

ECSS Secretariat

ESA-ESTEC

Requirements & Standards Section

Noordwijk, The Netherlands

**Foreword**

ECSS is a cooperative effort of the European Space Agency, national space agencies and European industry associations for the purpose of developing and maintaining common standards. Requirements in this Standard are defined in terms of what shall be accomplished, rather than in terms of how to organize and perform the necessary work. This allows existing organizational structures and methods to be applied where they are effective, and for the structures and methods to evolve as necessary without rewriting the standards.

This Standard has been prepared by the Working Group, reviewed by the ECSS Executive Secretariat and approved by the ECSS Technical Authority.

**Disclaimer**

ECSS does not provide any warranty whatsoever, whether expressed, implied, or statutory, including, but not limited to, any warranty of merchantability or fitness for a particular purpose or any warranty that the contents of the item are error-free. In no respect shall ECSS incur any liability for any damages, including, but not limited to, direct, indirect, special, or consequential damages arising out of, resulting from, or in any way connected to the use of this Standard, whether or not based upon warranty, business agreement, tort, or otherwise; whether or not injury was sustained by persons or property or otherwise; and whether or not loss was sustained from, or arose out of, the results of, the item, or any services that may be provided by ECSS.

Published by: ESA Requirements and Standards Section

ESTEC, P.O. Box 299,

2200 AG Noordwijk

The Netherlands

Copyright: 2025© by the European Space Agency for the members of ECSS

Change log

|  |  |
| --- | --- |
| ECSS-Q-80A  19 April 1996 | First issue |
| ECSS-Q-80B  10 October 2003 | Second issue |
| ECSS-Q-ST-80C  6 March 2009 | Third issue |
| ECSS-Q-ST-80C Rev.1  15 February 2017 | Third issue, Revision 1 |
| ECSS-Q-ST-80C Rev.2  30 April 2025 | Third issue, Revision 2  Changes with respect to ECSS-Q-ST-80C Rev.1 (15 February 2027) are identified with revision tracking.  Main changes   * Implementation of change requests * Addition of clause 6.2.9 “Software security” * Addition of clause 6.2.10 “Handling of security software” * Split of former clause 6.3.6 “Software delivery and acceptance” to new clauses 6.3.6 “Software delivery and installation” and 6.3.7 “Software acceptance”, including renumbering of existing and addition of new subclauses.   **Detailed Change Record:**  Deleted requirements:  6.3.7.4a (formerly 6.3.6.6a).  Added requirements:  5.1.5.4b; 5.2.6.1d; 5.4.5a; 6.1.4b; 6.2.4.8b and c; 6.2.4.12a; 6.2.6.13c; 6.2.7.3b; 6.2.9.1a; clause 6.2.9 “Software security”; clause 6.2.10 “Handling of security”; 6.3.1.3b; 6.3.2.3b; 6.3.5.14b; 6.3.5.28b; 6.3.5.33a; 6.3.6.1a and b; 6.3.6.3a; 6.3.7.1b; 7.5.2b; B.2.1<6.4>a and b.  Modified requirements:  5.1.5.4a; 5.2.7.2a Note; 5.4.1.2a (editorial); 5.5.3a; 5.5.6a (Note added); 5.6.1.2a; 5.7.1a; 5.7.2.2.a; 5.7.2.3a; 5.7.2.4a; 5.7.3.3a Note; 6.1.3a; 6.1.5a; 6.2.1.1a; 6.2.3.2 Note; 6.2.4.7a; 6.2.4.9a; 6.2.4.10a; 6.2.4.11a; 6.2.6.1a (Exp. Output); 6.2.6.13 Note added; 6.2.6.13b; 6.2.7.3 (Exp. Output removed), 6.2.7.4a; 6.2.7.8b; 6.3.4.1a; 6.3.4.4a; 6.3.4.6a; 6.3.5.1 Note; 6.3.5.2a; 6.3.5.28a (Exp. Output removed); 6.3.9.2a; 6.3.9.4a; 6.3.9.7a; 7.1.1a; 7.2.2.3a Note; 7.4.4a; 7.5.1a; 7.5.2a.  Editorial changes:   * Notes placed always after the text of the normative provision or Expected Output. * Addition of new 6.3.6.1 and 6.3.6.3 causes renumbering of previous clause numbers. * Addition of heading 6.3.7 “Software acceptance” causes renumbering of subsequent clause numbers. * Clause 7.5 renamed from “Firmware” to “Programmable devices”. * Annexes updated. |

Table of contents

[Change log 3](#_Toc198053329)

[1 Scope 10](#_Toc198053330)

[2 Normative references 11](#_Toc198053331)

[3 Terms, definitions and abbreviated terms 12](#_Toc198053332)

[3.1 Terms from other standards 12](#_Toc198053333)

[3.2 Terms specific to the present standard 12](#_Toc198053334)

[3.3 Abbreviated terms 17](#_Toc198053335)

[3.4 Nomenclature 19](#_Toc198053336)

[4 Space system software product assurance principles 20](#_Toc198053337)

[4.1 Introduction 20](#_Toc198053338)

[4.2 Organization of this Standard 21](#_Toc198053339)

[4.3 Tailoring of this Standard 23](#_Toc198053340)

[4.4 Security aspects of this Standard 23](#_Toc198053341)

[5 Software product assurance programme implementation 24](#_Toc198053342)

[5.1 Organization and responsibility 24](#_Toc198053343)

[5.1.1 Organization 24](#_Toc198053344)

[5.1.2 Responsibility and authority 24](#_Toc198053345)

[5.1.3 Resources 25](#_Toc198053346)

[5.1.4 Software product assurance manager/engineer 25](#_Toc198053347)

[5.1.5 Training 25](#_Toc198053348)

[5.2 Software product assurance programme management 26](#_Toc198053349)

[5.2.1 Software product assurance planning and control 26](#_Toc198053350)

[5.2.2 Software product assurance reporting 28](#_Toc198053351)

[5.2.3 Audits 28](#_Toc198053352)

[5.2.4 Alerts 28](#_Toc198053353)

[5.2.5 Software problems 29](#_Toc198053354)

[5.2.6 Nonconformances 29](#_Toc198053355)

[5.2.7 Quality requirements and quality models 30](#_Toc198053356)

[5.3 Risk management and critical item control 31](#_Toc198053358)

[5.3.1 Risk management 31](#_Toc198053359)

[5.3.2 Critical item control 31](#_Toc198053360)

[5.4 Supplier selection and control 32](#_Toc198053361)

[5.4.1 Supplier selection 32](#_Toc198053362)

[5.4.2 Supplier requirements 32](#_Toc198053363)

[5.4.3 Supplier monitoring 32](#_Toc198053364)

[5.4.4 Criticality classification 33](#_Toc198053365)

[5.4.5 Security sensitivity 33](#_Toc198053366)

[5.5 Procurement 34](#_Toc198053367)

[5.5.1 Procurement documents 34](#_Toc198053368)

[5.5.2 Review of procured software component list 34](#_Toc198053369)

[5.5.3 Procurement details 34](#_Toc198053370)

[5.5.4 Identification 34](#_Toc198053372)

[5.5.5 Inspection 34](#_Toc198053373)

[5.5.6 Exportability 34](#_Toc198053374)

[5.6 Tools and supporting environment 35](#_Toc198053375)

[5.6.1 Methods and tools 35](#_Toc198053376)

[5.6.2 Development environment selection 35](#_Toc198053377)

[5.7 Assessment and improvement process 36](#_Toc198053378)

[5.7.1 Process assessment 36](#_Toc198053379)

[5.7.2 Assessment process 37](#_Toc198053381)

[5.7.3 Process improvement 38](#_Toc198053385)

[6 Software process assurance 40](#_Toc198053387)

[6.1 Software development life cycle 40](#_Toc198053388)

[6.1.1 Life cycle definition 40](#_Toc198053389)

[6.1.2 Process quality objectives 40](#_Toc198053390)

[6.1.3 Life cycle definition review 40](#_Toc198053391)

[6.1.4 Life cycle resources 40](#_Toc198053392)

[6.1.5 Software validation process schedule 41](#_Toc198053393)

[6.2 Requirements applicable to all software engineering processes 41](#_Toc198053394)

[6.2.1 Documentation of processes 41](#_Toc198053395)

[6.2.2 Software dependability and safety 42](#_Toc198053396)

[6.2.3 Handling of critical software 45](#_Toc198053400)

[6.2.4 Software configuration management 47](#_Toc198053401)

[6.2.5 Process metrics 50](#_Toc198053402)

[6.2.6 Verification 51](#_Toc198053403)

[6.2.7 Reuse of existing software 54](#_Toc198053404)

[6.2.8 Automatic code generation 57](#_Toc198053405)

[6.2.9 Software security 58](#_Toc198053406)

[6.2.10 Handling of security sensitive software 60](#_Toc198053407)

[6.3 Requirements applicable to individual software engineering processes or activities 61](#_Toc198053408)

[6.3.1 Software related system requirements process 61](#_Toc198053409)

[6.3.2 Software requirements analysis 62](#_Toc198053410)

[6.3.3 Software architectural design and design of software items 63](#_Toc198053411)

[6.3.4 Coding 64](#_Toc198053412)

[6.3.5 Testing and validation 66](#_Toc198053413)

[6.3.6 Software delivery and installation 72](#_Toc198053414)

[6.3.7 Software acceptance 72](#_Toc198053415)

[6.3.8 Operations 74](#_Toc198053416)

[6.3.9 Maintenance 74](#_Toc198053417)

[7 Software product quality assurance 77](#_Toc198053419)

[7.1 Product quality objectives and metrication 77](#_Toc198053420)

[7.1.1 Deriving of requirements 77](#_Toc198053421)

[7.1.2 Quantitative definition of quality requirements 77](#_Toc198053422)

[7.1.3 Assurance activities for product quality requirements 77](#_Toc198053423)

[7.1.4 Product metrics 77](#_Toc198053424)

[7.1.5 Basic metrics 78](#_Toc198053426)

[7.1.6 Reporting of metrics 78](#_Toc198053427)

[7.1.7 Numerical accuracy 78](#_Toc198053428)

[7.1.8 Analysis of software maturity 79](#_Toc198053429)

[7.2 Product quality requirements 79](#_Toc198053431)

[7.2.1 Requirements baseline and technical specification 79](#_Toc198053432)

[7.2.2 Design and related documentation 80](#_Toc198053433)

[7.2.3 Test and validation documentation 80](#_Toc198053437)

[7.3 Software intended for reuse 81](#_Toc198053438)

[7.3.1 Customer requirements 81](#_Toc198053439)

[7.3.2 Separate documentation 81](#_Toc198053440)

[7.3.3 Self-contained information 81](#_Toc198053441)

[7.3.4 Requirements for intended reuse 81](#_Toc198053442)

[7.3.5 Configuration management for intended reuse 82](#_Toc198053443)

[7.3.6 Testing on different platforms 82](#_Toc198053444)

[7.3.7 Certificate of conformance 82](#_Toc198053445)

[7.4 Standard ground hardware and services for operational system 82](#_Toc198053447)

[7.4.1 Hardware procurement 82](#_Toc198053448)

[7.4.2 Service procurement 83](#_Toc198053449)

[7.4.3 Constraints 83](#_Toc198053450)

[7.4.4 Selection 83](#_Toc198053451)

[7.4.5 Maintenance 84](#_Toc198053452)

[7.5 Programmable devices 84](#_Toc198053453)

[7.5.1 Device programming 84](#_Toc198053454)

[7.5.2 Marking 84](#_Toc198053455)

[7.5.3 Calibration 84](#_Toc198053456)

[Annex A (informative) Software documentation 85](#_Toc198053457)

[Annex B (normative) Software product assurance plan (SPAP) - DRD 91](#_Toc198053458)

[B.1 DRD identification 91](#_Toc198053459)

[B.1.1 Requirement identification and source document 91](#_Toc198053460)

[B.1.2 Purpose and objective 92](#_Toc198053461)

[B.2 Expected response 93](#_Toc198053462)

[B.2.1 Scope and content 93](#_Toc198053463)

[B.2.2 Special remarks 97](#_Toc198053464)

[Annex C (normative) Software product assurance milestone report (SPAMR) - DRD 98](#_Toc198053465)

[C.1 DRD identification 98](#_Toc198053466)

[C.1.1 Requirement identification and source document 98](#_Toc198053467)

[C.1.2 Purpose and objective 98](#_Toc198053468)

[C.2 Expected response 99](#_Toc198053469)

[C.2.1 Scope and content 99](#_Toc198053470)

[C.2.2 Special remarks 100](#_Toc198053471)

[Annex D (normative) Tailoring of this Standard based on software criticality 101](#_Toc198053472)

[D.1 Software criticality categories 101](#_Toc198053473)

[D.2 Applicability matrix 102](#_Toc198053474)

[Annex E (informative) List of requirements with built-in tailoring capability 114](#_Toc198053475)

[Annex F (informative) Document organization and content at each milestone 115](#_Toc198053476)

[F.1 Introduction 115](#_Toc198053477)

[F.2 ECSS-Q-ST-80 Expected Output at SRR 115](#_Toc198053478)

[F.3 ECSS-Q-ST-80 Expected Output at PDR 118](#_Toc198053479)

[F.4 ECSS-Q-ST-80 Expected Output at CDR 123](#_Toc198053480)

[F.5 ECSS-Q-ST-80 Expected Output at QR 126](#_Toc198053481)

[F.6 ECSS-Q-ST-80 Expected Output at AR 128](#_Toc198053482)

[F.7 ECSS-Q-ST-80 Expected Output not associated with any specific milestone review 129](#_Toc198053483)

[Bibliography 132](#_Toc198053484)

**Figures**

[Figure 4‑1: Software related processes in ECSS Standards 21](#_Toc198053485)

[Figure 4‑2: Structure of this Standard 22](#_Toc198053486)

[Figure A-1 : Overview of software documents 85](#_Toc198053487)

**Tables**

[Table A-1 : ECSS-E-ST-40 and ECSS-Q-ST-80 Document requirements list (DRL) 86](#_Toc198053488)

[Table B-1 : SPAP traceability to ECSS-E-ST-40 and ECSS-Q-ST-80 clauses 91](#_Toc198053489)

[Table C-1 : SPAMR traceability to ECSS-Q-ST-80 clauses 98](#_Toc198053490)

[Table D-1 : Software criticality categories 101](#_Toc198053491)

[Table D-2 : Applicability matrix based on software criticality 102](#_Toc198053492)

# 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.

This Standard also applies to the development or reuse of non­deliverable 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.

# 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.

|  |  |
| --- | --- |
| ECSS-S-ST-00-01 | ECSS system — Glossary of terms |
| ECSS-E-ST-40 | Space engineering — Software general requirements |
| ECSS-E-ST-80 | Space engineering – Security in space systems lifecycles |
| ECSS-Q-ST-10 | Space product assurance – Product assurance management |
| ECSS-Q-ST-10-04 | Space product assurance – Critical-item control |
| ECSS-Q-ST-10-09 | Space product assurance – Nonconformance control system |
| ECSS-Q-ST-20 | Space product assurance – Quality assurance |
| ECSS-Q-ST-30 | Space product assurance – Dependability |
| ECSS-Q-ST-40 | Space product assurance – Safety |
| ECSS-M-ST-10 | Space project management – Project planning and implementation |
| ECSS-M-ST-10-01 | Space project management –Organization and conduct of reviews |
| ECSS-M-ST-40 | Space project management – Configuration and information management |
| ECSS-M-ST-80 | Space project management – Risk management |
| CSW-ESAISVV-2022-GBK-02897 | ESA ISVV Handbook - Independent software verification and validation handbook |
| ISO/IEC 33002:2015 | Information technology – Process assessment – Requirements for performing process assessment Process assessment |

# Terms, definitions and abbreviated terms

## Terms from other standards

1. For the purpose of this Standard, the terms and definitions from ECSS-S-ST-00-01 apply.
2. For the purpose of this Standard, the following terms and definitions from ECSS-E-ST-80 apply:
   1. security sensitivity

## Terms specific to the present standard

1. acceptance test

test of a system or functional unit usually performed by the customer on his premises after installation, with the participation of the supplier to ensure that the contractual requirements are met

1. Adapted from ISO/IEC 2382-20:1990
2. automatic code generation

generation of source code with a tool from a model

1. code coverage

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

1. competent assessor

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

1. Adapted from ISO/IEC 15504:1998, Part 9.
2. condition

boolean expression not containing boolean operators

1. configurable code

code that is only intended to be executed in certain specific configurations of the software product

1. This can be achieved either by use of compilation/link directives, parameter configuration (e.g. in a configuration file or database) or by target computer environment (e.g. hardware pin selection).
2. COTS, MOTS software

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

1. critical software

software of criticality category A, B or C

1. See ECSS-Q-ST-80 Annex D.1 – Software criticality categories.
2. deactivated code

code that is not intended to be executed in the target software product, or in the operational configuration of the target software product

1. Code related to defensive programming is not considered as deactivated code.
2. existing software

any software developed outside the business agreement to which this Standard is applicable, including software from previous developments provided by the supplier, software from previous developments provided by the customer, COTS and MOTS software, freeware and open source software

1. 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]

1. metric

a quantitative measure of the degree to which a system, component, or process possesses a given attribute

[ISO/IEC/IEEE 24765:2017]

1. migration

porting of a software product to a new environment

1. mission products

products and services delivered by the space system

1. For example: Communications services, science data.
2. operational

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

1. It is not related to the spacecraft operation.
2. portability (a quality characteristic)

capability of software to be transferred from one environment to another

1. processing unit

function which is defined to execute software.

1. The term covers the hardware functions such as processing core, Graphical Processing Unit (GPU), Vision Processing Unit (VPU), Tensor Processing Unit (TPU), Neural Processing Unit (NPU), Physics Processing Unit (PPU), Digital Signal Processor (DSP), Image Signal Processor (ISP). It also covers the software processing units such as interpreters, emulators and virtual machines.
2. quality characteristics (software)

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

1. A software quality characteristic can have multiple levels of sub-characteristics.
2. quality model (software)

defined set of characteristics, and of relationships between them, which provides a framework for specifying quality requirements and evaluating quality

[ISO/IEC 25000:2014]

1. 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]

1. 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]

1. 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]

1. software

set of instructions and data executed on a processing unit

* 1. 1: A processing unit can be hardware, e.g. a processor or software, e.g. a virtual machine or an interpreter.
  2. 2: Some processing units only require data, e.g. configuration of state machines or configuration data of a neural network.

1. software component

part of a software system

* 1. 1 Software component is used as a general term.
  2. 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 in ANSI/IEEE parlance, which defines a component as a “basic part of a system or program”; in this Standard, components are not always “basic” as they can be decomposed.

1. software item

see “software product”

1. software product

set of software, procedures, scripts, documentation and their associated data

1. The term “software item” is synonymous.
2. software problem

condition of a software product that causes difficulty or uncertainty in the use of the software

[CMU/SEI-92-TR-022]

1. software product assurance

totality of activities, standards, controls and procedures in the lifetime of a software product which establishes confidence that the delivered software product, or software affecting the quality of the delivered product, conforms to customer requirements

1. software unit

atomic level software component that can be subjected to stand-alone testing

1. test case

set of test inputs, execution conditions and expected results developed for a particular objective such as to exercise a particular program path or to verify compliance with a specified requirement

1. test procedure

detailed instructions for the set up, operation and evaluation of the results for a given test

1. threat

potential cause of an unwanted incident, which can result in harm to a system or organization

[ISO/IEC 27000:2018]

1. unit test

test of individual software unit

1. unreachable code

code that cannot be executed due to design or coding error

1. usability (a quality characteristic)

capability of the software to be understood, learned, used and liked by the user, when used under specified conditions

1. validation

<CONTEXT: software> process to confirm that the requirements are correctly and completely implemented in the final product

1. verification

<CONTEXT: software> process to confirm that adequate specifications and inputs exist for any activity, and that the outputs of the activities are correct and consistent with the specifications and input

1. The definition of verification at software level differs from the definition of verification at system level.
2. vulnerability

weakness which can be exploited by a threat source

1. walk-through

static analysis technique in which a designer or programmer leads members of the development team and other interested parties through a software product, and the participants ask questions and make comments about possible errors, violation of development standards, and other problems

[IEEE 1028-1997]

## Abbreviated terms

For the purpose of this Standard and of ECSS-E-ST-40, the abbreviated terms from ECSS-S-ST-00-01 and the following apply:

For the definition of DRD acronyms see Annex A.

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

| Abbreviation | Meaning |
| --- | --- |
| AR | acceptance review NOTE The term SW­AR can be used for clarity to denote ARs that solely involve software products. |
| CDR | critical design review NOTE The term SW­CDR can be used for clarity to denote CDRs that solely involve software products. |
| CMMI | capability maturity model integration |
| COTS | commercial-off-the-shelf |
| CPU | central processing unit NOTE The term CPU is commonly used to identify one or a group of processing units (PU). |
| DDF | design definition file |
| DDR | detailed design review |
| DJF | design justification file |
| DRD | document requirements definition |
| ECSS | European Cooperation for Space Standardization |
| eo | expected output |
| GS | ground segment |
| HMI | human machine interface |
| HSIA | hardware-software interaction analysis |
| HW | hardware |
| ICD | interface control document |
| IRD | interface requirements document |
| ISO | International Organization for Standardization |
| ISV | independent software validation |
| ISVV | independent software verification and validation |
| MGT | management file |
| MF | maintenance file |
| MOTS | modified off-the-shelf |
| OBCP | on-board control procedure |
| OP | operational plan |
| ORR | operational readiness review |
| OTS | off-the-shelf |
| PAF | product assurance file |
| PDR | preliminary design review NOTE The term SW­PDR can be used for clarity to denote PDRs that solely involve software products. |
| PRR | preliminary requirement review |
| PU | processing unit |
| QR | qualification review NOTE The term SW­QR can be used for clarity to denote QRs that solely involve software products. |
| RB | requirements baseline |
| SCAMPI | standard CMMI appraisal method for process improvement |
| SCMP | software configuration management plan |
| SDD | software design document |
| SDE | software development environment |
| SECOPS | Security operations |
| SF | security file |
| SOS | software operation support |
| SPA | software product assurance |
| SPAMR | software product assurance milestone report |
| SPAP | software product assurance plan |
| SPR | software problem report |
| SRB | software review board |
| SRR | system requirements review NOTE The term SW­SRR can be used for clarity to denote SRRs that solely involve software products. |
| SSMP | software security management plan |
| SVSR | software validation specification review |
| SW | software |
| SWE | software engineering |
| TRR | test readiness review |
| TS | technical specification |

## Nomenclature

The following nomenclature applies throughout this document:

1. The word “shall” is used in this Standard to express requirements. All the requirements are expressed with the word “shall”.
2. The word “should” is used in this Standard to express recommendations. All the recommendations are expressed with the word “should”.
3. It is expected that, during tailoring, recommendations in this document are either converted into requirements or tailored out.
4. The words “may” and “need not” are used in this Standard to express positive and negative permissions, respectively. All the positive permissions are expressed with the word “may”. All the negative permissions are expressed with the words “need not”.
5. The word “can” is used in this Standard to express capabilities or possibilities, and therefore, if not accompanied by one of the previous words, it implies descriptive text.
6. In ECSS “may” and “can” have completely different meanings: “may” is normative (permission), and “can” is descriptive.
7. The present and past tenses are used in this Standard to express statements of fact, and therefore they imply descriptive text.

# Space system software product assurance principles

## Introduction

The objectives of software product assurance are to provide adequate confidence to the customer and to the supplier that the developed or procured/reused software satisfies its requirements throughout the system’s lifetime. In particular, that the software is developed to perform properly, securely, and safely in its operational environment, meeting the project’s agreed quality objectives.

This Standard contributes to these objectives by defining the software product assurance requirements to be met in a particular space project. These requirements deal with quality management and framework, life cycle activities, process definition and quality characteristics of products.

One of the fundamental principles of this Standard is the customer­supplier relationship, assumed for all software developments. The organizational aspects of this are defined in ECSS-M-ST-10. In the general case, the customer is the procurer of two strongly associated products: the hardware and the software components of a system, subsystem, set, equipment, or assembly. The concept of the customer­supplier relationship is applied recursively, i.e. the customer can be a supplier to a higher level in the space system hierarchy.

The requirements of this Standard are applicable to the supplier, unless otherwise explicitly stated.

The supplier demonstrates compliance with the software product assurance requirements and provides the specified evidence of compliance.

To this end, the supplier specifies the software product assurance requirements for their suppliers, taking into account their responsibilities and the specific nature of their deliverables.

This Standard complements ECSS-E-ST-40 “Space engineering — Software general requirements”, with product assurance aspects, integrated in the space system software engineering processes as defined in ECSS-E-ST-40. Together the two standards specify all processes for space software development.

Figure 4‑1 schematically presents the different Software processes addressed by the set of the ECSS standards.



Figure 4‑1: Software related processes in ECSS Standards

## Organization of this Standard

This Standard is organized into three main parts:

* Software product assurance programme implementation
* Software process assurance
* Software product quality assurance.

The software documentation collecting the expected output of the ECSS-E-ST-40 and ECSS-Q-ST-80 requirements is summarized in Annex A.

Annex B and Annex C specify the DRDs (document requirements definitions) of the software product assurance documents (SPAP and SPAMR). The DRDs of other software engineering and management documents are included in ECSS-E-ST-40 and ECSS-M-ST-40.

The organization of this Standard is reflected in detail in Figure 4‑2.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Software product assurance programme implementation**   |  |  | | --- | --- | | 5.1 Organization and responsibility | 5.5 Procurement | | 5.2 Software product assurance programme management | 5.6 Tools and supporting environment | | 5.3 Risk management and critical item control | 5.7 Assessment and improvement process | | 5.4 Supplier selection and control | |
|  |
| **Software process assurance**   |  | | --- | | 6.1 Software development life cycle | | 6.2 Requirements applicable to all software engineering processes | | 6.3 Requirements applicable to individual software engineering processes or activities | |
|  |
| **Software product quality assurance**   |  | | --- | | 7.1 Product quality objectives and metrication | | 7.2 Product quality requirements | | 7.3 Software intended for reuse | | 7.4 Standard ground hardware and services for operational system | | 7.5 Programmable devices | |

Figure 4‑2: Structure of this Standard

Each requirement of this Standard is identified by a hierarchical number, plus a letter if necessary (e.g. 5.3.1.5, bullet a). For each requirement, the associated output is given in the “Expected Output” section. When several outputs are expected, they are identified by a letter (e.g. “a”, “b”, etc.). With each output, the destination file of the output is indicated in brackets, together with the corresponding document DRD (after a comma) and review(s) (after a semicolon). For example: “*[PAF, SPAP; SRR]*” denotes an output contained in the Software Product Assurance Plan, part of the Product Assurance File, and required at SRR. When no DRD is defined for an Expected Output, and/or the Expected Output is not to be provided at any specific milestone review, then the corresponding sections of that Expected Output are replaced by dashes (e.g. “*[PAF, -; -]*”).

This standard details, for the Software Product Assurance aspects, some of the general requirements already addressed by the ECSS Management, Product Assurance and Quality Assurance standards.

## Tailoring of this Standard

The general information and requirements for selecting and tailoring applicable standards are defined in ECSS-S-ST-00.

Several drivers for tailoring, include security, dependability and safety aspects, software development constraints, product quality objectives and business objectives.

Tailoring for dependability and safety aspects is based on the selection of requirements related to the verification, validation and levels of proofs demanded by the criticality of the software. Annex D contains a tailoring of this Standard based on software criticality.

Tailoring for security is based on the selection of requirements related to the verification, validation and levels of proofs demanded by the sensitivity of the software.

Tailoring for software development constraints considers the special characteristics of the software being developed and the development environment. The type of software development (e.g. database or real-time) and the target system (e.g. embedded processor, host system, programmable devices, or application-specific integrated circuits) are also taken into account (see Annex T of ECSS-E-ST-40). Specific requirements for verification, review, and inspection are imposed, for example, when full validation on the target computer is not feasible or performance goals are difficult to achieve.

Tailoring for product quality and business objectives is done by selecting requirements on quality of the product, as explained in clause 7 of this Standard, based on the quality objectives for the product specified by the customer.

## Security aspects of this Standard

Given the particularities related to security requirements, the following consideration is made:

Security assurance requirements significantly influence the security requirements for software product development. Higher levels of security assurance increase confidence in the security features but demand increased security controls on the development and evaluation of the software, to limit the potential for weaknesses and vulnerabilities present in the software.

# Software product assurance programme implementation

## Organization and responsibility

### Organization

ECSS-Q-ST-80\_0720001

The supplier shall ensure that an organizational structure is defined for software development, and that individuals have defined tasks and responsibilities.

### Responsibility and authority

#### .

ECSS-Q-ST-80\_0720002

The responsibility, the authority and the interrelation of personnel who manage, perform and verify work affecting software quality shall be defined and documented.

1. Software product assurance plan [PAF, SPAP; SRR].

#### .

ECSS-Q-ST-80\_0720003

The responsibilities and the interfaces of each organisation, either external or internal, involved in a project shall be defined and documented.

1. Software product assurance plan [PAF, SPAP; SRR].

#### .

ECSS-Q-ST-80\_0720004

The delegation of software product assurance tasks by a supplier to a lower level supplier shall be done in a documented and controlled way, with the supplier retaining the responsibility towards the customer.

1. Software product assurance plan [PAF, SPAP; SRR].

### Resources

#### .

ECSS-Q-ST-80\_0720005

The supplier shall provide adequate resources to perform the required software product assurance tasks.

1. Software product assurance plan [PAF, SPAP; SRR].

#### .

ECSS-Q-ST-80\_0720006

Reviews and audits of processes and of products shall be carried out by personnel not directly involved in the work being performed.

### Software product assurance manager/engineer

#### .

ECSS-Q-ST-80\_0720007

The supplier shall identify the personnel responsible for software product assurance for the project (SW PA manager/engineer).

1. Software product assurance plan [PAF, SPAP; SRR].

#### .

ECSS-Q-ST-80\_0720008

The software product assurance manager/engineer shall

report to the project manager (through the project product assurance manager, if any);

have organisational authority and independence to propose and maintain a software product assurance programme in accordance with the project software product assurance requirements;

have unimpeded access to higher management as necessary to fulfil his/her duties.

### Training

#### .

ECSS-Q-ST-80\_0720009

The supplier shall review the project requirements to establish and make timely provision for acquiring or developing the resources and skills for the management and technical staff.

1. Training plan [MGT, -; SRR].

#### .

ECSS-Q-ST-80\_0720010

The supplier shall maintain training records.

1. Records of training and experience [PAF, -; -].

#### .

ECSS-Q-ST-80\_0720011

The supplier shall ensure that the right composition and categories of appropriately trained personnel are available for the planned activities and tasks in a timely manner.

#### .

ECSS-Q-ST-80\_0720012

The supplier shall determine the training subjects based on the specific tools, techniques, methodologies, and computer resources to be used in developing and managing software product.

1. Personnel can undergo training to acquire skills and knowledge relevant to the specific field with which the software is to deal.

ECSS-Q-ST-80\_0720314

The supplier shall ensure that personnel involved in the planned activities and tasks are appropriately trained for:

software security analysis, security audits;

security engineering;

security assurance methods and tools;

security rules, policies and procedures applicable to the project.

1. It is good practice to make use of security certified organizations performing these types of training.

## Software product assurance programme management

### Software product assurance planning and control

#### .

ECSS-Q-ST-80\_0720013

The supplier shall develop a software product assurance plan in response to the software product assurance requirements in conformance with DRD in annex B.

ECSS-Q-ST-80\_0720014

The software product assurance plan shall be either a standalone document or a section of the supplier overall product assurance plan.

1. Software product assurance plan [PAF, SPAP; SRR, PDR].

#### .

ECSS-Q-ST-80\_0720015

Any internal manuals, standards or procedures referred to by the software product assurance plan shall become an integral part of the supplier’s software product assurance programme.

#### .

ECSS-Q-ST-80\_0720016

The software product assurance plan shall be revisited and updated as needed at each milestone to ensure that the activities to be undertaken in the following phase are fully defined.

1. Software product assurance plan [PAF, SPAP; CDR, QR, AR, ORR].

#### .

ECSS-Q-ST-80\_0720017

Before acceptance review, the supplier shall either supplement the software product assurance plan to specify the quality measures related to the operations and maintenance processes, or issue a specific software product assurance plan.

1. Software product assurance plan [PAF, SPAP; AR].

#### .

ECSS-Q-ST-80\_0720018

The supplier shall provide with the software product assurance plan a compliance matrix documenting conformance with the individual software product assurance requirements applicable for the project or business agreement.

1. Software product assurance plan [PAF, SPAP; SRR, PDR].

ECSS-Q-ST-80\_0720019

For each software product assurance requirement, the compliance matrix shall provide a reference to the document where the expected output of that requirement is located.

1. Software product assurance plan [PAF, SPAP; SRR, PDR].

### Software product assurance reporting

#### .

ECSS-Q-ST-80\_0720020

The supplier shall report on a regular basis on the status of the software product assurance programme implementation, if appropriate as part of the overall product assurance reporting of the project.

1. Software product assurance reports [PAF, -; -].

#### .

ECSS-Q-ST-80\_0720021

The software product assurance report shall include:

an assessment of the current quality of the product and processes, based on measured properties, with reference to the metrication as defined in the software product assurance plan;

verifications undertaken;

problems detected;

problems resolved.

1. Software product assurance reports [PAF, -; -].

#### .

ECSS-Q-ST-80\_0720022

The supplier shall deliver at each milestone review a software product assurance milestone report, covering the software product assurance activities performed during the past project phases.

1. Software product assurance milestone report [PAF, SPAMR; SRR, PDR, CDR, QR, AR, ORR].

### Audits

ECSS-Q-ST-80\_0720023

For software audits, ECSS-Q-ST-10 clause 5.2.3 shall apply.

1. Audit plan and schedule [PAF, -; SRR].

### Alerts

ECSS-Q-ST-80\_0720024

For software alerts, ECSS-Q-ST-10 clause 5.2.9 shall apply.

1. The following outputs are expected:

a. Preliminary alert information [PAF, -; -];

b. Alert information [PAF, -; -].

### Software problems

#### .

ECSS-Q-ST-80\_0720025

The supplier shall define and implement procedures for the logging, analysis and correction of all software problems, encountered during software development.

1. Software problem reporting procedures [PAF, -; PDR].

#### .

ECSS-Q-ST-80\_0720026

The software problem report shall contain the following information:

identification of the software item;

description of the problem;

recommended solution;

final disposition;

modifications implemented (e.g. documents, code, and tools);

tests re-executed.

1. Software problem reporting procedures [PAF, -; PDR].

#### .

ECSS-Q-ST-80\_0720027

The procedures for software problems shall define the interface with the nonconformance system (i.e. the circumstances under which a problem qualifies as a nonconformance).

1. Software problem reporting procedures [PAF, -; PDR].

#### .

ECSS-Q-ST-80\_0720028

The supplier shall ensure the correct application of problem reporting procedures.

### Nonconformances

#### .

ECSS-Q-ST-80\_0720029

For software nonconformance handling, ECSS-Q-ST-10-09 shall apply.

1. The following outputs are expected:

a. NCR SW procedure as part of the Software product assurance plan [PAF, SPAP; SRR];

b. Nonconformance reports [DJF, -; -].

ECSS-Q-ST-80\_0720030

When dealing with software nonconformance, the NRB shall include, at least, a representative from the software product assurance and the software engineering organizations.

1. Identification of SW experts in NRB [MGT, -; SRR]

ECSS-Q-ST-80\_0720312

The NRB shall dispose software nonconformances according to the following criteria:

use “as‐is”, when the software is found to be usable without eliminating the nonconformance;

fix, when the software product can be made fully in conformance with all specified requirements, by:

correction of the software,

addition of software patches, or

re‐design.

return to supplier, for procured software products (e.g. COTS).

1. Nonconformance reports [DJF, -; -].

ECSS-Q-ST-80\_0720315

When dealing with software nonconformance with a possible security impact, the NRB shall include a representative from the software security engineering organizations.

#### .

ECSS-Q-ST-80\_0720031

The software product assurance plan shall specify the point in the software life cycle from which the nonconformance procedures apply.

1. Software product assurance plan [PAF, SPAP; SRR, PDR].

### Quality requirements and quality models

#### .

ECSS-Q-ST-80\_0720032

Quality models shall be used to specify the software quality requirements.

1. Software product assurance plan [PAF, SPAP; PDR].

#### .

ECSS-Q-ST-80\_0720033

The following characteristics shall be used to specify the quality model:

functionality;

reliability;

maintainability;

reusability;

suitability for safety;

security;

usability;

efficiency;

portability;

software development effectiveness.

* 1. 1 Quality models are the basis for the identification of process metrics (see clause 6.2.5) and product metrics (see clause 7.1.4).
  2. 2 Quality models are also addressed by ISO/IEC 25000 and ECSS-Q-HB-80-04.
  3. 3 Software security vulnerabilities should be taken into account in the quality model.

## Risk management and critical item control

### Risk management

ECSS-Q-ST-80\_0720034

Risk management for software shall be performed by cross-reference to the project risk policy, as specified in ECSS-M-ST-80.

### Critical item control

#### .

ECSS-Q-ST-80\_0720035

For critical item control, ECSS-Q-ST-10-04 shall apply.

#### .

ECSS-Q-ST-80\_0720036

The supplier shall identify the characteristics of the software items that qualify them for inclusion in the Critical Item List.

## Supplier selection and control

### Supplier selection

#### .

ECSS-Q-ST-80\_0720037

For supplier selection ECSS-Q-ST-20 clause 5.4.1 shall apply.

1. The following outputs are expected:

a. Results of pre­award audits and assessments [PAF, -; -];

b. Records of procurement sources [PAF, -; -].

#### .

ECSS-Q-ST-80\_0720038

For the selection of suppliers of existing software, including software contained in OTS equipment and units, the expected output of clauses 6.2.7.2 to 6.2.7.6 shall be made available.

1. Software reuse file [DJF, SRF; -].

### Supplier requirements

#### .

ECSS-Q-ST-80\_0720039

The supplier shall establish software product assurance requirements for the next level suppliers, tailored to their role in the project, including a requirement to produce a software product assurance plan.

1. Software product assurance requirements for suppliers [PAF, -; SRR].

#### .

ECSS-Q-ST-80\_0720040

The supplier shall provide the software product assurance requirements applicable to the next level suppliers for customer’s acceptance.

1. Software product assurance requirements for suppliers [PAF, -; SRR].

### Supplier monitoring

#### .

ECSS-Q-ST-80\_0720041

The supplier shall monitor the next lower level suppliers’ conformance to the product assurance requirements.

#### .

ECSS-Q-ST-80\_0720042

The monitoring process shall include the review and approval of the next lower level suppliers’ product assurance plans, the continuous verification of processes and products, and the monitoring of the final validation of the product.

#### .

ECSS-Q-ST-80\_0720043

The supplier shall ensure that software development processes are defined and applied by the next lower level suppliers in conformance with the software product assurance requirements for suppliers.

1. Next level suppliers’ software product assurance plan [PAF, SPAP; PDR].

#### .

ECSS-Q-ST-80\_0720044

The supplier shall provide the next lower level suppliers’ software product assurance plan for customer’s acceptance.

1. Next level suppliers’ software product assurance plan [PAF, SPAP; PDR].

### Criticality classification

ECSS-Q-ST-80\_0720045

The supplier shall provide the lower level suppliers with the relevant results of the safety and dependability analyses performed at higher and his level (ref. clauses 6.2.2.1 and 6.2.2.2), including:

the criticality classification of the software products to be developed;

information about the failures that can be caused at higher level by the software products to be developed.

1. Safety and dependability analyses results for lower level suppliers [RB, -; SRR].

### Security sensitivity

ECSS-Q-ST-80\_0720316

The supplier shall provide its suppliers with the relevant results of the security analyses performed at higher and his level as specified in 6.2.9.2), including:

the security sensitivity of the software products to be developed;

information about the failures, attacks and related security impacts that can be caused at higher level by the software products during development and operation.

1. Security analyses results for lower level suppliers [SF,-;SRR].

## Procurement

### Procurement documents

ECSS-Q-ST-80\_0720046

For procurement documents, ECSS-Q-ST-20 clause 5.4.2 shall apply.

### Review of procured software component list

ECSS-Q-ST-80\_0720047

The choice of procured software shall be described and submitted for customer review.

1. Software development plan [MGT, SDP; SRR, PDR].

### Procurement details

ECSS-Q-ST-80\_0720048

For each of the software items the following data shall be provided:

ordering criteria

receiving inspection criteria;

back­up solutions if the product becomes unavailable;

contractual arrangements with the supplier for the development, maintenance and upgrades to new releases;

country source information if required by the customer.

1. Procurement data [MGT, -; SRR, PDR].
2. Examples of ordering criteria are: versions, options, security certifications and extensions.

### Identification

ECSS-Q-ST-80\_0720049

All the procured software shall be identified and registered by configuration management.

### Inspection

ECSS-Q-ST-80\_0720050

The supplier shall subject the procured software to a planned receiving inspection, in accordance with ECSS-Q-ST-20 clause 5.4.4, and the receiving inspection criteria as required by clause 5.5.3.

1. Receiving inspection report [PAF, -; PDR, CDR, QR].

### Exportability

ECSS-Q-ST-80\_0720051

Exportability constraints shall be identified.

1. Exportability constraints include constraints concerning applicable security export-control regulations.

## Tools and supporting environment

### Methods and tools

#### .

ECSS-Q-ST-80\_0720052

Methods and tools to be used for all the activities of the development cycle, (including requirements analysis, software specification, modelling, design, coding, validation, testing, configuration management, verification and product assurance) shall be identified by the supplier and agreed by the customer.

1. Software product assurance plan [PAF, SPAP; SRR, PDR].

#### .

ECSS-Q-ST-80\_0720053

The choice of development methods and tools shall be justified by demonstrating through testing or documented assessment that:

the development team has appropriate experience or training to apply them,

the tools and methods are appropriate for the functional and operational characteristics of the product, and

the tools are available throughout the development and maintenance lifetime of the product,

the tools and methods are appropriate to the security sensitivity of the product as determined by the security analysis and as defined in the software security management plan.

1. Software product assurance milestone report [PAF, SPAMR; SRR, PDR].

#### .

ECSS-Q-ST-80\_0720054

The correct use of methods and tools shall be verified and reported.

1. Software product assurance reports [PAF, -; -].

### Development environment selection

#### .

ECSS-Q-ST-80\_0720055

The software development environment shall be selected according to the following criteria:

availability;

compatibility;

performance;

maintenance;

durability and technical consistency with the operational equipment;

the assessment of the product with respect to requirements, including the criticality category;

the available support documentation;

the acceptance and warranty conditions;

the conditions of installation, preparation, training and use;

the maintenance conditions, including the possibilities of evolutions;

copyright and intellectual property rights constraints;

dependence on one specific supplier;

the assessment of the product with respect to the security sensitivity level of the products;

compliance with appropriate security requirements due to organizational or national security regulations, policies or directives.

1. Software development plan [MGT, SDP; SRR, PDR].

#### .

ECSS-Q-ST-80\_0720056

The suitability of the software development environment shall be justified.

1. Software development plan [MGT, SDP; SRR, PDR].

#### .

ECSS-Q-ST-80\_0720057

The availability of the software development environment to developers and other users shall be verified before the start of each development phase.

## Assessment and improvement process

### Process assessment

ECSS-Q-ST-80\_0720058

The supplier shall monitor and control the capability of the processes used during the development of the software, including the relevant processes corresponding to the services called from other organizational entities outside the project team.

1. Software process assessment records: Overall assessments and improvement programme plan [PAF, -; -].
   1. 1 The process assessment and improvement performed at organization level can be used to provide evidence of compliance for the project.
   2. 2 Process capability is the default process characteristic to be monitored and controlled. It is a characterization of the ability of a process to repeatability, predictability and consistently deliver its defined outcomes.

### Assessment process

#### .

ECSS-Q-ST-80\_0720059

The process assessment model and method to be used when performing any software process assessment shall be documented.

1. The following outputs are expected:

a. Software process assessment record: assessment model [PAF, -; -];

b. Software process assessment record: assessment method [PAF, -; -].

#### .

ECSS-Q-ST-80\_0720060

Assessments performed and process assessment models used shall be in conformance with ISO/IEC 33002:2015.

1. The following outputs are expected:

a. Software process assessment record: evidence of conformance of the process assessment model [PAF, -; -];

b. Software process assessment record: assessment method [PAF, -; -].

* 1. 1 Currently the model and method documented in ECSS-Q-HB-80-02 are not fully conformant to ISO/IEC 33002, however they can also be used with the customer agreement.
  2. 2 Currently the CMMI model is not fully conformant to ISO/IEC 15504, however it can be used, provided that the SCAMPI A method is applied.
  3. 3 Organisations that meet the requirements to be designated as Very Small Entity (VSE): enterprises, organizations, departments or projects having up to 25 people, can choose to use a model and a method conformant to ISO/IEC 29110-6-1.

#### .

ECSS-Q-ST-80\_0720061

The process assessment model, the method, the assessment scope, the results and the assessors shall be

established by the customer, and

verified that they are compliant with the project requirements.

* 1. 1 Examples of assessment scopes are: organizational unit evaluated, and processes evaluated.
  2. 2 ECSS-Q-HB-80-02 provides space specific process reference model and their indicators.

#### .

ECSS-Q-ST-80\_0720062

Assessments, shall be performed by a competent assessor.

1. Software process assessment record: competent assessor justification [PAF, -; -].
2. The competency of assessors could be established by the customer on the basis of ISO/IEC 33001 or through certification.

### Process improvement

#### .

ECSS-Q-ST-80\_0720063

The results of the assessment shall be used as feedback to improve as necessary the performed processes, to recommend changes in the direction of the project, and to determine technology advancement needs.

ECSS-Q-ST-80\_0720064

The suppliers shall ensure that the results of previous assessments are used in its project activity

1. Software process assessment records: improvement plan [PAF, -; -].

#### .

ECSS-Q-ST-80\_0720065

The process improvement shall be conducted according to a documented process improvement process.

1. Software process assessment records: improvement process [PAF, -; -].
   1. 1 For the definition of the process improvement process, see ECSS-Q-HB-80-02.
   2. 2 For CMMI, the process improvement is described in the OPF (Organizational Process Focus) process area.

#### .

ECSS-Q-ST-80\_0720066

Evidence of the improvement in performed processes shall be provided.

1. Software process assessment records: evidence of improvements [PAF, -; -].
   1. 1 See ECSS-Q-HB-80-02.
   2. 2 Depending on the context, improvements and changes at organisation level can be used to claim process improvement at project level.

# Software process assurance

## Software development life cycle

### Life cycle definition

ECSS-Q-ST-80\_0720067

The software development life cycle shall be defined or referenced in the software product assurance plan.

ECSS-Q-ST-80\_0720068

The following characteristics of the software life cycle shall be defined:

phases;

input and output of each phase;

status of completion of phase output;

milestones;

dependencies;

responsibilities;

role of the customer at each milestone review, in conformance with ECSS-M-ST-10 and ECSS-M-ST-10-01.

1. Software product assurance plan [PAF, SPAP; SRR, PDR].

### Process quality objectives

ECSS-Q-ST-80\_0720069

In the definition of the life cycle and associated milestones and documents, the quality objectives shall be used.

### Life cycle definition review

ECSS-Q-ST-80\_0720070

The software life cycle shall be reviewed against the contractual software engineering, product assurance and security assurance requirements.

### Life cycle resources

ECSS-Q-ST-80\_0720071

The software life cycle shall be reviewed for suitability and for the availability of resources to implement it by all functions involved in its application.

ECSS-Q-ST-80\_0720317

The disposal and release of the resources used after the software development concludes, shall be performed in accordance with the relevant security requirements.

### Software validation process schedule

ECSS-Q-ST-80\_0720072

A mandatory TRR, as specified in clause 5.3.5.1 of ECSS-E-ST-40, shall be held, in the form of a Project Review as specified in clause 5.3.3.2 ECSS-E-ST-40, before the beginning of the validation against the requirements baseline, to check that:

the software status is compatible with the commencement of validation activities;

the necessary resources, software product assurance plans, test and validation documentation, simulators or other technical means are available and ready for use.

1. Confirmation of readiness of test activities [DJF; TRR].

## Requirements applicable to all software engineering processes

### Documentation of processes

#### .

ECSS-Q-ST-80\_0720073

The following activities shall be covered either in software-specific plans or in project general plans:

development;

specification, design and customer documents to be produced;

configuration and documentation management;

verification, testing and validation activities;

maintenance and retirement;

security management.

1. Software project plans [MGT, MF, DJF, SF].

#### .

ECSS-Q-ST-80\_0720074

All plans shall be finalized before the start of the related activities.

1. Software project plans [MGT, MF, DJF].

#### .

ECSS-Q-ST-80\_0720075

All plans shall be updated for each milestone to reflect any changes during development.

1. Software project plans [MGT, MF, DJF].

#### .

ECSS-Q-ST-80\_0720076

The software product assurance plan shall identify all plans to be produced and used, the relationship between them and the time­scales for their preparation and update.

1. Software product assurance plan [PAF, SPAP; SRR, PDR].

#### .

ECSS-Q-ST-80\_0720077

Each plan shall be reviewed against the relevant contractual requirements.

#### .

ECSS-Q-ST-80\_0720078

Procedures and project standards shall address all types of software products included in the project.

1. Procedures and standards [PAF, -; PDR].

#### .

ECSS-Q-ST-80\_0720079

All procedures and project standards shall be finalized before starting the related activities.

1. Procedures and standards [PAF, -; PDR].

#### .

ECSS-Q-ST-80\_0720080

Each procedure or standard shall be reviewed against the relevant plans and contractual requirements.

#### .

ECSS-Q-ST-80\_0720081

Before any activity is started, each procedure or standard for that activity shall be reviewed by all functions involved in its application, for suitability and for the availability of resources to implement it.

### Software dependability and safety

#### .

ECSS-Q-ST-80\_0720082

For the system-level analyses leading to the criticality classification of software products based on the severity of failures consequences, ECSS-Q-ST-40 clause 6.5.6.3, and ECSS-Q-ST-30 clause 5.4, shall apply.

1. Criticality classification of software products [PAF, -; SRR, PDR].

#### .

ECSS-Q-ST-80\_0720083

The supplier shall perform a software dependability and safety analysis of the software products, using the results of system-level safety and dependability analyses, in order to determine the criticality of the individual software components.

1. Software dependability and safety analysis report [PAF, -; PDR].

#### .

ECSS-Q-ST-80\_0720084

The supplier shall identify the methods and techniques for the software dependability and safety analysis to be performed at technical specification and design level.

ECSS-Q-ST-80\_0720085

Methods and techniques for software dependability and safety analysis shall be agreed between the supplier and customer.

1. ECSS-Q-HB-80-03 provides indication on methods and techniques that can be applied such as:
   * + software failure modes and effects analysis (for the performing of this analysis, see also ECSS-Q-ST-30-02);
     + software fault tree analysis;
     + software common cause failure analysis.

#### .

ECSS-Q-ST-80\_0720086

Based on the results of the software criticality analysis, the supplier shall apply engineering measures to reduce the number of critical software components and mitigate the risks associated with the critical software (ref. clause 6.2.3).

#### .

ECSS-Q-ST-80\_0720087

The supplier shall report on the status of the implementation and verification of the SW dependability and safety analysis recommendations.

1. Software dependability and safety analysis report [PAF, -; CDR, QR, AR].

#### .

ECSS-Q-ST-80\_0720088

The supplier shall update the software dependability and safety analysis at each software development milestone, to confirm the criticality category of software components.

1. Software dependability and safety analysis report [PAF, -; CDR, QR, AR].

#### .

ECSS-Q-ST-80\_0720089

The supplier shall provide the results of the software dependability and safety analysis for integration into the system-level dependability and safety analyses, addressing in particular:

additional failure modes identified at software design level;

recommendations for system-level activities.

1. For example: introduction of hardware inhibits, and modifications of the system architecture.

#### .

ECSS-Q-ST-80\_0720090

As part of the software requirements analysis activities (ref. clause 6.3.2), the supplier shall contribute to the Hardware-Software Interaction Analysis (HSIA) by identifying, for each hardware failure included in the HSIA, the requirements that specify the software behaviour in the event of that hardware failure.

#### .

ECSS-Q-ST-80\_0720091

During the verification and validation of the software requirements resulting from the Hardware-Software Interaction Analysis, the supplier shall verify that the software reacts correctly to hardware failures, and no undesired software behaviour occurs that lead to system failures.

#### .

ECSS-Q-ST-80\_0720313

If it cannot be prevented that software components cause failures of higher criticality components, due to failure propagation or use of shared resources, then all the involved components shall be classified at the highest criticality category among them.

1. Failures of higher-criticality software components caused by lower-criticality components can be prevented by design measures such as separate hardware platforms, isolation of software processes or prohibition of shared memory (segregation and partitioning).

### Handling of critical software

#### .

ECSS-Q-ST-80\_0720092

<<deleted>>

ECSS-Q-ST-80\_0720093

<<deleted>>

#### .

ECSS-Q-ST-80\_0720094

The supplier shall define, justify and apply measures to assure the dependability and safety of critical software.

1. Software product assurance plan [PAF, SPAP; PDR, CDR].
2. These measures can include:
   * + use of software design or methods that have performed successfully in a similar application;
     + insertion of features for failure isolation and handling (ref. ECSS-Q-HB-80-03, software failure modes and effects analysis);
     + defensive programming techniques, such as input verification and consistency checks;
     + use of a “safe subset” of programming language;
     + use of formal design language for formal proof;
     + 100 % code branch coverage at unit testing level;
     + full inspection of source code;
     + witnessed or independent testing;
     + gathering and analysis of failure statistics;
     + removing deactivated code or showing through a combination of analysis and testing that the means by which such code can be inadvertently executed are prevented, isolated, or eliminated;
     + use of dynamic code verification techniques.

#### .

ECSS-Q-ST-80\_0720095

The application of the chosen measures to handle the critical software shall be verified.

1. Software product assurance milestone report [PAF, SPAMR; PDR, CDR, QR, AR].

#### .

ECSS-Q-ST-80\_0720096

Critical software shall be subject to regression testing after:

any change of functionality of the underlying platform hardware;

any change of the tools that affect directly or indirectly the generation of the executable code.

* 1. 1 In case of minor changes in tools that affect the generation of the executable code, a binary comparison of the executable code generated by the different tools can be used to verify that no modifications are introduced.
  2. 2 Example for item 1: instruction set of a processor.

#### .

ECSS-Q-ST-80\_0720097

The need for additional verification and validation of critical software shall be analysed after:

any change of functionality or performance of the underlying platform hardware;

any change in the environment in which the software or the platform hardware operate.

1. Software product assurance plan [PAF, SPAP; PDR, CDR].

#### .

ECSS-Q-ST-80\_0720098

Identified unreachable code shall be removed and the need for re-verification and re-validation shall be analysed.

#### .

ECSS-Q-ST-80\_0720099

Unit and integration testing shall be (re-)executed on non-instrumented code.

#### .

ECSS-Q-ST-80\_0720100

Validation testing shall be (re-)executed on non-instrumented code.

### Software configuration management

#### .

ECSS-Q-ST-80\_0720101

ECSS-M-ST-40 shall be applied for software configuration management, complemented by the following requirements.

#### .

ECSS-Q-ST-80\_0720102

The software configuration management system shall allow any reference version to be re­generated from backups.

1. Software configuration management plan [MGT, SCMP; SRR, PDR].

#### .

ECSS-Q-ST-80\_0720103

The software configuration file and the software release document shall be provided with each software delivery.

1. The following outputs are expected:

a. Software configuration file [DDF, SCF; -];

b. Software release document [DDF, SRelD; -].

#### .

ECSS-Q-ST-80\_0720104

The software configuration file shall be available and up to date for each project milestone.

1. Software configuration file [DDF, SCF; CDR, QR, AR, ORR].

#### .

ECSS-Q-ST-80\_0720105

Any components of the code generation tool that are customizable by the user shall be put under configuration control.

ECSS-Q-ST-80\_0720106

The change control procedures defined for the project shall address the specific aspects of these components.

1. The following outputs are expected:

a. Software configuration file [DDF, SCF; CDR, QR, AR, ORR];

b. Software configuration management plan [MGT, SCMP; SRR, PDR].

#### .

ECSS-Q-ST-80\_0720107

The supplier shall ensure that all authorized changes are implemented in accordance with the software configuration management plan.

1. Authorized changes - Software configuration file [DDF, SCF; CDR, QR, AR, ORR].

#### .

ECSS-Q-ST-80\_0720108

The following documents shall be controlled (see ECSS-Q-ST-10 clause 5.2.5):

procedural documents describing the quality system to be applied during the software life cycle;

planning documents describing the planning and progress of the activities;

documents describing a particular software product, including:

development phase inputs,

development phase outputs,

verification and validation plans and results,

test case specifications, test procedures and test reports,

traceability matrices,

validation control documentation,

documentation for the software and system operators and users,

maintenance documentation, and

Retirement documentation.

#### .

ECSS-Q-ST-80\_0720109

The supplier shall identify a method and tool to protect the supplied software against corruption.

1. For example: source code, executable and data.

ECSS-Q-ST-80\_0720318

The supplier shall select methods and tools to protect the supplied software in accordance with the security analysis.

ECSS-Q-ST-80\_0720319

The security analysis can define security requirements on software integrity checks.

1. The following outputs are expected:

a. Software product assurance plan [PAF, SPAP; SRR, PDR];

b. Software configuration file [DDF, SCF; CDR, QR, AR, ORR];

c. Software security management plan [SF, SSMP; SRR, PDR].

#### .

ECSS-Q-ST-80\_0720110

The supplier shall specify the necessary elements and mechanisms to prove integrity and authenticity of the SW product.

1. The following outputs are expected:

a. Software product assurance plan [PAF, SPAP; SRR, PDR];

b. Software security management plan, [SF, SSMP; , SRR].

1. A valid method for integrity verification could be the use of checksum.

#### .

ECSS-Q-ST-80\_0720111

The integrity verification value shall be provided in the software configuration file with each software delivery.

1. Software configuration file [DDF, SCF; -].

#### .

ECSS-Q-ST-80\_0720112

For every software delivery from the supplier to the customer, the following information shall be provided as a minimum:

the software name;

the version number;

the reference to the software configuration file;

information and media in accordance with any requirements for delivery and protective security marking;

any annotation and caveats used to indicate or further constraint the distribution limitations.

1. The following outputs are expected:

a. Software product assurance plan [PAF, SPAP; SRR, PDR];

b. Labels [DDF, -; -];

c. Software security management plan [SF, SSMP; SRR];

d. Protective security markings [SF,-,-].

1. Protective security marking, e.g. security classification, is mandated by organisational or national security regulations, policies or directives.

#### .

ECSS-Q-ST-80\_0720320

Procedures shall be established for software branching and merging and the corresponding use of software configuration management tools.

1. Software configuration management plan [MGT, SCMP; SRR, PDR].

### Process metrics

#### .

ECSS-Q-ST-80\_0720113

Metrics shall be used to manage the development and to assess the quality of the development processes.

1. Process metrics are based on quality models (see clause 5.2.7).

#### .

ECSS-Q-ST-80\_0720114

Process metrics shall be collected, stored and analysed on a regular basis by applying quality models and procedures.

1. Software product assurance plan [PAF, SPAP; SRR, PDR].

#### .

ECSS-Q-ST-80\_0720115

The following basic process metrics shall be used within the supplier’s organization:

duration: how phases and tasks are being completed versus the planned schedule;

effort: how much effort is consumed by the various phases and tasks compared to the plan.

1. Internal metrics report.

#### .

ECSS-Q-ST-80\_0720116

Process metrics shall be used within the supplier’s organization and reported to the customer, including:

number of problems detected during verification;

number of problems detected during integration and validation testing and use.

1. See also software problem reporting described in clause 5.2.5.

#### .

ECSS-Q-ST-80\_0720117

Metrics reports shall be included in the software product assurance reports.

1. Software product assurance reports [PAF, -; -].

### Verification

#### .

ECSS-Q-ST-80\_0720118

Activities for the verification of the quality requirements shall be specified in the definition of the verification plan.

1. Software verification plan [DJF, SVerP; SRR, PDR].
2. Verification includes various techniques such as review, inspection, walk­through, cross-reading, desk-checking, model simulation, and many types of analysis such as traceability analysis, formal proof or fault tree analysis.

#### .

ECSS-Q-ST-80\_0720119

The outputs of each development activity shall be verified for conformance against pre-defined criteria.

ECSS-Q-ST-80\_0720120

Only outputs which have been subjected to planned verifications shall be used as inputs for subsequent activities.

1. Software product assurance reports [PAF, -; -].

#### .

ECSS-Q-ST-80\_0720121

A summary of the assurance activities concerning the verification process and their findings shall be included in software product assurance reports.

1. Software product assurance reports [PAF, -; -].

#### .

ECSS-Q-ST-80\_0720122

The completion of actions related to software problem reports generated during verification shall be verified and recorded.

1. Software problem reports [DJF, -; SRR, PDR, CDR, QR, AR, ORR].

#### .

ECSS-Q-ST-80\_0720123

Software containing deactivated code shall be verified specifically to ensure that the deactivated code cannot be activated or that its accidental activation cannot harm the operation of the system.

1. Software verification report [DJF, SVR; CDR, QR, AR].

#### .

ECSS-Q-ST-80\_0720124

Software containing configurable code shall be verified specifically to ensure that any unintended configuration cannot be activated at run time or included during code generation.

1. Software verification report [DJF, SVR; CDR, QR, AR].

#### .

ECSS-Q-ST-80\_0720125

The supplier shall ensure that:

the planned verification activities are adequate to confirm that the products of each phase are conformant to the applicable requirements;

the verification activities are performed according to the plan.

1. Software product assurance reports [PAF, -; -].

#### .

ECSS-Q-ST-80\_0720126

Reviews and inspections shall be carried out according to defined criteria, and according to the defined level of independence of the reviewer from the author of the reviewed item.

#### .

ECSS-Q-ST-80\_0720127

Each review and inspection shall be based on a written plan or procedure.

1. For projects reviews, ECSS-E-ST-40 clause 5.3.3.3, bullet b and Annex P are applicable.

#### .

ECSS-Q-ST-80\_0720128

The review or inspection plans or procedures shall specify:

the reviewed or inspected items;

the person in charge;

the participants;

the means of review or inspection (e.g. tools or check list);

the nature of the report.

1. Review and inspection plans or procedures [PAF, -; -].

#### .

ECSS-Q-ST-80\_0720129

Review and inspection reports shall:

refer to the corresponding review/inspection procedure or plan;

identify the reviewed item, the author, the reviewer, the review criteria and the findings of the review.

1. Review and inspection reports [PAF, -; -].

#### .

ECSS-Q-ST-80\_0720130

Traceability matrices (as defined in ECSS-E-ST-40 clause 5.8) shall be verified at each milestone.

1. Software product assurance milestone report [PAF, SPAMR; SRR, PDR, CDR, QR, AR, ORR].

#### .

ECSS-Q-ST-80\_0720131

Independent software verification shall be performed by a third party.

1. This requirement is applicable where the risks associated with the project justify the costs involved. The customer can consider a less rigorous level of independence, e.g. an independent team in the same organization.

ECSS-Q-ST-80\_0720132

Independent software verification shall be a combination of reviews, inspections, analyses, simulations, and auditing.

1. Guidance on ISVV purpose, scope, and activities is provided in CSW-ESAISVV-2022-GBK-02897.

ECSS-Q-ST-80\_0720321

Independent software verification shall be performed in addition to, and shall not fully or partly replace, verification activities to be carried out by the supplier in accordance with ECSS-E-ST-40, clause 5.8.

1. The following outputs are expected:

a. ISVV plan [DJF, -; SRR, PDR];

b. ISVV report [DJF, -; PDR, CDR, QR, AR].

### Reuse of existing software

#### General

The requirements in 6.2.7 do not apply to tools and software development environment, for which requirements of clause 5.6 apply.

#### .

ECSS-Q-ST-80\_0720133

Analyses of the advantages to be obtained with the selection of existing software (ref. 3.2.10) instead of new development shall be carried out.

1. The following outputs are expected:

a. Software reuse approach, including approach to delta qualification [PAF, SPAP; SRR, PDR];

b. Software reuse file [DJF, SRF; SRR, PDR].

#### .

ECSS-Q-ST-80\_0720134

The existing software shall be assessed with regards to the applicable functional, performance and quality requirements.

ECSS-Q-ST-80\_0720322

The existing software shall be assessed with regards to the applicable security requirements, addressing the following aspects:

Authorisation for use;

Security sensitivity;

Security assurance requirements;

Security evaluations, certifications and accreditations.

1. The following outputs are expected:

a. Software reuse approach, including approach to delta qualification [PAF, SPAP; SRR, PDR];

b. Software reuse file [DJF, SRF; SRR, PDR].

1. For example, Common Criteria (CC), FIPS, system accreditation, council approval.

#### .

ECSS-Q-ST-80\_0720135

The quality level of the existing software shall be analysed with respect to the project requirements, according to the criticality and sensitivity of the system function implemented, taking into account the following aspects:

software requirements documentation;

software architectural and detailed design documentation;

forward and backward traceability between system requirements, software requirements, design and code;

unit tests documentation and coverage;

integration tests documentation and coverage;

validation documentation and coverage;

verification reports;

performance;

operational performances;

residual nonconformances, waivers; and alerts;

user operational documentation;

code quality (adherence to coding standards, metrics);

known security vulnerabilities in the reused product, including vulnerabilities introduced by dependencies at all levels between components.

* 1. 1 Examples of performance are memory occupation, CPU load.
  2. 2 Example of user operation documentation is a user manual.

#### .

ECSS-Q-ST-80\_0720136

The results of the reused software analysis shall be recorded in the software reuse file, together with an assessment of the possible level of reuse and a description of the assumptions and the methods applied when estimating the level of reuse.

1. Results of the reused software analysis, such as detailed reference to requirement and design documents, test reports and coverage results.

#### .

ECSS-Q-ST-80\_0720137

The analysis of the suitability of existing software for reuse shall be complemented by an assessment of the following aspects:

the acceptance and warranty conditions;

the available support documentation;

the conditions of installation, preparation, training and use;

the identification and registration by configuration management;

maintenance responsibility and conditions, including the possibilities of changes;

the durability and validity of methods and tools used in the initial development, that are envisaged to be used again;

the copyright and intellectual property rights constraints (modification rights);

the licensing conditions;

exportability constraints.

1. Software reuse file [DJF, SRF; SRR, PDR].

#### .

ECSS-Q-ST-80\_0720138

Corrective actions shall be identified, documented in the reuse file and applied to the reused software not meeting the applicable requirements related to the aspects as specified in clauses 6.2.7.2 to 6.2.7.6.

1. Software reuse file [DJF, SRF; SRR, PDR].

#### .

ECSS-Q-ST-80\_0720139

Reverse engineering techniques shall be applied to generate missing documentation and to reach the required verification and validation coverage.

ECSS-Q-ST-80\_0720140

For software products whose life cycle data from previous development are not available and reverse engineering techniques are not fully applicable, the following methods shall be applied:

generation of validation and verification documents based on the available user documentation (e.g. user manual) and execution of tests in order to achieve the required level of test coverage;

use of the product service history to provide evidence of the product’s suitability for the current application, including information about:

relevance of the product service history for the new operational environment;

configuration management and change control of the software product;

effectiveness of problem reporting;

actual error rates and maintenance records;

impact of modifications;

number, types, priorities and correction rates of vulnerabilities.

1. Software reuse file [DJF, SRF; SRR, PDR].

#### .

ECSS-Q-ST-80\_0720141

The software reuse file shall be updated at project milestones to reflect the results of the identified corrective actions for reused software not meeting the project requirements.

1. Software reuse file [DJF, SRF; CDR, QR, AR].

#### .

ECSS-Q-ST-80\_0720142

All the reused software shall be kept under configuration control.

#### .

ECSS-Q-ST-80\_0720143

The detailed configuration status of the reused software baseline shall be provided to the customer in the reuse file for acceptance.

1. Software reuse file [DJF, SRF; SRR, PDR, CDR, QR, AR].

### Automatic code generation

#### .

ECSS-Q-ST-80\_0720144

For the selection of tools for automatic code generation, the supplier shall evaluate the following aspects:

evolution of the tools in relation to the tools that use the generated code as an input;

customization of the tools to comply with project standards;

portability requirements for the generated code;

collection of the required design and code metrics;

verification of software components containing generated code;

configuration control of the tools including the parameters for customisation;

compliance with open standards.

1. Examples for item 1: compilers or code management systems.

#### .

ECSS-Q-ST-80\_0720145

The requirements on testing applicable to the automatically generated code shall ensure the achievement of the same objectives as those for manually generated code.

1. Validation and testing documentation [DJF, SValP; PDR], [DJF, SVS; CDR, QR, AR], [DJF, SUITP; PDR, CDR].

#### .

ECSS-Q-ST-80\_0720146

The required level of verification and validation of the automatic generation tool shall be at least the same as the one required for the generated code, if the tool is used to skip verification or testing activities on the target code.

#### .

ECSS-Q-ST-80\_0720147

Modelling standards for automatic code generation tools shall be defined and applied.

1. Modelling standards [PAF, -; SRR, PDR].

#### .

ECSS-Q-ST-80\_0720148

Adherence to modelling standards shall be verified.

1. Software product assurance reports [PAF, -; -].

#### .

ECSS-Q-ST-80\_0720149

Clause 6.3.4 shall apply to automatically generated code, unless the supplier demonstrates that the automatically generated code does not need to be manually modified.

#### .

ECSS-Q-ST-80\_0720150

The verification and validation documentation shall address separately the activities to be performed for manually and automatically generated code.

1. Validation and testing documentation [DJF, SValP; PDR], [DJF, SVS; CDR, QR, AR], [DJF, SUITP; PDR, CDR].

### Software security

#### .

ECSS-Q-ST-80\_0720323

The software product assurance plan shall include security assurance.

1. Software product assurance plan [PAF, SPAP; SRR, PDR, CDR].

#### .

ECSS-Q-ST-80\_0720324

The supplier shall perform a software security analysis of the software products, using the results of system-level security analyses, where software products are part of a system, in order to determine the sensitivity of the individual software components.

1. Software security analysis report [SF, -; PDR].

#### .

ECSS-Q-ST-80\_0720325

The supplier shall identify the methods and techniques for the software security analysis.

ECSS-Q-ST-80\_0720326

Methods and techniques for software security analysis shall be agreed between the supplier and customer.

1. Examples of software security analysis are: Requirements analysis, security risk analysis, design analysis and code analysis.

#### .

ECSS-Q-ST-80\_0720327

Based on the results of the software security analysis, the supplier shall apply engineering measures to reduce the number of security sensitive software components and mitigate the risks associated with security sensitive software (ref. clause 6.2.10).

#### .

ECSS-Q-ST-80\_0720328

The supplier shall report on the status of the implementation and verification of the software security analysis recommendations.

1. Software security analysis report [SF, -; CDR, QR, AR]

#### .

ECSS-Q-ST-80\_0720329

The supplier shall update the software security analysis at each software development milestone, to confirm the security sensitivity and related security risks of software components.

1. Software security analysis report [SF, -; CDR, QR, AR].

#### .

ECSS-Q-ST-80\_0720330

The supplier shall provide the results of the software security analysis for integration into the system-level security analyses, addressing in particular:

additional security failure modes or vulnerabilities identified at software level;

recommendations for system-level security activities and changes to the system including requirements

1. Software security analysis report [SF, -; PDR, CDR].

### Handling of security sensitive software

#### .

ECSS-Q-ST-80\_0720331

The supplier shall define and implement measures to avoid propagation of failures, including the ones caused by deliberate action, between software components.

1. This can be achieved by design measures such as separate hardware platforms, fail secure isolation of software processes or prohibition of shared memory (segregation and partitioning).

ECSS-Q-ST-80\_0720332

The consequences for security of the malfunction of higher criticality components, and the consequences for safety and dependability of the malfunction of higher sensitivity components shall be analysed and any conflict resolved.

1. The following outputs are expected:

a. Software dependability and safety analysis report [PAF, -; PDR, CDR, QR, AR];

b. Software security analysis report [SF, -; PDR, CDR, QR, AR].

#### .

ECSS-Q-ST-80\_0720333

For security sensitive software, measures shall be defined, justified and applied, in addition to those in clause 6.2.3.2.

1. Software security management plan [SF, , SSMP;; PDR, CDR, QR, AR].
2. These measures can include:
   * + use of secure coding practices;
     + use of security baseline(s);
     + use of fuzzing, static, dynamic security testing;
     + use of vulnerability assessment and penetration testing.

#### .

ECSS-Q-ST-80\_0720334

Security sensitive software shall be subject to regression testing after:

any change of functionality of the underlying platform hardware;

any change of the tools that affect directly or indirectly the generation of the executable code;

any change in the security of the target operating environment.

1. The following outputs are expected:

a. Software product assurance plan [PAF, SPAP; PDR, CDR];

b. Software security management plan [SF, SSMP; PDR, CDR].

* 1. 1 This can be the result of changing security characteristics of the existing environment. Alternatively, this can be due to the migration of the software from one operating environment to another.
  2. 2 In case of minor changes in tools that affect the generation of the executable code, a binary comparison of the executable code generated by the different tools can be used to verify that no modifications are introduced.
  3. 3 Example for item 1: instruction set of a processor.

#### .

ECSS-Q-ST-80\_0720335

The need for additional V&V of security sensitive software shall be analysed after:

any change of functionality or performance of the underlying platform hardware;

any change in the environment in which the software or the platform hardware operate;

any change in knowledge of security threats, security vulnerabilities or security sensitivity of the system;

any change in the infrastructure and tools used to build environment.

1. The following outputs are expected:

a. Software product assurance plan [PAF, SPAP; PDR, CDR];

b. Software security management plan [SF, SSMP; PDR, CDR].

## Requirements applicable to individual software engineering processes or activities

### Software related system requirements process

#### .

ECSS-Q-ST-80\_0720151

For the definition of the software related system requirements to be specified in the requirements baseline, ECSS-E-ST-40 clause 5.2 shall apply.

#### .

ECSS-Q-ST-80\_0720152

The requirements baseline shall be subject to documentation control and configuration management as part of the development documentation.

#### .

ECSS-Q-ST-80\_0720153

For the definition of the requirements baseline, all results from the safety and dependability analyses (including results from the HSIA ECSS-Q-ST-30 clause 6.4.2.3) shall be used.

ECSS-Q-ST-80\_0720336

For the definition of the requirements baseline, all results from the security analysis shall be used.

### Software requirements analysis

#### .

ECSS-Q-ST-80\_0720154

The requirements baseline shall be analyzed to fully and unambiguously define the software requirements in the technical specification.

#### .

ECSS-Q-ST-80\_0720155

The technical specification shall be subject to documentation control and configuration management as part of the development documentation.

#### .

ECSS-Q-ST-80\_0720156

For the definition of the technical specification, all results from the safety and dependability analyses (including results from the HSIA ECSS-Q-ST-30 clause 6.4.2.3) shall be used.

ECSS-Q-ST-80\_0720337

For the definition of the technical specification, all results from the software security analysis shall be used.

#### .

ECSS-Q-ST-80\_0720157

In addition to the functional requirements, the technical specification shall include all non-functional requirements necessary to satisfy the requirements baseline, including, as a minimum, the following:

performance,

safety,

reliability,

robustness,

quality,

maintainability,

configuration management,

security,

privacy,

metrication, and

verification and validation.

1. Performance requirements include requirements on numerical accuracy.

#### .

ECSS-Q-ST-80\_0720158

Prior to the technical specification elaboration, customer and supplier shall agree on the following principles and rules as a minimum:

assignment of persons (on both sides) responsible for establishing the technical specification;

methods and tools for agreeing on requirements and approving changes;

efforts to prevent misunderstandings such as definition of terms, explanations of background of requirements;

recording and reviewing discussion results on both sides.

### Software architectural design and design of software items

#### .

ECSS-Q-ST-80\_0720159

The design definition file shall be subject to documentation control and configuration management.

#### .

ECSS-Q-ST-80\_0720160

Mandatory and advisory design standards shall be defined and applied.

1. Design standards [PAF, -; SRR, PDR].

#### .

ECSS-Q-ST-80\_0720161

For software in which numerical accuracy is relevant to mission success specific rules on design and code shall be defined to ensure that the specified level of accuracy is obtained.

1. Software product assurance plan [PAF, SPAP; PDR].
2. For example: for an attitude and orbit control subsystem, scientific data generation components.

#### .

ECSS-Q-ST-80\_0720162

Adherence to design standards shall be verified.

1. Software product assurance reports [PAF, -; -].

#### .

ECSS-Q-ST-80\_0720163

The supplier shall define means, criteria and tools to ensure that the complexity and modularity of the design meet the quality requirements.

ECSS-Q-ST-80\_0720164

The design evaluation shall be performed in parallel with the design process, in order to provide feedback to the software design team.

1. Software product assurance plan [PAF, SPAP; PDR].

#### .

ECSS-Q-ST-80\_0720165

Synthesis of the results obtained in the software complexity and modularity evaluation and corrective actions implemented shall be described in the software product assurance reports.

1. Software product assurance reports [PAF, -; -].

#### .

ECSS-Q-ST-80\_0720166

The supplier shall review the design documentation to ensure that it contains the appropriate level of information for maintenance activities.

1. The following outputs are expected:

a. Software product assurance plan [PAF, SPAP; PDR];

b. Software product assurance reports [PAF, -; -].

### Coding

#### .

ECSS-Q-ST-80\_0720167

Coding standards (including security, consistent naming conventions and adequate commentary rules) shall be specified and observed.

1. Coding standards [PAF, -; PDR].

#### .

ECSS-Q-ST-80\_0720168

The standards shall be consistent with the product quality requirements.

1. Coding standards depend on the software quality objectives (see clause 5.2.7).

#### .

ECSS-Q-ST-80\_0720169

The tools to be used in implementing and checking conformance with coding standards shall be identified in the product assurance plan before coding activities start.

1. Software product assurance plan [PAF, SPAP; PDR].

#### .

ECSS-Q-ST-80\_0720170

Coding standards shall be reviewed with the customer to ensure that they reflect product quality and security requirements.

1. Coding standards and description of tools [PAF, -; PDR].

#### .

ECSS-Q-ST-80\_0720171

Use of low-level programming languages shall be justified.

1. Software development plan [MGT, SDP; PDR].

#### .

ECSS-Q-ST-80\_0720172

The supplier shall define measurements, criteria and tools to ensure that the software code meets the quality and security requirements.

1. Software product assurance plan [PAF, SPAP; PDR].

ECSS-Q-ST-80\_0720173

The code evaluation shall be performed in parallel with the coding process, in order to provide feedback to the software programmers.

#### .

ECSS-Q-ST-80\_0720174

Synthesis of the code analysis results and corrective actions implemented shall be described in the software product assurance reports.

1. Software product assurance reports [PAF, -; -].

#### .

ECSS-Q-ST-80\_0720175

The code shall be put under configuration control immediately after successful unit testing.

### Testing and validation

#### .

ECSS-Q-ST-80\_0720176

Testing shall be performed in accordance with a strategy for each testing level (i.e. unit, integration, validation against the technical specification, validation against the requirements baseline, acceptance), which includes:

the types of tests to be performed;

the tests to be performed in accordance with the plans and procedures;

the means and organizations to perform assurance function for testing and validation.

1. For examples for item 1 are: functional, boundary, performance, security, and usability tests.

#### .

ECSS-Q-ST-80\_0720177

Based on the dependability and safety criticality, and the security sensitivity of the software, test coverage goals for each testing level shall be agreed between the customer and the supplier and their achievement monitored by metrics:

for unit level testing;

for integration level testing;

for validation against the technical specification and validation against the requirements baseline.

1. Software product assurance plan [PAF, SPAP; PDR, CDR].

#### .

ECSS-Q-ST-80\_0720178

The supplier shall ensure through internal review that the test procedures and data are adequate, feasible and traceable and that they satisfy the requirements.

1. Software product assurance reports [PAF, -; -].

#### .

ECSS-Q-ST-80\_0720179

Test readiness reviews shall be held before the commencement of test activities, as defined in the software development plan.

1. Test readiness review reports [DJF, -; TRR].

#### .

ECSS-Q-ST-80\_0720180

Test coverage shall be checked with respect to the stated goals.

1. Software product assurance reports [PAF, -; -].

ECSS-Q-ST-80\_0720181

Feedback from the results of test coverage evaluation shall be continuously provided to the software developers.

#### .

ECSS-Q-ST-80\_0720182

The supplier shall ensure that nonconformances and software problem reports detected during testing are properly documented and reported to those concerned.

1. Nonconformance reports and software problem reports [DJF, -; CDR, QR, AR, ORR].

#### .

ECSS-Q-ST-80\_0720183

The test coverage of configurable code shall be checked to ensure that the stated requirements are met in each tested configuration.

1. Statement of compliance with test plans and procedures [PAF, -; CDR, QR, AR, ORR].

#### .

ECSS-Q-ST-80\_0720184

The completion of actions related to software problem reports generated during testing and validation shall be verified and recorded.

1. Software problem reports [DJF, -; SRR, PDR, CDR, QR, AR, ORR].

#### .

ECSS-Q-ST-80\_0720185

Provisions shall be made to allow witnessing of tests by the customer.

#### .

ECSS-Q-ST-80\_0720186

Provisions shall be made to allow witnessing of tests by supplier personnel independent of the development.

1. For example: specialist software product assurance personnel.

#### .

ECSS-Q-ST-80\_0720187

The supplier shall ensure that:

tests are conducted in accordance with approved test procedures and data,

the configuration under test is correct,

the tests are properly documented, and

the test reports are up to date and valid.

1. Statement of compliance with test plans and procedures [PAF, -; CDR, QR, AR, ORR].

#### .

ECSS-Q-ST-80\_0720188

The supplier shall ensure that tests are repeatable by verifying the storage and recording of tested software, support software, test environment, supporting documents and problems found.

1. Software product assurance reports [PAF, -; -].

#### .

ECSS-Q-ST-80\_0720189

The supplier shall confirm in writing that the tests are successfully completed.

1. Testing and validation reports [DJF, -; CDR, QR, AR, ORR].

#### .

ECSS-Q-ST-80\_0720190

Review boards looking to engineering and product assurance aspects shall be convened after the completion of test phases, as defined in the software development plan.

ECSS-Q-ST-80\_0720338

Review boards looking to security aspects shall be convened after the completion of test phases, as defined in the software security management plan.

#### .

ECSS-Q-ST-80\_0720191

Areas affected by any modification shall be identified and re­tested (regression testing).

#### .

ECSS-Q-ST-80\_0720192

In case of re­testing, all test related documentation (test procedures, data and reports) shall be updated accordingly.

1. Updated test documentation [DJF, -; CDR, QR, AR, ORR].

#### .

ECSS-Q-ST-80\_0720193

The need for regression testing and additional verification of the software shall be analysed after any change of the platform hardware.

1. Updated test documentation [DJF, -; CDR, QR, AR, ORR].

#### .

ECSS-Q-ST-80\_0720194

The need for regression testing and additional verification of the software shall be analysed after a change or update of any tool used to generate it.

1. For example: source code or object code.

#### .

ECSS-Q-ST-80\_0720195

Validation shall be carried out by staff who have not taken part in the design or coding of the software being validated.

1. This can be achieved at the level of the whole software product, or on a component by component basis.

#### .

ECSS-Q-ST-80\_0720196

Validation of the flight software against the requirement baseline on the flight equipment model shall be performed on a software version without any patch.

#### .

ECSS-Q-ST-80\_0720197

The supplier shall review the test documentation to ensure that it is up to date and organized to facilitate its reuse for maintenance.

#### .

ECSS-Q-ST-80\_0720198

Tests shall be organized as activities in their own right in terms of planning, resources and team composition.

1. Test and validation documentation [DJF, SValP; PDR], [DJF, SUITP; PDR, CDR].

#### .

ECSS-Q-ST-80\_0720199

The necessary resources for testing shall be identified early in the life cycle, taking into account the operating and maintenance requirements.

1. Test and validation documentation [DJF, SValP; PDR], [DJF, SUITP; PDR, CDR].

#### .

ECSS-Q-ST-80\_0720200

Test tool development or acquisition (hardware and software) shall be planned for in the overall project plan.

1. Test and validation documentation [DJF, SValP; PDR], [DJF, SUITP; PDR, CDR].

#### .

ECSS-Q-ST-80\_0720201

The supplier shall establish and review the test procedures and data before starting testing activities and also document the constraints of the tests concerning physical, performance, functional, controllability and observability limitations.

1. Test and validation documentation [DJF, SValP; PDR], [DJF, SVS; CDR, QR, AR], [DJF, SUITP; PDR, CDR].

#### .

ECSS-Q-ST-80\_0720202

Before offering the product for delivery and customer acceptance, the supplier shall validate its operation as a complete product, under conditions similar to the application environment as specified in the requirements baseline.

#### .

ECSS-Q-ST-80\_0720203

When testing under the operational environment is performed, the following concerns shall be addressed:

the features to be tested in the operational environment;

the specific responsibilities of the supplier and customer for carrying out and evaluating the test;

restoration of the previous operational environment (after test).

1. Test and validation documentation [DJF, -; AR].

#### .

ECSS-Q-ST-80\_0720204

Independent software validation shall be performed by a third party.

1. This requirement is applicable where the risks associated with the project justify the costs involved. The customer can consider a less rigorous level of independence, e.g. an independent team in the same organization.

ECSS-Q-ST-80\_0720339

Independent software validation shall be performed in addition to, and not fully or partly replace, validation activities to be carried out by the supplier in accordance with ECSS-E-ST-40, clause 5.6.

1. The following outputs are expected:

a. ISVV plan [DJF, -; SRR, PDR];

b. ISVV report [DJF, -; PDR, CDR, QR, AR].

1. Guidance on ISVV purpose, scope, and activities is provided in CSW-ESAISVV-2022-GBK-02897.

#### .

ECSS-Q-ST-80\_0720205

The validation shall include testing in the different configurations possible or in a representative set of them when it is evident that the number of possible configurations is too high to allow validation in all of them.

1. Test and validation documentation [DJF, SValP; PDR], [DJF, SVS; CDR, QR, AR].

#### .

ECSS-Q-ST-80\_0720206

Software containing deactivated code shall be validated specifically to ensure that the deactivated code cannot be activated or that its accidental activation cannot harm the operation of the system.

1. Testing and validation reports [DJF, -; CDR, QR, AR].

#### .

ECSS-Q-ST-80\_0720207

Software containing configurable code shall be validated specifically to ensure that unintended configuration cannot be activated at run time or included during code generation.

1. Testing and validation reports [DJF, -; CDR, QR, AR].

#### .

ECSS-Q-ST-80\_0720208

Activities for the validation of the quality requirements shall be specified in the definition of the validation specification.

1. Software validation specification [DJF, SVS; CDR, QR, AR].

#### .

ECSS-Q-ST-80\_0720340

The software validation control information, as specified in clause 5.6.5 of ECSS-E-ST-40, shall be reviewed by the supplier and made available to the customer at milestone reviews.

1. Software validation control information [DJF, -; CDR, QR, AR].
2. This information is expected to be provided in electronic format.

### Software delivery and installation

#### .

ECSS-Q-ST-80\_0720341

The Delivery Review Board convened for software deliveries, as specified in clause 5.3.5.3 of ECSS-E-ST-40, shall include a representative from the software product assurance and the software engineering organizations.

ECSS-Q-ST-80\_0720342

The software product assurance reporting to be provided to the Delivery Review Board, including product metrics measurement results, shall be defined based on delivery's purpose and scope.

#### .

ECSS-Q-ST-80\_0720209

The roles, responsibilities and obligations of the supplier and customer during installation shall be established.

1. Installation procedure [DDF, SCF; AR].

#### .

ECSS-Q-ST-80\_0720343

The customer shall verify that the executable code was regenerated from configuration managed source code components and installed in accordance with predefined procedures on the target environment.

#### .

ECSS-Q-ST-80\_0720210

The installation shall be performed in accordance with the installation procedure.

### Software acceptance

#### .

ECSS-Q-ST-80\_0720211

The customer shall establish an acceptance test plan specifying the intended acceptance tests including specific tests suited to the target environment (see ECSS-E-ST-40 clause 5.7.3.1).

1. Acceptance test plan [DJF, -; QR, AR].
   1. 1 The acceptance tests can be partly made up of tests used during previous test activities.
   2. 2 The acceptance test plan takes into account the requirement for operational demonstration, either as part of acceptance or after acceptance.

ECSS-Q-ST-80\_0720344

The customer shall confirm whether the residual security risks are acceptable.

1. Joint review report [DJF, -; AR].
2. The confirmation could be based on acceptance testing and other software analysis techniques: for example, analysis and penetration testing.

#### .

ECSS-Q-ST-80\_0720212

The customer shall ensure that the acceptance tests are performed in accordance with the approved acceptance test plan (see ECSS-E-ST-40 clause 5.7.3.2).

#### .

ECSS-Q-ST-80\_0720213

Before the software is presented for customer acceptance, the supplier shall ensure that:

the delivered software complies with the contractual requirements (including any specified content of the software acceptance data package);

the source and object code supplied correspond to each other;

all agreed changes are implemented;

all nonconformances are either resolved or declared.

#### .

ECSS-Q-ST-80\_0720214

<<deleted>>

#### .

ECSS-Q-ST-80\_0720215

Any discovered problems shall be documented in nonconformance reports.

1. Nonconformance reports [DJF, -; AR].

#### .

ECSS-Q-ST-80\_0720216

On completion of the acceptance tests, a report shall be drawn up and be signed by the supplier’s representatives, the customer’s representatives, the software quality engineers of both parties and the representative of the organization charged with the maintenance of the software product.

1. Acceptance test report [DJF, -; AR].

#### .

ECSS-Q-ST-80\_0720217

The customer shall certify conformance to the procedures and state the conclusion concerning the test result for the software product under test (accepted, conditionally accepted, rejected).

1. Acceptance test report [DJF, -; AR].

### Operations

#### .

ECSS-Q-ST-80\_0720218

During operations, the quality of the mission products related to software shall be agreed with the customer and users.

1. Quality of mission products can include parameters such as: error­free data, availability of data and permissible outages; permissible information degradation.

#### .

ECSS-Q-ST-80\_0720219

During the demonstration that the software conforms to the operational requirements, the following shall be covered as a minimum:

availability and maintainability of the host system (including reboot after maintenance interventions);

safety features;

human-computer interface;

operating procedures;

ability to meet the mission product quality requirements.

1. Validation of the operational requirements [PAF, -; ORR].
2. Operating procedures can include specific security operating procedures (SECOPS).

#### .

ECSS-Q-ST-80\_0720220

The product assurance plan for system operations shall include consideration of software.

1. Input to product assurance plan for systems operation [PAF, -; ORR]

### Maintenance

#### .

ECSS-Q-ST-80\_0720221

The organization responsible for maintenance shall be identified to allow a smooth transition into the operations and maintenance.

1. An organization, with representatives from both supplier and customer, can be set up to support the maintenance activities. Attention is drawn to the importance of the flexibility of this organization to cope with the unexpected occurrence of problems and the identification of facilities and resources to be used for the maintenance activities.

#### .

ECSS-Q-ST-80\_0720222

The maintenance organization shall specify the assurance, verification, validation and security analysis activities applicable to maintenance interventions.

1. Maintenance plan [MF, -; QR, AR, ORR].

#### .

ECSS-Q-ST-80\_0720223

The maintenance plans shall be verified against specified requirements for maintenance of the software product.

1. The maintenance plans and procedures can address corrective, improving, adaptive and preventive maintenance, differentiating between “routine” and “emergency” maintenance activities.

#### .

ECSS-Q-ST-80\_0720224

The maintenance plans and procedures shall include the following as a minimum:

scope of maintenance;

identification of the first version of the software product for which maintenance is to be done;

support organization;

maintenance life cycle;

maintenance activities;

quality measures to be applied during the maintenance;

security measures to be applied during the maintenance.

maintenance records and reports.

1. Maintenance plan [MF, -; QR, AR, ORR].

#### .

ECSS-Q-ST-80\_0720225

Rules for the submission of maintenance reports shall be established and agreed as part of the maintenance plan.

1. Maintenance plan [MF, -; QR, AR, ORR].

#### .

ECSS-Q-ST-80\_0720226

All maintenance activities shall be logged in predefined formats and retained.

1. Maintenance records [MF, -; -].

#### .

ECSS-Q-ST-80\_0720227

Maintenance records shall be established for each software product, including, as a minimum, the following information,:

list of requests for assistance or problem reports that have been received and the current status of each;

security related aspects of requests for assistance or problem reports and the current status of each;

organization responsible for responding to requests for assistance or implementing the appropriate corrective actions;

priorities assigned to the corrective actions;

results of the corrective actions;

statistical data on failure occurrences and maintenance activities.

* 1. 1 The record of the maintenance activities can be utilized for evaluation and enhancement of the software product and for improvement of the quality system itself.
  2. 2 Related to item 2, problem reports include evidence of specific security issues (for example data leakage and presence of malware) and incident description while preserving the privacy of the case.

# Software product quality assurance

## Product quality objectives and metrication

### Deriving of requirements

ECSS-Q-ST-80\_0720228

The software quality requirements (including security, safety and dependability requirements) shall be derived from the requirements defined at system level.

1. The following outputs are expected:

a. Requirement baseline [RB, SSS; SRR];

b. Technical specification [TS, SRS; PDR].

### Quantitative definition of quality requirements

ECSS-Q-ST-80\_0720229

Quality requirements shall be expressed in quantitative terms or constraints.

1. The following outputs are expected:

a. Requirement baseline [RB, SSS; SRR];

b. Technical specification [TS, SRS; PDR].

### Assurance activities for product quality requirements

ECSS-Q-ST-80\_0720230

The supplier shall define assurance activities to ensure that the product meets the quality requirements as specified in the technical specification.

1. Software product assurance plan [PAF, SPAP; SRR, PDR].

### Product metrics

ECSS-Q-ST-80\_0720231

In order to verify the implementation of the product quality requirements, the supplier shall define a metrication programme based on the identified quality model (see clause 5.2.7), specifying:

the metrics to be collected and stored;

the means to collect metrics (measurements);

the target values, with reference to the product quality requirements;

the analyses to be performed on the collected metrics, including the ones to derive:

descriptive statistics;

trend analysis (such as trends in software problems).

how the results of the analyses performed on the collected metrics are fed back to the development team and used to identify corrective actions;

the schedule of metrics collection, storing, analysis and reporting, with reference to the whole software life cycle.

* 1. 1 Guidance for software metrication programme implementation can be found in ECSS-Q-HB-80-04.
  2. 2 Example to item 4(a): the number of units at each level of complexity.

### Basic metrics

ECSS-Q-ST-80\_0720232

The following basic products metrics shall be used:

size (code);

complexity (design, code);

fault density and failure intensity;

test coverage;

number of failures.

1. Software product assurance plan [PAF, SPAP; SRR, PDR].

### Reporting of metrics

ECSS-Q-ST-80\_0720233

The results of metrics collection and analysis shall be included in the software product assurance reports, in order to provide the customer with an insight into the level of quality obtained.

1. Software product assurance reports [PAF, -; -].

### Numerical accuracy

ECSS-Q-ST-80\_0720234

Numerical accuracy shall be estimated and verified.

1. Numerical accuracy analysis [DJF, SVR; PDR, CDR, QR].

### Analysis of software maturity

ECSS-Q-ST-80\_0720235

The supplier shall define the organization and means implemented to collect and analyse data required for the study of software maturity.

1. For example: failures, corrections, duration of runs.

## Product quality requirements

### Requirements baseline and technical specification

#### .

ECSS-Q-ST-80\_0720236

The software quality requirements shall be documented in the requirements baseline and technical specification.

1. The following outputs are expected:

a. Requirement baseline [RB, SSS; SRR];

b. Technical specification [TS, SRS; PDR].

#### .

ECSS-Q-ST-80\_0720237

The software requirements shall be:

correct;

unambiguous;

complete;

consistent;

verifiable;

traceable.

#### .

ECSS-Q-ST-80\_0720238

For each requirement the method for verification and validation shall be specified.

1. The following outputs are expected:

a. Requirement baseline [RB, SSS; SRR];

b. Technical specification [TS, SRS; PDR].

### Design and related documentation

#### .

ECSS-Q-ST-80\_0720239

The software design shall meet the non-functional requirements as documented in the technical specification.

#### .

ECSS-Q-ST-80\_0720240

The software shall be designed to facilitate testing.

#### .

ECSS-Q-ST-80\_0720241

Software with a long planned lifetime shall be designed with minimum dependency on the operating system and the hardware, in order to aid portability.

1. The following outputs are expected:

a. Software product assurance plan [PAF, SPAP; SRR, PDR];

b. Justification of design choices [DDF, SDD; PDR, CDR].

1. This requirement is applicable to situations where the software lifetime can lead to the obsolescence and non-availability of the original operating system and/or hardware, thereby jeopardizing the maintainability of the software.

### Test and validation documentation

#### .

ECSS-Q-ST-80\_0720242

Detailed test and validation documentation (data, procedures and expected results) defined in the ECSS-E-ST-40 DJF shall be consistent with the defined test and validation strategy (see clause 6.3.5 and ECSS-E-ST-40 clauses 5.5.3, 5.5.4, 5.6 and 5.8).

#### .

ECSS-Q-ST-80\_0720243

The test documentation shall cover the test environment, tools and test software, personnel required and associated training requirements.

#### .

ECSS-Q-ST-80\_0720244

The criteria for completion of each test and any contingency steps shall be specified.

#### .

ECSS-Q-ST-80\_0720245

Test procedures, data and expected results shall be specified.

#### .

ECSS-Q-ST-80\_0720246

The hardware and software configuration shall be identified and documented as part of the test documentation.

#### .

ECSS-Q-ST-80\_0720247

For any requirements not covered by testing a verification report shall be drawn up documenting or referring to the verification activities performed.

1. Software verification report [DJF, SVR; CDR, QR, AR].

## Software intended for reuse

### Customer requirements

ECSS-Q-ST-80\_0720248

For the development of software intended for reuse, ECSS-E-ST-40 clauses 5.2.4.7 and 5.4.3.6 shall apply.

### Separate documentation

ECSS-Q-ST-80\_0720249

The information related to the components developed for reuse shall be separated from the others in the technical specification, design justification file, design definition file and product assurance file.

### Self-contained information

ECSS-Q-ST-80\_0720250

The information related to components developed for reuse in the technical specification, the design justification file, the design definition file and the product assurance file shall be self-contained.

### Requirements for intended reuse

ECSS-Q-ST-80\_0720251

The technical specification of components developed for reuse shall include requirements for maintainability, portability and verification of those components.

1. Technical specification for reusable components [TS, -; PDR].

### Configuration management for intended reuse

ECSS-Q-ST-80\_0720252

The configuration management system shall include provisions for handling specific aspects of software developed for reuse, such as:

longer lifetime of the components developed for reuse compared to the other components of the project;

evolution or change of the development environment for the next project that intends to use the components;

transfer of the configuration and documentation management information to the next project reusing the software.

1. Software configuration management plan [MGT, SCMP; SRR, PDR].

### Testing on different platforms

ECSS-Q-ST-80\_0720253

Where the components developed for reuse are developed to be reusable on different platforms, the testing of the software shall be performed on all those platforms.

1. Verification and validation documentation for reusable components [DJF, -; CDR].

### Certificate of conformance

ECSS-Q-ST-80\_0720254

The supplier shall provide a certificate of conformance that the tests have been successfully completed on all the relevant platforms.

1. Verification and validation documentation for reusable components [DJF, -; CDR].
2. In case not all platforms are available, the certificate of conformance states the limitations of the validation performed.

## Standard ground hardware and services for operational system

### Hardware procurement

ECSS-Q-ST-80\_0720255

The subcontracting and procurement of hardware shall be carried out according to the requirements of ECSS-Q-ST-20 clause 5.4.

1. The following outputs are expected:

a. Justification of selection of operational ground equipment [DJF, -; SRR, PDR];

b. Receiving inspection reports [PAF, -; SRR, PDR].

### Service procurement

ECSS-Q-ST-80\_0720256

The procurement of support services to be used in operational phases shall be justified as covering service level agreements, quality of services and escalation procedures, as needed for system exploitation and maintenance.

1. Justification of selection of operational support services [DJF, -; SRR, PDR].

### Constraints

ECSS-Q-ST-80\_0720257

The choice of procured hardware and services shall address the constraints associated with both the development and the actual use of the software.

1. Justification of selection of operational ground equipment [DJF, -; SRR, PDR].

### Selection

ECSS-Q-ST-80\_0720258

The ground computer equipment and supporting services for implementing the final system shall be selected according to the project requirements regarding:

performance;

maintenance;

durability and technical consistency with the operational equipment;

the assessment of the product with respect to requirements, including the criticality category;

the assessment of the product with respect to security sensitivity of the product;

compliance with applicable organizational, national or international standards, security regulations, policies or directives;

the available support documentation;

the acceptance and warranty conditions;

the conditions of installation, preparation, training and use;

the maintenance conditions, including the possibilities of evolutions;

copyright constraints;

availability;

compatibility;

site operational constraints.

1. Justification of selection of operational ground equipment [DJF, -; SRR, PDR].
2. to item 5: The security sensitivity of the product is determined by the security analysis.

### Maintenance

ECSS-Q-ST-80\_0720259

Taking account of the provider’s maintenance and product policy, it shall be ensured that the hardware and support services can be maintained throughout the specified life of the software product within the operational constraints.

## Programmable devices

### Device programming

ECSS-Q-ST-80\_0720260

The supplier shall establish procedures for device programming and duplication of programmed services.

1. Software product assurance plan [PAF, SPAP; PDR].

### Marking

ECSS-Q-ST-80\_0720261

The programmed device shall be indelibly marked to allow the identification (by reference) of the hardware component and of the software component.

ECSS-Q-ST-80\_0720345

Where protective security marking is required, devices and all software configuration items shall be marked accordingly.

1. Software product assurance plan [PAF, SPAP; PDR].

### Calibration

ECSS-Q-ST-80\_0720262

The supplier shall ensure that the programming equipment is calibrated.

1. (informative)   
   Software documentation

This annex defines the structure of the software documents to be produced, as depicted in Figure A-1.



: Overview of software documents

Table A-1 represents the document requirements list, identifying the software documentation to be produced in accordance with the requirements defined in this Standard and in ECSS-E-ST-40 as output of the relevant processes.

: ECSS-E-ST-40 and ECSS-Q-ST-80 Document requirements list (DRL)

| Related file | DRL item (e.g. Plan, document, file, report, form, matrix) | DRL item having a DRD | SRR | PDR | CDR | QR | AR | ORR |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| RB | Software system specification (SSS) | ECSS-E-ST-40 Annex B | ****** |  |  |  |  |  |
| Interface requirements document (IRD) | ECSS-E-ST-40 Annex C | ****** |  |  |  |  |  |
| Safety and dependability analysis results for lower-level suppliers | - | ****** |  |  |  |  |  |
| TS | Software requirements specification (SRS) | ECSS-E-ST-40 Annex D |  | ****** |  |  |  |  |
| Software interface control document (ICD) | ECSS-E-ST-40 Annex E |  | ****** | ****** |  |  |  |
| DDF | Software design document (SDD) | ECSS-E-ST-40 Annex F |  | ****** | ****** |  |  |  |
| Software configuration file (SCF) | ECSS-M-ST-40 Annex E |  | ****** | ****** | ****** | ****** | ****** |
| Software release document (SRelD) | ECSS-E-ST-40 Annex G |  |  |  | ****** | ****** |  |
| Software user manual (SUM) | ECSS-E-ST-40 Annex H |  |  | ****** | ****** | ****** |  |
| Software source code and media labels | - |  |  | ****** |  |  |  |
| Software product and media labels | - |  |  |  | ****** | ****** | ****** |
| Training material | - |  |  |  | ****** |  |  |
| DJF | Software verification plan (SVerP) | ECSS-E-ST-40 Annex I | ****** | ****** |  |  |  |  |
| Software validation plan (SValP) | ECSS-E-ST-40 Annex J |  | ****** |  |  |  |  |
| Independent software verification & validation plan | - | ****** | ****** |  |  |  |  |
| Software integration test plan (SUITP) | ECSS-E-ST-40C Annex K |  | ****** | ****** |  |  |  |
| Software unit test plan (SUITP) | ECSS-E-ST-40 Annex K |  |  | ****** |  |  |  |
| Software validation specification (SVS) with respect to TS | ECSS-E-ST-40 Annex L |  |  | ****** |  |  |  |
| Software validation specification (SVS) with respect to RB | ECSS-E-ST-40 Annex L |  |  |  | ****** | ****** |  |
| Acceptance test plan | - |  |  |  | ****** | ****** |  |
| Software unit test report | - |  |  | ****** |  |  |  |
| Software integration test report | - |  |  | ****** |  |  |  |
| Software validation report with respect to TS | - |  |  | ****** |  |  |  |
| Software validation report with respect to RB | - |  |  |  | ****** | ****** |  |
| Acceptance test report | - |  |  |  |  | ****** |  |
| Installation report | - |  |  |  |  | ****** |  |
| Software verification report (SVR) | ECSS-E-ST-40 Annex M | ****** | ****** | ****** | ****** | ****** | ****** |
| Independent software verification & validation report | - |  | ****** | ****** | ****** | ****** | ****** |
| Software reuse file (SRF) | ECSS-E-ST-40 Annex N | ****** | ****** | ****** |  |  |  |
| Software problem reports and nonconformance reports | - | ****** | ****** | ****** | ****** | ****** | ****** |
| Joint review report | - | ****** | ****** | ****** | ****** | ****** |  |
| Justification of selection of operational ground equipment and services | - | ****** | ****** |  |  |  |  |
| MGT | Software development plan (SDP) | ECSS-E-ST-40 Annex O | ****** | ****** |  |  |  |  |
| Software review plan (SRevP) | ECSS-E-ST-40 Annex P | ****** | ****** |  |  |  |  |
| Software configuration management plan | ECSS-M-ST-40 Annex A | ****** | ****** |  |  |  |  |
| Training plan | - | ****** |  |  |  |  |  |
| Interface management procedures | - | ****** |  |  |  |  |  |
| Identification of NRB SW members | - | ****** |  |  |  |  |  |
| Procurement data | - | ****** | ****** |  |  |  |  |
| MF | Software maintenance plan | ECSS-E-ST-40 Annex T |  |  |  | ****** | ****** | ****** |
| Maintenance records | - |  |  |  | ****** | ****** | ****** |
| SPR and NCR - Modification analysis report - Problem analysis report - Modification documentation - Baseline for change - Joint review reports | - |  |  |  |  |  |  |
| Migration plan and notification | - |  |  |  |  |  |  |
| Retirement plan and notification | - |  |  |  |  |  |  |
| OP | Software operation support plan | - |  |  |  |  |  | ****** |
| Operational testing results | - |  |  |  |  |  | ****** |
| SPR and NCR - User’s request record - Post operation review report | - |  |  |  |  |  | ****** |
| SF | Software security management plan (SSMP) |  | ****** | ****** |  |  |  |  |
| Software security analysis report (SSAR) |  | ****** | ****** | ****** | ****** | ****** |  |
| Security analyses results for lower level suppliers |  | ****** |  |  |  |  |  |
| Security risk treatment plan (SRTP) | - | ****** | ****** | ****** | ****** | ****** | ****** |
| PAF | Software product assurance plan (SPAP) | ECSS-Q-ST-80 Annex B | ****** | ****** | ****** | ****** | ****** | ****** |
| Software product assurance requirements for suppliers | - | ****** |  |  |  |  |  |
| Audit plan and schedule | - | ****** |  |  |  |  |  |
| Review and inspection plans or procedures | - |  |  |  |  |  |  |
| Procedures and standards | - |  | ****** |  |  |  |  |
| Modelling and design standards |  | ****** | ****** |  |  |  |  |
| Coding standards and description of tools | - |  | ****** |  |  |  |  |
| Software problem reporting procedures | - |  | ****** |  |  |  |  |
| Software dependability and safety analysis report - Criticality classification of software components | - |  | ****** | ****** | ****** | ****** |  |
| Software product assurance reports | - |  |  |  |  |  |  |
| Software product assurance milestone report (SPAMR) | ECSS-Q-ST-80 Annex C | ****** | ****** | ****** | ****** | ****** | ****** |
| Statement of compliance with test plans and procedures | - |  |  | ****** | ****** | ****** | ****** |
| Records of training and experience | - |  |  |  |  |  |  |
| (Preliminary) alert information | - |  |  |  |  |  |  |
| Result of pre-award audits and assessments, and of procurement sources | - |  |  |  |  |  |  |
| Software process assessment plan | - |  |  |  |  |  |  |
| Software process assessment records | - |  |  |  |  |  |  |
| Review and inspection reports | - |  |  |  |  |  |  |
| Receiving inspection reports | - | ****** | ****** | ****** | ****** |  |  |
| Input to product assurance plan for systems operation | - |  |  |  |  |  | ****** |

1. (normative)   
   Software product assurance plan (SPAP) - DRD
   1. DRD identification 
      1. Requirement identification and source document

The software product assurance plan (SPAP) is called from the normative provisions summarized in Table B-1.

: SPAP traceability to ECSS-E-ST-40 and ECSS-Q-ST-80 clauses

| ECSS Standard | Clause | DRD section |
| --- | --- | --- |
| ECSS-Q-ST-80 | 5.1.2.1 | <5.1>.a, <5.1>.b |
| 5.1.2.2 | <5.1>.a, <5.1>.b |
| 5.1.2.3 | <5.1>.b |
| 5.1.3.1 | <5.3> |
| 5.1.4.1 | <5.1>.b |
| 5.2.1.1 | All |
| 5.2.1.3 | All |
| 5.2.1.4 | <5.10> |
| 5.2.1.5 | <8> |
| 5.2.6.1c | <6.5> |
| 5.2.7.2 | <5.5> |
| 5.4.3.3 | All |
| 5.4.3.4 | All |
| 5.6.1.1 | <5.8> |
| 6.1.1 | <6.1> |
| 6.2.1.4 | <6.2> |
| 6.2.3.2 | <6.3>.c |
| 6.2.3.4 | <6.3>.c |
| 6.2.3.5 | <6.3>.c |
| 6.2.4.8 | <6.5>.d |
| 6.2.4.9 | <6.5>.d |
| 6.2.5.1 | <6.6>.e |
| 6.2.5.2 | <6.6>.e |
| 6.2.7.2 | <6.7>.f |
| 6.2.7.3 | <6.7>.f |
| 6.2.7.4 | <6.7>.f |
| 6.2.7.5 | <6.7>.f |
| 6.2.9.1 | <6.4>.a |
| 6.2.10.3 | <6.4>.a |
| 6.2.10.4 | <6.4>.a |
| 6.3.3.3 | <6.8>.a.2.h.3 |
| 6.3.3.5 | <6.8>.a.2.g.2 |
| 6.3.3.7 | <6.8>.a.2.g.2 |
| 6.3.4.3 | <6.8>.a.3.h.2 |
| 6.3.4.6 | <6.8>.a.3.g.3 |
| 6.3.5.1 | <6.8>.a.4.g.4 |
| 6.3.5.2 | <6.8>.a.4.g.4 |
| 7.1.3 | <7.b.6>.b.4 |
| 7.1.5 | <7>.b.1 |
| 7.1.6 | <7>.b.1 |
| 7.2.2.3 | <7>.a |
| 7.5.1 | <6.9>.c.19.h.3 |
| 7.5.2 | <6.9>.c.19.h.3 |

* + 1. Purpose and objective

The software product assurance plan is a constituent of the product assurance file (PAF).

The purpose of the software product assurance plan is to provide information on the organizational aspects and the technical approach to the execution of the software product assurance programme

* 1. Expected response 
     1. Scope and content

Introduction

ECSS-Q-ST-80\_0720263

The SPAP shall contain a description of the purpose, objective, content and the reason prompting its preparation.

Applicable and reference documents

ECSS-Q-ST-80\_0720264

The SPAP shall list the applicable and reference documents to support the generation of the document.

Terms, definitions and abbreviated terms

ECSS-Q-ST-80\_0720265

The SPAP shall include any additional terms, definition or abbreviated terms used.

System Overview

ECSS-Q-ST-80\_0720266

The SPAP shall include or refer to a description of the system and software products being developed.

Software product assurance programme implementation

Organization

ECSS-Q-ST-80\_0720267

The SPAP shall describe the organization of software product assurance activities, including responsibility, authority and the interrelation of personnel who manage, perform and verify work affecting software quality.

ECSS-Q-ST-80\_0720268

The following topics shall be included:

organizational structure;

interfaces of each organisation, either external or internal, involved in the project;

relationship to the system level product assurance and safety;

independence of the software product assurance function;

delegation of software product assurance tasks to a lower level supplier, if any.

Responsibilities

ECSS-Q-ST-80\_0720269

The SPAP shall describe the responsibilities of the software product assurance function.

Resources

ECSS-Q-ST-80\_0720270

The SPAP shall describe the resources to be used to perform the software product assurance function.

ECSS-Q-ST-80\_0720271

The description in B.2.1<5.3>a. shall include human resources and skills, hardware and software tools.

Reporting

ECSS-Q-ST-80\_0720272

The SPAP shall describe the reporting to be performed by software product assurance.

Quality models

ECSS-Q-ST-80\_0720273

The SPAP shall describe the quality models applicable to the project and how they are used to specify the quality requirements.

Risk management

ECSS-Q-ST-80\_0720274

The SPAP shall describe the contribution of the software product assurance function to the project risk management.

Supplier selection and control

ECSS-Q-ST-80\_0720275

The SPAP shall describe the contribution of the software product assurance function to the next level suppliers selection and control.

Methods and tools

ECSS-Q-ST-80\_0720276

The SPAP shall describe the methods and tools used for all the activities of the development cycle, and their level of maturity.

Process assessment and improvement

ECSS-Q-ST-80\_0720277

The SPAP shall state the scope and objectives of process assessment.

ECSS-Q-ST-80\_0720278

The SPAP shall describe the methods and tools to be used for process assessment and improvement.

Operations and maintenance (optional)

ECSS-Q-ST-80\_0720279

The SPAP shall specify the quality measures related to the operations and maintenance processes (alternatively, a separate SPAP is produced).

Software process assurance

Software development cycle

ECSS-Q-ST-80\_0720280

The SPAP shall refer to the software development cycle description in the software development plan.

ECSS-Q-ST-80\_0720281

If not covered in the software development plan, the life cycle shall be described.

ECSS-Q-ST-80\_0720282

The life cycle shall include a milestone immediately before the starting of the software validation.

Projects plans

ECSS-Q-ST-80\_0720283

The SPAP shall describe all plans to be produced and used in the project.

ECSS-Q-ST-80\_0720284

The relationship between the project plans and a timely planning for their preparation and update shall be described.

Software dependability and safety

ECSS-Q-ST-80\_0720285

The SPAP shall contain a description and justification of the measures to be applied for the handling of critical software, including the dependability and safety analyses to be performed and the standards applicable for critical software.

Software security

ECSS-Q-ST-80\_0720346

The SPAP shall contain a description of and justification of the measures to be applied for handling of security related software, including the security analysis to be performed and the standards applicable to sensitive software.

ECSS-Q-ST-80\_0720347

The SPAP shall contain at least a security assurance section containing at least references to the documents and section titles that capture the related security and any additional security assurance activities documented in the security file.

Software documentation and configuration management

ECSS-Q-ST-80\_0720286

The SPAP shall describe the contribution of the software product assurance function to the proper implementation of documentation and configuration management.

ECSS-Q-ST-80\_0720287

The nonconformance control system shall be described or referenced. The point in the software life cycle from which the nonconformance procedures apply shall be specified.

ECSS-Q-ST-80\_0720288

The SPAP shall identify method and tool to protect the supplied software, a checksum-type key calculation for the delivered operational software, and a labelling method for the delivered media.

Process metrics

ECSS-Q-ST-80\_0720289

The SPAP shall describe the process metrics derived from the defined quality models, the means to collect, store and analyze them, and the way they are used to manage the development processes.

Reuse of software

ECSS-Q-ST-80\_0720290

The SPAP shall describe the approach for the reuse of existing software, including delta qualification.

Product assurance planning for individual processes and activities

ECSS-Q-ST-80\_0720291

The following processes and activities shall be covered, taking into account the project scope and life cycle:

software requirements analysis;

software architectural design and design of software items;

coding;

testing and validation (including regression testing);

verification;

software delivery and acceptance;

operations and maintenance.

Procedures and standards

ECSS-Q-ST-80\_0720292

The SPAP shall describe or list by reference all procedures and standards applicable to the development of the software in the project.

ECSS-Q-ST-80\_0720293

The software product assurance measures to ensure adherence to the project procedures and standards shall be described.

ECSS-Q-ST-80\_0720294

The standards and procedures to be described or listed in accordance with B.2.1<6.9>a shall be as a minimum those covering the following aspects:

project management;

risk management;

configuration and documentation management;

verification and validation;

requirements engineering;

design;

coding;

metrication;

nonconformance control;

audits;

alerts;

procurement;

reuse of existing software;

use of methods and tools;

numerical accuracy;

delivery, installation and acceptance;

operations;

maintenance;

device programming and marking.

Software product quality assurance

ECSS-Q-ST-80\_0720295

The SPAP shall describe the approach taken to ensure the quality of the software product.

ECSS-Q-ST-80\_0720296

The description of the approach specified in B.2.1<7>a shall include the:

specification of the product metrics, their target values and the means to collect them;

definition of a timely metrication programme;

analyses to be performed on the collected metrics;

way the results are fed back to the development team;

documentation quality requirements;

assurance activities meant to ensure that the product meets the quality requirements.

Compliance matrix to software product assurance requirements

ECSS-Q-ST-80\_0720297

The SPAP shall include the compliance matrix to the applicable software product assurance requirements (e.g. ECSS-Q-ST-80 clauses, as tailored by a product assurance requirements document), or provide a reference to it.

ECSS-Q-ST-80\_0720298

For each software product assurance requirement, the following information shall be provided:

requirement identifier;

compliance   
(C = compliant, NC = non–compliant, NA = not applicable);

reference to the project documentation covering the requirement (e.g. section of the software product assurance plan);

remarks.

* + 1. Special remarks

The response to this DRD may be combined with the response to the project product assurance plan, as defined in ECSS-Q-ST-10.

1. (normative)   
   Software product assurance milestone report (SPAMR) - DRD

A-A-

* 1. DRD identification
     1. Requirement identification and source document

The software product assurance milestone report (SPAMR) is called from the normative provisions summarized in Table C-1.

: SPAMR traceability to ECSS-Q-ST-80 clauses

| ECSS Standard | Clause | DRD section |
| --- | --- | --- |
| ECSS-Q-ST-80 | 5.2.2.3 | All |
| 5.6.1.2 | <5>.a |
| 5.6.1.3 | <5>.b |
| 6.2.5.4 | <7> |
| 6.2.5.5 | <7> |
| 6.2.6.3 | <4> |
| 6.2.6.7 | <4> |
| 6.2.8.5 | <6> |
| 6.3.3.4 | <6> |
| 6.3.3.6 | <6>.a.1 |
| 6.3.3.7 | <6>.a.2 |
| 6.3.4.7 | <6> |
| 6.3.5.3 | <8> |
| 6.3.5.5 | <8> |
| 6.3.5.12 | <8> |
| 7.1.6 | <7> |
| 7.1.8 | <7> |

* + 1. Purpose and objective

The software product assurance milestone report is a constituent of the product assurance file (PAF).

The main purpose of the software product assurance milestone report is to collect and present at project milestones the reporting on the software product assurance activities performed during the past project phases.

* 1. Expected response
     1. Scope and content

Introduction

ECSS-Q-ST-80\_0720299

The SPAMR shall contain a description of the purpose, objective, content and the reason prompting its preparation.

Applicable and reference documents

ECSS-Q-ST-80\_0720300

The SPAMR shall list the applicable and reference documents to support the generation of the document.

Terms, definitions and abbreviated terms

ECSS-Q-ST-80\_0720301

The SPAMR shall include any additional terms, definition or abbreviated terms used.

Verification activities performed

ECSS-Q-ST-80\_0720302

The SPAMR shall contain reporting on verification activities performed by the product assurance function, including:

reviews;

inspections;

walk-throughs;

review of traceability matrices;

documents reviewed.

ECSS-Q-ST-80\_0720303

The SPAMR shall contain reporting on the verification of the measures applied for the handling of critical software.

Methods and tools

ECSS-Q-ST-80\_0720304

The SPAMR shall include or reference a justification of the suitability of the methods and tools applied in all the activities of the development cycle, including requirements analysis, software specification, design, coding, validation, testing, configuration management, verification and product assurance.

ECSS-Q-ST-80\_0720305

The SPAMR shall include reporting on the correct use of methods and tools.

Adherence to design and coding standards

ECSS-Q-ST-80\_0720306

The SPAMR shall include reporting on the adherence of software products to the applicable modelling, design and coding standards, including:

reporting on the application of measures meant to ensure that the design complexity and modularity meet the quality requirements;

reporting on design documentation w.r.t. suitability for maintenance.

Product and process metrics

ECSS-Q-ST-80\_0720307

The SPAMR shall include reporting on the collected product and process metrics, the relevant analyses performed, the corrective actions undertaken and the status of these actions.

ECSS-Q-ST-80\_0720308

The results of the software maturity analysis shall also be reported.

Testing and validation

ECSS-Q-ST-80\_0720309

The SPAMR shall include reporting on adequacy of the testing and validation documentation (including feasibility, traceability repeatability), and on the achieved test coverage w.r.t. stated goals.

SPRs and SW NCRs

ECSS-Q-ST-80\_0720310

The SPAMR shall include reporting on the status of software problem reports and nonconformances relevant to software.

References to progress reports

ECSS-Q-ST-80\_0720311

Whenever relevant and up-to-date information has been already delivered as part of the regular PA progress reporting, a representative summary shall be provided, together with a detailed reference to the progress report(s) containing that information.

* + 1. Special remarks

The response to this DRD may be combined with the response to the project product assurance report, as defined in ECSS-Q-ST-10.

1. (normative)   
   Tailoring of this Standard based on software criticality
   1. Software criticality categories

Criticality categories are assigned to software products as specified in ECSS-Q-ST-30 clause 5.4, and ECSS-Q-ST-40 clause 6.5.6.3.

Table D-1 describes the relationship between the criticality category of the software products, the highest criticality of the functions implemented by the software and the existing system compensating provisions, as described in ECSS-Q-ST-30, clause 5.4, and ECSS-Q-ST-40, clause 6.5.6.3.

To any software product type described in the right column, the corresponding criticality category in the left column is assigned. E.g. both "Software involved in category I functions AND: no compensating provisions exist" and "Software included in compensating provisions for category I functions" are category A software.

For criticality classification of software components, clause 6.2.2 of this Standard applies.

: Software criticality categories

| Software criticality category | Definition |
| --- | --- |
| A | Software involved in category I functions  AND: no compensating provisions exist |
| Software included in compensating provisions for category I functions |
| B | Software involved in category I functions  AND: at least one of the following compensating provisions is available, meeting the requirements defined in ECSS-Q-ST-30 clause 5.4 and ECSS-Q-ST-40 clause 6.5.6.3:  - A hardware implementation  - A software implementation; this software implementation shall be classified as criticality A  - An operational procedure |
| Software involved in category II functions  AND: no compensating provisions exist |
| Software included in compensating provisions for category II functions |
| C | Software involved in category II functions  AND: at least one of the following compensating provisions is available, meeting the requirements defined in ECSS-Q-ST-30 clause 5.4 and ECSS-Q-ST-40 clause 6.5.6.3:  - A hardware implementation  - A software implementation; this software implementation shall be classified as criticality B  - An operational procedure |
| Software involved in category III functions  AND: no compensating provisions exist |
| Software included in compensating provisions for category III functions |
| D | Software involved in category III functions  AND: at least one of the following compensating provisions is available, meeting the requirements defined in ECSS-Q-ST-30 clause 5.4 and ECSS-Q-ST-40 clause 6.5.6.3:  - A hardware implementation  - A software implementation; this software implementation shall be classified as criticality C  - An operational procedure |
| Software involved in category IV functions  AND: no compensating provisions exist |

* 1. Applicability matrix

The following applicability matrix represents a tailoring of the requirements of this Standard based on the software criticality categories defined as per D.1.

For each clause of this Standard and for each software criticality category, an indication is given whether that clause is applicable (Y), not applicable (N), or applicable under the conditions thereby specified to that software criticality category.

Security is a transversal activity that might be applied independently of software criticality.

: Applicability matrix based on software criticality

| Clause | Description | A | B | C | D |
| --- | --- | --- | --- | --- | --- |
| 5 | Software product assurance programme implementation | - | - | - | - |
| 5.1 | Organization and responsibility | - | - | - | - |
| 5.1.1 | Organization | Y | Y | Y | Y |
| 5.1.2 | Responsibility and authority | - | - | - | - |
| 5.1.2.1 |  | Y | Y | Y | Y |
| 5.1.2.2 |  | Y | Y | Y | Y |
| 5.1.2.3 |  | Y | Y | Y | Y |
| 5.1.3 | Resources | - | - | - | - |
| 5.1.3.1 |  | Y | Y | Y | Y |
| 5.1.3.2 |  | Y | Y | Y | N |
| 5.1.4 | Software product assurance manager/engineer | - | - | - | - |
| 5.1.4.1 |  | Y | Y | Y | Y |
| 5.1.4.2 |  | Y | Y | Y | Y |
| 5.1.5 | Training | - | - | - | - |
| 5.1.5.1 |  | Y | Y | Y | Expected output not required |
| 5.1.5.2 |  | Y | Y | Y | N |
| 5.1.5.3 |  | Y | Y | Y | Y |
| 5.1.5.4 |  | Y | Y | Y | Y |
| 5.2 | Software product assurance programme management | - | - | - | - |
| 5.2.1 | Software product assurance planning and control | - | - | - | - |
| 5.2.1.1 |  | Y | Y | Y | Y |
| 5.2.1.2 |  | Y | Y | Y | Y |
| 5.2.1.3 |  | Y | Y | Y | Y |
| 5.2.1.4 |  | Y | Y | Y | Y |
| 5.2.1.5 |  | Y | Y | Y | Y |
| 5.2.2 | Software product assurance reporting | - | - | - | - |
| 5.2.2.1 |  | Y | Y | Y | Y |
| 5.2.2.2 |  | Y | Y | Y | Y |
| 5.2.2.3 |  | Y | Y | Y | Y |
| 5.2.3 | Audits | Y | Y | Y | Audits planned and performed only when necessary |
| 5.2.4 | Alerts | Y | Y | Y | Y |
| 5.2.5 | Software problems | - | - | - | - |
| 5.2.5.1 |  | Y | Y | Y | Y |
| 5.2.5.2 |  | Y | Y | Y | Y |
| 5.2.5.3 |  | Y | Y | Y | Y |
| 5.2.5.4 |  | Y | Y | Y | Y |
| 5.2.6 | Nonconformances | - | - | - | - |
| 5.2.6.1 |  | Y | Y | Y | Y |
| 5.2.6.2 |  | Y | Y | Y | Y |
| 5.2.7 | Quality requirements and quality models | - | - | - | - |
| 5.2.7.1 |  | Y | Y | Y | Y |
| 5.2.7.2 |  | Y | Y | Y | Relevant characteristics only (e.g. suitability for safety is not relevant for cat. D software) |
| 5.3 | Risk management and critical item control | - | - | - | - |
| 5.3.1 | Risk management | Y | Y | Y | Y |
| 5.3.2 | Critical item control | - | - | - | - |
| 5.3.2.1 |  | Y | Y | Y | Y |
| 5.3.2.2 |  | Y | Y | Y | Y |
| 5.4 | Supplier selection and control | - | - | - | - |
| 5.4.1 | Supplier selection | - | - | - | - |
| 5.4.1.1 |  | Y | Y | Y | Expected output not required |
| 5.4.1.2 |  | Y | Y | Y | Y |
| 5.4.2 | Supplier requirements | - | - | - | - |
| 5.4.2.1 |  | Y | Y | Y | Y |
| 5.4.2.2 |  | Y | Y | Y | N |
| 5.4.3 | Supplier monitoring | - | - | - | - |
| 5.4.3.1 |  | Y | Y | Y | Y |
| 5.4.3.2 |  | Y | Y | Y | Y |
| 5.4.3.3 |  | Y | Y | Y | Y |
| 5.4.3.4 |  | Y | Y | Y | N |
| 5.4.4 | Criticality classification | Y | Y | Y | Y |
| 5.4.5 | Security sensitivity | Y | Y | Y | Y |
| 5.5 | Procurement | - | - | - | - |
| 5.5.1 | Procurement documents | Y | Y | Y | Y |
| 5.5.2 | Review of procured software component list | Y | Y | Y | Y |
| 5.5.3 | Procurement details | Y | Y | Y | Y |
| 5.5.4 | Identification | Y | Y | Y | Y |
| 5.5.5 | Inspection | Y | Y | Y | Y |
| 5.5.6 | Exportability | Y | Y | Y | Y |
| 5.6 | Tools and supporting environment | - | - | - | - |
| 5.6.1 | Methods and tools | - | - | - |  |
| 5.6.1.1 |  | Y | Y | Y | The proposed methods and tools shall have been successfully used at least in one project before (possibly a non-space project) |
| 5.6.1.2 |  | Y | Y | Y | Expected output not required |
| 5.6.1.3 |  | Y | Y | Y | Expected output not required |
| 5.6.2 | Development environment selection | - | - | - | - |
| 5.6.2.1 |  | Y | Y | Y | Expected output not required |
| 5.6.2.2 |  | Y | Y | Y | Expected output not required |
| 5.6.2.3 |  | Y | Y | Y | Y |
| 5.7 | Assessment and improvement process | - | - | - | - |
| 5.7.1 | Process assessment | Y | Y | Y | N |
| 5.7.2 | Assessment process | - | - | - | - |
| 5.7.2.1 |  | Y | Y | Y | N |
| 5.7.2.2 |  | Y | Y | Y | N |
| 5.7.2.3 |  | Y | Y | Y | N |
| 5.7.2.4 |  | Y | Y | Y | N |
| 5.7.3 | Process improvement | - | - | - | - |
| 5.7.3.1 |  | Y | Y | Y | N |
| 5.7.3.2 |  | Y | Y | Y | N |
| 5.7.3.3 |  | Y | Y | Y | N |
| 6 | Software process assurance | - | - | - | - |
| 6.1 | Software development life cycle | - | - | - | - |
| 6.1.1 | Life cycle definition | Y | Y | Y | Y |
| 6.1.2 | Quality objectives | Y | Y | Y | Y |
| 6.1.3 | Life cycle definition review | Y | Y | Y | Y |
| 6.1.4 | Life cycle resources | Y | Y | Y | Y |
| 6.1.5 | Software validation process schedule | Y | Y | Y | Y |
| 6.2 | Requirements applicable to all software engineering processes | - | - | - | - |
| 6.2.1 | Documentation of processes | - | - | - | - |
| 6.2.1.1 |  | Y | Y | Y | Y |
| 6.2.1.2 |  | Y | Y | Y | Y |
| 6.2.1.3 |  | Y | Y | Y | Y |
| 6.2.1.4 |  | Y | Y | Y | Y |
| 6.2.1.5 |  | Y | Y | Y | Y |
| 6.2.1.6 |  | Y | Y | Y | Y |
| 6.2.1.7 |  | Y | Y | Y | Y |
| 6.2.1.8 |  | Y | Y | Y | Y |
| 6.2.1.9 |  | Y | Y | Y | N |
| 6.2.2 | Software dependability and safety | - | - | - | - |
| 6.2.2.1 |  | Y | Y | Y | Y |
| 6.2.2.2 |  | Y | Y | Y | N |
| 6.2.2.3 |  | Y | Y | Y | N |
| 6.2.2.4 |  | Y | Y | Y | N |
| 6.2.2.5 |  | Y | Y | Y | N |
| 6.2.2.6 |  | Y | Y | Y | N |
| 6.2.2.7 |  | Y | Y | Y | N |
| 6.2.2.8 |  | Y | Y | Y | Y |
| 6.2.2.9 |  | Y | Y | Y | Y |
| 6.2.2.10 |  | Y | Y | Y | Y |
| 6.2.3 | Handling of critical software | - | - | - | - |
| 6.2.3.2 |  | Y | Y | Y | N |
| 6.2.3.3 |  | Y | Y | Y | N |
| 6.2.3.4 |  | Y | Y | Y | N |
| 6.2.3.5 |  | Y | Y | Y | N |
| 6.2.3.6 |  | Y | Y | Y | N |
| 6.2.3.7 |  | Y | Y | N | N |
| 6.2.3.8 |  | Y | Y | Y | N |
| 6.2.4 | Software configuration management | - | - | - | - |
| 6.2.4.1 |  | Y | Y | Y | Y |
| 6.2.4.2 |  | Y | Y | Y | Y |
| 6.2.4.3 |  | Y | Y | Y | Y |
| 6.2.4.4 |  | Y | Y | Y | Y |
| 6.2.4.5 |  | Y | Y | Y | Y |
| 6.2.4.6 |  | Y | Y | Y | Y |
| 6.2.4.7 |  | Y | Y | Y | Y |
| 6.2.4.8 |  | Y | Y | Y | Y |
| 6.2.4.9 |  | Y | Y | Y | Y |
| 6.2.4.10 |  | Y | Y | Y | Y |
| 6.2.4.11 |  | Y | Y | Y | Y |
| 6.2.4.12 |  | Y | Y | Y | Y |
| 6.2.5 | Process metrics | - | - | - | - |
| 6.2.5.1 |  | Y | Y | Y | Y |
| 6.2.5.2 |  | Y | Y | Y | Y |
| 6.2.5.3 |  | Y | Y | Y | Y |
| 6.2.5.4 |  | Y | Y | Y | Limited to number of problems detected during validation |
| 6.2.5.5 |  | Y | Y | Y | Y |
| 6.2.6 | Verification | - | - | - | - |
| 6.2.6.1 |  | Y | Y | Y | Y |
| 6.2.6.2 |  | Y | Y | Y | Y |
| 6.2.6.3 |  | Y | Y | Y | Y |
| 6.2.6.4 |  | Y | Y | Y | Y |
| 6.2.6.5 |  | Y | Y | Y | N |
| 6.2.6.6 |  | Y | Y | Y | N |
| 6.2.6.7 |  | Y | Y | Y | Y |
| 6.2.6.8 |  | Y | Y | Y | Y |
| 6.2.6.9 |  | Y | Y | Y | Y |
| 6.2.6.10 |  | Y | Y | Y | Y |
| 6.2.6.11 |  | Y | Y | Y | Y |
| 6.2.6.12 |  | Y | Y | Y | Y |
| 6.2.6.13 |  | Y | Y | N | N |
| 6.2.7 | Reuse of existing software | - | - | - | - |
| 6.2.7.1 |  | Y | Y | Y | Y |
| 6.2.7.2 |  | Y | Y | Y | Y |
| 6.2.7.3 |  | Y | Y | Y | Y |
| 6.2.7.4 |  | Y | Y | Y | Bullets 3, 4, 5 and 7 not applicable. Bullet 2 limited to architectural design |
| 6.2.7.5 |  | Y | Y | Y | Y |
| 6.2.7.6 |  | Y | Y | Y | Y |
| 6.2.7.7 |  | Y | Y | Y | Limited to the extent to ensure maintainability of the software |
| 6.2.7.8 |  | Y | Y | Y | Limited to the extent to ensure maintainability of the software |
| 6.2.7.9 |  | Y | Y | Y | Y |
| 6.2.7.10 |  | Y | Y | Y | Y |
| 6.2.7.11 |  | Y | Y | Y | Y |
| 6.2.8 | Automatic code generation | - | - | - | - |
| 6.2.8.1 |  | Y | Y | Y | Y |
| 6.2.8.2 |  | Y | Y | Y | Y |
| 6.2.8.3 |  | Y | Y | Y | Y |
| 6.2.8.4 |  | Y | Y | Y | Y |
| 6.2.8.5 |  | Y | Y | Y | Y |
| 6.2.8.6 |  | Y | Y | Y | Y |
| 6.2.8.7 |  | Y | Y | Y | Y |
| 6.2.9 | Software security | - | - | - | - |
| 6.2.9.1 |  | To be applied based on security assurance and sensitivity levels. | | | |
| 6.2.9.2 |  | To be applied based on security assurance and sensitivity levels. | | | |
| 6.2.9.3 |  | To be applied based on security assurance and sensitivity levels. | | | |
| 6.2.9.4 |  | To be applied based on security assurance and sensitivity levels. | | | |
| 6.2.9.5 |  | To be applied based on security assurance and sensitivity levels. | | | |
| 6.2.9.6 |  | To be applied based on security assurance and sensitivity levels. | | | |
| 6.2.9.7 |  | To be applied based on security assurance and sensitivity levels. | | | |
| 6.2.10 | Handling of security sensitive software | - | - | - | - |
| 6.2.10.1 |  | To be applied based on security assurance and sensitivity levels. | | | |
| 6.2.10.2 |  | To be applied based on security assurance and sensitivity levels. | | | |
| 6.2.10.3 |  | To be applied based on security assurance and sensitivity levels. | | | |
| 6.2.10.4 |  | To be applied based on security assurance and sensitivity levels. | | | |
| 6.3 | Requirements applicable to individual software engineering processes or activities | - | - | - | - |
| 6.3.1 | Software related system requirements process | - | - | - | - |
| 6.3.1.1 |  | Y | Y | Y | Y |
| 6.3.1.2 |  | Y | Y | Y | Y |
| 6.3.1.3 |  | Y | Y | Y | Y |
| 6.3.2 | Software requirements analysis | - | - | - | - |
| 6.3.2.1 |  | Y | Y | Y | Y |
| 6.3.2.2 |  | Y | Y | Y | Y |
| 6.3.2.3 |  | Y | Y | Y | Y |
| 6.3.2.4 |  | Y | Y | Y | Y |
| 6.3.2.5 |  | Y | Y | Y | Y |
| 6.3.3 | Software architectural design and design of software items | - | - | - | - |
| 6.3.3.1 |  | Y | Y | Y | Documentation control only |
| 6.3.3.2 |  | Y | Y | Y | Only recommended |
| 6.3.3.3 |  | Y | Y | Y | N |
| 6.3.3.4 |  | Y | Y | Y | Only if design standards are applied (6.3.3.2) |
| 6.3.3.5 |  | Y | Y | Y | N |
| 6.3.3.6 |  | Y | Y | Y | N |
| 6.3.3.7 |  | Y | Y | Y | Y |
| 6.3.4 | Coding | - | - | - | - |
| 6.3.4.1 |  | Y | Y | Y | Y |
| 6.3.4.2 |  | Y | Y | Y | Y |
| 6.3.4.3 |  | Y | Y | Y | N |
| 6.3.4.4 |  | Y | Y | Y | N – This is Y for security sensitive software |
| 6.3.4.5 |  | Y | Y | Y | Y |
| 6.3.4.6 |  | Y | Y | Y | Y |
| 6.3.4.7 |  | Y | Y | Y | Y |
| 6.3.4.8 |  | Y | Y | Y | The code shall be put under configuration control at the beginning of validation testing |
| 6.3.5 | Testing and validation | - | - | - | - |
| 6.3.5.1 |  | Y | Y | Y | No formal unit testing and integration activity required |
| 6.3.5.2 |  | Y | Y | Y | No formal unit testing and integration activity required  This is Y for security sensitive software |
| 6.3.5.3 |  | Y | Y | Y | Test procedures and data verified by sample |
| 6.3.5.4 |  | Y | Y | Y | Applicable to validation and acceptance tests only |
| 6.3.5.5 |  | Y | Y | Y | Y |
| 6.3.5.6 |  | Y | Y | Y | Y |
| 6.3.5.7 |  | Y | Y | Y | Y |
| 6.3.5.8 |  | Y | Y | Y | Y |
| 6.3.5.9 |  | Y | Y | Y | N |
| 6.3.5.10 |  | Y | Y | Y | N |
| 6.3.5.11 |  | Y | Y | Y | Y |
| 6.3.5.12 |  | Y | Y | Y | Y |
| 6.3.5.13 |  | Y | Y | Y | Y |
| 6.3.5.14 |  | Y | Y | Y | Applicable to validation and acceptance tests only |
| 6.3.5.15 |  | Y | Y | Y | Y |
| 6.3.5.16 |  | Y | Y | Y | Y |
| 6.3.5.17 |  | Y | Y | Y | Y |
| 6.3.5.18 |  | Y | Y | Y | Y |
| 6.3.5.19 |  | Y | Y | Y | N |
| 6.3.5.20 |  | Y | Y | Y | Y |
| 6.3.5.21 |  | Y | Y | Y | Y |
| 6.3.5.22 |  | Y | Y | Y | Y |
| 6.3.5.23 |  | Y | Y | Y | Y |
| 6.3.5.24 |  | Y | Y | Y | Y |
| 6.3.5.25 |  | Y | Y | Y | Y |
| 6.3.5.26 |  | Y | Y | Y | Y |
| 6.3.5.27 |  | Y | Y | Y | Y |
| 6.3.5.28 |  | Y | Y | N | N |
| 6.3.5.29 |  | Y | Y | Y | Y |
| 6.3.5.30 |  | Y | Y | Y | N |
| 6.3.5.31 |  | Y | Y | Y | N |
| 6.3.5.32 |  | Y | Y | Y | Y |
| 6.3.5.33 |  | Y | Y | Y | Y |
| 6.3.6 | Software delivery and installation | - | - | - | - |
| 6.3.6.1 |  | Y | Y | Y | Y |
| 6.3.6.2 |  | Y | Y | Y | Y |
| 6.3.6.3 |  | Y | Y | Y | Y |
| 6.3.6.4 |  | Y | Y | Y | Y |
| 6.3.7 | Software acceptance | - | - | - | - |
| 6.3.7.1 |  | Y | Y | Y | Y |
| 6.3.7.2 |  | Y | Y | Y | Y |
| 6.3.7.3 |  | Y | Y | Y | Y |
| 6.3.7.5 |  | Y | Y | Y | Y |
| 6.3.7.6 |  | Y | Y | Y | Y |
| 6.3.7.7 |  | Y | Y | Y | Y |
| 6.3.8 | Operations | - | - | - | - |
| 6.3.8.1 |  | Y | Y | Y | Y |
| 6.3.8.2 |  | Y | Y | Bullet on safety features not applicable | Bullet on safety features not applicable |
| 6.3.8.3 |  | Y | Y | Y | Y |
| 6.3.9 | Maintenance | - | - | - | - |
| 6.3.9.1 |  | Y | Y | Y | Y |
| 6.3.9.2 |  | Y | Y | Y | Y |
| 6.3.9.3 |  | Y | Y | Y | Y |
| 6.3.9.4 |  | Y | Y | Y | Y |
| 6.3.9.5 |  | Y | Y | Y | Y |
| 6.3.9.6 |  | Y | Y | Y | Y |
| 6.3.9.7 |  | Y | Y | Y | Statistical data not collected |
| 7 | Software product quality assurance | - | - | - | - |
| 7.1 | Product quality objectives and metrication | - | - | - | - |
| 7.1.1 | Deriving of requirements | Y | Y | Y | Y |
| 7.1.2 | Quantitative definition of quality requirements | Y | Y | Y | Y |
| 7.1.3 | Assurance activities for product quality requirements | Y | Y | Y | Y |
| 7.1.4 | Product metrics | Y | Y | Y | Bullet 4.(a) not applicable |
| 7.1.5 | Basic metrics | Y | Y | Y | Design-relevant and fault density/failure intensity metrics not required |
| 7.1.6 | Reporting of metrics | Y | Y | Y | Y |
| 7.1.7 | Numerical accuracy | Y | Y | Y | Y |
| 7.1.8 | Analysis of software maturity | Y | Y | Y | N |
| 7.2 | Product quality requirements | - | - | - | - |
| 7.2.1 | Requirements baseline and technical specification | - | - | - | - |
| 7.2.1.1 |  | Y | Y | Y | Y |
| 7.2.1.2 |  | Y | Y | Y | Y |
| 7.2.1.3 |  | Y | Y | Y | Y |
| 7.2.2 | Design and related documentation | - | - | - | - |
| 7.2.2.1 |  | Y | Y | Y | Y |
| 7.2.2.2 |  | Y | Y | Y | Y |
| 7.2.2.3 |  | Y | Y | Y | Y |
| 7.2.3 | Test and validation documentation | - | - | - | - |
| 7.2.3.1 |  | Y | Y | Y | Y |
| 7.2.3.2 |  | Y | Y | Y | Y |
| 7.2.3.3 |  | Y | Y | Y | Y |
| 7.2.3.4 |  | Y | Y | Y | Y |
| 7.2.3.5 |  | Y | Y | Y | Y |
| 7.2.3.6 |  | Y | Y | Y | Y |
| 7.3 | Software intended for reuse | - | - | - | - |
| 7.3.1 | Customer requirements | Y | Y | Y | Y |
| 7.3.2 | Separate documentation | Y | Y | Y | Y |
| 7.3.3 | Self-contained information | Y | Y | Y | Y |
| 7.3.4 | Requirements for intended reuse | Y | Y | Y | Y |
| 7.3.5 | Configuration management for intended reuse | Y | Y | Y | Y |
| 7.3.6 | Testing on different platforms | Y | Y | Y | Y |
| 7.3.7 | Certificate of conformance | Y | Y | Y | Y |
| 7.4 | Standard hardware for operational system | - | - | - | - |
| 7.4.1 | Hardware procurement | Y | Y | Y | Y |
| 7.4.2 | Service procurement | Y | Y | Y | Y |
| 7.4.3 | Constraints | Y | Y | Y | Y |
| 7.4.4 | Selection | Y | Y | Y | Y |
| 7.4.5 | Maintenance | Y | Y | Y | Y |
| 7.5 | Programmable devices | - | - | - | - |
| 7.5.1 | Device programming | Y | Y | Y | Y |
| 7.5.2 | Marking | Y | Y | Y | Y |
| 7.5.3 | Calibration | Y | Y | Y | Y |

1. (informative)   
   List of requirements with built-in tailoring capability

The following requirements are applicable under specific conditions, as described in the requirement’s text.

|  |  |
| --- | --- |
| 5.1.4.2 | The software product assurance *manager/engineer* shall report to the project manager (through the project product assurance manager, *if any*) |
| 5.2.2.1 | The supplier shall report on a regular basis on the status of the software product assurance programme implementation, *if appropriate* as part of the overall product assurance reporting of the project. |
| 6.2.3.4 | *In case* of minor changes in tools that affect the generation of the executable code, a binary comparison of the executable code generated by the different tools can be used to verify that no modifications are introduced |
| 6.2.6.13 | This requirement is applicable *where* the risks associated with the project justify the costs involved. The customer *may* consider a less rigorous level of independence, e.g. an independent team in the same organization. |

The following requirements foresee an agreement between the customer and the supplier.

|  |  |
| --- | --- |
| 6.3.2.5 | Prior to the technical specification elaboration, customer and supplier shall agree on the following principles and rules as a minimum: […]. |
| 6.3.5.2 | Based on the criticality of the software, test coverage goals for each testing level shall be agreed between the customer and the supplier and their achievement monitored by metrics: […]. |

1. (informative)  
   Document organization and content at each milestone
   1. Introduction

The following table shows the organization of the Expected Output of the clauses of this Standard, sorted per review, then per destination file, then per DRD.

When no DRD is available, “-” is shown.

* 1. ECSS-Q-ST-80 Expected Output at SRR

| Clause | Expected Output | Dest. File | DRD | Section |
| --- | --- | --- | --- | --- |
| 7.1.1.a | Requirement baseline | RB | SSS | <5.9> |
| 7.1.2.a | Requirement baseline | RB | SSS | <5.9> |
| 7.2.1.1.a | Requirement baseline | RB | SSS | <5.9> |
| 7.2.1.3.a | Requirement baseline | RB | SSS | <5.1> |
| 5.4.4 | Safety and dependability analyses results for lower level suppliers | RB | - |  |
| 5.4.5 | Security analysis for lower level suppliers | SF | - |  |
| 6.2.4.8.c | Software security management plan | SF | - |  |
| 6.2.4.9.b | Software security management plan | SF | - |  |
| 6.2.4.11.c | Software security management plan | SF | - |  |
| 5.1.2.1 | Software product assurance plan | PAF | SPAP | <5.1> |
| 5.1.2.2 | Software product assurance plan | PAF | SPAP | <5.1>, <5.2> |
| 5.1.2.3 | Software product assurance plan | PAF | SPAP | <5.1> |
| 5.1.3.1 | Software product assurance plan | PAF | SPAP | <5.3> |
| 5.1.3.2 | Software product assurance plan | PAF | SPAP | <5.1>, <5.3> |
| 5.1.4.1 | Software product assurance plan | PAF | SPAP | <5.1>, <5.3> |
| 5.2.1.1 | Software product assurance plan | PAF | SPAP | All |
| 5.2.1.5 | Software product assurance plan | PAF | SPAP | <8> |
| 5.2.6.1.a | NCR SW procedure as part of the Software product assurance plan | PAF | SPAP |  |
| 5.2.6.2. | Software product assurance plan | PAF | SPAP | <6.5> |
| 5.6.1.1 | Software product assurance plan | PAF | SPAP | <5.8> |
| 6.1.1 | Software product assurance plan | PAF | SPAP | <6.1> |
| 6.1.5 | Software product assurance plan | PAF | SPAP | <6.1> |
| 6.2.1.4 | Software product assurance plan | PAF | SPAP | <6.2> |
| 6.2.4.8.a | Software product assurance plan | PAF | SPAP | <6.5> |
| 6.2.4.9.a | Software product assurance plan | PAF | SPAP | <6.5> |
| 6.2.4.11.a | Software product assurance plan | PAF | SPAP | <6.55> |
| 6.2.5.1 | Software product assurance plan | PAF | SPAP | <6.6> |
| 6.2.5.2 | Software product assurance plan | PAF | SPAP | <6.6> |
| 6.2.9.1 | Software product assurance plan | PAF | SPAP | <6.4> |
| 6.2.7.2.a | Software reuse approach, including approach to delta qualification | PAF | SPAP | <6.7> |
| 6.2.7.3.a | Software reuse approach, including approach to delta qualification | PAF | SPAP | <6.7> |
| 6.2.7.4.a | Software reuse approach, including approach to delta qualification | PAF | SPAP | <6.7> |
| 6.2.7.5.a | Software reuse approach, including approach to delta qualification | PAF | SPAP | <6.7> |
| 7.1.3 | Software product assurance plan | PAF | SPAP | <7> |
| 7.1.4 | Software product assurance plan | PAF | SPAP | <7> |
| 7.1.5 | Software product assurance plan | PAF | SPAP | <7> |
| 7.2.2.3.a | Software product assurance plan | PAF | SPAP | <6.8> |
| 5.2.2.3 | Software product assurance milestone report | PAF | SPAMR | All |
| 5.6.1.2 | Software product assurance milestone report | PAF | SPAMR | <5> |
| 6.2.6.12 | Software product assurance milestone report | PAF | SPAMR | <4> |
| 5.2.3 | Audit plan and schedule | PAF | - |  |
| 5.4.2.1 | Software product assurance requirements for suppliers | PAF | - |  |
| 5.4.2.2 | Software product assurance requirements for suppliers | PAF | - |  |
| 6.2.2.1 | Criticality classification of software products | PAF | - |  |
| 6.2.8.4 | Modelling standards | PAF | - |  |
| 6.3.3.2 | Design standards | PAF | - |  |
| 7.4.1.b | Receiving inspection report | PAF | - |  |
| 5.5.2 | Software development plan | MGT | SDP | <4.8> |
| 5.6.2.1 | Software development plan | MGT | SDP | <5.4> |
| 5.6.2.2 | Software development plan | MGT | SDP | <5.4> |
| 6.2.4.2 | Software configuration management plan | MGT | SCMP |  |
| 6.2.4.12 | Software configuration management plan | MGT | SCMP |  |
| 7.3.5 | Configuration management for reusable components | MGT | SCMP |  |
| 5.1.5.1 | Training plan | MGT | - |  |
| 5.2.6.1.b | Identification of SW experts in NRB | MGT | - |  |
| 5.5.3 | Procurement data | MGT | - |  |
| 6.2.7.2.b | Software reuse file | DJF | SRF | <6> |
| 6.2.7.3.b | Software reuse file | DJF | SRF | <4>, <5> |
| 6.2.7.4.b | Software reuse file | DJF | SRF | <5> |
| 6.2.7.5.b | Software reuse file | DJF | SRF | <6> |
| 6.2.7.6 | Software reuse file | DJF | SRF | <4>, <5> |
| 6.2.7.7 | Software reuse file | DJF | SRF | <8> |
| 6.2.7.8 | Software reuse file | DJF | SRF | <8> |
| 6.2.7.11 | Software reuse file | DJF | SRF | <9> |
| 6.2.6.4 | Software problem reports | DJF | - |  |
| 6.2.6.13.a | ISVV plan | DJF | - |  |
| 6.3.5.8 | Software problem reports | DJF | - |  |
| 6.3.5.28.a | ISVV plan | DJF | - |  |
| 7.4.1.a | Justification of selection of operational ground equipment | DJF | - |  |
| 7.4.2 | Justification of selection of operational support services | DJF | - |  |
| 7.4.3 | Justification of selection of operational ground equipment | DJF | - |  |
| 7.4.4 | Justification of selection of operational ground equipment | DJF | - |  |

* 1. ECSS-Q-ST-80 Expected Output at PDR

| Clause | Expected output | Dest. File | DRD | Section |
| --- | --- | --- | --- | --- |
| 6.3.2.4 | Software requirements specification | TS | SRS | <5> |
| 7.1.1.b | Technical specification | TS | SRS | <5.10> |
| 7.1.2.b | Technical specification | TS | SRS | <5.10> |
| 7.2.1.1.b | Technical specification | TS | SRS | <5.10> |
| 7.2.1.3.b | Technical specification | TS | SRS | <6> |
| 7.3.4 | Technical specification for reusable components | TS | - |  |
| 6.2.4.8 | Software security management plan | SF | - |  |
| 6.2.6.13.d | Software security management plan | SF | - |  |
| 6.2.10.2 | Software security management plan | SF | - |  |
| 6.2.10.3.b | Software security management plan | SF | - |  |
| 6.2.10.4.b | Software security management plan | SF | - |  |
| 6.3.5.28.c | Software security management plan | SF | - |  |
| 6.2.9.2 | Software security analysis report | SF | - |  |
| 6.2.9.7 | Software security analysis report | SF | - | - |
| 6.2.10.1.b | Software security analysis report | SF | - |  |
| 5.2.1.1 | Software product assurance plan | PAF | SPAP | All |
| 5.2.1.5 | Software product assurance plan | PAF | SPAP | <8> |
| 5.2.6.2. | Software product assurance plan | PAF | SPAP | <6.4> |
| 5.2.7.1 | Software product assurance plan | PAF | SPAP | <5.5> |
| 5.2.7.2 | Software product assurance plan | PAF | SPAP | <5.5> |
| 5.4.3.3 | Next level suppliers’ software product assurance plan | PAF | SPAP | All |
| 5.4.3.4 | Next level suppliers’ software product assurance plan | PAF | SPAP | All |
| 5.6.1.1 | Software product assurance plan | PAF | SPAP | <5.8> |
| 6.1.1 | Software product assurance plan | PAF | SPAP | <6.1> |
| 6.1.5 | Software product assurance plan | PAF | SPAP | <6.1> |
| 6.2.1.4 | Software product assurance plan | PAF | SPAP | <6.2> |
| 6.2.2.5.a | Software product assurance plan | PAF | SPAP | <6.3> |
| 6.2.2.10.a | Software product assurance plan | PAF | SPAP | <6.3> |
| 6.2.3.2 | Software product assurance plan | PAF | SPAP | <6.3> |
| 6.2.3.4 | Software product assurance plan | PAF | SPAP | <6.8> |
| 6.2.3.5 | Software product assurance plan | PAF | SPAP | <6.8> |
| 6.2.4.8.a | Software product assurance plan | PAF | SPAP | <6.5> |
| 6.2.4.9 | Software product assurance plan | PAF | SPAP | <6.5> |
| 6.2.4.11.a | Software product assurance plan | PAF | SPAP | <6.5> |
| 6.2.5.1 | Software product assurance plan | PAF | SPAP | <6.6> |
| 6.2.5.2 | Software product assurance plan | PAF | SPAP | <6.6> |
| 6.2.7.2.a | Software reuse approach, including approach to delta qualification | PAF | SPAP | <6.7> |
| 6.2.7.3.a | Software reuse approach, including approach to delta qualification | PAF | SPAP | <6.7> |
| 6.2.7.4.a | Software reuse approach, including approach to delta qualification | PAF | SPAP | <6.7> |
| 6.2.7.5.a | Software reuse approach, including approach to delta qualification | PAF | SPAP | <6.7> |
| 6.2.9.1 | Software product assurance plan | PAF | SPAP | <6.4> |
| 6.2.10.3.a | Software product assurance plan | PAF | SPAP | <6.4> |
| 6.2.10.4.a | Software product assurance plan | PAF | SPAP | <6.4> |
| 6.3.3.3 | Software product assurance plan | PAF | SPAP | <6.9> |
| 6.3.3.5 | Software product assurance plan | PAF | SPAP | <6.8> |
| 6.3.3.7.a | Software product assurance plan | PAF | SPAP | <6.8> |
| 6.3.4.3 | Software product assurance plan | PAF | SPAP | <6.7> |
| 6.3.4.6 | Software product assurance plan | PAF | SPAP | <6.7> |
| 6.3.5.1 | Software product assurance plan | PAF | SPAP | <6.6> |
| 6.3.5.2 | Software product assurance plan | PAF | SPAP | <6.6> |
| 7.1.3 | Software product assurance plan | PAF | SPAP | <7> |
| 7.1.4 | Software product assurance plan | PAF | SPAP | <7> |
| 7.1.5 | Software product assurance plan | PAF | SPAP | <7> |
| 7.2.2.3.a | Software product assurance plan | PAF | SPAP | <6.8> |
| 7.5.1 | Software product assurance plan | PAF | SPAP | <6.9> |
| 7.5.2 | Software product assurance plan | PAF | SPAP | <6.8> |
| 5.2.2.3 | Software product assurance milestone report | PAF | SPAMR | All |
| 5.6.1.2 | Software product assurance milestone report | PAF | SPAMR | <5> |
| 6.2.3.3 | Software product assurance milestone report | PAF | SPAMR | <4> |
| 6.2.6.12 | Software product assurance milestone report | PAF | SPAMR | <4> |
| 5.2.5.1 | Software problem reporting procedures | PAF | - |  |
| 5.2.5.2 | Software problem reporting procedures | PAF | - |  |
| 5.2.5.3 | Software problem reporting procedures | PAF | - |  |
| 5.5.5 | Receiving inspection report | PAF | - |  |
| 6.2.1.6 | Procedures and standards | PAF | - |  |
| 6.2.1.7 | Procedures and standards | PAF | - |  |
| 6.2.2.1 | Criticality classification of software products | PAF | - |  |
| 6.2.2.2 | Software dependability and safety analysis report | PAF | - |  |
| 6.2.2.3 | Criticality classification of software components | PAF | - |  |
| 6.2.2.7 | Software dependability and safety analysis report | PAF | - |  |
| 6.2.2.10.b | Software dependability and safety analysis report | PAF | - |  |
| 6.2.10.1.a | Software dependability and safety analysis report | PAF | - |  |
| 6.2.8.4 | Modelling standards | PAF | - |  |
| 6.3.3.2 | Design standards | PAF | - |  |
| 6.3.4.1 | Coding standards | PAF | - |  |
| 6.3.4.2 | Coding standards | PAF | - |  |
| 6.3.4.4 | Coding standards and description of tools | PAF | - |  |
| 7.4.1.b | Receiving inspection report | PAF | - |  |
| 5.5.2 | Software development plan | MGT | SDP | <4.8> |
| 5.6.2.1 | Software development plan | MGT | SDP | <5.4> |
| 5.6.2.2 | Software development plan | MGT | SDP | <5.4> |
| 6.3.4.5 | Software development plan | MGT | SDP | <5.4> |
| 6.2.4.2 | Software configuration management plan | MGT | SCMP |  |
| 7.3.5 | Configuration management for reusable components | MGT | SCMP |  |
| 5.5.3 | Procurement data | MGT | - |  |
| 7.1.7 | Numerical accuracy analysis | DJF | SVR | <6> |
| 6.2.6.1 | Software verification plan | DJF | SVerP | <6.3> |
| 6.2.8.2 | Validation and testing documentation | DJF | SValP | <4.1> |
| 6.2.8.7 | Validation and testing documentation | DJF | SValP | <4.1> |
| 6.3.5.22 | Test and validation documentation | DJF | SValP | <4> |
| 6.3.5.23 | Test and validation documentation | DJF | SValP | <4.4> |
| 6.3.5.24 | Test and validation documentation | DJF | SValP | <4.6> |
| 6.3.5.25 | Test and validation documentation | DJF | SValP | <5> |
| 6.3.5.29 | Test and validation documentation | DJF | SValP | <6> |
| 6.2.8.2 | Validation and testing documentation | DJF | SUITP | <7.6> |
| 6.2.8.7 | Validation and testing documentation | DJF | SUITP | <7.6> |
| 6.3.5.22 | Test and validation documentation | DJF | SUITP | <5> |
| 6.3.5.23 | Test and validation documentation | DJF | SUITP | <5.3> |
| 6.3.5.24 | Test and validation documentation | DJF | SUITP | <5.5> |
| 6.3.5.25 | Test and validation documentation | DJF | SUITP | <9.2>, <10> |
| 6.2.7.2.b | Software reuse file | DJF | SRF | <6> |
| 6.2.7.3.b | Software reuse file | DJF | SRF | <4>, <5> |
| 6.2.7.4.b | Software reuse file | DJF | SRF | <5> |
| 6.2.7.5.b | Software reuse file | DJF | SRF | <6> |
| 6.2.7.6 | Software reuse file | DJF | SRF | <4>, <5> |
| 6.2.7.7 | Software reuse file | DJF | SRF | <8> |
| 6.2.7.8 | Software reuse file | DJF | SRF | <8> |
| 6.2.7.11 | Software reuse file | DJF | SRF | <9> |
| 6.2.6.4 | Software problem reports | DJF | - |  |
| 6.2.6.13.a | ISVV plan | DJF | - |  |
| 6.2.6.13.b | ISVV report | DJF | - |  |
| 6.3.5.8 | Software problem reports | DJF | - |  |
| 6.3.5.28.a | ISVV plan | DJF | - |  |
| 6.3.5.28.b | ISVV report | DJF | - |  |
| 7.4.1.a | Justification of selection of operational ground equipment | DJF | - |  |
| 7.4.2 | Justification of selection of operational support services | DJF | - |  |
| 7.4.3 | Justification of selection of operational ground equipment | DJF | - |  |
| 7.4.4 | Justification of selection of operational ground equipment | DJF | - |  |
| 7.2.2.3.b | Justification of design choices | DDF | SDD | <4.5> |

* 1. ECSS-Q-ST-80 Expected Output at CDR

| Clause | Expected output | Dest. File | DRD | Section |
| --- | --- | --- | --- | --- |
| 6.2.9.5 | Software security analysis report | SF | - | - |
| 6.2.9.6 | Software security analysis report | SF | - | - |
| 6.2.9.7 | Software security analysis report | SF | - | - |
| 6.2.10.1.b | Software security analysis report | SF | - |  |
| 6.3.5.28.c | Software security analysis report | SF | - |  |
| 6.2.10.2 | Software security management plan | SF | - |  |
| 6.2.10.3.b | Software security management plan | SF | - |  |
| 6.2.10.4.b | Software security management plan | SF | - |  |
| 5.2.1.3 | Software product assurance plan | PAF | SPAP | All |
| 6.2.2.5.a | Software product assurance plan | PAF | SPAP | <6.3> |
| 6.2.3.2 | Software product assurance plan | PAF | SPAP | <6.3> |
| 6.2.3.4 | Software product assurance plan | PAF | SPAP | <6.8> |
| 6.2.3.5 | Software product assurance plan | PAF | SPAP | <6.8> |
| 5.2.2.3 | Software product assurance milestone report | PAF | SPAMR | All |
| 6.2.3.3 | Software product assurance milestone report | PAF | SPAMR | <4> |
| 6.2.6.12 | Software product assurance milestone report | PAF | SPAMR | <4> |
| 5.5.5 | Receiving inspection report | PAF | - |  |
| 6.2.2.5 | Software dependability and safety analysis report | PAF | - |  |
| 6.2.2.6 | Software dependability and safety analysis report | PAF | - |  |
| 6.2.2.7 | Software dependability and safety analysis report | PAF | - |  |
| 6.2.2.10.b | Software dependability and safety analysis report | PAF | - |  |
| 6.2.10.1.a | Software dependability and safety analysis report | PAF | - |  |
| 6.2.2.10.a | Software product assurance plan | PAF | SPAP | <6.3> |
| 6.2.9.1 | Software product assurance plan | PAF | SPAP | <6.4> |
| 6.2.10.3.a | Software product assurance plan | PAF | SPAP | <6.4> |
| 6.2.10.4.a | Software product assurance plan | PAF | SPAP | <6.4> |
| 6.3.5.7 | Statement of compliance with test plans and procedures | PAF | - |  |
| 6.3.5.11 | Statement of compliance with test plans and procedures | PAF | - |  |
| 6.2.8.2 | Validation and testing documentation | DJF | SVS | <5.6> |
| 6.2.8.7 | Validation and testing documentation | DJF | SVS | <5.6> |
| 6.3.5.25 | Test and validation documentation | DJF | SVS | <7.2>, <8> |
| 6.3.5.29 | Test and validation documentation | DJF | SVS | <6> |
| 6.3.5.33 | Software validation control information | DJF | - |  |
| 6.3.5.32 | Software validation specification | DJF | SVS | <5> |
| 6.2.6.5 | Software verification report | DJF | SVR | <4.4> |
| 6.2.6.6 | Software verification report | DJF | SVR | <4.4> |
| 7.1.7 | Numerical accuracy analysis | DJF | SVR | <6> |
| 7.2.3.6 | Software verification report | DJF | SVR | <4.5> |
| 6.2.8.2 | Validation and testing documentation | DJF | SUITP | <7.6> |
| 6.2.8.7 | Validation and testing documentation | DJF | SUITP | <7.6> |
| 6.3.5.22 | Test and validation documentation | DJF | SUITP | <5> |
| 6.3.5.23 | Test and validation documentation | DJF | SUITP | <5.3> |
| 6.3.5.24 | Test and validation documentation | DJF | SUITP | <5.5> |
| 6.3.5.25 | Test and validation documentation | DJF | SUITP | <9.2>, <10> |
| 6.2.7.9 | Software reuse file | DJF | SRF | <8> |
| 6.2.7.11 | Software reuse file | DJF | SRF | <9> |
| 6.2.6.4 | Software problem reports | DJF | - |  |
| 6.2.6.13.b | ISVV report | DJF | - |  |
| 6.3.5.6 | Nonconformance reports and software problem reports | DJF | - |  |
| 6.3.5.8 | Software problem reports | DJF | - |  |
| 6.3.5.13 | Testing and validation reports | DJF | - |  |
| 6.3.5.16 | Updated test documentation | DJF | - |  |
| 6.3.5.17 | Updated test documentation | DJF | - |  |
| 6.3.5.18 | Updated test documentation | DJF | - |  |
| 6.3.5.28.b | ISVV report | DJF | - |  |
| 6.3.5.30 | Testing and validation reports | DJF | - |  |
| 6.3.5.31 | Testing and validation reports | DJF | - |  |
| 7.3.6 | Verification and validation documentation for reusable components | DJF | - |  |
| 7.3.7 | Verification and validation documentation for reusable components | DJF | - |  |
| 6.2.4.4 | Software configuration file | DDF | SCF | All |
| 6.2.4.5 | Software configuration file | DDF | SCF | All |
| 6.2.4.8.b | Software configuration file | DDF | SCF | All |
| 7.2.2.3.b | Justification of design choices | DDF | SDD | <4.5> |

* 1. ECSS-Q-ST-80 Expected Output at QR

| Clause | Expected output | Dest. File | DRD | Section |
| --- | --- | --- | --- | --- |
| 6.2.9.5 | Software security analysis report | SF | - | - |
| 6.2.9.6 | Software security analysis report | SF | - | - |
| 6.2.10.1.b | Software security analysis report | SF | - |  |
| 6.2.10.2 | Software security management plan | SF | - |  |
| 5.2.1.3 | Software product assurance plan | PAF | SPAP | All |
| 5.2.2.3 | Software product assurance milestone report | PAF | SPAMR | All |
| 6.2.3.3 | Software product assurance milestone report | PAF | SPAMR | <4> |
| 6.2.6.12 | Software product assurance milestone report | PAF | SPAMR | <4> |
| 5.5.5 | Receiving inspection report | PAF | - |  |
| 6.2.2.5 | Software dependability and safety analysis report | PAF | - |  |
| 6.2.2.6 | Software dependability and safety analysis report | PAF | - |  |
| 6.2.2.10.b | Software dependability and safety analysis report | PAF | - |  |
| 6.2.10.1.a | Software dependability and safety analysis report | PAF | - |  |
| 6.3.5.7 | Statement of compliance with test plans and procedures | PAF | - |  |
| 6.3.5.11 | Statement of compliance with test plans and procedures | PAF | - |  |
| 6.3.8.1 | Maintenance plan | MF | - |  |
| 6.3.8.2 | Maintenance plan | MF | - |  |
| 6.3.8.4 | Maintenance plan | MF | - |  |
| 6.3.8.5 | Maintenance plan | MF | - |  |
| 6.3.7.1 | Acceptance test plan | DJF | - |  |
| 6.2.8.2 | Validation and testing documentation | DJF | SVS | <5.6> |
| 6.2.8.7 | Validation and testing documentation | DJF | SVS | <5.6> |
| 6.3.5.25 | Test and validation documentation | DJF | SVS | <7.2>, <8> |
| 6.3.5.29 | Test and validation documentation | DJF | SVS | <6> |
| 6.3.5.32 | Software validation specification | DJF | SVS | <5> |
| 6.2.6.5 | Software verification report | DJF | SVR | <4.4> |
| 6.2.6.6 | Software verification report | DJF | SVR | <4.4> |
| 7.1.7 | Numerical accuracy analysis | DJF | SVR | <6> |
| 7.2.3.6 | Software verification report | DJF | SVR | <4.5> |
| 6.2.7.9 | Software reuse file | DJF | SRF | <8> |
| 6.2.7.11 | Software reuse file | DJF | SRF | <9> |
| 6.2.6.4 | Software problem reports | DJF | - |  |
| 6.2.6.13.c | ISVV report | DJF | - |  |
| 6.3.5.6 | Nonconformance reports and software problem reports | DJF | - |  |
| 6.3.5.8 | Software problem reports | DJF | - |  |
| 6.3.5.13 | Testing and validation reports | DJF | - |  |
| 6.3.5.16 | Updated test documentation | DJF | - |  |
| 6.3.5.17 | Updated test documentation | DJF | - |  |
| 6.3.5.18 | Updated test documentation | DJF | - |  |
| 6.3.5.28.b | ISVV report | DJF | - |  |
| 6.3.5.30 | Testing and validation reports | DJF | - |  |
| 6.3.5.31 | Testing and validation reports | DJF | - |  |
| 6.2.4.4 | Software configuration file | DDF | SCF | All |
| 6.2.4.5 | Software configuration file | DDF | SCF | All |
| 6.2.4.8.b | Software configuration file | DDF | SCF | All |

* 1. ECSS-Q-ST-80 Expected Output at AR

| Clause | Expected output | Dest. File | DRD | Section |
| --- | --- | --- | --- | --- |
| 6.2.6.13.d | Software security analysis report | SF | - | - |
| 6.2.9.5 | Software security analysis report | SF | - | - |
| 6.2.9.6 | Software security analysis report | SF | - | - |
| 6.2.10.1.b | Software security analysis report | SF | - | - |
| 6.2.10.2 | Software security management plan | SF | - |  |
| 5.2.1.3 | Software product assurance plan | PAF | SPAP | All |
| 5.2.1.4 | Software product assurance plan | PAF | SPAP | <5.10> |
| 5.2.2.3 | Software product assurance milestone report | PAF | SPAMR | All |
| 6.2.3.3 | Software product assurance milestone report | PAF | SPAMR | <4> |
| 6.2.6.12 | Software product assurance milestone report | PAF | SPAMR | <4> |
| 6.2.2.5 | Software dependability and safety analysis report | PAF | - |  |
| 6.2.2.6 | Software dependability and safety analysis report | PAF | - |  |
| 6.2.2.10 | Software dependability and safety analysis report | PAF | - |  |
| 6.2.10.1.a | Software dependability and safety analysis report | PAF | - |  |
| 6.3.5.7 | Statement of compliance with test plans and procedures | PAF | - |  |
| 6.3.5.11 | Statement of compliance with test plans and procedures | PAF | - |  |
| 6.3.8.1 | Maintenance plan | MF | - |  |
| 6.3.8.2 | Maintenance plan | MF | - |  |
| 6.3.8.4 | Maintenance plan | MF | - |  |
| 6.3.8.5 | Maintenance plan | MF | - |  |
| 6.2.8.2 | Validation and testing documentation | DJF | SVS | <5.6> |
| 6.2.8.7 | Validation and testing documentation | DJF | SVS | <5.6> |
| 6.3.5.25 | Test and validation documentation | DJF | SVS | <7.2>, <8> |
| 6.3.5.29 | Test and validation documentation | DJF | SVS | <6> |
| 6.3.7.1 | Joint review report | DJF | - |  |
| 6.3.5.32 | Software validation specification | DJF | SVS | <5> |
| 6.2.6.5 | Software verification report | DJF | SVR | <4.4> |
| 6.2.6.6 | Software verification report | DJF | SVR | <4.4> |
| 7.2.3.6 | Software verification report | DJF | SVR | <4.5> |
| 6.2.7.9 | Software reuse file | DJF | SRF | <8> |
| 6.2.7.11 | Software reuse file | DJF | SRF | <9> |
| 6.2.6.4 | Software problem reports | DJF | - |  |
| 6.2.6.13.c | ISVV report | DJF | - |  |
| 6.3.5.6 | Nonconformance reports and software problem reports | DJF | - |  |
| 6.3.5.8 | Software problem reports | DJF | - |  |
| 6.3.5.13 | Testing and validation reports | DJF | - |  |
| 6.3.5.16 | Updated test documentation | DJF | - |  |
| 6.3.5.17 | Updated test documentation | DJF | - |  |
| 6.3.5.18 | Updated test documentation | DJF | - |  |
| 6.3.5.27 | Test and validation documentation | DJF | - |  |
| 6.3.5.28.b | ISVV report | DJF | - |  |
| 6.3.5.30 | Testing and validation reports | DJF | - |  |
| 6.3.5.31 | Testing and validation reports | DJF | - |  |
| 6.3.7.1 | Acceptance test plan | DJF | - |  |
| 6.3.7.5 | Nonconformance reports | DJF | - |  |
| 6.3.7.6 | Acceptance test report | DJF | - |  |
| 6.3.7.7 | Acceptance test report | DJF | - |  |
| 6.2.4.4 | Software configuration file | DDF | SCF | All |
| 6.2.4.5 | Software configuration file | DDF | SCF | All |
| 6.2.4.8.b | Software configuration file | DDF | SCF | All |
| 6.3.6.2 | Installation procedure | DDF | SCF | <4.2> |

* 1. ECSS-Q-ST-80 Expected Output not associated with any specific milestone review

| Clause | Expected output | Dest. File | DRD | Section |
| --- | --- | --- | --- | --- |
| 6.2.4.11.d | Protective security markings | SF | - |  |
| 5.1.5.2 | Records of training and experience | PAF | - |  |
| 5.2.2.1 | Software product assurance report | PAF | - |  |
| 5.2.2.2 | Software product assurance report | PAF | - |  |
| 5.2.4.a | Preliminary alert information | PAF | - |  |
| 5.2.4.b | Alert information | PAF | - |  |
| 5.4.1.1.a | Results of pre­award audits and assessments | PAF | - |  |
| 5.4.1.1.b | Records of procurement sources | PAF | - |  |
| 5.6.1.3 | Software product assurance reports | PAF | - |  |
| 5.7.1 | Software process assessment records: Overall assessments and improvement programme plan | PAF | - |  |
| 5.7.2.1.a | Software process assessment record: assessment model | PAF | - |  |
| 5.7.2.1.b | Software process assessment record: assessment method | PAF | - |  |
| 5.7.2.2.a | Software process assessment record: evidence of conformance of the process assessment model | PAF | - |  |
| 5.7.2.2.b | Software process assessment record: assessment method | PAF | - |  |
| 5.7.2.3 | Software process assessment record: Software process assessment recognition evidence | PAF | - |  |
| 5.7.2.4 | Software process assessment record: competent assessor justification | PAF | - |  |
| 5.7.3.1 | Software process assessment records: improvement plan | PAF | - |  |
| 5.7.3.2 | Software process assessment records: improvement process | PAF | - |  |
| 5.7.3.3 | Software process assessment records: evidence of improvements | PAF | - |  |
| 6.2.5.4 | Software product assurance reports | PAF | - |  |
| 6.2.5.5 | Software product assurance reports | PAF | - |  |
| 6.2.6.2 | Software product assurance reports | PAF | - |  |
| 6.2.6.3 | Software product assurance reports | PAF | - |  |
| 6.2.6.7 | Software product assurance reports | PAF | - |  |
| 6.2.6.9 | Review and inspection plans or procedures | PAF | - |  |
| 6.2.6.10 | Review and inspection plans or procedures | PAF | - |  |
| 6.2.6.11 | Review and inspection reports | PAF | - |  |
| 6.2.8.5 | Software product assurance reports | PAF | - |  |
| 6.3.3.4 | Software product assurance reports | PAF | - |  |
| 6.3.3.6 | Software product assurance reports | PAF | - |  |
| 6.3.3.7.b | Software product assurance reports | PAF | - |  |
| 6.3.4.7 | Software product assurance reports | PAF | - |  |
| 6.3.5.3 | Software product assurance reports | PAF | - |  |
| 6.3.5.5 | Software product assurance reports | PAF | - |  |
| 6.3.5.12 | Software product assurance reports | PAF | - |  |
| 7.1.6 | Software product assurance reports | PAF | - |  |
| 7.1.7 | Software product assurance reports | PAF | - |  |
| 6.3.8.6 | Maintenance records | MF | - |  |
| 6.3.8.7 | Maintenance records | MF | - |  |
| 5.2.6.1.a.b | Nonconformance reports | DJF | - |  |
| 5.4.1.2 | Software reuse file | DJF | SRF | All |
| 6.2.4.3.a | Software configuration file | DDF | SCF | All |
| 6.2.4.3.b | Software release document | DDF | SRelD | All |
| 6.2.4.10 | Software configuration file | DDF | SCF | All |
| 6.2.4.11.b | Labels | DDF | - |  |

Bibliography

|  |  |
| --- | --- |
| ECSS-S-ST-00 | ECSS system — Description, implementation and general requirement |
| ECSS-Q-HB-80-02 | Space product assurance — Software process assessment and improvement |
| ECSS-Q-HB-80-03 | Space product assurance — Software dependability and safety methods and techniques |
| ECSS-Q-HB-80-04 | Space product assurance — Software metrication programme definition and implementation |
| ECSS-Q-ST-30-02 | Space product assurance — Failure modes, effects (and criticality) analysis |
| IEEE 610.12:1990 | IEEE Standard Glossary of software engineering terminology |
| IEEE 1028-1997 | IEEE Standard for Software Reviews |
| ISO 9000:2000 | Quality management systems — Fundamentals and vocabulary |
| ISO/IEC 25000:2014 | Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — Guide to SQuaRESQuaRE |
| ISO/IEC 25010:2011 | Systems and software engineering - Systems and software Quality Requirements and Evaluation (SQuaRE) - System and software quality models |
| ISO/IEC/IEEE 24765:2017 | Systems and software engineering — Vocabulary (https://www.iso.org/standard/71952.html) |
| RTCA/DO-178B | Software considerations in airborne systems and equipment certification |
| CMU/SEI-92-TR-022 | Software Quality Measurement: A framework for counting problems and defects |
| CMU/SEI-2006-TR-008 | CMMI for Development, Version 1.2 |