which define unwanted or unexpected behaviour.

6.3.4 Development of descriptive and guidance documentation

Although it is not usual for guides or tutorials to be published for simple standards, it can be useful to develop such documentation for standards that specify more complex technologies, particularly when the specification spans a number of different standards. Such guides include:

- technology overviews;
- implementation handbooks;
- design guides;
- future standardization proposals; and
- testing frameworks.

The development of descriptive and guidance documentation requires a good understanding of the standardized technology and, as with a requirements catalogue (see clause 6.3.1), a close inspection of the base text. Again, little additional effort is required for the validation to be quite thorough but is important.

6.3.5 Product development

In much the same way as prototype development can be used a means of validating a standard (see clause 6.2.4.2), the development of marketable products which implement the standard can also provide a measure of validation. However, this approach may not be as effective for the following reasons:

- validation of the standard is coincidental rather than the main objective of the exercise. Consequently:
 - there may not be a predefined schedule of validation tests to be carried out during development;
 - there is less likelihood that necessary changes to the standard will be identified and reported back to the responsible TB;
- correcting and reworking the standard at this stage is likely to be more costly than if validation had been undertaken explicitly earlier in the process.

6.4 Validation best practice

6.4.1 Identify the most appropriate validation level

6.4.1.1 Defined validation levels

It is not practical to consider using all of the validation methods specified in the present document but it is useful to consider validation at different levels. In this way it may be easier to determine which combination of validation methods is most appropriate for each standard or set of standards.

Validation of standards can be defined in three levels where Level 1 represents the basic validation expected for a **standard and Level 3 the most rigorous. These levels are defined as follows:**

- Level 1 validation (Basic):
 - informal review of the standard(s) has taken place within the responsible committee or working group;

- all formal language segments have been syntax checked by software tools:
 - ASN.1;
 - SDL;
 - UML:
 - class diagrams;
 - sequence diagrams;
 - state diagrams;
- the standard has been approved by the TC/WG using the ETSI Remote Consensus procedure;
- the editorial consistency of the standard has been scrutinized by ETSI EditHelp service; and
- for standards that specify protocols or services only, a Plugtests™ event has been held during the development of the standard. Only informally specified tests were executed.
- Level 2 validation (Strong):
 - Level 1 validation;
 - formal peer reviews have been undertaken throughout development of the standard; and
 - at least one of the following implicit validation method(s) have been undertaken:
 - a test specification has been developed;
 - a requirements catalogue has been compiled;
 - a design guide or other supporting documentation has been produced;
 - one or more implementations of the standard have been incorporated into commercial products; or
 - the behaviour specified in a protocol or service standard has been examined using one or more of the following methods:
 - behaviour models have been specified (using SDL, UML or other modelling language), compiled and simulated and the resultant message sequences inspected;
 - one or more prototype implementation has been developed specifically for the purpose of evaluating the standard;
 - an additional Plugtests™ event based on more structured test specifications has been organized towards the end of development.
- Level 3 validation (Rigorous):
 - Level 2 validation; and
 - the behaviour specified in a protocol or service standard has been evaluated using one or more of the following methods:
 - model-based validation tools have been used to assess the behaviour, integrity and completeness of the specification;
 - implementations of the standard have been evaluated in a programme of ETSI Plugtests events using an evolving set of structured tests throughout the development period.

NOTE: The validation levels described here are intended primarily for use with respect to base (i.e. non-test) specifications. An established definition of validation levels related to test specifications can be found in Annex B.

6.4.1.2 Select the required method(s)

The most effective and appropriate method of validating a standard will depend upon a number of different factors which include:

- the content of the standard, for example:
 - protocol and service standards can be validated using all of the methods specified in the present document;
 - standards specifying physical characteristics can be validated most easily by peer review or prototyping;
- the complexity of the specification, for example:
 - a standard specifying a simple protocol may not justify the expense of a Plugtests™ event;
 - a complex technology specified across a number of standards may be difficult to validate fully in a peer review (even though the individual parts would benefit from separate reviews);
- the availability of personnel with the necessary skills and experience;
- the availability of the required resources;
- the personal preferences of the rapporteur and technical committee responsible for the standard(s).

These and other factors should be taken into consideration when making the choice of validation method. There is not, however, a right and wrong answer regarding this choice and it may be effective to select more than one method (a combination of explicit and implicit validation methods for example). It is important to choose methods that are effective and which the responsible technical committee is willing to use.

6.4.1.3 Plan validation into the standards development process

When considering the development of a standard or a set of standards it is useful to know not only what the specific technical requirements are which need to be met by the contents of the documents but also what validation activity is to be applied to the finished drafts. It follows that the validation of a standard should be planned together with its development. The definition of a comprehensive validation plan should give consideration to the following aspects:

- what type of validation will be used?
 - peer review;
 - model-based;
 - Plugtests™ event;
 - implicit validation (through test development etc.);
 - other;
 - a combination of methods; or
 - none;

NOTE 1: The type of validation depends on the expected results of the standard development process. If the resultant deliverable will specify function or behaviour, the object of validation may be an executable validation model. In that case formal validation supported by automated software tools should be considered in the plan.

- what are the prerequisites for validation?
 - standards;
 - guidance material;
 - reporting pro-formas;

- procedures;
- resources:
 - implementations;
 - testing facilities;
 - personnel for testing or reviewing;
 - office or lab space;
- when should validation activities take place?
 - in relation to the development of the standard;
 - in relation to the availability of related standards;
 - in relation to the availability of products implementing the standard(s);
 - at various points within the development or once only;
- what equipment is necessary for successful validation?
 - software for model development and simulation;
 - processing platforms for validation tools;
 - special test equipment;
- what will be the output of the validation activity?
 - validation results;
 - change requests.

It will not be necessary to plan all of the above items into the development schedule and tasks and resources necessary will depend upon the validation methods selected. The list is intended to act as an "*aide memoire*" at the start of the standards development process to ensure that the most effective and efficient approach to validation is chosen and integrated into the overall activity.

EG 202 107 [i.3] contains some useful algorithms for estimating the amount of effort required for many of the validation methods described in the present document.

NOTE 2: EG 202 107 [i.3] refers only to the use of the Specification and Design Language, SDL [i.18], for model-based validation but its guidelines on estimating effort should apply equally well to other approaches.

6.4.1.4 Resolve problems detected during validation

One of the activities that is common to all of the validation methods described in the present document is the reporting of problems found in the validated standard back to the responsible technical committee. There is no single problem reporting system for ETSI documents and each technical committee manages change requests (CR) in different ways. Any problems or improvements identified during the validation process should be reported using whatever problem management system is in common use within the committee responsible for the standard. It is then the responsibility of that committee to review each CR, accept or reject it, respond to the originator and, if required, devise a change to the standard.

Annex A: Sample review forms

A.1 Introduction

Although there is no mandate in the present document to use specific forms to record decisions, review comments and change proposals, the following tables can be used for this purpose if considered to be useful.

A.1.1 A form for recording reviewers comments

The form shown in Table A.1 can be used by document reviewers to record comments and possible solutions prior to a review meeting.

Table A.1: Reviewers comment form

Document Review Comment					
Work Item:		Version:		Review Date:	
Comment No:				Clause:	
Reviewer:					
Comment:					
Proposal:					
Accepted <input type="checkbox"/>	Rejected <input type="checkbox"/>	Assigned to:		For Version:	

Annex B: Test Suite Validation Guidelines

B.1 Introduction

The following definitions of validation levels for standardized test specifications have been developed by the ETSI Centre for Testing and Interoperability (CTI). The intention here is to provide guidance to Technical Bodies on:

- the selection of a validation level for a particular test suite to be developed;
- the completion of the validation aspects within the Terms of Reference (ToR) for a Specialist Task Force (STF) to develop the test suite; and
- the construction of a validation report.

B.2 Validation levels

B.2.1 Level 1 validation (Basic)

B.2.1.1 Description

The aim of Level 1 (Basic) validation is to provide reasonable confidence that the test suite is correctly constructed and is capable of executing. The characteristics of Level 1 are as follows:

- the test suite has been compiled on at least one TTCN-3 test tool (preferably more than one);
- the tests have not been executed against real Systems Under Test (SUT);
- no assurance is given that the tests will execute on an industrial test platform;
- it is anticipated that industry will implicitly validate the execution of the test suite and provide feedback to ETSI.

The quality of a test suite validated in this way cannot be guaranteed to be high but this can be improved over time with good feedback from industry. Level 1 validation can usually be achieved at a relatively low and predictable cost.

B.2.2 Level 2 validation (Strong)

B.2.2.1 Description

The aim of Level 2 (Strong) validation is to provide reasonable confidence that the test suite is correctly constructed and that it tests the core requirements effectively when executed with a representative System Under test (SUT). The characteristics of Level 2 validation are as follows:

- the test suite has been compiled on at least one TTCN-3 test tool (preferably more than one);
- the test suite has been executed with a small number of SUTs (possibly only one SUT);
- it is unlikely that all the tests have run to completion;
- output traces from the test execution have not been analyzed.

Level 2 validation raises the quality of the test specifications and gives some assurance that they will compile and execute on an industrial test platform. The costs involved in this level of validation will be greater than at Level 1 and, due to the nature of testing, will be more difficult to predict accurately.

B.2.3 Level 3 validation (Rigorous)

B.2.3.1 Description

The aim of Level 3 (Rigorous) validation is to provide assurance that the complete test suite is correctly constructed and that it tests the full range of requirements effectively when executed in any combination of test system and SUT. The characteristics of Level 3 are as follows:

- the test suite has been compiled on more than one TTCN-3 tool;
- the complete suite of tests has been implemented and executed on more than one test platform;
- the complete suite of tests have been executed against SUTs from a range of different suppliers;
- the operation and output traces of all the tests have been validated.

Level 3 validation guarantees that the quality of the whole test suite will be high but it can add significantly to the cost of the test specifications.

B.3 Completing the Terms of Reference

The Terms of Reference (ToR) for a Specialist Task Force (STF) to develop test specifications requires the following information to be included:

- the extent of validation of the test specifications;
- the division of responsibilities;
- the cost of validation.

B.3.1 The extent of validation

A declaration should be made of the validation level (see clause B.2) that is intended to be used within the project. This should include an assessment of the risk that the given level of validation may not be achieved. It should also specify how the declared validation level is to be achieved by identifying the following:

- full details on the validation process;
- a list of test equipment and/or platforms required;
- the availability of suitable SUTs to test against etc;
- a timeline for completing the task(s), noting all critical activities;
- details of the resources from outside the ETSI Funded Work Program that will be provided by committee members;
- assurances that the external resources will be made available when they are required; and
- the methods that will be used for providing feedback to the responsible technical committee.

B.3.2 Responsibilities

Details should be provided of where responsibilities lie for the provision of resources (both human and equipment) and how those resources are to be used. This should clearly indicate:

- which tasks are to be undertaken by the members of the STF itself;
- which tasks are to be undertaken by experts outside the STF;
- which tasks are to be performed jointly by STF experts and external experts; and
- where each task is likely to be carried out (particularly if not at ETSI's own offices).

Each validation level carries its own overlapping set of responsibilities that need to be considered and these are identified in Table B.1.

Table B.1: Responsibilities associated with each level of validation

Rigorous Validation (Level 3)	
Strong Validation (Level 2)	
Basic Validation (Level 1)	
Provision of TTCN-3 analyser/compiler	
Compilation of the test suite	
Reporting of errors in the tools	
Reporting of errors in the ATS	
Updating of ATS	
Compilation of validation report	
	Setting-up and maintaining the test platform
	Provision and installation of SUT(s)
	Execution of the tests
	Reporting of errors in the platform
	Validation of test verdicts
	Analysis of output traces

B.3.3 Costs associated with validation

An estimate of the cost of the validation tasks should be provided in terms of effort, travel and equipment costs (for example, hire charges, insurance and transport). This information should be included in the overall total of the STF.

Each level of validation implies a different level of expense to be considered. These are summarized in Table B.2.

Table B.2: Validation cost considerations

Validation Level	Costing information
Basic Validation	<ul style="list-style-type: none"> • Low cost; • Easily calculated depending on: <ul style="list-style-type: none"> - the size and complexity of the test suite; and - the number of compilers that will be used.
Strong Validation	<ul style="list-style-type: none"> • Reasonable cost; • More difficult to estimate; • Depends on: <ul style="list-style-type: none"> - the number of tests validated; - the maturity of the test suite; - the complexity of the test suite; - the number of compilers that will be used; and - the number of SUTs expected to be used.
Rigorous Validation	<ul style="list-style-type: none"> • Higher cost; • Difficult to estimate accurately; • Depends on: <ul style="list-style-type: none"> - the size and complexity of the test suite; - the maturity of the test suite; - the number of compilers that will be used; - the number of SUTs expected to be used; and - the equipment and other resources required for the validation exercise.

B.4 The validation report

Once the validation of a test suite has been completed it is necessary to produce a report detailing the results of the validation exercise. There is no specific requirement for the report to be made available in any particular form but in most cases it can be usefully placed as an annex to the ATS Technical Specification (TS) document.

The amount of information included in the validation report will depend on the validation level chosen. However, all validation reports should include the following:

- an indication of the validation level achieved;
- a declaration of the edition of the TTCN-3 standard that the test suite was based upon;
- for each TTCN-3 tool used:
 - identification of the TTCN-3 Tool:
 - supplier;
 - tool name;
 - version;
 - configuration settings; and
 - other tool-specific information;
 - compilation status:
 - overall success or failure;
 - details of compilation failures including justifications for accepting them if necessary;
- non standard modifications to the test suite:
 - Details of any use of non-standard TTCN-3 within the test suite with justifications for its inclusion;

- TTCN-3 quality:
 - if the TTCN-3 has been analyzed with the ETSI T3Q quality assessment tool then a copy of the T3Q configuration file and details of the results of the analysis should be included here;
- validation method:
 - a brief description of the validation process followed.

If the validation level was either Strong (Level 2) or Rigorous (Level 3) then the following additional items should be included in the validation report:

- for each test platform used, an indication of:
 - the manufacturer;
 - the product name;
 - any version or revision information;
 - other useful product-specific information;
- for each SUT used in the validation, an indication of:
 - the manufacturer;
 - the product name;
 - any version or revision information;
 - other useful product-specific information;
- Validation Status of the test suite:
 - for each tests case executed:
 - a test case identifier;
 - the verdict achieved at the end of validation;
 - the implication on the validation of the test suite of each verdict regardless of whether "Pass", "Fail" or "Inconclusive";
 - activities required, if any, to correct the test suite.

History

Document history		
V1.1.1	May 1997	Publication
V1.2.1	May 1999	Publication
V2.1.1	November 2011	Membership Approval Procedure MV 20120127: 2011-11-28 to 2012-01-27
V2.1.1	February 2012	Publication