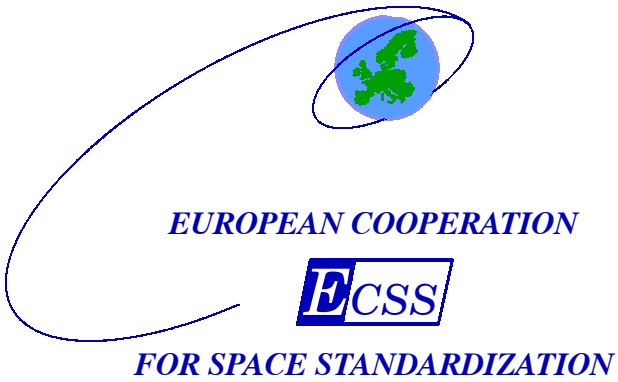
ECSS-Q-80B
10 October 2003



Space product assurance

Software product assurance

ECSS Secretariat
ESA-ESTEC
Requirements & Standards Division
Noordwijk, The Netherlands

Published by: ESA Publications Division
ESTEC, P.O. Box 299,
2200 AG Noordwijk,
The Netherlands

ISSN: 1028-396X

Price: € 20

Printed in The Netherlands

Foreword

This Standard is one of the series of ECSS Standards intended to be applied together for the management, engineering and product assurance in space projects and applications. 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.

This Standard defines the principles and requirements applicable to space software product assurance. ECSS-E-40, Part 2, defines the content of the document requirement definitions (DRDs) that are referenced in this Standard.

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.

The formulation of this Standard takes into account the existing ISO 9000 family of documents, and the ISO/IEC 12207 standard.

Significant changes between this version and the previous version are:

- removal of software engineering requirements which are now covered by ECSS-E-40,
- identification of software PA documentation at the project milestones,
- additional requirements to address software technology evolution.

This Standard has been prepared by the ECSS Software Working Group, reviewed by the Product Assurance Panel, and approved by the ECSS Steering Board.

This version B cancels and replaces ECSS-Q-80A.

(This page is intentionally left blank)

Contents

Figures

1

Scope

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

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

This Standard supplements ECSS-Q-00 “Product assurance” and ECSS-Q-20 “Quality assurance”, and it has interfaces with:

ECSS-M-00	Space project management — Project management
ECSS-M-30	Space project management — Project phasing and planning
ECSS-M-40	Space project management — Configuration management
ECSS-M-50	Space project management — Information/documentation management
ECSS-Q-30	Space product assurance — Dependability
ECSS-Q-40	Space product assurance — Safety
ECSS-E-40	Space engineering — Software

When viewed from the perspective of a specific project context, the requirements defined in this Standard should be tailored to match the requirements of a particular profile and circumstances of a project.

Tailoring of this Standard to a specific contract or project, when software product assurance requirements are prepared, is defined in subclause 4.5.

NOTE Tailoring is a process by which individual requirements or specifications, standards and related documents are evaluated and made applicable to a specific project, by selection and in some exceptional cases, modification of existing or addition of new requirements.
[ECSS-M-00-02A, clause 3]

(This page is intentionally left blank)

2

Normative references

The following normative documents contain provisions which, through reference in this text, constitute provisions of this ECSS Standard. For dated references, subsequent amendments to, or revisions 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 most recent editions of the normative documents indicated below. For undated references the latest edition of the publication referred to applies.

ECSS-P-001	Glossary of terms
ECSS-Q-00	Space product assurance — Policy and principles
ECSS-Q-20B	Space product assurance — Quality assurance
ECSS-Q-30B	Space product assurance — Dependability
ECSS-Q-40B	Space product assurance — Safety
ECSS-M-00-03	Space project management — Risk management
ECSS-M-30A	Space project management — Project phasing and planning
ECSS-M-30-01	Space project management — Organization and conduct of reviews
ECSS-M-40	Space project management — Configuration management
ECSS-E-40B	Space engineering — Software

(This page is intentionally left blank)

3

Terms, definitions and abbreviated terms

3.1 Terms and definitions

The following terms and definitions are specific to this Standard in the sense that they are complementary or additional with respect to those contained in ECSS-P-001.

3.1.1
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

[adapted from ISO/IEC 2382-20:1990]

3.1.2
configurable code
code (source code or executable code) that can be tailored by setting values of parameters

3.1.3
critical software
software supporting a safety or dependability critical function that if incorrect or inadvertently executed can result in catastrophic or critical consequences

NOTE For the definition of catastrophic and critical see ECSS-Q-30 and ECSS-Q-40.

3.1.4
deactivated code
code that, although incorporated through correct design and coding, is not intended to execute in any software product configuration

3.1.5
logical model
implementation-independent model of software items used to analyse and document software requirements

3.1.6
integration testing
testing in which software components, hardware components, or both are combined and tested to evaluate the interaction between them

[IEEE 610.12:1990]

3.1.7
margin philosophy
rationale for margins allocated to the performance parameters and computer resources of a development, and how these margins shall be managed during the execution of the project

3.1.8
metric
defined measurement method and the measurement scale

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

NOTE 2 Metrics include methods for categorising qualitative data.

[ISO/IEC 9126-1:2001]

3.1.9
migration
porting of a software product to a new environment

3.1.10
portability (a quality characteristic)
capability of software to be transferred from one environment to another

3.1.11
quality characteristics (software)
set of attributes of a software product by which its quality is described and evaluated

NOTE A software quality characteristic can have multiple levels of subcharacteristics.

3.1.12
quality model (software)
set of characteristics and the relationships between them which provide the basis for specifying quality requirements and evaluating quality

[ISO/IEC 9126-1:2001]

3.1.13
regression testing (software)
selective retesting of a system or component to verify that modifications have not caused unintended effects and that the system or component still complies with its specified requirements

[IEEE 610.12:1990]

3.1.14
reusability
degree to which a software module or other work product can be used in more than one computer program or software system

[IEEE 610.12:1990]

3.1.15
singular input
individual parameter stress testing

3.1.16
software
see 3.1.21 software product

3.1.17
software component
part of a software system

NOTE 1 Software component is used as a general term

NOTE 2 Components can be assembled and decomposed to form new components. In the production activities, components are implemented as modules, 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.

3.1.18
software item
see 3.1.21 software product

3.1.19
software intensive system
space system in which the dominant part of the constituents are software elements

NOTE In such systems, subsystems consist mainly of software. For this type of system, the majority of interfaces are software-software interfaces.

3.1.20
software observability
property of a system for which the values of status variables can be determined throughout observations of the output variables

3.1.21
software product
set of computer programs, procedures, documentation and their associated data

3.1.22
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

3.1.23
software unit
separately compilable piece of source code

NOTE In this Standard no distinction is made between a software unit and a database; both are covered by the same requirements.

3.1.24
state (of the software)
level of maturity of the software product along the life cycle

NOTE It can have four different statuses, "specified", "defined", "qualified" or "accepted", depending on the success in achieving the PDR, CDR, QR or AR milestones.

3.1.25
stress test
test that evaluates a system or software component at or beyond its value limits

3.1.26
unit test
test of software unit to ensure that there are no programming errors

3.1.27
unreachable code
code that cannot be executed due to design or coding error

3.1.28
usability (a quality characteristic)
capability of the software to be understood, learned, used and liked by the user, when used under specified conditions

3.1.29
validation
confirmation, through the provision of objective evidence, that the requirements for a specific intended use or application have been fulfilled

[ISO 9000:2000]

NOTE The validation process (for software) is the process to confirm that the requirements baseline functions and performances are correctly and completely implemented in the final product.

3.1.30
verification
confirmation, through the provision of objective evidence, that specified requirements have been fulfilled

[ISO 9000:2000]

NOTE The verification process (for software) is the 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.

3.2 Abbreviated terms

The following abbreviated terms are defined and used within this Standard:

Abbreviation	Meaning
AR	acceptance review
	NOTE The term SW-AR is often used for clarity to denote ARs that solely involve software products.
CDR	critical design review
	NOTE The term SW-CDR is often used for clarity to denote CDRs that solely involve software products.
COTS	commercial off-the-shelf
CPU	central processing unit
DDF	design definition file
DDR	detailed design review
DJF	design justification file
ECSS	European Cooperation for Space Standardization

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
MMI	man-machine interface
MOTS	modified off-the-shelf
OP	operational plan
ORR	operational readiness review
PDR	preliminary design review

NOTE The term SW-PDR is often used for clarity to denote PDRs that solely involve software products.

QR	qualification review

NOTE The term SW-QR is often used for clarity to denote QRs that solely involve software products.

RB	requirements baseline
SDE	software development environment
SPA	software product assurance
SPR	software problem report
SRB	software review board
SRR	system requirements review

NOTE The term SW-SRR is often used for clarity to denote SRRs that solely involve software products.

SW	software
SWE	software engineering
TS	technical specification
V&V	verification and validation

(This page is intentionally left blank)

4

Space system software product assurance

4.1 Introduction

Clause 4 introduces the structure of this Standard and the framework of the space software product assurance process that form its basis. It is intended to be an informative clause (not normative), and therefore it does not contain explicit requirements to be tailored.

The objectives of software product assurance are to provide adequate confidence to the customer and to the suppliers that developed or reused software satisfies the requirements throughout the system lifetime. In particular the software is developed to perform properly and safely in the operational environment meeting the quality objectives agreed for the project.

This Standard (as tailored for a particular project) 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 and process definition and quality characteristics of products.

The customer ensures that the software product assurance requirements derived from this Standard by the tailoring process (see subclause 4.5) express his/her requirements completely and unambiguously.

The fundamental principle of this Standard is the customer-supplier relationship, assumed for all software developments. The organizational aspects of this are defined in ECSS-M-20. The customer is, in the general case, the procurer of two strongly associated products: the hardware and the software for a system, subsystem, set, equipment or assembly (see ECSS-E-00). The concept of the customer-supplier relationship is applied recursively, i.e. the customer can himself be a supplier to a higher level in the space system as shown in Figure 1.

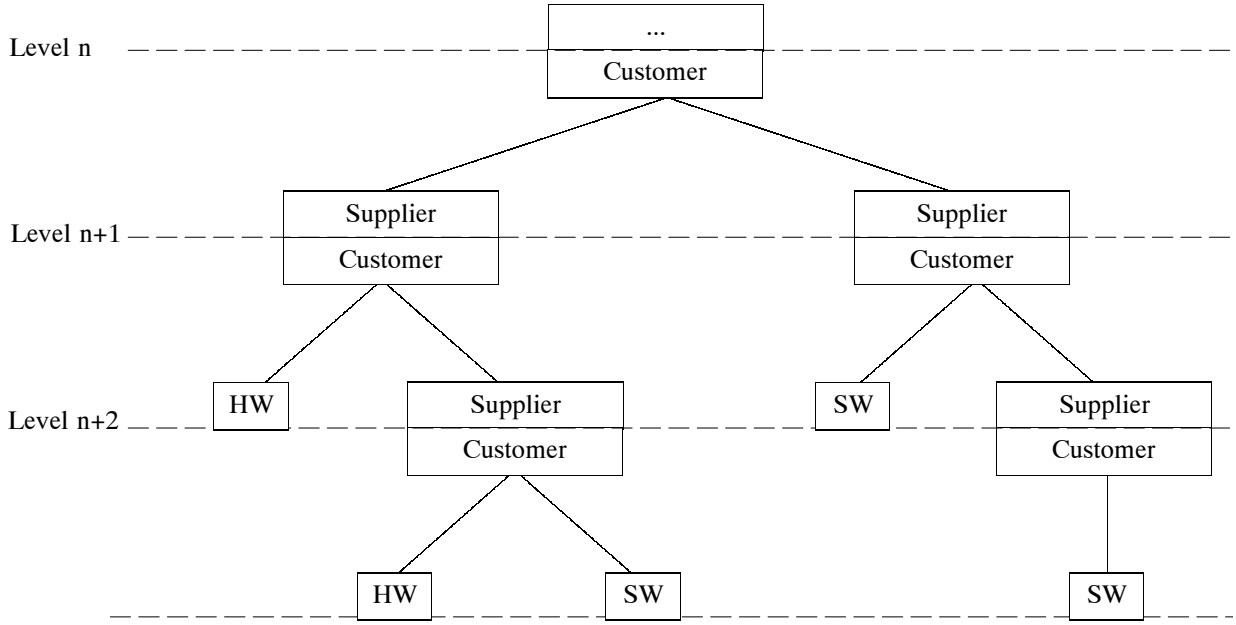


Figure 1: The recursive customer-supplier model

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 his/her suppliers, taking into account their responsibilities and the specific nature of their deliverables.

This Standard defines the detailed requirements for implementation of the software product assurance policy as derived from the general product assurance policy and specified in ECSS-Q-00. This Standard complements ECSS-E-40, Space engineering — Software, with product assurance aspects, integrated in the space system software engineering process as defined in ECSS-E-40. Together the two standards specify all processes for space software development.

The coverage of all software life cycle processes by the ECSS Standards is illustrated in Figure 2.

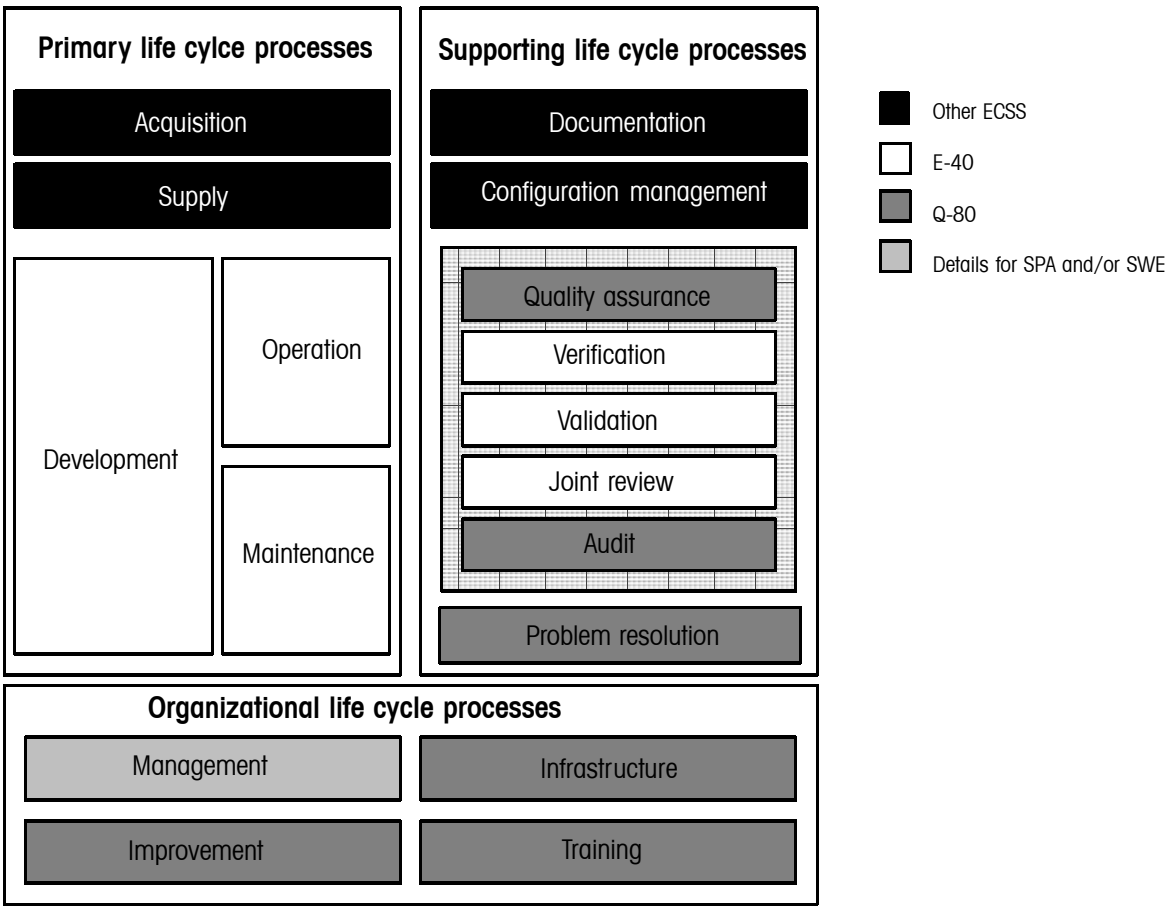


Figure 2: Software life cycle processes in ECSS Standards

4.2 Software product assurance within the overall engineering process

The context of space software product assurance is the overall space system product assurance process and the space software engineering process. This subclause 4.2 defines the general relationship between the software product assurance processes and the general product assurance processes and software engineering processes of space systems.

This Standard covers all aspects of space software product assurance including the implementation aspects of the software product assurance process, and both software process and product related assurance activities. It defines the scope of the space software product assurance process and its interfaces with management, engineering and other system level product assurance activities, which are addressed in the management (-M), engineering (-E) and product assurance (-Q) branches of the ECSS system, and explains how they apply to the software product assurance process.

Software development has important differences from other disciplines as explained in ECSS-E-40B subclause 4.1.

Because of the complexity of software products, these activities are carried out in a disciplined manner to build quality into the product from the very beginning.

Software product assurance consists in both the assurance of the process (software process assurance) and the assurance of the quality of the product (software product quality assurance).

4.3 Organization of this Standard

This Standard is organized into three main parts:

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

An overview of the software documentation is included in annex A.

In the preparation of this Standard the ISO/IEC 12207 standard has been used extensively, providing a common internationally recognized framework for the terminology and software life cycle processes description.

The organization of this Standard is reflected in detail in Figure 3:

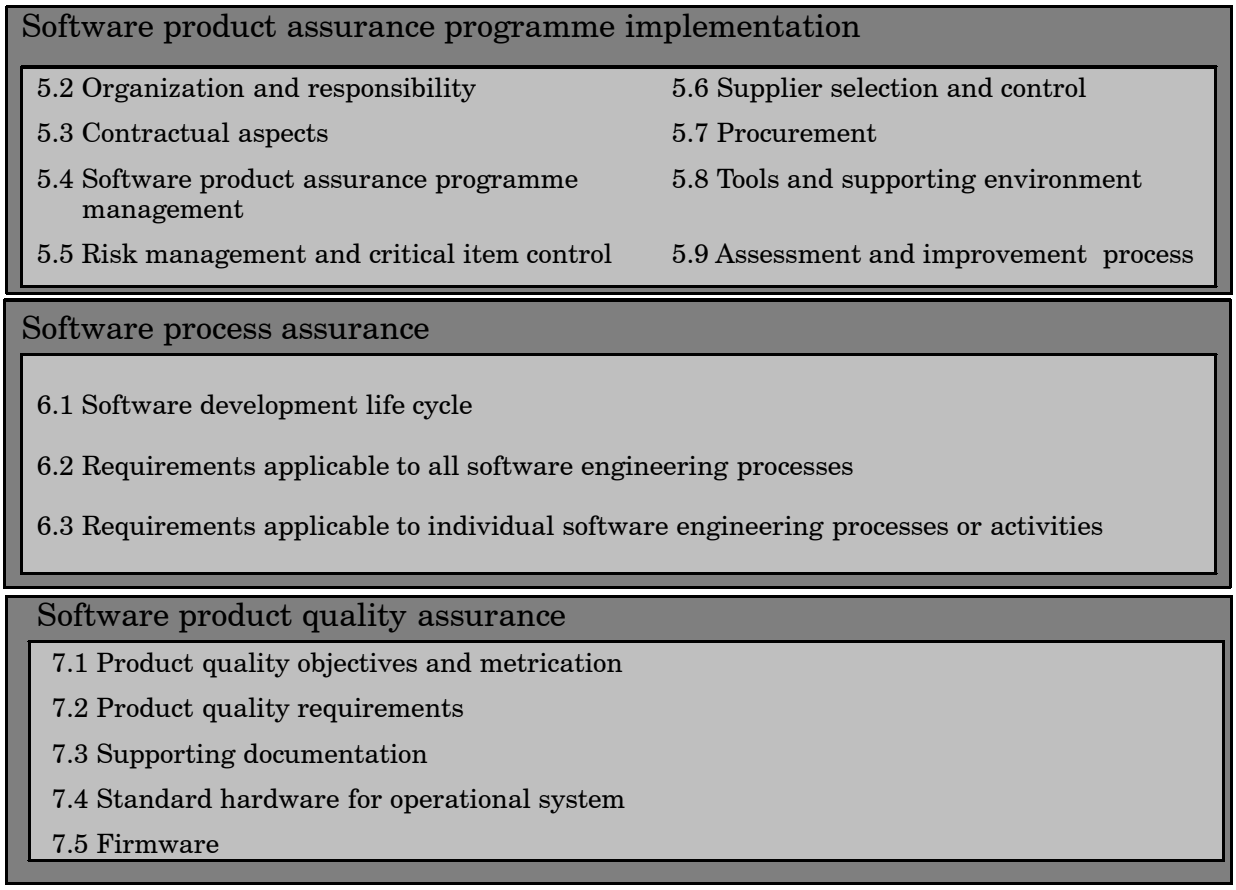


Figure 3: Structure of this Standard

4.4 Relation to ECSS-M, -E and -Q standards

4.4.1 General

This subclause discusses how this Standard interfaces with other ECSS series, namely the ECSS-Q series of standards (product assurance), ECSS-E series of standards (engineering) and the ECSS-M series of standards (management).

Where requirements are adequately covered in normative standards, they are not repeated here except if necessary for clarity, but are explicitly made applicable. See clause 2 for list of ECSS references.

4.4.2 Software engineering

The interface of this Standard to the ECSS-E branch is via ECSS-E-40; equally, the interface of ECSS-E-40 to the ECSS-Q branch is via this Standard.

ECSS-E-40 covers all aspects of space software engineering from requirements definition to retirement.

It defines the scope of the space software engineering processes and their interfaces with management and product assurance, which are addressed in the management (-M) and product assurance (-Q) branches of the ECSS system.

4.4.3 Product assurance

4.4.3.1 General

In ECSS-Q-00A clause 1, the PA discipline is defined as covering: PA management, quality assurance, safety assurance, reliability, availability and maintainability assurance, software product assurance, EEE components, materials, mechanical parts and processes.

ECSS-Q standards define the requirements to be applied to the product assurance of space projects. The following subclauses describe how the ECSS-Q standards apply to the product assurance of software projects.

In addition, requirements that cannot be found in other standards of the Q-branch are defined in this Standard.

4.4.3.2 ECSS-Q-00: Policy and principles

ECSS-Q-00 is a top-level standard that defines product assurance policy, objectives, principles and rules for the establishment and implementation of product assurance programmes to be applied to all aspects of a space project including software.

4.4.3.3 ECSS-Q-20: Quality assurance

ECSS-Q-20 defines the quality assurance (QA) requirements for the establishment and implementation of QA programmes for projects covering mission definition, design, development, production and operations of space systems, including disposal.

4.4.3.4 ECSS-Q-30: Dependability

ECSS-Q-30 defines the requirements for a dependability (reliability, availability and maintainability) assurance programme for space projects.

4.4.3.5 ECSS-Q-40: Safety

ECSS-Q-40 defines the safety programme and the technical safety requirements for space projects. It is intended to protect flight and ground personnel, the launch vehicle, associated payloads, ground support equipment, the general public, public and private property, and the environment from hazards associated with space systems.

4.4.4 Project management

4.4.4.1 General

The ECSS-M branch defines the requirements to be applied to the management of space projects. The following subclauses describe how the ECSS-M standards apply to the management of software projects.

In addition, requirements that cannot be found in the M-branch because they are specific to software product assurance are defined in this Standard.

4.4.4.2 ECSS-M-00: Policy and principles

ECSS-M-00 is a top-level standard that defines project management principles and general requirements to be applied to all aspects of a space project including

software. ECSS-M-00 also addresses risk management, and discusses the terms "customer" and "supplier" which are used in this Standard.

4.4.4.3 ECSS-M-30: Project phasing and planning

ECSS-M-30 defines the phasing and planning requirements for a space project with a "system view" of the entire project. Project phases as defined in ECSS-M-30 are top-level (i.e. mission) phases, used to structure the whole space project. They do not apply recursively to software development.

Requirements concerning phasing and reviews, which are specific to software, are defined in ECSS-E-40B subclauses 4.2 and 5.3.

4.4.4.4 ECSS-M-40: Configuration management

ECSS-M-40 defines the requirements for configuration management for space projects including software.

One aspect of software configuration management is that all configuration items can be regarded as documents (even the code). Therefore, the software configuration management can easily be automated.

Specific software product assurance activities regarding the implementation of configuration management for software are defined in this Standard.

4.4.4.5 ECSS-M-50: Information/documentation management

The objectives of information and documentation management are to ensure the accessibility of information to all parties of the project and to ensure the coherence of this information. These objectives also apply to software projects. The relevant requirements are defined in ECSS-M-50.

Specific software product assurance requirements regarding the implementation of information/documentation management for software are defined in this Standard.

4.5 Tailoring of this Standard

The general requirements for the selection and tailoring of applicable standards are defined in ECSS-M-00-02.

There are several drivers for tailoring, such as 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 critical software. The application of software dependability and safety techniques as described in subclause 6.2.2 of this Standard is also considered. The existence of software of different criticality in the development is accounted for in tailoring.

Tailoring for software development constraints takes into account the special characteristics of the software being developed, and of 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) is also taken into account (see annex C of ECSS-E-40B). Specific requirements for verification, review and inspection are imposed, for example, when full validation on the target computer is not feasible or where 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. This process requires the customer to specify the quality objectives for the product.

5

Software product assurance programme implementation

5.1 Introduction

This clause 5 defines the requirements for the implementation of a software product assurance programme.

Each requirement can be identified by a hierarchical number. The text of the requirement is followed, where necessary, by further explanation of the aim. For each requirement, the associated output is given in the output section. With each output (e.g. "a.", "b."), the destination (document) of that output is indicated in brackets together with the corresponding review. For example: "[DDF, DJF; QR]" denotes an output to the design definition file and the design justification file. The output in this example is requested for the qualification review.

5.2 Organization and responsibility

5.2.1 Organization

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

5.2.2 Responsibility and authority

ECSS-Q-00A subclause 3.3.1 is applicable.

EXPECTED OUTPUT:

a. *Responsibility, authority and interrelation of personnel managing, performing and verifying work affecting quality [PAF; SRR];*

b. *External and internal interfaces and responsibilities of each organization [PAF; SRR];*

c. *Lower level supplier performing delegated product assurance tasks [PAF; SRR].*

5.2.3 Resources

ECSS-Q-00A subclause 3.3.2 is applicable.

EXPECTED OUTPUT: *Software product assurance resource requirements [PAF; SRR].*

5.2.4 Software product assurance manager

5.2.4.1

One person shall be appointed as software product assurance manager for the project.

EXPECTED OUTPUT: *Identification of the software product assurance manager [PAF; SRR].*

5.2.4.2

ECSS-Q-00A subclause 3.3.3 is applicable with "product assurance manager" being replaced by "software product assurance manager".

NOTE 1 In a large project, product assurance organization at system level is mirrored at software level. The software product assurance manager generally reports to the project manager via the product assurance manager. He or she liaises with the software engineers and dependability and safety engineers.

NOTE 2 In a software-intensive project, the software product assurance manager and the product assurance manager can be the same person.

5.2.5 Training

5.2.5.1

A review of the project requirements shall be conducted to establish and make timely provision for acquiring or developing the resources and skills for the management and technical staff.

EXPECTED OUTPUT: *Training plan [MGT; SRR].*

5.2.5.2

Training records shall be maintained.

EXPECTED OUTPUT: *Records of training and experience [PAF].*

5.2.5.3

It shall be ensured that the right composition and categories of appropriately trained personnel are available for the planned activities and tasks in a timely manner.

5.2.5.4

The training subjects shall be determined by the specific tools, techniques, methodologies and computer resources to be used in the development and management of the software product. Personnel can undergo training to acquire skills and knowledge relevant to the specific field with which the software is to deal.

5.3 Contractual aspects

ECSS-Q-00A subclause 3.3.4 is applicable.

5.4 Software product assurance programme management

5.4.1 Software product assurance planning and control

5.4.1.1
The supplier shall develop a software product assurance plan in response to the software product assurance requirements.

NOTE 1 The software product assurance plan may be included in the system product assurance plan.

NOTE 2 Depending on the software characteristics such as type, criticality, size, application, and the organization of the project, the supplier may either propose different software product assurance plans, or an adaptation in the application of his software product assurance plan.

EXPECTED OUTPUT: *Software product assurance plan [PAF; SRR, PDR].*

5.4.1.2
The software product assurance plan shall meet the requirements of a product assurance plan as set out in ECSS-Q-00A subclause 3.3.3.

NOTE If the software product assurance plan refers to supplier internal manuals, standards or procedures, then these documents become an integral part of the supplier's product assurance plan.

EXPECTED OUTPUT: *Software product assurance plan [PAF; SRR, PDR].*

5.4.1.3
The software product assurance plan shall be updated at each milestone in such a way that the activities to be undertaken in the following phase are fully defined.

EXPECTED OUTPUT: *Software product assurance plan [PAF; CDR, QR, AR, ORR].*

5.4.1.4
The software product assurance plan shall specify or reference the following items:

- quality objectives, expressed in measurable terms whenever possible;
- the software development life cycle, the related milestones and the input and output criteria for each development phase;
- types of verification and validation activities (including tests) to be carried out;
- detailed planning of verification and validation activities (including tests) to be carried out, including schedules, resources and approval authorities;
- specific responsibilities for quality activities such as reviews and tests, configuration management and change control, nonconformance control and corrective action;
- methods, tools and rules to be applied;
- the procedures for determining the criticality category of software processes, functions, objects (according to the design methodology adopted), packages, units, files;
- specific actions and measures for supplier control.

EXPECTED OUTPUT: *Software product assurance plan [PAF; SRR, PDR].*

5.4.1.5
Before acceptance review, the software product assurance plan shall be supplemented to specify the quality measures related to the operations and maintenance processes (alternatively a specific software product assurance plan shall be issued).

EXPECTED OUTPUT: *Quality measures for the operations and maintenance processes in the software product assurance plan [PAF; AR].*

5.4.1.6
The supplier shall provide with the software product assurance plan a compliance matrix documenting conformance to the software product assurance requirements applicable for the project or contract.

EXPECTED OUTPUT: *Compliance matrix [PAF; SRR, PDR].*

5.4.2 Software product assurance reporting

5.4.2.1
ECSS-Q-20B subclause 4.4 is applicable.

NOTE The software product assurance reporting can be included in the system product assurance reporting.

5.4.2.2
The software product assurance report shall include an assessment of the current quality of the software development process (see subclause 6.2.5).

EXPECTED OUTPUT: *Assessment of the quality of the software development process in the software product assurance report [PAF; SRR, PDR, CDR, QR, AR, ORR].*

5.4.2.3
The software product assurance report shall include an assessment of the current quality of the product, based on measured properties, verifications undertaken, problems detected and problems resolved.

EXPECTED OUTPUT: *Assessment of the quality of the software product in the software product assurance report [PAF; SRR, PDR, CDR, QR, AR, ORR].*

5.4.2.4
The assessment shall be made with reference to the metrication as defined in the software product assurance plan.

EXPECTED OUTPUT: *Software product assurance report [PAF; SRR, PDR, CDR, QR, AR, ORR].*

5.4.3 Audits

ECSS-Q-20B subclause 4.6 is applicable.

EXPECTED OUTPUT: *Audit plan and schedule [PAF; SRR].*

5.4.4 Alerts

ECSS-Q-20B subclause 5.7 is applicable.

EXPECTED OUTPUT: *a. Preliminary alert information [PAF];*

b. Alert information [PAF].

5.4.5 Nonconformances

5.4.5.1
ECSS-Q-20B subclause 5.6 is applicable with "Nonconformance review board" being replaced by "Software review board".

EXPECTED OUTPUT:
a. *Description of nonconformance control system [PAF; SRR];*
b. *Nonconformance [DJF];*
c. *Nonconformance record [DJF].*

5.4.5.2
The Software review board shall be established at all contractual levels and include, at least, a representative from the software product assurance and the software engineering organizations.

EXPECTED OUTPUT: *Identification of SRB and members [MGT; SRR]*

5.4.5.3
The software product assurance plan shall specify the point in the software life cycle from which the nonconformance procedures apply.

EXPECTED OUTPUT: *Identification of the point in the software life cycle from which the nonconformance procedures apply - software product assurance plan [PAF; SRR, PDR].*

5.4.6 Software problems

5.4.6.1
The supplier shall define and implement procedures for the logging, analysis and correction of all software problems encountered during software development.

EXPECTED OUTPUT: *Software problem reporting procedures [PAF; PDR].*

5.4.6.2
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).

EXPECTED OUTPUT: *Software problem reporting procedures [PAF; PDR].*

5.4.6.3
The supplier shall ensure the correct application of problem reporting procedures.

5.5 Risk management and critical item control

5.5.1 Risk management

Risk management for software shall be performed by cross-reference to the project risk policy, as specified by ECSS-M-00-03.

5.5.2 Critical item control

For critical item control ECSS-Q-00A subclause 3.3.5 is applicable.

5.6 Supplier selection and control

5.6.1 Supplier selection

For supplier selection ECSS-Q-20B subclause 7.2 is applicable.

NOTE The assessment of the supplier's continuous capability to furnish software products and services of the type and quality level being procured can be performed based on internationally recognized approaches such as ISO/IEC 15504.

EXPECTED OUTPUT: *a. Results of pre-award audits [PAF];*

b. Records of procurement sources [PAF].

5.6.2 Supplier requirements

5.6.2.1

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 product assurance plan.

EXPECTED OUTPUT: *Software product assurance requirements for suppliers [PAF; SRR].*

5.6.2.2

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

EXPECTED OUTPUT: *Software product assurance requirements for suppliers [PAF; SRR].*

5.6.3 Supplier monitoring

5.6.3.1

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

5.6.3.2

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

5.6.3.3

The supplier shall ensure that a software development process is defined and applied by the next lower level suppliers in compliance with the software product assurance requirements for suppliers.

EXPECTED OUTPUT: *Next level suppliers' software product assurance plan [PAF; PDR].*

5.6.3.4

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

EXPECTED OUTPUT: *Next level suppliers' software product assurance plan [PAF; PDR].*

5.6.4 Criticality classification

The supplier shall ensure that the procured software is correctly classified for dependability and safety criticality, if this classification forms part of the subcontract.

EXPECTED OUTPUT: *Evidence of dependability and safety criticality classification [PAF; SRR].*

5.7 Procurement

5.7.1 Requirements

The customer shall identify the procurement process for projects where the use of COTS, OTS or MOTS is intended (such as IEEE 1062).

EXPECTED OUTPUT: *Software procurement process for COTS, OTS or MOTS [RB; SRR].*

5.7.2 Selection

For the selection of COTS, OTS or MOTS software components to be used for or integrated into the system, the requirements of subclause 6.2.7 are applicable.

5.7.3 Approval

The choice of procured software shall be described and submitted for customer approval in the form of a software component list.

EXPECTED OUTPUT: *Software component list [DJF; SRR, PDR].*

5.7.4 Procurement details

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

- ordering criteria (e.g. versions, options and extensions);
- receiving inspection criteria;
- arrangements for maintenance and upgrades to new releases;
- back-up solutions if the product becomes unavailable;
- contractual arrangements for the development and maintenance phases.

EXPECTED OUTPUT: *Procurement data [MGT; SRR, PDR].*

5.7.5 Identification

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

5.7.6 Inspection

The supplier shall subject the procured software to a planned receiving inspection against pre-defined criteria before its acceptance.

NOTE Supplementary specific tests associated with the operational environment can be specified.

EXPECTED OUTPUT: *Receiving inspection report [PAF; PDR, CDR, QR].*

5.7.7 Exportability

Any constraints regarding exportability shall be taken into account.

5.8 Tools and supporting environment

5.8.1 Development computer selection

The development computer equipment 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 constraints.

5.8.2 Choice description

The choices of development computer equipment shall be described.

EXPECTED OUTPUT: *Descriptions of choices of development equipment in the software development plan [MGT; SRR, PDR].*

5.8.3 Methods and tools

5.8.3.1

Mature methods and tools shall be used for all activities of the development cycle, including requirements analysis, software specification, design, coding, validation, testing, configuration management, verification and product assurance.

EXPECTED OUTPUT: *Justification included or referenced in the product assurance file [PAF; PDR].*

5.8.3.2

The choice of development methods and tools shall be justified by demonstrating 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 (in an appropriate hardware environment) throughout the development and maintenance lifetime of the product.

NOTE This can be demonstrated through testing or documented assessment.

EXPECTED OUTPUT: *Justification included or referenced in the product assurance file [PAF; PDR].*

5.8.3.3

Suitability of the software development environment shall be justified.

EXPECTED OUTPUT: *Evidence of suitability of the software development environment [DJF; SRR, PDR].*

5.8.3.4
The availability of the software development environment to developers and other users shall be verified before the start of each development phase.

5.8.3.5
The correct use of methods and tools shall be verified and reported.

EXPECTED OUTPUT: *Software product assurance report [PAF; SRR, PDR, CDR, QR, AR, ORR].*

5.8.4 Tool selection

5.8.4.1
If tools are used for automatic code generation, the relevant requirements of subclauses 6.3.3 and 6.3.4.24 of this Standard are applicable. Furthermore, the supplier shall take the following aspects into account:

- evolution of the tools in relation to the tools that use the generated code as an input (e.g. compilers or code management systems);
- 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.

5.8.4.2
The required level of verification of the automatic generation tool shall be at least the same as that for the generated code if the tool is used to skip the verification of the target code.

5.9 Assessment and improvement process

5.9.1 Assessment process

The supplier shall monitor and control the effectiveness 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.

NOTE See ECSS-Q-80-02 for further details

EXPECTED OUTPUT: *Software process assessment plan [PAF].*

5.9.2 Assessment procedure

A process assessment procedure shall be developed, documented and applied.

EXPECTED OUTPUT: *Software process assessment procedure [PAF].*

5.9.3 Assessment records

Assessment records shall be kept and maintained.

EXPECTED OUTPUT: *Software process assessment records [PAF].*

5.9.4 Assessment data

Data shall be collected and analysed to gain an understanding of the strengths and weaknesses of the employed processes.

EXPECTED OUTPUT: *Software process assessment records: strengths and weaknesses [PAF].*

5.9.5 Quality data

Quality data shall be collected, maintained, and used to support the process assessments.

EXPECTED OUTPUT: *Software process assessment records: quality data [PAF].*

5.9.6 Process improvement

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

EXPECTED OUTPUT: *Software process assessment records: improvement plan [PAF].*

5.9.7 Process or project documentation

Process or project documentation shall be updated to reflect the improvements.

EXPECTED OUTPUT: *Software process assessment record: updates to process or project documentation [PAF].*

6

Software process assurance

6.1 Software development life cycle

6.1.1 Life cycle definition

The software development life cycle shall be defined or referenced in the software product assurance plan. If the software development plan does not contain the definition of the software life cycle, then it shall be included in the software product assurance plan

EXPECTED OUTPUT: *Reference to software development life cycle definition [PAF; SRR, PDR].*

6.1.2 Life cycle definition review

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

6.1.3 Life cycle resources

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.

NOTE The life cycle is associated with choices of techniques used during the development of the software product (e.g. database management system, extensive product reuse, and man-machine interface generators) and with the risks inherent in the project (e.g. highly changeable product specifications, stringent schedule constraints and project size).

6.1.4 Quality objectives

In the definition of the life cycle and associated milestones and documents, the quality objectives shall be taken into account (see ECSS-E-40B subclause 5.3.2 and ECSS-Q-80B subclause 7.1).

6.1.5 Phase outputs

The state of completion of phase outputs shall be specified when the life cycle is defined.

EXPECTED OUTPUT: *Reference to the software development life cycle definition [PAF; SRR, PDR].*

6.1.6 Special characteristics

The outputs of each software life cycle phase shall identify those characteristics of the product that are crucial to its safe and proper functioning.

EXPECTED OUTPUT: *Reference to the software development life cycle definition [PAF; SRR, PDR].*

6.1.7 Milestones

The milestones shall be specified when the software life cycle is defined.

NOTE If the need for additional milestones is identified (e.g. Detailed Design Review, see ECSS-E-40B, subclause 5.3.2.9), the product assurance documentation presented during previous reviews can be updated accordingly.

EXPECTED OUTPUT: *Reference to the software development life cycle definition [PAF; SRR, PDR].*

6.1.8 Role of customer

The role of the customer at these milestones shall be defined in accordance with ECSS-M-30A subclause 4.5 and clause 7 and also with ECSS-M-30-01.

EXPECTED OUTPUT: *Reference to the software development life cycle definition [PAF; SRR, PDR].*

6.1.9 Validation process schedule

A milestone shall be scheduled immediately before the software validation process starts, to check that the software status is compatible with the commencement of validation activities and that the necessary resources, software product assurance plans, test and validation documentation, simulators or other technical means are available.

EXPECTED OUTPUT: *Reference to the software development life cycle definition [PAF; SRR, PDR].*

6.2 Requirements applicable to all software engineering processes

6.2.1 Documentation of processes

6.2.1.1

The following activities shall be covered in project plans:

- development;
- specification, design and customer documents to be produced;
- configuration and documentation management;
- verification and validation activities (including testing);
- maintenance.

NOTE In case software-specific plans are not produced, these activities may be addressed either in software-specific plans or in project general plans.

EXPECTED OUTPUT: *Software project plans [MGT, MF, DJF]*

6.2.1.2

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

EXPECTED OUTPUT: *Software project plans [MGT, MF, DJF].*

6.2.1.3
All plans shall be updated for each milestone to reflect any changes during development.

EXPECTED OUTPUT: *Software project plans [MGT, MF, DJF].*

6.2.1.4
The software product assurance plan shall identify all plans to be produced and used, the relationship between them and the time-scales for their preparation and update.

EXPECTED OUTPUT: *Identification of software plans, their interrelations and schedule for preparation [PAF; SRR, PDR].*

6.2.1.5
Each plan shall be reviewed against the relevant contractual requirements.

6.2.1.6
The following activities shall be covered by development procedures and project standards:

- recording of development metrics;
- classification of software product according to its functional criticality;
- use of program design language, if it is used in the detailed design;
- use of coding languages.

EXPECTED OUTPUT: *Procedures and standards [PAF; PDR].*

6.2.1.7
Procedures and project standards shall include provision for all classes of software included in the project.

EXPECTED OUTPUT: *Procedures and standards [PAF; PDR].*

6.2.1.8
All procedures and project standards shall be finalized before starting the activities.

EXPECTED OUTPUT: *Procedures and standards [PAF; PDR].*

6.2.1.9
Each procedure or standard shall be reviewed against the relevant plans and contractual requirements.

6.2.1.10
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.

6.2.2 Software dependability and safety analysis

6.2.2.1
The system dependability analysis (see ECSS-Q-30B, clause 8) shall take the interaction of the software with its environment (e.g. system hardware and human intervention) into account

EXPECTED OUTPUT: *Critical functions identification and analysis [RB; SRR].*

6.2.2.2
The supplier shall carry out a software dependability and safety analysis (see ECSS-Q-40B, subclause 6.4.4.1 and ECSS-Q-30B, subclause 6.4) to assign criticality levels to software components.

EXPECTED OUTPUT: a. *Software criticality analysis report [PAF; SRR, PDR];*

b. *Input to the system safety data package [PAF; SRR, PDR].*

6.2.2.3
The criticality of the software components shall be determined based on the criticality classification of functions (see ECSS-Q-30B, subclause 6.4) and the identification of safety critical functions (see ECSS-Q-40B, subclause 5.4)

EXPECTED OUTPUT: *List of critical software components [PAF; PDR].*

6.2.2.4
The software dependability and safety analysis shall be updated for each development milestone.

EXPECTED OUTPUT: a. *Software criticality analysis report [PAF; CDR, QR, AR, ORR];*

b. *Input to the system safety data package [PAF; CDR, QR, AR, ORR].*

6.2.2.5
The list of critical software components shall be verified and reviewed for continuing validity at each development milestone.

EXPECTED OUTPUT: *List of critical software components [PAF; CDR, QR, AR, ORR].*

6.2.2.6
The supplier shall design critical software components to facilitate dependability and safety analysis and software testing.

6.2.2.7
On the basis of the results of the software criticality analysis, the supplier shall, without introducing undesirable software complexity, minimize the number of critical software components by appropriate software design.

6.2.3 Handling of critical software

6.2.3.1
The supplier shall define and apply measures to assure the reliability of critical software.

These measures can include:

- use of software design or methods that have performed successfully in a similar application;
- failure mode analysis of the software, with the insertion of appropriate features for failure isolation and handling (see ECSS-Q-80-03);
- defensive programming techniques, such as input verification and consistency checks;
- prohibiting the use of language commands and features that are unpredictable;
- 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 could be inadvertently executed are prevented, isolated, or eliminated.

EXPECTED OUTPUT: *Definition of measures and verification activities in software product assurance plan [PAF; PDR, CDR].*

6.2.3.2
The application of the chosen measures to handle the critical software shall be verified.

EXPECTED OUTPUT: *Verification and validation documentation [DJF; PDR, CDR, QR, AR].*

6.2.3.3
Critical software shall be subject to regression testing after:

- any change of functionality (e.g. instruction set of a processor) of the underlying platform hardware, and
- any change of the tools that affect directly or indirectly the generation of the executable code.

NOTE 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.

EXPECTED OUTPUT: *Definition of measures and verification activities in software product assurance plan [PAF; PDR, CDR].*

6.2.3.4
The need for additional verification of critical software shall be analysed after:

- any change of functionality or performance of the underlying platform hardware, and
- any change in the environment in which the software or the platform hardware operate.

EXPECTED OUTPUT: *Definition of measures and verification activities in software product assurance plan [PAF; PDR, CDR].*

6.2.3.5
The measures used to handle critical software and their justification shall be documented in the design.

EXPECTED OUTPUT:
a. *Measures in the design [DDF; PDR, CDR];*
b. *Verification and validation documentation[DJF; PDR, CDR].*

6.2.3.6
Unreachable code shall be removed and the need for re-verification and re-validation shall be analysed.

6.2.3.7
Testing on instrumented code shall be re-run on the non-instrumented code.

6.2.3.8
Integration and validation testing shall be executed on the non-instrumented code.

6.2.3.9
The supplier shall ensure that failure of non-critical software, which is not subject to the assurance measures stated in subclauses from 6.2.3.1 to 6.2.3.8, does not cause failure of critical software.

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

EXPECTED OUTPUT: *a. Measures in the design [DDF; PDR, CDR];*

b. Verification and validation documentation [DJF; PDR, CDR].

6.2.4 Software configuration management

6.2.4.1
ECSS-M-40 shall be applied for software configuration management complemented by the following requirements.

NOTE Subclause 6.2.4 contains requirements on product assurance of software configuration management which are complementary to the ECSS-M-40A (subclause 5.3.2) requirements.

6.2.4.2
The software configuration management system shall allow any reference version to be re-generated from backups.

EXPECTED OUTPUT: *Software configuration management [MGT; SRR, PDR].*

6.2.4.3
The software configuration file shall be submitted to the customer for approval at acceptance testing.

EXPECTED OUTPUT: *Software configuration file [DDF; AR].*

6.2.4.4
The software configuration file shall be available and up to date for each project milestone.

EXPECTED OUTPUT: *Software configuration file [DDF; PDR, CDR, QR, AR, ORR].*

6.2.4.5
Any components of the code generation tool that are customizable by the user shall be put under configuration control and the change control procedures defined for the project shall take into account the specific aspects of these components.

EXPECTED OUTPUT: *Software configuration file [DDF; PDR, CDR, QR, AR, ORR].*

6.2.4.6
The supplier shall ensure that all authorized changes are implemented in accordance with the software configuration management plan.

EXPECTED OUTPUT: *Authorized changes - Software configuration file [DDF; PDR, CDR, QR, AR, ORR].*

6.2.4.7
The following documents shall be controlled (see ECSS-Q-20B subclause 5.1):

- procedural documents describing the quality system to be applied during the software life cycle;
- planning documents describing the planning and progress of the contract 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,

- documentation for the software and system operators and users, and
- maintenance documentation.

6.2.4.8
The supplier shall establish a mechanism to protect all supplied software (e.g. source, executable or data) against corruption.

EXPECTED OUTPUT: *a. Identification and protection method or tool [PAF; SRR, PDR];*

b. Identification and protection method in the software configuration file [DDF; CDR, QR, AR, ORR].

6.2.4.9
For all software products in operational use, the supplier shall use a checksum-type identification key calculation and checking software on each executable binary or each file delivered (e.g. source or database).

EXPECTED OUTPUT: *Identification and protection method or tool [PAF; SRR, PDR].*

6.2.4.10
The checksum value shall be specified in the software configuration file.

NOTE The checksum value is calculated:
- prior to each delivery;
- at reception to check identification.

EXPECTED OUTPUT: *Checksum value in the software configuration file [DDF; CDR, QR, AR, ORR].*

6.2.4.11
If the protection mechanism used is based on a tool, this tool shall be agreed between the customer and the supplier in advance.

EXPECTED OUTPUT: *Identification and protection method or tool [PAF; SRR, PDR].*

6.2.4.12
The software media deliverable to the customer shall be marked by the supplier during the preparation of each delivery, indicating the following minimum information:

- the software name;
- the version number;
- the reference to the software configuration file.

EXPECTED OUTPUT: *a. Labelling method [PAF; SRR, PDR];*

b. Labels [DDF; CDR, QR, AR].

6.2.5 Process metrics

6.2.5.1
Process metrics shall be collected, stored, analysed, and reported on a regular basis by applying quality models and procedures.

NOTE Subclause 6.2.5 deals with process metrics only. Note that subclause 7.1 deals with product metrics.

EXPECTED OUTPUT: *Details of metrics in software product assurance plan [PAF; SRR, PDR].*

6.2.5.2
Metrics shall be used to manage the development and to assess the quality of the development process.

EXPECTED OUTPUT: *Details of metrics in software product assurance plan [PAF; SRR, PDR].*

6.2.5.3
The collection, storage, analysis and reporting of metrics shall be defined in the software product assurance plan.

EXPECTED OUTPUT: *Details of metrics in software product assurance plan [PAF; SRR, PDR].*

6.2.5.4
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.

EXPECTED OUTPUT: *Metrics reports in software product assurance reports [PAF; PDR, CDR, QR, AR, ORR].*

6.2.5.5
The following basic process metrics shall be used within the supplier's organization, and reported to the customer:

- number of problems detected during inspection;
- number of problems detected during integration and validation testing and use.

NOTE See also software problem reporting described in subclause 5.4.6.

EXPECTED OUTPUT: *Metrics reports in software product assurance reports [PAF; PDR, CDR, QR, AR, ORR].*

6.2.5.6
Metrics reports shall be included in the software product assurance reports.

EXPECTED OUTPUT: *Metrics reports in software product assurance reports [PAF; PDR, CDR, QR, AR, ORR].*

6.2.6 Verification

6.2.6.1
Activities for the verification of the quality requirements shall be specified in the definition of the verification plan (see ECSS-E-40B subclause 5.8.2).

NOTE Verification includes various techniques such as review, inspection, testing, walk-through, cross-reading, desk-checking, and many types of analysis such as traceability analysis, formal proof or fault tree analysis. The term "review" includes both joint reviews with the customer and internal reviews.

EXPECTED OUTPUT: *Software verification plan - verification of quality requirements [DJF; PDR].*

6.2.6.2
The outputs of each development activity shall be verified for conformance against the inputs to that activity, to demonstrate that they:

- conform to appropriate development standards,
- contain or reference acceptance criteria for forwarding to subsequent activities, and
- identify those characteristics of the product that are crucial to its safe and proper functioning (e.g. margin philosophy of the computing resources or performances of operating systems on which the application runs).

NOTE Only outputs which have been subjected to planned verifications are accepted as inputs for subsequent activities.

EXPECTED OUTPUT: *a. SPA reports [PAF; SRR, PDR, CDR, QR, AR, ORR];*

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

6.2.6.3
A summary of the assurance activities concerning the verification process and their findings shall be included in software product assurance reports.

EXPECTED OUTPUT: *Software problem reports [DJF; SRR, PDR, CDR, QR, AR, ORR].*

6.2.6.4
The verification results, including any software problem reports, and any further actions to ensure that the specified requirements are met, shall be recorded and checked when the action is completed.

EXPECTED OUTPUT: *Software problem reports [DJF; SRR, PDR, CDR, QR, AR, ORR].*

6.2.6.5
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.

EXPECTED OUTPUT: *a. SPA reports [PAF; CDR, QR, AR, ORR];*

b. Software problem reports [DJF; CDR, QR, AR, ORR].

6.2.6.6
Software containing configurable code shall be verified specifically to ensure that configuration of table sizes, other configurable compilation or any other configurable run time code cannot harm the operation of the system and that configurable generation or configurable run time activation of code cannot occur accidentally.

EXPECTED OUTPUT: *a. SPA reports [PAF; CDR, QR, AR, ORR];*

b. Software problem reports [DJF; CDR, QR, AR, ORR].

6.2.6.7
The supplier shall:

- ensure that the planned verification activities are adequate to ensure that the products of each phase are in conformity,
- ensure that verification activities are performed according to the plan, and
- ensure that the planned verification activities include full verification of critical software (see subclause 6.2.2) at each stage of its development.

EXPECTED OUTPUT: *Software problem reports [DJF; SRR, PDR, CDR, QR, AR, ORR].*

6.2.6.8
Each review and inspection shall be based on a written procedure.

EXPECTED OUTPUT: *Review and inspection procedures [PAF; SRR, PDR, CDR, QR, AR, ORR].*

6.2.6.9
The review or inspection procedures shall specify:

- the inspected items,
- the person in charge,
- participants,
- the means of inspection (e.g. tools or check list), and
- the nature of the report.

EXPECTED OUTPUT: *Review and inspection procedures [PAF; SRR, PDR, CDR, QR, AR, ORR].*

6.2.6.10
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.

6.2.6.11
Review and inspection records shall be kept. These records shall identify the reviewed item, the author, the reviewer, the review criteria and the finding of the review.

EXPECTED OUTPUT: *Review and inspection records [PAF; SRR, PDR, CDR, QR, AR, ORR].*

6.2.6.12
Traceability matrices shall be verified at each milestone.

EXPECTED OUTPUT: *Review and inspection records [PAF; SRR, PDR, CDR, QR, AR, ORR].*

6.2.6.13
Independent software verification shall be performed for highly critical software.

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

NOTE 2 ISVV is not considered to be merely “independent” testing of the product. The concept of ISVV includes the necessity of setting up an independent team of highly qualified staff composed of specialists from all disciplines including software product assurance. This team, independently of the development team, performs verification activities such as conducting reviews, inspections, testing and auditing.

EXPECTED OUTPUT: *ISVV plan [DJF; SRR, PDR] and ISVV report [DJF; PDR, CDR, QR, AR, ORR].*

6.2.7 Reuse of existing software

6.2.7.1
Analyses of the advantages to be obtained by using existing software shall be carried out and finalized at the architectural design stage.

NOTE 1 Reused software includes software from previous developments which is used for the project development as is or with adaptation. It also includes software supplied by the customer for use in the project development.

NOTE 2 Reused software includes any software developed outside the contract to which this Standard is applicable.

NOTE 3 In addition to the above products, reused software includes products such as: freeware and open source software products.

EXPECTED OUTPUT: *Justification of selection of reused software in the software reuse file [DJF; SRR, PDR].*

6.2.7.2
These analyses shall serve to refine or validate a priori choices and to define the associated actions to conform to quality requirements.

EXPECTED OUTPUT: *Justification of selection of reused software in the software reuse file [DJF; SRR, PDR].*

6.2.7.3
The choice of reused software shall take into account:

- the assessment of the product with respect to all applicable requirements (including the quality requirements);
- the criticality of the function provided;
- the acceptance (demonstration of correct operation) and warranty conditions;
- the available support documentation;
- the conditions of installation, preparation, training and use;
- the identification and registration by configuration management;
- maintenance responsibility;
- the maintenance conditions, including the possibilities of changes;
- the copyright constraints (modification rights);
- the licencing conditions.

EXPECTED OUTPUT: *Justification of selection of reused software in the software reuse file [DJF; SRR, PDR].*

6.2.7.4
The reused software shall be analysed to check the following aspects and define any necessary corrective actions:

- the durability and validity of methods and tools, used in the initial development, that it is envisaged to use again;
- for each reused component:
  - its validation level or operational behaviour;
  - its documentation status;
  - its quality status, i.e. residual nonconformances, complexity analyses, or waivers.

EXPECTED OUTPUT: *Justification of selection of reused software in the software reuse file [DJF; SRR, PDR].*

6.2.7.5
All the elements on which the decision to reuse software is based shall be recorded in a software reuse file.

EXPECTED OUTPUT: *Software reuse file [DJF; SRR, PDR, CDR].*

6.2.7.6
This software reuse file shall contain:

- a qualitative summary review of the reused components and an assessment of the possible level of reuse,
- a description of the assumptions and the method of calculating the level of reuse, and
- planned corrective actions regarding the reused software.

EXPECTED OUTPUT: *Software reuse file [DJF; SRR, PDR, CDR].*

6.2.7.7
The supplier shall provide the software reuse file to the customer for acceptance.

EXPECTED OUTPUT: *Software reuse file [DJF; SRR, PDR, CDR].*

6.2.7.8
Reused software shall be subject to the same requirements as procured software, as set out in subclause 5.7 of this Standard.

6.2.7.9
If existing software proposed for reuse was developed with less rigour than specified by the project standards for the criticality level of its intended use, the supplier shall provide the evidence of the product's suitability by use of (a combination of) approaches such as:

- analysis of software life cycle data from the previous development to ensure the adequacy of the development processes,
- reverse engineering to generate adequate software documentation, and
- use of the product service history which includes information concerning:
  - configuration management and change control of the software product,
  - effectiveness of problem reporting,
  - stability and maturity of the software,
  - relevance for the product service history for the new environment,
  - actual error rates and maintenance records, and
  - impact of modifications.

EXPECTED OUTPUT: *Evaluation report of proposed product in the software reuse file [DJF; SRR, PDR, CDR].*

6.2.7.10
If the analysis of available data indicates remaining risks, the supplier shall propose and agree with the customer the additional verification tasks to be performed.

EXPECTED OUTPUT: *Action plan in the software reuse file [DJF; SRR, PDR, CDR].*

6.2.7.11
The customer shall review the available data for adequacy and confirm the suitability of the proposed existing software for reuse in the higher criticality function.

6.3 Requirements applicable to individual software engineering processes or activities

6.3.1 Software requirements analysis

6.3.1.1
The software development shall start with analysis to fully and unambiguously define the software requirements in the technical specification on the basis of these inputs.

NOTE 1 The requirements baseline on the software can be presented to the supplier in various forms, e.g. a customer requirements specification, a system requirements specification or a general description of the project objectives.

NOTE 2 The supplier may develop these requirements in close cooperation with the customer and the supplier may obtain the customer's approval before entering the development stage. In some cases, the technical specification is provided by the customer.

6.3.1.2
The technical specification shall be subject to documentation control and configuration management as part of the development documentation.

6.3.1.3
In the definition of the technical specification all results from the dependability analysis (including results from the HSIA (ECSS-Q-30B subclause 8.2.2)) shall be taken into account.

6.3.1.4
The technical specification shall include all non-functional requirements necessary to satisfy the requirements baseline in addition to the functional requirements, including, as a minimum, the following: performance, safety, reliability, robustness, quality, maintainability, configuration management, security, privacy, metrication, verification and validation.

NOTE Performance requirements include requirements on numerical accuracy.

EXPECTED OUTPUT: *Software requirements specification - Non-functional requirements [TS; PDR].*

6.3.1.5
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 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.

6.3.2 Software architectural design and design of software items

6.3.2.1
A design methodology appropriate to the type of software product being developed and suitable tools shall be used, and identified in the software development plan.

The design method(s) and tools documentation shall be made available to the customer for acceptance.

EXPECTED OUTPUT: *Definition of methodology and tools in the software development plan [MGT; SRR, PDR].*

6.3.2.2
The design definition file shall be subject to documentation control and configuration management.

6.3.2.3
Mandatory and advisory design standards shall be defined and applied.

EXPECTED OUTPUT: *Design standards [PAF; SRR, PDR].*

6.3.2.4
For software in which numerical accuracy is relevant to mission success (e.g. for an attitude and orbit control subsystem, scientific data generation components) specific rules on design and code shall be defined to ensure that the specified level of accuracy is obtained.

EXPECTED OUTPUT: *Design and coding rules for numerical accuracy [PAF; PDR].*

6.3.2.5
Adherence to design standards shall be verified.

EXPECTED OUTPUT: *Results in software product assurance reports [PAF; PDR, CDR].*

6.3.2.6
The complexity and modularity of the design shall be checked to ensure achievement of quality requirements.

NOTE These checks can be implemented in parallel with the design process so as to ensure that they are taken into account by the designers.

EXPECTED OUTPUT: *Results in software product assurance reports [PAF; PDR, CDR].*

6.3.2.7
The nature of the checks, the criteria, the tools used and the feedback process to the design team shall be described in the software product assurance plan.

EXPECTED OUTPUT: *Description of checks in the software product assurance plan [PAF; PDR].*

6.3.2.8
Synthesis of results obtained and corrective actions implemented shall be described in quality assessment reports.

EXPECTED OUTPUT: *Results in software product assurance reports [PAF; PDR, CDR].*

6.3.2.9
The supplier shall review the design documentation to ensure that it contains the appropriate level of information for maintenance activities.

EXPECTED OUTPUT:
a. *Description of checks in the software product assurance plan [PAF; SRR, PDR];*
b. *Results in software product assurance reports [PAF; PDR, CDR, QR, AR, ORR].*

6.3.3 Coding

6.3.3.1
Coding standards (including consistent naming conventions, and adequate commentary rules) shall be specified and observed.

EXPECTED OUTPUT: *Coding standards [PAF; PDR].*

6.3.3.2
The standards shall be designed to ensure consistency with the product quality requirements.

NOTE Coding standards depend on the software quality objectives (see subclause 7.1).

EXPECTED OUTPUT: *Coding standards and description of tools [PAF; PDR].*

6.3.3.3
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.

EXPECTED OUTPUT: *Coding standards and description of tools [PAF; PDR].*

6.3.3.4
Coding standards shall be reviewed with the customer to ensure that they reflect product quality requirements.

EXPECTED OUTPUT: *Coding standards and description of tools [PAF; PDR].*

6.3.3.5
Use of low-level programming languages shall be justified.

EXPECTED OUTPUT: *Document justifying suitability of the language [DJF; PDR].*

6.3.3.6
Measurement and analysis on code (e.g. complexity) shall be performed to ensure conformance to quality requirements.

NOTE 1 This measurement can be implemented in parallel with the coding process to ensure that the results are easily taken into account by the developers.

NOTE 2 Automatic means to measure adherence of the code to the coding standards can be used.

EXPECTED OUTPUT: *Description of measurements and tools [PAF; PDR, CDR, QR, AR, ORR].*

6.3.3.7
The nature of the measurements, the aspects of the coding standards to be checked, the tools used and the relationship to the coding team shall be described in the software product assurance plan.

EXPECTED OUTPUT: *Description of measurements and tools [PAF; PDR].*

6.3.3.8
Synthesis of code analysis results and corrective actions implemented shall be described in software product assurance reports.

EXPECTED OUTPUT: *Description of measurements and synthesis in software product assurance reports [PAF; CDR, QR, AR, ORR].*

6.3.3.9
The code shall be put under configuration control immediately after successful unit testing.

6.3.3.10
Unreachable code shall be removed, or a justification for the decision not to remove it shall be provided.

EXPECTED OUTPUT: *Justification for retention of unreachable code [DJF; CDR, QR, AR]*

6.3.4 Testing and validation

6.3.4.1
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, e.g. functional, boundary, performance, and usability tests,
- the tests to be performed in accordance with the plans and procedures, and
- means and organizations to perform assurance function for testing and validation.

EXPECTED OUTPUT: *Assurance activities for testing in the software product assurance plan [PAF; PDR, CDR].*

6.3.4.2
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:

- for unit level testing (such as branch coverage and decision coverage);
- for integration level testing (such as call graph and parameter passing);
- for validation against the technical specification and validation against the requirements baseline (such as all requirements, all external interfaces and using nominal, boundary, zero/nil and out of range data value).

EXPECTED OUTPUT: *Assurance activities for testing in the software product assurance plan [PAF; PDR, CDR].*

6.3.4.3
The supplier shall ensure through internal review that the test procedures and data are adequate, feasible and traceable and that they satisfy requirements.

EXPECTED OUTPUT: *Collected data and analysis of the results in the software product assurance report [PAF; CDR, QR, AR, ORR].*

6.3.4.4
Test readiness reviews shall be held before the commencement of key test activities.

6.3.4.5
For each test activity, test coverage shall be checked with respect to the stated goals.

NOTE This check can be made with an automatic tool to measure the coverage obtained.

EXPECTED OUTPUT: *Collected data and analysis of the results in the software product assurance report [PAF; CDR, QR, AR, ORR].*

6.3.4.6
The measurements shall be performed throughout the test programme.

NOTE The supplier can implement these measurements in a way that allows developers to take the results as inputs.

EXPECTED OUTPUT: *Collected data and analysis of the results in the software product assurance report [PAF; CDR, QR, AR, ORR].*

6.3.4.7
The supplier shall ensure that nonconformances and software problem reports detected during testing are properly documented and reported to those concerned.

EXPECTED OUTPUT: *Nonconformance report and SPR [DJF; CDR, QR, AR, ORR].*

6.3.4.8
The test coverage of configurable code shall be checked to ensure that the stated requirements are met in each tested configuration.

EXPECTED OUTPUT: *Statement of compliance with test plans and procedures [PAF; CDR, QR, AR, ORR].*

6.3.4.9
The supplier shall ensure that problem reports and subsequent actions are properly closed out.

6.3.4.10
Provisions shall be made to allow witnessing of tests by the customer.

6.3.4.11
Provisions shall be made to allow witnessing of tests by supplier personnel independent of the development (e.g. specialist software product assurance personnel).

6.3.4.12
The supplier shall ensure that tests are conducted in accordance with approved test procedures and data, that the configuration under test is correct, that the tests are properly documented and that the test reports are up to date and valid.

EXPECTED OUTPUT: *Statement of compliance with test plans and procedures [PAF; CDR, QR, AR, ORR].*

6.3.4.13
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.

EXPECTED OUTPUT: *Statement of compliance with test plans and procedures [PAF; CDR, QR, AR, ORR].*

6.3.4.14
The supplier shall confirm in writing that the tests are successfully completed.

EXPECTED OUTPUT: *Statement of compliance with test plans and procedures [PAF; CDR, QR, AR, ORR].*

6.3.4.15
Review boards looking to engineering and product assurance aspects shall be convened after the completion of key test phases.

6.3.4.16
Areas affected by any modifications shall be identified and re-tested (regression testing).

6.3.4.17
In case of re-testing, all test related documentation (test procedures and data, reports) shall be updated accordingly.

EXPECTED OUTPUT: *Updated test documentation [DJF; CDR, QR, AR, ORR].*

6.3.4.18
The need for regression testing and additional verification of software shall be analysed after any change of platform hardware.

EXPECTED OUTPUT: *Updated test documentation [DJF; CDR, QR, AR, ORR].*

6.3.4.19
The need for regression testing and additional verification of software shall be analysed after a change or update of any tool used to generate it (e.g. source code or object code).

EXPECTED OUTPUT: *Updated test documentation [DJF; CDR, QR, AR, ORR].*

6.3.4.20
Validation shall be carried out by staff who have not taken part in the design or coding of the software being validated.

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

6.3.4.21
Validation of the flight software against the requirements baseline on the flight equipment model shall be performed on a software version without any patch.

EXPECTED OUTPUT: *Contribution to the test and validation documentation [DJF; PDR, CDR, QR, AR, ORR].*

6.3.4.22
The supplier shall review the test documentation to ensure that it is up to date and organized to facilitate its reuse for maintenance.

6.3.4.23
The testing strategy for the project shall consider the specific aspects of testing of automatically generated code.

EXPECTED OUTPUT: *Contribution to the test and validation documentation [DJF; PDR, CDR, QR, AR, ORR].*

6.3.4.24
The requirements on testing applicable to the automatically generated code shall ensure the achievement of the same objectives as those for manually generated code unless different objectives are specifically stated in the verification requirements.

NOTE Relaxation of testing requirements for automatically generated code can only be granted if the code generation tool was qualified.

EXPECTED OUTPUT: *Contribution to the test and validation documentation [DJF; PDR, CDR, QR, AR, ORR].*

6.3.4.25
Tests shall be organized as activities in their own right in terms of planning, resources and team composition. The necessary resources shall be identified early in the life cycle taking into account the operating and maintenance requirements. Test tool development or acquisition (hardware and software) shall be planned for in the overall project plan.

EXPECTED OUTPUT: *Contribution to the test and validation documentation [DJF; PDR, CDR, QR, AR, ORR].*

6.3.4.26
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.

EXPECTED OUTPUT: *Contribution to the test and validation documentation [DJF; PDR, CDR, QR, AR, ORR].*

6.3.4.27
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.

6.3.4.28
Where testing under the operational environment occurs, 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).

EXPECTED OUTPUT: *Contribution to the test and validation documentation [DJF; AR].*

6.3.4.29
Independent software validation shall be performed for highly critical software. (see subclause 6.2.6.13 for guidance on ISVV).

EXPECTED OUTPUT:
a. *ISVV plan [DJF; SRR, PDR];*
b. *ISVV report [DJF; PDR, CDR, QR, AR, ORR].*

6.3.4.30
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.

EXPECTED OUTPUT:
a. *SPA reports [PAF; CDR, QR, AR, ORR];*
b. *Software problem reports [DJF; CDR, QR, AR, ORR].*

6.3.4.31
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.

EXPECTED OUTPUT:
a. *SPA reports [PAF; CDR, QR, AR, ORR];*
b. *Software problem reports [DJF; CDR, QR, AR, ORR].*

6.3.4.32
Software containing configurable code shall be validated specifically to ensure that configuration of table sizes, other configurable compilation or any other configurable run time code cannot harm the operation of the system and that configurable generation or configurable run time activation of code does not occur accidentally.

EXPECTED OUTPUT:
a. *SPA reports [PAF; CDR, QR, AR, ORR];*
b. *Software problem reports [DJF; CDR, QR, AR, ORR].*

6.3.5 Software delivery and acceptance

6.3.5.1
The roles, responsibilities and obligations of the supplier and customer during installation shall be established.

EXPECTED OUTPUT: *Contribution to the installation plan [DJF; AR].*

6.3.5.2
The installation shall be performed in accordance with the installation plan.

6.3.5.3
The customer shall establish an acceptance test plan specifying the intended acceptance tests including specific tests taking into account the target environment (see ECSS-E-40B subclause 5.7.3.1).

NOTE 1 The acceptance tests can be partly made up of tests used during previous test activities.

NOTE 2 The acceptance test plan takes into account the requirement for operational demonstration, either as part of acceptance or after acceptance.

EXPECTED OUTPUT: *Acceptance test plan [DJF; AR].*

6.3.5.4
The customer shall ensure that the acceptance tests are performed in accordance with the approved acceptance test plan (see ECSS-E-40B subclause 5.7.3.1).

NOTE The method of handling problems detected during the acceptance procedure and their disposition are agreed between the customer and supplier and are documented.

6.3.5.5
The supplier shall ensure before the software is presented for customer acceptance that:

- the delivered software complies with the contractual requirements (including any specified content of the software acceptance data pack);
- the source and object code supplied correspond to each other;
- all agreed changes are implemented;
- all nonconformances are either resolved or declared.

EXPECTED OUTPUT: *Acceptance test plan [DJF; AR].*

6.3.5.6
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.

6.3.5.7
Any discovered problems shall be documented in nonconformance reports.

EXPECTED OUTPUT: *Nonconformance report [DJF; AR].*

6.3.5.8
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.

EXPECTED OUTPUT: *Acceptance test report [DJF; AR].*

6.3.5.9
The acceptance test report shall certify conformance to the procedures and state the conclusion concerning the test result for the software product under test (accepted, conditionally accepted, rejected).

EXPECTED OUTPUT: *Acceptance test report [DJF; AR].*

6.3.6 Operations

6.3.6.1
During operations, the quality of the mission products related to software shall be agreed with the customer and users.

NOTE Quality of products can include such parameters as:
- error-free data;
- availability of data and permissible outages;
- permissible information degradation.

EXPECTED OUTPUT: *Contribution to the operational plan [OP; AR].*

6.3.6.2
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.

EXPECTED OUTPUT: *Contribution to the validation of the operational requirements [PAF; AR].*

6.3.6.3
The product assurance plan for system operations (see ECSS-Q-20B subclause 11.4.1) shall include consideration of software.

EXPECTED OUTPUT: *Input to product assurance plan for systems operation [PAF; ORR]*

6.3.7 Maintenance

6.3.7.1
The organization responsible for maintenance shall be identified early in the development life cycle to allow a smooth transition into the operations and maintenance.

NOTE 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.

EXPECTED OUTPUT: *Maintenance plan - assurance [MF; QR, AR, ORR].*

6.3.7.2
The maintenance organization shall specify the assurance, verification and validation activities applicable to maintenance interventions.

EXPECTED OUTPUT: *Maintenance plan - assurance [MF; QR, AR, ORR].*

6.3.7.3
The maintenance plans shall be verified against specified requirements for maintenance of the software product.

NOTE The maintenance plans and procedures can address corrective, improving, adaptive and preventive maintenance, differentiating between "routine" and "emergency" maintenance activities.

6.3.7.4
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;
- maintenance records and reports.

EXPECTED OUTPUT: *Maintenance plan - assurance [MF; QR, AR, ORR].*

6.3.7.5
Rules for the submission of maintenance reports shall be established and agreed as part of the maintenance plan.

EXPECTED OUTPUT: *Rules for submission of maintenance reports - Maintenance plan - assurance [MF; QR, AR, ORR].*

6.3.7.6
All maintenance activities shall be logged in predefined formats and retained.

EXPECTED OUTPUT: *Maintenance records [MF; QR, AR, ORR].*

6.3.7.7
Maintenance records, including as a minimum the following information, shall be established for each software product:

- list of requests for assistance or problem reports that have been received 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.

NOTE 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.

EXPECTED OUTPUT: *Maintenance records [MF; QR, AR, ORR].*

7

Software product quality assurance

7.1 Product quality objectives and metrication

7.1.1 Assurance activities for product quality requirements

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

EXPECTED OUTPUT: *Software product assurance plan - assurance activities [PAF; SRR, PDR].*

7.1.2 Deriving of requirements

The software quality requirements shall be derived from the reliability, safety, maintainability and quality requirements of the system.

EXPECTED OUTPUT: *Software quality requirements [TS; PDR].*

7.1.3 Quality models

7.1.3.1

Quality models shall be used to specify the quality requirements.

NOTE This can be done by reference to a quality model such as ISO/IEC 9126.

EXPECTED OUTPUT: *Software quality models [PAF; PDR].*

7.1.3.2

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

- functionality;
- reliability;
- maintainability;
- reusability;
- operability;
- documentation quality;
- suitability for safety;
- security.

EXPECTED OUTPUT: *Software quality models [PAF; PDR].*

7.1.3.3
Quality requirements shall be expressed in quantitative terms or constraints.

EXPECTED OUTPUT: *Software quality requirements [TS; PDR].*

7.1.4 Product metrics

7.1.4.1
The supplier shall define a metrication programme to verify and prove that the project rules for design, code and documentation are properly applied and complied with.

7.1.4.2
The supplier shall define the relevant metrics, their associated target values and the means to collect or measure them, to assess the actual quality characteristics of the product for comparison with those required.

NOTE Guidance on metric set selection and target values is provided in ECSS-Q-80-04.

EXPECTED OUTPUT: *Products metrics specification and justification in the software product assurance plan [PAF; SRR, PDR].*

7.1.5 Measurement

Measurements shall be performed throughout the development and the results obtained shall be used to define corrective actions.

EXPECTED OUTPUT: *Report of the analysis and remedial actions in the software product assurance report [PAF; PDR, CDR, QR, AR, ORR].*

7.1.6 Measurement results

The results shall be used to provide the customer with an insight into the level of quality obtained through software product assurance reports.

EXPECTED OUTPUT: *Report of the analysis and metrics in the software product assurance report [PAF; PDR, CDR, QR, AR, ORR].*

7.1.7 Basic metrics

The following basic products metrics shall be used:

- size (design, code);
- complexity (design, code);
- fault density and failure intensity;
- test coverage;
- number of failures.

EXPECTED OUTPUT: *Product metrics specification and justification in the software product assurance plan [PAF; SRR, PDR].*

7.1.8 Metrication process

Metrics chosen shall be collected, stored, analysed and reported.

EXPECTED OUTPUT: *Report of the analysis and metrics in the software product assurance report [PAF; PDR, CDR, QR, AR, ORR].*

7.1.9 Metrics analysis

The metrics shall be analysed against target values or quality requirements and remedial actions shall be taken to ensure conformance to quality requirements.

EXPECTED OUTPUT: *Report of the analysis and remedial actions in the software product assurance report [PAF; PDR, CDR, QR, AR, ORR].*

7.1.10 Numerical accuracy

Numerical accuracy shall be estimated and verified.

EXPECTED OUTPUT: *Numerical accuracy analysis [DJF; PDR, CDR, QR].*

7.1.11 Analysis of software behaviour

The supplier shall define the organization and means implemented to collect and analyse data required for the study of software behaviour (e.g. failures, corrections, duration of runs).

EXPECTED OUTPUT: *Report of the analysis of software behaviour in the software product assurance report [PAF; PDR, CDR, QR, AR, ORR].*

7.1.12 Metrics trend

Data shall be collected with respect to predetermined procedures, and processed to derive:

- descriptive statistics (e.g. the number of units at each level of complexity), and
- trend analysis (such as trends in software problems).

EXPECTED OUTPUT: *Report of the analysis and metrics in the software product assurance report [PAF; PDR, CDR, QR, AR, ORR].*

7.1.13 Improvement actions

Metrics data shall be used to determine further actions to improve the software.

EXPECTED OUTPUT: *Records of data collection, analysis and results and actions for improvement [PAF; PDR, CDR, QR, AR, ORR].*

7.2 Product quality requirements

7.2.1 Technical specification

7.2.1.1

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

EXPECTED OUTPUT: *Software quality requirements [TS; PDR].*

7.2.1.2

The software requirements shall be a complete, unambiguous set of requirements.

7.2.1.3

Requirements shall be stated in terms that allow verification and validation.

NOTE Preferably requirements are specified in measurable/ quantitative terms

7.2.1.4
For each requirement the method for verification and validation shall be specified.

EXPECTED OUTPUT: *Verification and validation method for each requirement [DJF; PDR].*

7.2.2 Design and related documentation

7.2.2.1
The software design shall meet the quality requirements as documented in the technical specification.

7.2.2.2
The product shall be designed to facilitate testing and to meet the non-functional requirements.

7.2.2.3
Software with a long planned lifetime shall be designed with minimum dependency on the operating system and the hardware, in order to aid portability.

EXPECTED OUTPUT:
a. *Product quality requirements reflected in coding and design standards [PAF; SRR, PDR];*
b. *Justification of design choices [DJF; PDR, CDR].*

7.2.3 Software intended for reuse

7.2.3.1
The development of software to be reused shall follow ECSS-E-40B subclauses 5.2.5.6, 5.4.3.9 and 5.4.3.10

7.2.3.2
The information related to components developed for reuse shall be separated from others in the technical specification, design justification file, design definition file and the product assurance file.

7.2.3.3
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.

7.2.3.4
The technical specification of components developed for reuse shall include requirements for maintainability, portability and verification of those components for the projects intending to reuse them.

EXPECTED OUTPUT: *Technical specification for reusable components in the software reuse file [DJF; SRR, PDR].*

7.2.3.5
The configuration management system shall take the specific aspects of software developed for reuse into account, 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, and
- transfer of the configuration and documentation management information to the next project.

EXPECTED OUTPUT: *Configuration management for software reuse file [MGT; SRR, PDR].*

7.2.3.6
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.

EXPECTED OUTPUT: *Test reports for the software reuse file [DJF; CDR].*

7.2.3.7
The supplier shall certify that the tests have been successfully completed on all the relevant platforms.

EXPECTED OUTPUT: *Test reports for the software reuse file [DJF; SRR, PDR].*

7.3 Supporting documentation

7.3.1 Test and validation documentation

7.3.1.1
Detailed test and validation documentation (data, procedures and expected results) defined in the ECSS-E-40 DJF shall be consistent with the defined test and validation strategy (see subclause 6.3.4 and ECSS-E-40B subclauses 5.5.3, 5.5.4, 5.6 and 5.8).

7.3.1.2
The test documentation shall cover the test environment, tools and test software, personnel required and associated training requirements.

7.3.1.3
The criteria for completion of each test and any contingency steps shall be specified.

7.3.1.4
Test procedures, data and expected results shall be specified.

7.3.1.5
The hardware and software configuration shall be identified and documented as part of the test documentation.

7.3.1.6
For any requirements not covered by testing a verification report shall be drawn up documenting or referring to the verification activities performed.

EXPECTED OUTPUT: *Verification reports [DJF; CDR, QR, AR].*

7.3.2 Reports and analysis

Software product assurance reports shall report on the execution and the results of all assurance, verification and validation activities.

EXPECTED OUTPUT: *Software product assurance report [PAF; SRR, PDR, CDR, QR, AR, ORR].*

7.4 Standard hardware for operational system

7.4.1 Procurement

The subcontracting and procurement of hardware shall be carried out according to the requirements of ECSS-Q-20B clause 7.

EXPECTED OUTPUT:
a. *Justification of selection of ground equipment [DJF; SRR, PDR];*
b. *Receiving inspection report [DJF; SRR, PDR].*

7.4.2 Constraints

The choice of procured hardware shall take account of the constraints associated with both the development and the actual use.

EXPECTED OUTPUT: *Justification of selection of ground equipment [DJF; SRR, PDR].*

7.4.3 Selection

The ground computer equipment 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 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, and
- compatibility.

EXPECTED OUTPUT: *Justification of selection of ground equipment [DJF; SRR, PDR].*

7.4.4 Maintenance

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

7.4.5 Documentation

Justification of the selection of ground computer equipment shall be referenced in the software product assurance plan.

EXPECTED OUTPUT: *Justification of selection of ground equipment [DJF; SRR, PDR].*

7.5 Firmware

7.5.1 Device programming

The supplier shall establish procedures for firmware device programming and duplication of firmware devices.

EXPECTED OUTPUT: *Procedures described or referenced in the software product assurance plan [PAF; PDR].*

7.5.2 Marking

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

EXPECTED OUTPUT: *Marking described or referenced in the software product assurance plan [PAF; PDR].*

7.5.3 Calibration

The supplier shall ensure that the firmware programming equipment is calibrated.

(This page is intentionally left blank)

Annex A (normative)

Software documentation

ECSS-E-40 Part 1B, Annex A defines the contents of the software documents to be produced. The contents are defined by the outputs of the clauses in this standard and in ECSS-E-40, and the list of the outputs for each milestone of the project is provided.

The overall structure is given in Figure A-1.

ECSS-E-40 Part 2B, defines the content of the document requirements definitions (DRDs) which are called up by this Standard and by ECSS-E-40 Part 1B.

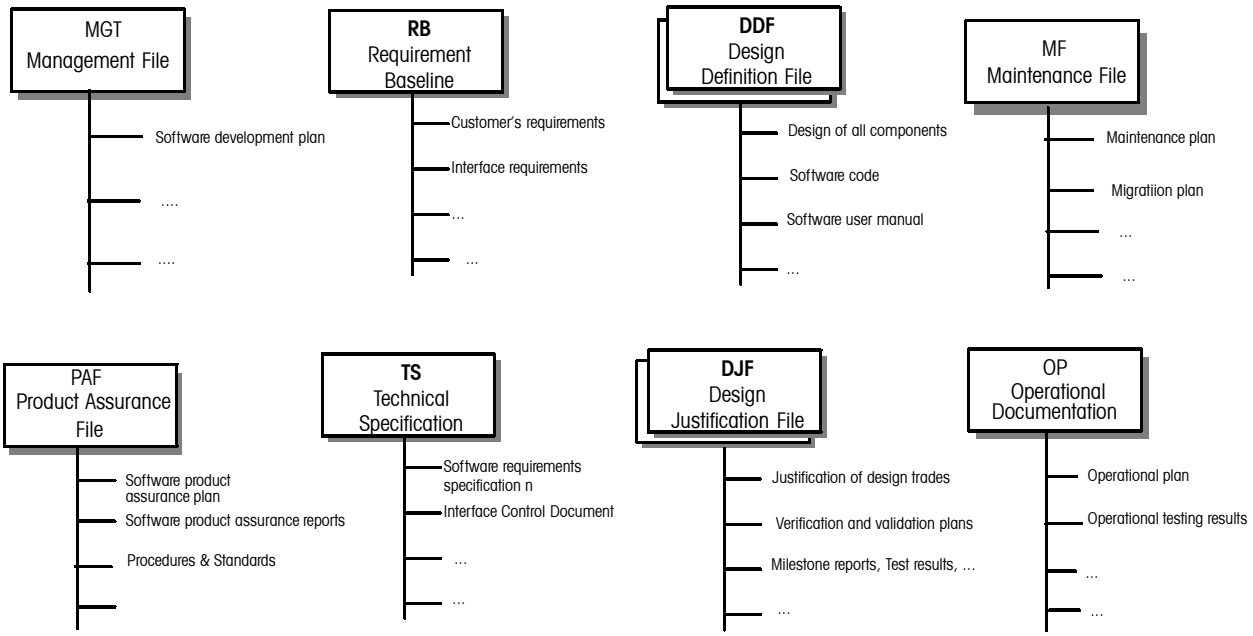


Figure A-1: Overview of software documents

(This page is intentionally left blank)

Annex B (informative)

References to other ECSS Standards

Referenced ECSS Standard	***clause*** **page(s)**
ECSS-Q-00	*cl4* 18, 21
ECSS-Q-00A	*cl4* 21; *cl5* 23, 24, 25, 27
ECSS-Q-20	*cl4* 21
ECSS-Q-20B	*cl5* 26, 27, 28; *cl6* 38, 53; *cl7* 59
ECSS-Q-30	*cl4* 21
ECSS-Q-30B	*cl6* 35, 36, 45
ECSS-Q-40	*cl4* 21
ECSS-Q-40B	*cl6* 36
ECSS-Q-80	*cl4* 21
ECSS-Q-80B	*cl6* 33
ECSS-Q-80-02	*cl5* 31; *cl6* 36
ECSS-Q-80-04	*cl7* 56
ECSS-E-00	*cl4* 17
ECSS-E-40	*cl4* 18, 21; *cl7* 59
ECSS-E-40B	*cl4* 19, 22; *cl6* 33, 34, 40, 52; *cl7* 58, 59
ECSS-M-00	*cl4* 21, 22
ECSS-M-00-02	*cl4* 22
ECSS-M-00-03	*cl5* 27
ECSS-M-20	*cl4* 17
ECSS-M-30	*cl4* 22
ECSS-M-30A	*cl6* 34
ECSS-M-30-01	*cl6* 34
ECSS-M-40	*cl4* 22; *cl6* 38
ECSS-M-40A	*cl6* 38
ECSS-M-50	*cl4* 22

(This page is intentionally left blank)

Bibliography

ECSS-Q-80-02 [1)]	Space product assurance — Software process assessment and improvement
ECSS-Q-80-03 [1)]	Space product assurance — Software dependability and safety methods and techniques
ECSS-Q-80-04 [1)]	Space product assurance — Software metrication programme definition and implementation
ECSS-M-00	Space project management — Policy and principles
ECSS-M-00-02A	Space project management — Tailoring of space standards
ECSS-M-10	Space project management — Project breakdown structures
ECSS-M-20	Space project management — Project organisation
ECSS-M-50	Space project management — Information/documentation management
ECSS-M-60	Space project management — Cost and schedule management
ECSS-M-70	Space project management — Integrated logistic support
ECSS-E-00	Space engineering — Policy and principles
ECSS-E-10	Space engineering — System engineering
ISO 9000:2000	Quality management systems — Fundamentals and vocabulary
ISO 9126-1:2001	Software engineering — Product quality — Part 1: Quality model
ISO/IEC 2382-20:1990	Information technology — Vocabulary — Part 20: System development
ISO/IEC 12207:1995	Information technology — Software life cycle processes
ISO/IEC TR 15504:1998	Information technology — Software process assessment
IEEE 610.12:1990	IEEE Standard Glossary of software engineering terminology
IEEE 1062:1993	IEEE recommended practices for software acquisition

1) To be published.

(This page is intentionally left blank)

ECSS Document Improvement Proposal

1. Document I.D.	2. Document date	3. Document title
ECSS-Q-80B	10 October 2003	Software product assurance

4. Recommended improvement (identify clauses, subclauses and include modified text or graphic, attach pages as necessary)

5. Reason for recommendation

6. Originator of recommendation

Name:	Organization:	
Address:	Phone: Fax: e-mail:	**7. Date of submission:**

8. Send to ECSS Secretariat

Name:	Address:	Phone: +31-71-565-3952
W. Kriedte ESA-TOS/QR	ESTEC, P.O. Box 299 2200 AG Noordwijk The Netherlands	Fax: +31-71-565-6839 e-mail: Werner.Kriedte@esa.int

Note: The originator of the submission should complete items 4, 5, 6 and 7.

An electronic version of this form is available in the ECSS website at: http://www.ecss.nl/
At the website, select “Standards” - “ECSS forms” - “ECSS Document Improvement Proposal”

(This page is intentionally left blank)