



SLOVENSKI STANDARD
oSIST prEN 16602-80:2017
01-januar-2017

Zagotavljanje varnih proizvodov v vesoljski tehniki - Zagotavljanje varne programske opreme

Space product assurance - Software product assurance

Raumfahrtproduktsicherung - Software-Produktsicherung

Assurance produit des projets spatiaux - Assurance produit logiciel

Ta slovenski standard je istoveten z: prEN 16602-80

ICS:

35.080	Programska oprema	Software
49.140	Vesoljski sistemi in operacije	Space systems and operations

oSIST prEN 16602-80:2017

en,fr,de

EUROPEAN STANDARD
NORME EUROPÉENNE
EUROPÄISCHE NORM

DRAFT
prEN 16602-80

November 2016

ICS 35.080; 49.140

English version

Space product assurance - Software product assurance

Assurance produit des projets spatiaux - Assurance
produit logiciel

Raumfahrtproduktsicherung - Software-
Produktsicherung

This draft European Standard is submitted to CEN members for enquiry. It has been drawn up by the Technical Committee CEN/CLC/TC 5.

If this draft becomes a European Standard, CEN and CENELEC members are bound to comply with the CEN/CENELEC Internal Regulations which stipulate the conditions for giving this European Standard the status of a national standard without any alteration.

This draft European Standard was established by CEN and CENELEC in three official versions (English, French, German). A version in any other language made by translation under the responsibility of a CEN and CENELEC member into its own language and notified to the CEN-CENELEC Management Centre has the same status as the official versions.

CEN and CENELEC members are the national standards bodies and national electrotechnical committees of Austria, Belgium, Bulgaria, Croatia, Cyprus, Czech Republic, Denmark, Estonia, Finland, Former Yugoslav Republic of Macedonia, France, Germany, Greece, Hungary, Iceland, Ireland, Italy, Latvia, Lithuania, Luxembourg, Malta, Netherlands, Norway, Poland, Portugal, Romania, Slovakia, Slovenia, Spain, Sweden, Switzerland, Turkey and United Kingdom.

<https://standards.iteh.ai/catalog/standards/sist/6ec6d20d-6b56-48ff-99b6-47dd1efad2f1/sist->
Recipients of this draft are invited to submit, with their comments, notification of any relevant patent rights of which they are aware and to provide supporting documentation. Recipients of this draft are invited to submit, with their comments, notification of any relevant patent rights of which they are aware and to provide supporting documentation.

Warning : This document is not a European Standard. It is distributed for review and comments. It is subject to change without notice and shall not be referred to as a European Standard.



**CEN-CENELEC Management Centre:
Avenue Marnix 17, B-1000 Brussels**

Table of contents

Foreword	7
1 Scope	8
2 Normative references	9
3 Terms, definitions and abbreviated terms	10
3.1 Terms for other standards	10
3.2 Terms specific to the present standard	10
3.3 Abbreviated terms	15
4 Space system software product assurance principles	18
4.1 Introduction	18
4.2 Organization of this Standard	19
4.3 Tailoring of this Standard	21
5 Software product assurance programme implementation	22
5.1 Organization and responsibility	22
5.1.1 Organization	22
5.1.2 Responsibility and authority	22
5.1.3 Resources	23
5.1.4 Software product assurance manager/engineer	23
5.1.5 Training	23
5.2 Software product assurance programme management	24
5.2.1 Software product assurance planning and control	24
5.2.2 Software product assurance reporting	25
5.2.3 Audits	26
5.2.4 Alerts	26
5.2.5 Software problems	26
5.2.6 Nonconformances	27
5.2.7 Quality requirements and quality models	28
5.3 Risk management and critical item control	28

5.3.1	Risk management	28
5.3.2	Critical item control.....	28
5.4	Supplier selection and control.....	29
5.4.1	Supplier selection.....	29
5.4.2	Supplier requirements	29
5.4.3	Supplier monitoring	29
5.4.4	Criticality classification	30
5.5	Procurement.....	30
5.5.1	Procurement documents	30
5.5.2	Review of procured software component list	30
5.5.3	Procurement details	31
5.5.4	Identification.....	31
5.5.5	Inspection	31
5.5.6	Exportability	31
5.6	Tools and supporting environment.....	31
5.6.1	Methods and tools.....	31
5.6.2	Development environment selection	32
5.7	Assessment and improvement process	33
5.7.1	Process assessment.....	33
5.7.2	Assessment process	33
5.7.3	Process improvement	34
6	Software process assurance	36
6.1	Software development life cycle.....	36
6.1.1	Life cycle definition.....	36
6.1.2	Process quality objectives	36
6.1.3	Life cycle definition review.....	36
6.1.4	Life cycle resources	36
6.1.5	Software validation process schedule	37
6.2	Requirements applicable to all software engineering processes	37
6.2.1	Documentation of processes	37
6.2.2	Software dependability and safety.....	38
6.2.3	Handling of critical software	40
6.2.4	Software configuration management.....	42
6.2.5	Process metrics	44
6.2.6	Verification	45
6.2.7	Reuse of existing software	48
6.2.8	Automatic code generation.....	51

prEN 16602-80:2016 (E)

6.3	Requirements applicable to individual software engineering processes or activities.....	52
6.3.1	Software related system requirements process.....	52
6.3.2	Software requirements analysis	52
6.3.3	Software architectural design and design of software items	54
6.3.4	Coding	55
6.3.5	Testing and validation	56
6.3.6	Software delivery and acceptance.....	61
6.3.7	Operations	62
6.3.8	Maintenance	63
7	Software product quality assurance.....	65
7.1	Product quality objectives and metrication	65
7.1.1	Deriving of requirements	65
7.1.2	Quantitative definition of quality requirements.....	65
7.1.3	Assurance activities for product quality requirements.....	65
7.1.4	Product metrics	65
7.1.5	Basic metrics.....	66
7.1.6	Reporting of metrics.....	66
7.1.7	Numerical accuracy.....	66
7.1.8	Analysis of software maturity.....	67
7.2	Product quality requirements	67
7.2.1	Requirements baseline and technical specification	67
7.2.2	Design and related documentation.....	68
7.2.3	Test and validation documentation.....	68
7.3	Software intended for reuse.....	69
7.3.1	Customer requirements.....	69
7.3.2	Separate documentation	69
7.3.3	Self-contained information.....	69
7.3.4	Requirements for intended reuse	69
7.3.5	Configuration management for intended reuse.....	69
7.3.6	Testing on different platforms.....	70
7.3.7	Certificate of conformance	70
7.4	Standard ground hardware and services for operational system.....	70
7.4.1	Hardware procurement	70
7.4.2	Service procurement.....	70
7.4.3	Constraints.....	71
7.4.4	Selection.....	71

7.4.5	Maintenance	71
7.5	Firmware	71
7.5.1	Device programming	71
7.5.2	Marking	72
7.5.3	Calibration.....	72
Annex A (informative) Software documentation.....		73
Annex B (normative) Software product assurance plan (SPAP) - DRD		79
B.1	DRD identification.....	79
B.1.1	Requirement identification and source document.....	79
B.1.2	Purpose and objective.....	80
B.2	Expected response.....	81
B.2.1	Scope and content.....	81
B.2.2	Special remarks	85
Annex C (normative) Software product assurance milestone report (SPAMR) - DRD		86
C.1	DRD identification.....	86
C.1.1	Requirement identification and source document.....	86
C.1.2	Purpose and objective.....	87
C.2	Expected response.....	87
C.2.1	Scope and content.....	87
C.2.2	Special remarks	88
Annex D (normative) Tailoring of this Standard based on software criticality.....		89
D.1	Software criticality categories	89
D.2	Applicability matrix.....	90
Annex E (informative) List of requirements with built-in tailoring capability.....		101
Annex F (informative) Document organization and content at each milestone.....		102
F.1	Introduction.....	102
F.2	ECSS-Q-ST-80 Expected Output at SRR	102
F.3	ECSS-Q-ST-80 Expected Output at PDR	104
F.4	ECSS-Q-ST-80 Expected Output at CDR.....	109
F.5	ECSS-Q-ST-80 Expected Output at QR	111
F.6	ECSS-Q-ST-80 Expected Output at AR.....	112

prEN 16602-80:2016 (E)

F.7 ECSS-Q-ST-80 Expected Output not associated with any specific milestone review	114
--	-----

Bibliography..... 116**Figures**

Figure 4-1: Software related processes in ECSS Standards.....	19
Figure 4-2: Structure of this Standard.....	20
Figure A-1 : Overview of software documents	73

Tables

Table A-1 : ECSS-E-ST-40 and ECSS-Q-ST-80 Document requirements list (DRL)	74
Table B-1 : SPAP traceability to ECSS-E-ST-40 and ECSS-Q-ST-80 clauses	79
Table C-1 : SPAMR traceability to ECSS-E-ST-40 and ECSS-Q-ST-80 clauses	86
Table D-1 : Software criticality categories.....	89
Table D-2 : Applicability matrix based on software criticality	90

iTeh STANDARD PREVIEW (standards.iteh.ai)

SIST EN 16602-80:2019

<https://standards.iteh.ai/catalog/standards/sist/6ec6d20d-6b56-48ff-99b6-47dd1efad2f1/sist-en-16602-80-2019>

Foreword

This document (prEN 16602-80:2016) has been prepared by Technical Committee CEN/CLC/TC 5 “Space”, the secretariat of which is held by DIN (Germany).

This document (prEN 16602-80:2016) originates from ECSS-Q-ST-80C Rev.1 DIR1.

This document is currently submitted to the CEN ENQUIRY.

This document has been developed to cover specifically space systems and will therefore have precedence over any EN covering the same scope but with a wider domain of applicability (e.g.: aerospace).

iTeh STANDARD PREVIEW
(standards.iteh.ai)

[SIST EN 16602-80:2019](https://standards.iteh.ai/catalog/standards/sist/6ec6d20d-6b56-48ff-99b6-47dd1efad2f1/sist-en-16602-80-2019)

<https://standards.iteh.ai/catalog/standards/sist/6ec6d20d-6b56-48ff-99b6-47dd1efad2f1/sist-en-16602-80-2019>

1 Scope

This Standard defines a set of software product assurance requirements to be used for the development and maintenance of software for space systems. Space systems include manned and unmanned spacecraft, launchers, payloads, experiments and their associated ground equipment and facilities. Software includes the software component of firmware.

This Standard also applies to the development or reuse of non-deliverable software which affects the quality of the deliverable product or service provided by a space system, if the service is implemented by software.

ECSS-Q-ST-80 interfaces with space engineering and management, which are addressed in the Engineering (-E) and Management (-M) branches of the ECSS System, and explains how they relate to the software product assurance processes.

This standard may be tailored for the specific characteristic and constraints of a space project in conformance with ECSS-S-ST-00.

Tailoring of this Standard to a specific business agreement or project, when software product assurance requirements are prepared, is also addressed in clause 4.3.

2

Normative references

The following normative documents contain provisions which, through reference in this text, constitute provisions of this ECSS Standard. For dated references, subsequent amendments to, or revision of any of these publications do not apply, However, parties to agreements based on this ECSS Standard are encouraged to investigate the possibility of applying the more recent editions of the normative documents indicated below. For undated references, the latest edition of the publication referred to applies.

EN reference	Reference in text	Title
EN 16601-00-01	ECSS-S-ST-00-01	ECSS system – Glossary of terms
EN 16003-40	ECSS-E-ST-40	Space engineering – Software general requirements
EN 16602-10	ECSS-Q-ST-10	Space product assurance – Product assurance management
EN 16602-10-04	ECSS-Q-ST-10-04	Space product assurance – Critical-item control
EN 16602-10-09	ECSS-Q-ST-10-09	Space product assurance – Nonconformance control system
EN 16602-20	ECSS-Q-ST-20	Space product assurance – Quality assurance
EN 16602-30	ECSS-Q-ST-30	Space product assurance – Dependability
EN 16602-40	ECSS-Q-ST-40	Space product assurance – Safety
EN 16601-10	ECSS-M-ST-10	Space project management – Project planning and implementation
EN 16601-10-01	ECSS-M-ST-10-01	Space project management – Organization and conduct of reviews
EN 16601-40	ECSS-M-ST-40	Space project management – Configuration and information management
EN 16601-80	ECSS-M-ST-80	Space project management – Risk management
	ISO/IEC 15504 Part 2:2003	Software engineering - Process assessment – Part 2: Performing an assessment - First Edition

Terms, definitions and abbreviated terms

3.1 Terms for other standards

For the purpose of this Standard, the terms and definitions from ECSS-ST-00-01 apply in particular for the term:

acceptance test

software product

NOTE The terms and definitions are common for the ECSS-E-ST-40 and ECSS-Q-ST-80 Standards.

3.2 Terms specific to the present standard

3.2.1 automatic code generation

generation of source code with a tool from a model

3.2.2 code coverage

percentage of the software that has been executed (covered) by the test suite

3.2.3 competent assessor

person who has demonstrated the necessary skills, competencies and experience to lead a process assessment in conformance with ISO/IEC 15504

NOTE Adapted from ISO/IEC 15504:1998, Part 9.

3.2.4 condition

boolean expression not containing boolean operators

3.2.5 configurable code

code (source code or executable code) that can be tailored by setting values of parameters

NOTE This definition covers in particular classes of configurable code obtained by the following configuration means:

- configuration based on the use of a compilation directive;

- configuration based on the use of a link directive;
- configuration performed through a parameter defined in a configuration file;
- configuration performed through data defined in a database with impact on the actually executable parts of the software (e.g. parameters defining branch structures that result in the non-execution of existing parts of the code).

3.2.6 COTS, OTS, MOTS software

for the purpose of this Standard, commercial-off-the-shelf, off-the-shelf and modified-off-the-shelf software for which evidence of use is available

3.2.7 critical software

software of criticality category A, B or C

NOTE See ECSS-Q-ST-80C, Annex D.1 – Software criticality categories.

3.2.8 deactivated code

code that, although incorporated through correct design and coding, is intended to execute in certain software product configurations only, or in none of them

[adapted from RTCA/DO-178B]

3.2.9 decision

boolean expression composed of conditions and zero or more boolean operators that are used in a control construct.

NOTE 1 For example: “if.....thenelse” or the “case” statement are control construct.

NOTE 2 A decision without a boolean operator is a condition.

NOTE 3 If a condition appears more than once in a decision, each occurrence is a distinct condition.

3.2.10 decision coverage

measure of the part of the program within which every point of entry and exit is invoked at least once and every decision has taken “true” and “false” values at least once.

NOTE Decision coverage includes, by definition, statement coverage.

3.2.11 existing software

any software developed outside the business agreement to which this Standard is applicable, including software from previous developments provided by the

prEN 16602-80:2016 (E)

supplier, software from previous developments provided by the customer, COTS, OTS and MOTS software, freeware and open source software

3.2.12 integration testing

testing in which software components, hardware components, or both are combined and tested to evaluate the interaction between them

[IEEE 610.12:1990]

3.2.13 logical model

implementation-independent model of software items used to analyse and document software requirements

3.2.14 margin philosophy

rationale for margins allocated to the performance parameters and computer resources of a development, and the way to manage these margins during the execution of the project

3.2.15 metric

defined measurement method and the measurement scale

NOTE 1 Metrics can be internal or external, and direct or indirect.

NOTE 2 Metrics include methods for categorising qualitative data.

[ISO/IEC 9126-1:2001]

3.2.16 migration

porting of a software product to a new environment

3.2.17 mission products

products and services delivered by the space system

NOTE For example: Communications services, science data.

3.2.18 modified condition and decision coverage

measure of the part of the program within which every point of entry and exit has been invoked at least once, every decision in the program has taken “true” and “false” values at least once, and each condition in a decision has been shown to independently affect that decision’s outcome

NOTE A condition is shown to independently affect a decision’s outcome by varying that condition while holding fixed all other possible conditions.

3.2.19 operational

for the purpose of this Standard, related to the software operation

NOTE It is not related to the spacecraft operation.

3.2.20 portability (a quality characteristic)

capability of software to be transferred from one environment to another

3.2.21 quality characteristics (software)

set of attributes of a software product by which its quality is described and evaluated

NOTE A software quality characteristic can have multiple levels of sub-characteristics.

3.2.22 quality model (software)

set of characteristics and the relationships between them which provide the basis for specifying quality requirements and evaluating quality

[ISO/IEC 9126-1:2001]

3.2.23 real-time

pertaining to a system or mode of operation in which computation is performed during the actual time that an external process occurs, in order that the computation results can be used to control, monitor, or respond in a timely manner to the external process

[IEEE 610.12:1990]

3.2.24 regression testing (software)

selective retesting of a system or component to verify that modifications have not caused unintended effects and that the system or component still complies with its specified requirements

[IEEE 610.12:1990]

3.2.25 reusability

degree to which a software unit or other work product can be used in more than one computer program or software system

[IEEE 610.12:1990]

3.2.26 singular input

input corresponding to a singularity of the function

3.2.27 software

see "software product" in ECSS-S-ST-00-01

3.2.28 software component

part of a software system

NOTE 1 Software component is used as a general term.

NOTE 2 Components can be assembled and decomposed to form new components. In the production activities, components are implemented as units, tasks or programs, any of which can be configuration items. This usage of the term is more general than