ECSS–Q–80A
19 April 1996

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,
2200AG Noordwijk,
The Netherlands.

Price: 35 Dutch Guilders

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.

Requirements in this standard are defined in terms of what must be accomplished, rather than in terms of how to organise and perform the necessary work. This allows existing organisational 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.

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

(This page is intentionally left blank)

Contents List

Figures

1

General

1.1 Scope

This standard defines a set of requirements to be used in Software Product Assurance 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.

This standard shall be tailored as defined in 1.5 to a specific contract or project when Software Product Assurance Requirements are prepared.

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

ECSS–M–00 Programme Management

ECSS–M–40 Configuration Management

ECSS–M–50 Documentation Management

ECSS–Q–30 Dependability Management

ECSS–Q–40 Safety.

1.2 Objectives

The objectives of software product assurance are to provide adequate confidence to the customer and to the contractors that developed or re–used software satisfies the requirements throughout the system lifetime. In particular the software should perform properly and safely in the operational environment and the software product should meet quality objectives.

This standard (as tailored for a particular contract) 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.

1.3 Definitions and Abbreviations

1.3.1 Definitions

For the purposes of this standard, the definitions given in ECSS–P–001 Issue 1 apply. In particular, it should be noted that the following terms have a specific definition for use in ECSS standards.

The following terms and definitions are specific to this standard and shall be applied.

“Acceptance Testing

The 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 [ISO 2382].”

“Assessment

An action of applying specific documented assessment criteria to a specific software module, package, or product for the purpose of determining acceptance or release of the software module, package or product [ISO 9126].”

“Critical Software

A defined set of software components which have been evaluated and whose continuous operation has been determined to be essential for safe and reliable operation of the system. Critical software is composed of two independent elements: Reliability Critical Software and Safety Critical Software. [PSS–01–21]

- **Reliability Critical Software**
  Software which has been evaluated and found to have a Functional Effect Severity Category at level 1 or 2.
- **Safety Critical Software**
  Software which has been evaluated and found to: a) have a hazard Consequence category at level 1, or b) perform emergency caution and warning functions, or c) initiates escape and rescue functions.”

“Integration Test (IT)

a. The progressive linking and testing of programs or modules in order to ensure their proper functioning in the complete system [ISO 2382].

b. Testing in which software components, hardware components, or both are combined and tested to evaluate the interaction between them [IEEE].”

“Metric

A quantitative measure of the degree to which a system, component, or process possesses a given attribute [IEEE].”

“Portability (a Quality Characteristic)

A set of attributes that bear on the ability of software to be transferred from one environment to another [ISO 9126].”

“Quality Characteristics (Software)

A set of attributes of a software product by which its quality is described and evaluated. A software quality characteristic may be refined into multiple levels of subcharacteristics [ISO 9126].”

“Quality Model (Software)

A model that relates Quality Characteristics to attributes of processes or products. Quality models permit the early prediction of Quality Characteristics by measuring related attributes.”

“Regression testing (Software)

Selective retesting to detect faults introduced during modification of a system or system component, to verify that the modifications have not caused unintended

adverse effects, or to verify that a modified system or system component still meets its specified requirements [IEEE]."

"Reusability

The degree to which a software module or other work product can be used in more than one computer program or software system [IEEE]."

"Software

Computer programs, procedures, rules and possibly associated documentation and data pertaining to the operation of a computer system. [IEEE]"

"Software Product

Complete set of computer programs, procedures and associated documentation and data designated for delivery to a user [ISO 9000–3]."

"Software Product Assurance

The 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, will conform adequately to customer requirements."

"System Testing

Testing conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements [IEEE]"

"Unit Test

A test of individual programs or modules in order to ensure that there are no analysis or programming errors [ISO 2382]."

"Usability (a Quality Characteristic)

The ease with which a user can learn to operate, prepare inputs for, and interpret outputs of a system or component [IEEE]."

1.3.2 Abbreviations

The following abbreviation is defined and used within this standard.

Abbreviation	Meaning
ISVV	Independent software verification and validation

1.4 Interfaces with Other Disciplines

Where requirements are adequately covered in applicable documents they are not repeated here except if necessary for clarity but are explicitly made applicable. See clause 1.8 for lists of applicable and reference documents.

1.5 Applicability – Tailoring Guidelines

This standard is intended to be applied to the customer/supplier interfaces for a specific contract, in its tailored form.

The purpose of this clause is to give to the customer general guidance on tailoring the standard for a specific contract or project.

Tailoring may be done by deleting requirements, limiting applicability to a specific part of the system or modifying outputs or proofs to be provided, refining or specifying existing requirements or in exceptional cases adding new requirements.

Tailoring should include the definition of project-specific standards and procedures as appropriate.

There are several drivers for tailoring, i.e. dependability and safety aspects, software development constraints, product quality objectives and commercial parameters. The type of software development (e.g. database, real-time) and the

target system (e.g. embedded processor, host system, programmable devices, application-specific integrated circuits) should also be taken into account.

Tailoring for dependability and safety aspects should be based on the selection of requirements which are related to the verification, validation and levels of proofs demanded of the critical software. The application of software dependability and safety techniques as described in sub–clause 3.2.2 of this standard should also be considered.

Tailoring for software development constraints should take account of the special characteristics of the software being developed, and of the development environment. Specific requirements for verification, review and inspection should be 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 objectives and commercial parameters should be done by selecting requirements on quality of the product as explained in clause 4 of this standard. This process requires the customer to specify the quality objectives for the product.

The existence of software of differing criticality in the development should be accounted for in tailoring. This ensures that inspection effort, etc is directed at software which is truly critical, whilst software of lower criticality can be developed using more economical methods and procedures. The customer and supplier should consider defining and applying levels of software criticality, based on the dependability and safety analysis and other drivers, to aid the tailoring process.

1.6 Responsibilities

The customer is responsible for ensuring that the Software Product Assurance requirements derived from this standard by the tailoring process (see 1.5) express his requirements completely and unambiguously.

The supplier is responsible for the compliance of himself and of his subcontractors with the Software Product Assurance requirements and for providing the specified evidence of compliance.

To this end the supplier is also responsible for specifying the software product assurance requirements for his subcontractors, taking into account their responsibilities and the specific nature of their deliverables.

1.7 Normative and Informative Documents

1.7.1 Normative Documents

This ECSS Standard incorporates, by dated or undated reference, provisions from other publications. These normative references are cited at the appropriate place in the text and publications are listed hereafter. For dated references, subsequent revisions of any of these apply to this ECSS standard only when incorporated in it by revision. For undated references the latest edition of the publication referred to applies.

This clause lists the documents which may be applicable during the preparation of the tailored standard for a specific project.

After tailoring of this standard (see 1.5) to the characteristics of the project, this clause should be updated in the light of the requirements that have been selected and the other tailored standards.

The following documents are applicable to the extent specified within the text of the document:

ECSS–Q–00	“Space product assurance: Policy and principles”
ECSS–Q–20	“Space product assurance: Quality assurance”

ECSS–M–00	“Programme Management”
ECSS–P–001	“Glossary of Terms”

1.7.2 Informative Documents

ECSS–Q–80–1	“Guidelines for tailoring ECSS–Q–80 to a project”
ECSS–Q–80–2	“Guidelines on software product assurance programme implementation”
ECSS–Q–80–3	“Guidelines for software dependability and safety techniques”
ECSS–Q–80–4	“Guidelines for software metric programme definition and implementation”
ECSS–M–40	“Configuration Management”
ECSS–Q–30	“Dependability Management”
ECSS–Q–40	“Safety Management”
ISO 9126	“Information technology – software product evaluation – quality characteristics and guidelines for their use”

1.8 Structure and Presentation of this Standard

The structure of this standard is shown in figure 1.

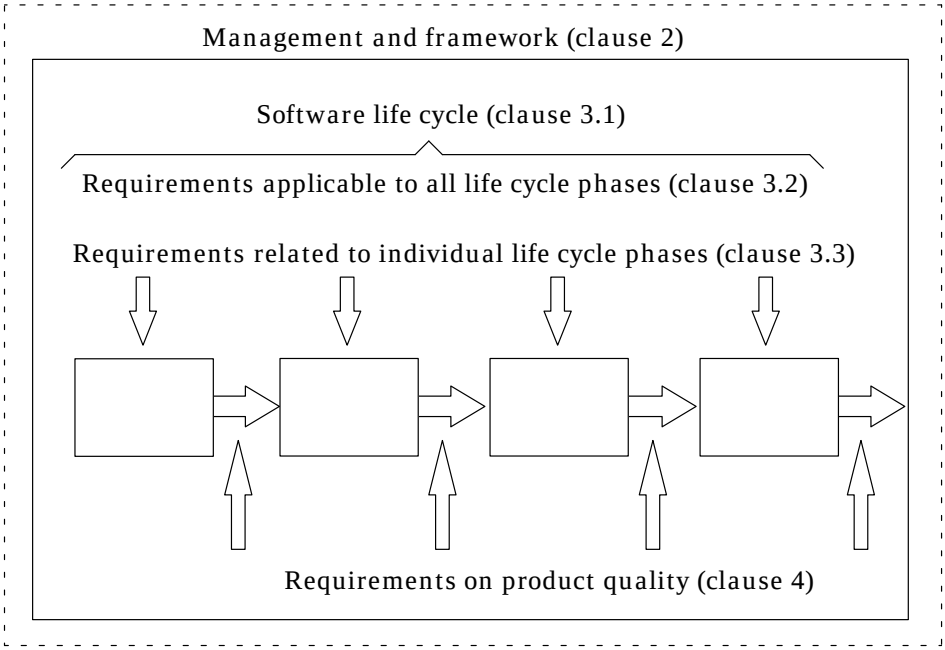

Figure 1: Structure of this ECSS standard

Clause 2 of this standard presents requirements on the management and framework of software product assurance.

Clause 3 of this standard presents requirements on software life–cycle activities and processes. These include requirements that are independent of life–cycle phases and those that are related to individual phases.

Clause 4 of this standard presents requirements on the quality of software products, including both executable code and related products such as documentation and test data.

Each requirement has a corresponding Required output identified. This is included to assist compliance and also to assist the customer in the selection of requirements during the tailoring process.

Certain requirements are labelled as “Assurance activities”, which are often implemented by an independent software product assurance function. This labelling is for guidance (see also ECSS–Q–80–2), and it is not intended to impose any particular organisation on the supplier except as specified in clause 2.1.

This standard includes requirements on the process which generate a framework in which the standard is to be applied. These requirements are marked “(RE)”.

2

Requirements on Management and Framework

2.1 Organisation and Responsibility

Assurance activity:

2.1.1

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

2.1.2 Responsibility and Authority

ECSS–Q–00 sub–clause 3.3.1 is applicable.

2.1.3 Resources

ECSS–Q–00 sub–clause 3.3.2 is applicable.

2.1.4 Software Product Assurance Manager

a. One person shall be appointed as Software Product Assurance manager for the project.

b. ECSS–Q–00 sub–clause 3.3.3 is applicable with 'product assurance manager' being replaced by 'software product assurance manager'.

The software product assurance programme to be implemented is defined by this standard.

In a large system project, product assurance organisation at system level should be mirrored at software level. The software product assurance manager should report to the project manager via the product assurance manager. He or she should liaise with the software engineers and dependability and safety engineers.

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

2.1.5 Training

a. Personnel performing specific assigned tasks shall be qualified on the basis of appropriate education, training and/or experience.

REQUIRED OUTPUT: *Records of training/experience*

The subjects to be addressed should be determined in the light of the specific tools, techniques, methodologies and computer resources to be used in the development and management of the software product. It may also be necessary to require that personnel undergo training to acquire skills and knowledge relevant to the specific field with which the software is to deal.

2.2 Contractual Aspects

ECSS–Q–00 sub–clause 3.3.4 is applicable.

2.3 Software Product Assurance Programme Management

2.3.1 Software Product Assurance Planning and Control

Assurance activity:

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

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

Depending on the software characteristics such as type, criticality, size, application, and the organisation 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.

REQUIRED OUTPUT: *Software product assurance plan for approval by the customer*

b. The software product assurance plan shall meet the requirements of a product assurance plan as set out in ECSS–Q–00 sub–clause 3.3.3.

c. 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.

d. 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, modules, files;
- specific actions and measures for subcontractor control.

Before acceptance testing, the software product assurance plan should be supplemented to specify the quality measures applied during the warranty phase. These measures should be restated in the maintenance plan.

e. The supplier shall provide with his software product assurance plan a compliance matrix documenting his compliance with the software product assurance requirements applicable for the project/contract.

REQUIRED OUTPUT: *Compliance matrix*

2.3.2 Software Product Assurance Reporting

Assurance activity:

a. ECSS–Q–20 clause 2.4 is applicable.

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

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

The software product assurance reporting may be included in the system product assurance reporting.

REQUIRED OUTPUT: *Software product assurance report*

2.3.3 Audits

ECSS–Q–20 clause 2.6 is applicable.

2.3.4 Alerts

ECSS–Q–20 clause 3.7 is applicable.

2.3.5 Nonconformances

a. ECSS–Q–20 clause 3.6 is applicable with “Material Review Boards” being replaced by “Software Review Boards”.

b. 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 organisations.

c. The software product assurance plan shall specify the point in the software life cycle from which the nonconformance procedures have to be applied.

2.3.6 Software Problems

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

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

REQUIRED OUTPUT: *Software problem reporting procedures*

Assurance activity:

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

Trends in software problems should be identified and analysed (see also clause 4.1).

2.4 Risk Assessment and Critical Item Control

2.4.1

For risk assessment ECSS–M–00, is applicable.

2.4.2

For critical item control ECSS–Q–20 clause 2.8 is applicable.

2.5 Subcontractor Control

Assurance activity:

2.5.1

The supplier shall establish appropriate software product assurance requirements for the subcontractors, including a requirement to produce a subcontractor product assurance plan.

REQUIRED OUTPUT: *Software product assurance requirements for subcontractors for customer's acceptance*

Assurance activity:

2.5.2

a. The supplier shall monitor the subcontractors for compliance with the product assurance requirements.

b. The monitoring process shall include the review and approval of the subcontractors' product assurance plan, the continuing verification of process and products, and the monitoring of the final validation of the product.

The supplier should ensure that appropriate software process development is defined and applied by the subcontractor in compliance with the software product assurance requirements for subcontractors.

REQUIRED OUTPUT: *Subcontractors' Software product assurance plan for customer's acceptance*

Assurance activity:

2.5.3

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

3

Requirements on Life–Cycle Activities and Processes

3.1 Software Life Cycle

Software life cycle definition

3.1.1

(RE) The supplier shall define and follow a software development and maintenance life cycle for each software product.

The software life cycle should take into account the technical characteristics and development constraints for the software.

3.1.2

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

REQUIRED OUTPUT: *Software development and maintenance life–cycle definition*

Assurance activity:

3.1.3

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

Assurance activity:

3.1.4

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.

The life cycle should be associated with choices of techniques used during the development of the software product (database management system, extensive product re-use, man/machine interface generators, etc) and with the risks inherent in the project (highly changeable product specifications, stringent schedule constraints, project size, etc.).

Development phases

3.1.5

(RE) In order to ensure effective assurance, software development shall be broken down into successive phases from the statement of requirements to the entry of the software into service.

These phases, and the milestones marking the transitions between them, constitute the software development life cycle.

3.1.6

(RE) A software specification phase shall be included at the beginning of the development life cycle.

3.1.7

(RE) A validation phase, covering the whole software product, shall be included at the end of the development life cycle.

3.1.8

(RE) The maintenance phase and the interfaces between the development and maintenance (e.g. the documents to be produced, allocation of responsibilities) shall be identified in the software life cycle.

The software development cycle should be included in the system development cycle into which it will be integrated.

3.1.9

The development cycle of embedded software shall be compatible with that of the hardware into which it will be embedded.

3.1.10

(RE) The development life cycle shall define the required inputs and outputs for each development phase.

These should be chosen to permit effective verification of the outputs against the inputs.

3.1.11

(RE) The required outputs of each phase shall include documents, in outline or complete versions, and the results of verification activities on the phase technical outputs.

The required outputs should be related to the tools and methods used for the software development.

3.1.12

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

Milestones

3.1.13

(RE) A series of milestones (reviews or technical meetings) shall be defined for each software product.

3.1.14

The role of the customer at these milestones shall be defined.

3.1.15

(RE) Milestone meetings shall examine the work carried out in the previous phase and assess the level of preparation for the next phase.

3.1.16

Milestones allowing an overall system view of the activities performed on each component shall be established at complete software level.

Assurance activity:

3.1.17

(RE) A milestone shall be scheduled immediately before the software validation phase, to check that the software status is compatible with the commencement of validation activities and that the necessary resources, software product assurance plans, test case specifications and procedures, simulators etc. are available.

3.1.18

(RE) A milestone shall be scheduled at the completion of software validation, for a summary review of the test reports.

REQUIRED OUTPUT: *Definition of milestones*

3.2 Requirements Applicable to All Life–Cycle Phases

3.2.1 Process Documentation

Plans

The following activities should be covered in project plans:

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

a. (RE) All plans shall be finalised before the phase for which they are applicable is started.

b. (RE) All plans shall be updated for each milestone to reflect any changes during development.

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

REQUIRED OUTPUT: *Identification of plans and their preparation/update timescales*

Assurance activity:

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

Assurance activity:

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

Procedures and standards

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

- configuration management of all products and documentation;
- 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.

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

h. All procedures and project standards shall be finalised before the phase for which they are applicable is started.

REQUIRED OUTPUT: *Procedures and Standards*

Assurance activity:

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

Assurance activity:

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

Document preparation

k. Typical plans and writing guides shall be prepared before the phase in which the documents have to be produced.

REQUIRED OUTPUT: *The typical plans and writing guides*

3.2.2 Software Dependability and Safety

The following dependability and safety requirements concern software products involved in fulfilling critical functions, as determined by the system level functional analysis (see ECSS–Q–30).

Various analyses of safety and reliability (e.g. sneak analysis, fault tree analysis) may include software components. The supplier should ensure that the methods used are appropriate, that the software designs used are accurate and that the analyses are updated along with any modifications to the software design.

The supplier should endeavour to design critical software components as simply as possible, to facilitate dependability and safety analysis and software testing.

a. A functional analysis at system level shall be used to identify the critical software modules.

b. Criticality allocation for software shall be made by cross-reference to the project risk policy, as required by ECSS–M–00.

c. The functional analysis shall take into account the interaction of the software with its environment (system hardware, human intervention, etc.) jointly with the system level analysis.

REQUIRED OUTPUT: *List of critical modules for customer's acceptance*

Assurance activity:

d. The list of critical modules for software shall be verified for continuing validity at each development milestone.

e. The functional analysis shall, if necessary, be updated for each development milestone.

f. The supplier shall carry out a software criticality analysis to assign criticality levels to software components.

g. On the basis of the results of the software criticality analysis, the supplier shall, to the extent possible without introducing undesirable software complexity, minimise the number of critical software components by appropriate software technical design.

Handling of critical modules

h. The supplier shall define and apply measures to assure the reliability of critical modules.

These measures may include (but need not be restricted to):

- use of software design or methods which 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 level 3 standard ECSS–Q–80–3);
- defensive programming techniques, such as input verification and consistency checks;
- prohibiting the use of language commands and features which are known to be unpredictable or difficult to program;
- use of formal design language for formal proof and/or automatic code generation;
- full inspection of source code;
- witnessed or independent testing;
- gathering and analysis of failure statistics.

Assurance activity:

i. The application of the chosen measures to critical modules shall be verified.

REQUIRED OUTPUT: *Definition of measures and verification activities in software product assurance plan*

Software of different criticality

j. The supplier shall ensure that failure of noncritical software, which is not subject to the above–stated assurance measures, will not cause failure of critical software.

This should be achieved by design measures such as separate hardware platforms, isolation of software processes, prohibition of shared memory, etc.

k. The measures taken and justification shall be documented in the design.

3.2.3 Software Configuration and Documentation Management

The requirements for software configuration management are described in the standard ECSS–M–40, and this clause contains requirements on product assurance of software configuration management which are complementary to the ECSS–M–40 requirements.

The configuration management implementation document should explicitly address the configuration management of software, with respect to all of the requirements of ECSS–M–40.

a. (RE) The software configuration management system shall allow any reference version to be re-generated from backups.

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

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

REQUIRED OUTPUT: *Software configuration file.*

Assurance of change control

The following activities are in addition to the configuration management audits required by ECSS–M–40. They apply to both development and maintenance phases.

d. The supplier shall ensure that all authorised changes are implemented according to the requirements of the software configuration management plan.

e. The supplier shall ensure that only appropriately authorised changes are made.

Software configuration management tool

The use of a computer-based configuration management tool is recommended. The following requirements apply to any computer-based tool that is used.

f. The configuration management tool shall be compatible with the customer's interface requirements to ensure effective delivery and continuity in post-acceptance configuration management.

g. (RE) The configuration management tool shall be identified in the configuration management implementation document.

REQUIRED OUTPUT: *Identification of the software configuration management tool*

Control of documents

h. The following documents shall be controlled (see ECSS–Q–20 clause 3.1):

- 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,
  * maintenance documentation.

Protection and marking

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

REQUIRED OUTPUT: *Identification and protection method or tool*

For all software products in operational use, the supplier should use a checksum-type identification key calculation and checking software on each executable binary or each file considered to be a supply (source, database).

The checksum value should be specified in the software configuration file.

The identification key should be used:

- prior to each delivery;
- at reception to check identification.

If the protection mechanism used is based on a supplier-specific tool, this tool should be supplied to the customer along with the delivered software products.

j. (RE) 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.

REQUIRED OUTPUT: *Labelling of media*

3.2.4 Process Metrics

This section deals with process metrics only – note that clause 4.1 deals with product metrics.

a. Process metrics shall be collected, stored, analysed, and reported on a regular basis.

This should be done by applying suitable quality models and procedures.

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

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

REQUIRED OUTPUT: *Details of metrics in software product assurance plan*

d. The following basic process metrics shall be used within the contractor's organisation:

- duration: how phases/tasks are being completed versus the planned schedule;
- effort: how much effort is consumed by the various phases/tasks compared to the plan.

e. The following basic process metrics shall be used within the contractor's organisation, and reported to the customer:

- problems detected during inspection;
- problems detected during integration and system testing and use.
- (See also software problem reporting described in sub–clause 2.3.6)

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

REQUIRED OUTPUT: *Metrics reports in software product assurance reports*

g. Metrics reports shall be submitted to each milestone from the start of design to the completion of acceptance.

REQUIRED OUTPUT: *Metrics reports at milestones*

3.2.5 Verification

Verification is the process of evaluating the products of a given phase to ensure correctness and consistency with respect to the products and standards provided as input to that phase.

Verification includes various techniques such as review, inspection, testing, walkthrough, cross-reading, desk-checking, and many types of analysis such as traceability analysis, formal proof, fault tree analysis, etc. (Note that the term "review" includes both joint reviews with the customer and internal reviews.)

a. (RE) Verification activities shall be carried out according to a comprehensive verification plan.

b. (RE) The plan shall identify all tools, facilities, training and skills required to carry out the verification activities.

c. (RE) The first version of the verification plan shall be provided at the first milestone.

REQUIRED OUTPUT: *Verification plan*

The verification plan may be part of a verification and validation plan. This would include plans for all testing (test designs).

General requirements

d. The outputs of each development phase shall be verified for conformance against the inputs to that phase, to demonstrate that they:
   - meet the relevant requirements;
   - conform to appropriate development standards;
   - contain or reference acceptance criteria for forwarding to subsequent phases;
   - identify those characteristics of the product that are crucial to its safe and proper functioning (e.g. operating margins of the computing resources or performances of operating systems on which the application is to run).

e. A summary of verification activities undertaken for each phase, and of their results, shall be included in regular software product assurance reports.

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

REQUIRED OUTPUT: *Verification reports and software problem reports*

Only outputs which have been subjected to planned verifications should be submitted to configuration management and accepted as inputs for subsequent phases.

Assurance activity:

g. The supplier shall:
   - ensure that the planned verification activities are adequate to ensure the products of each phase are compliant;
   - ensure that verification activities are performed according to the plan;
   - ensure that the planned verification activities include full verification of critical software (see sub–clause 3.2.2) at each stage of its development.

Inspections and Reviews

h. (RE) Inspection and review activities shall be described in the Software verification plan.

i. (RE) Each inspection shall be based on a written procedure.

j. (RE) The inspection procedures shall specify:
   - the inspected items;
   - the person in charge;
   - participants;
   - the means of inspection (tools, check list, etc...);
   - the nature of the report.

k. (RE) In addition to the contractually specified joint reviews, the supplier shall include a schedule for internal reviews as part of the verification process.

REQUIRED OUTPUT: *Inspection and review schedule as part of the verification plan*

l. Reviews shall be carried out according to defined criteria, by someone other than the author of the reviewed item.

m. Review records shall be kept which identify the reviewed item, the author, the reviewer, the review criteria and the finding of the review.

REQUIRED OUTPUT: *Review records*

Tracing

n. (RE) Traceability matrices documenting consistency shall be established covering:
   - system/user requirements to software requirements;
   - software requirements to design;
   - design to code;
   - integration tests to major components of the architecture;
   - software requirements to system test documentation;
   - system/user requirements to validation / acceptance documentation.

REQUIRED OUTPUT: *Traceability Matrices*

Assurance activity:

o. The traceability matrices shall be verified at the completion of each phase.

Independent Software Verification and Validation (ISVV)

Note that this requirement should only be applied 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 organisation.

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

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 should, independently of the development team, perform verification activities such as conducting reviews, inspections, testing, auditing, etc..

REQUIRED OUTPUT: *ISVV plan and ISVV report*

3.2.6 Methods and Tools

Proven methods and operational tools should be used for all phases of the development cycle, including requirements analysis, software specification, coding, validation and testing.

a. The choice of development methods and tools shall be justified by demonstrating that:
   - the development team has appropriate experience and/or training to apply them;
   - the tools and methods are appropriate for the functional and operational characteristics of the product;
   - the tools will be available (in an appropriate hardware environment) throughout the development and maintenance lifetime of the product.

This may be demonstrated through testing or documented assessment.

REQUIRED OUTPUT: *Justification included or referenced in the software product assurance plan*

b. The supplier shall set up a software development environment integrating the chosen development tools, and the tools associated with software configuration management, verification, product assurance and inspection activities.

REQUIRED OUTPUT: *Description of the software development environment*

Assurance activity:

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

Assurance activity:

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

3.2.7 Re–used Software

Re-used software includes software from previous developments which is intended to be used for the project development as it is or with adaptation. It also includes software supplied by the customer for use in the project development.

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

b. These analyses shall serve to refine or validate a priori choices and to define the associated actions required to meet quality requirements.

c. The choice of re-used software shall take into account:

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

Assurance activity:

d. The re-used software shall be analysed to check the following aspects of its condition 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 re-used component:
  * its validation level or operational behaviour,
  * its documentation status,
  * its quality status, i.e. residual nonconformances, complexity analyses, waivers, etc...

e. All the elements on which the decision to re-use software is based shall be recorded in a re-used software file.

f. This re-used file shall contain:

- a qualitative summary review of the re-used components and an assessment of the possible level of re-use;
- a description of the assumptions and the method of calculating the level of re-use;
- planned corrective actions regarding the re-used software.

g. All the associated corrective actions to be implemented shall be determined through the re-used software analysis.

REQUIRED OUTPUT: *Re-used software file submitted to the customer for acceptance*

3.3 Requirements Related to Individual Life-Cycle Phases

3.3.1 Software Requirements Specification

Requirements on the software may 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. The software development needs

to begin with analysis to fully and unambiguously define the software requirements on the basis of these inputs.

The supplier should develop these requirements in close co-operation with the customer and the supplier should obtain the customer's approval before entering the development stage. In some cases, the software requirements specification is provided by the customer. The requirements specification should be subject to documentation control and configuration management as part of the development documentation.

a. (RE) Software requirements shall be analysed before any software development is started and documented in a software requirements specification.

b. (RE) The software requirements specification shall be prepared and submitted for customer approval (at a defined milestone) before software development begins.

REQUIRED OUTPUT: *The software requirements specification*

c. When the software forms part of a system, the (functional) software requirements shall be derived from analyses at system level.

d. (RE) In addition to the functional requirements, these requirements shall include all aspects necessary to satisfy the customer's need.

These should include, but are not limited to, the following: performance, safety, reliability, quality, maintainability, configuration management, security, privacy, metrication, verification and validation.

e. (RE) The following interface requirements shall be fully specified, either directly or by reference (i.e. interface control document), in the software requirements specification:

- between the software product and other software products
- between the software product and hardware products
- interface requirements relating to the man/machine interface.

During the development of the software requirements specification, attention should be paid to the following:

- assignment of persons (on both sides) responsible for establishing the software requirements 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.

3.3.2 Design

The design and implementation activities are those which transform the customer's requirements specification into a software product. Because of the complexity of software products, these activities should be carried out in a disciplined manner, in order to produce a product according to specification rather than depending on the validation tests for assurance of quality.

The level of information disclosure to be provided to the customer should be mutually agreed to by the parties, as design and implementation processes are frequently proprietary to the supplier.

The design specification should be subject to documentation control and configuration management as part of the development documentation. The implementation process (i.e. coding, integration and testing) should not proceed until the consequences of all known deficiencies are satisfactorily resolved or the risk of proceeding otherwise is known.

a. (RE) A systematic design methodology appropriate to the type of software product being developed and suitable tools shall be used, and defined or referenced in the software product assurance plan.

REQUIRED OUTPUT: *Definition of methodology and tools*

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

REQUIRED OUTPUT: *Design standards*

Assurance activity:

c. Adherence to design standards shall be verified.

Assurance activity:

d. The complexity and modularity of the design shall be checked to ensure achievement of maintainability and reusability goals.

It is recommended that these checks be implemented in parallel to the design process so as to ensure that the results are easily taken as inputs by the designers.

e. The nature of the checks, the criteria, the tools used and the relationship to the design team shall be described in the software Product Assurance Plan.

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

REQUIRED OUTPUT: *: Description of checks in the software product assurance plan and results in software product assurance reports*

3.3.3 Coding

a. (RE) Standards such as programming rules, programming languages, consistent naming conventions, coding and adequate commentary rules shall be specified and observed.

b. The standards shall be designed to ensure consistency with the product quality requirements (see clause 4.2).

The coding standards should be appropriate to the software quality objectives (see clause 4.1).

c. A description of the tools to be used in implementing the standards shall be provided to the customer before coding activities start.

Assurance activity:

d. Coding standards shall be reviewed to ensure that they reflect product quality requirements and quality objectives.

REQUIRED OUTPUT: *Coding standards and description of tools for customer's acceptance*

e. (RE) If no high-level programming language is selected this choice shall be justified.

REQUIRED OUTPUT: *Document justifying suitability of the language.*

Assurance activity:

f. Complexity measures shall be performed on code in order to ensure achievement of maintainability and re–usability goals.

This measurement should be implemented in parallel with the coding process to ensure that the results are easily taken as inputs by the developers.

REQUIRED OUTPUT: *Description of complexity measures in the software product assurance plan and results in software product assurance reports*

As far as possible, automatic means to measure adherence of the code to the coding standards should be used.

This measurement should be performed in parallel with coding activities so as to ensure that the results could be easily taken as inputs by the developers.

g. 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 verification plan.

h. Synthesis of results obtained and corrective actions implemented shall be described in software product assurance reports.

REQUIRED OUTPUT: *: Description of measurements and synthesis in software product assurance reports*

Assurance activity:

i. For software for which long life maintenance occurs, the supplier shall review the design documentation to ensure that it contains the appropriate level of information for maintenance activities.

j. The code shall be taken under configuration control before formal testing is started.

3.3.4 Testing (Including Validation Tests)

Test planning

a. (RE) Testing shall be prepared, performed and evaluated in accordance with a defined strategy (test design).

b. (RE) The test strategy shall include:

- testing for software modules, integration, system testing and acceptance testing;
- the types of tests to be performed, e.g. functional, boundary, performance, and usability tests;
- test coverage goals.

c. The assurance activities for testing, e.g. witnessing tests or verifying records, shall be defined in the software product assurance plan and carried out to identified procedures.

REQUIRED OUTPUT: *Assurance activities for testing in the software product assurance plan*

d. (RE) Detailed test case specifications and procedures shall be prepared on the basis of the defined strategy.

REQUIRED OUTPUT: *Test documentation for customer's acceptance*

Assurance activity:

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

The document that describes the test plan may be an independent document or a part of another document, or may be composed of several documents. Test planning documents should complement or be combined with the verification plan.

Test execution

Assurance activity:

f. Test readiness reviews shall be held before the commencement of key test phases, as defined by the customer (e.g. validation phase, system test phase).

Note that software product assurance personnel may attend this review to carry out the assurance activities appropriate to this phase.

Assurance activity:

g. At each test phase, test coverage shall be checked with respect to the stated objectives.

As far as possible, this check should be made with an automatic tool to measure coverage obtained.

h. The measurements shall be performed throughout the test programme (unit, integration and system test).

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

i. The means and organisation to perform this task shall be described in the Software Product Assurance Plan.

j. The level of coverage obtained shall be described in the quality assessment reports.

REQUIRED OUTPUT: *Required output Collected data and analysis of the results in the software product assurance report*

Assurance activity:

k. The supplier shall ensure that anomalies and changes in test conditions are properly documented.

Assurance activity:

l. During the validation and acceptance tests, the supplier shall ensure that results and events are accurately and fully recorded in the test report.

REQUIRED OUTPUT: *Test reports and associated tracking elements*

m. (RE) Any discovered problems and their possible impacts on any other parts of the software shall be reported as problem reports and those responsible notified.

Assurance activity:

n. The supplier shall ensure that problem reports and subsequent actions are properly closed off.

Assurance activity:

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

Assurance activity:

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

Assurance activity:

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

Assurance activity:

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

s. The fact that tests have been successfully completed to plan shall be certified by the supplier.

REQUIRED OUTPUT: *Test certification*

t. (RE) Review boards shall be convened after the completion of key test phases.

Note that software product assurance personnel may attend this review to carry out the assurance activities appropriate to this phase.

Regression testing

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

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

REQUIRED OUTPUT: *Updated test documentation*

Validation

w. (RE) Validation tests shall cover all requirements of the software requirements specification.

x. (RE) Verification of the user documentation shall be part of the validation activities.

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

This may be achieved at the level of the whole system, or on a subsystem by subsystem basis.

z. Validation of embedded software for flight equipment shall include tests of the equipment model without “patching” the software under test.

Assurance activity:

aa. For software for which long life maintenance occurs, the supplier shall review the test documentation to ensure that it is up to date and organised to facilitate its re-use in the maintenance phase.

Independent Software Verification and Validation (ISVV)

ab. Independent software validation shall be performed for highly critical software. (See sub–clause 3.2.5 for guidance on ISVV.)

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

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

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

Where testing under field conditions occurs, the following concerns should be addressed:

- the features to be tested in the field environment;
- the specific responsibilities of the supplier and customer for carrying out and evaluating the test;
- restoration of the user environment (after test).

3.3.5 Delivery, Installation and Acceptance

When the supplier is ready to deliver the validated product, the customer should judge whether or not the product is acceptable according to previously agreed criteria and in a manner specified in the contract. Provisions should be made for verifying the correctness and completeness of copies of the software product delivered.

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

a. The planning and documentation of the installation shall cover as a minimum the schedule, access (security badges, passwords, escorts), availability of skilled personnel, availability and access to target environment/equipment, the need for testing as part of each installation, and formal approval of each installation upon completion.

REQUIRED OUTPUT: *Installation plan*

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

b. (RE) The supplier shall establish an acceptance test plan (design and test case specifications) specifying the intended acceptance tests which may be supplemented by the customer, if necessary, with tests suited to the target environment.

REQUIRED OUTPUT: *Acceptance test plan for acceptance by the customer*

The acceptance tests may be partly made up of tests used during previous test activities.

The acceptance test plan should take into account the requirement for operational demonstration, either as part of acceptance or after acceptance.

c. 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)
   - that the source and object code supplied correspond to each other
   - that all agreed changes have been implemented
   - that all nonconformances are either resolved or declared.

d. Acceptance shall include generation of the executable code from configuration managed source code components and its installation on the target environment.

e. Any discovered problems shall be documented in nonconformance reports.

f. 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 and the representative of the organisation charged with the maintenance of the software product.

g. 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).

REQUIRED OUTPUT: *Acceptance test report*

3.3.6 Operations and Maintenance

Operations

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

Quality of products should include parameters such as:

- error-free data;
- availability of data and permissible outages;
- permissible information degradation.

b. (RE) The supplier shall ensure that prior to the operations phase, the software has been demonstrated capable of meeting the operational requirements.

This demonstration may be part of the acceptance tests of the system.

This demonstration should be representative in terms of:

- hardware operating environment;
- situations to which the software is designed to be fault tolerant;
- system configuration;
- sequence of operations and phases;
- operator interventions.

c. (RE) The demonstration shall cover at least:
   - availability and maintainability of the host system (including re-boot after maintenance interventions, etc.);
   - safety features;
   - human/computer interface;
   - operating procedures;
   - ability to meet the mission product quality requirements.

REQUIRED OUTPUT: *Demonstration plan to be approved by the customer*

d. The quality assurance plan for system operations (see ECSS–Q–20 sub–clause 9.4.1) shall include consideration of software.

Maintenance

When maintenance of the software occurs after initial delivery and installation, the organisation charged with the maintenance of the software product should be identified early in the life cycle thus allowing a smooth transition into the operations and maintenance phase.

It may be necessary to establish an organisation, with representatives from both supplier and customer, to support the maintenance activities. Since activities in the maintenance stage cannot always be carried out on a scheduled basis, this organisation should be flexible enough to cope with the unexpected occurrence of problems. It may also be necessary to identify facilities and resources to be used for the maintenance activities.

e. (RE) The organisation responsible for maintenance shall establish and maintain plans and procedures for performing maintenance activities and support of the operation of the software product.

Assurance activity:

f. The maintenance organisation shall specify the assurance, verification and validation activities applicable to maintenance interventions.

REQUIRED OUTPUT: *Maintenance plans and procedures*

Assurance activity:

g. These plans and procedures shall be verified against specified requirements for maintenance of the software product.

h. (RE) As far as practical, software maintenance shall be performed using the same procedures, methods, tools and standards as used for the development.

i. (RE) All changes to the software product shall be documented in accordance with the procedures for document control and configuration management.

The maintenance plans and procedures may have to address corrective, evolutionary, improving, adaptive and preventive maintenance. The maintenance procedures should differentiate between “routine” and “emergency” maintenance activities.

The maintenance plans and procedures should include the following:

- scope of maintenance;
- identification of the initial status of the software product;
- support organisation;
- maintenance activities;
- maintenance records and reports.

All maintenance activities should be recorded in pre-defined formats and retained. Rules for the submission of maintenance reports should be established and agreed as part of the maintenance plan.

j. 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;
- organisation responsible for responding to requests for assistance or implementing the appropriate corrective actions;
- priorities that have been assigned to the corrective actions;
- results of the corrective actions;
- statistical data on failure occurrences and maintenance activities.

The record of the maintenance activities may be utilised for evaluation and enhancement of the software product and for improvement of the quality system itself.

4

Requirements on Product Quality

4.1 Product Quality Objectives and Metrication

Product quality objectives

Assurance activity:

4.1.1

The supplier shall define assurance activities to ensure that the product meets the quality objectives as specified in the contract.

Assurance activity:

4.1.2

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

Assurance activity:

4.1.3

Quality models shall be used to specify the required quality objectives.

This can be done by reference to a quality model such as MacCall or ISO 9126. The following subjects should be considered:

- portability especially for long–life software for ground segments;
- maintainability especially for long–life software fore ground segments;
- re–usability;
- correctness.

As far as possible, quantitative objectives or constraints should be expressed.

4.1.4

To achieve these objectives, the contractors shall apply rules on design, code and documentation and define a metrication programme to verify and prove that the objectives are reached.

Requirements on how these rules and metrics are to be set up are defined in the following sub–clauses of this clause.

REQUIRED OUTPUT: *Required output: Description of the metrics chosen, the measurement tools used and the planning of measurement*

Product metrics

4.1.5

The supplier shall define the relevant metrics, and the means to obtain them, to predict / assess / estimate the actual quality characteristics of the product for comparison with those required.

Guidance on metric set selection is provided in ECSS–Q–80–4.

4.1.6

These measurements shall be performed throughout the development and the results obtained used to define corrective actions when necessary.

4.1.7

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

4.1.8

The following basic products metrics shall be considered for use:

– size (design, code);

– complexity (design, code);

– fault density and/or failure intensity;

– test coverage;

– number of failures.

REQUIRED OUTPUT: *Products metrics specification and justification in software product assurance plan*

4.1.9

Metrics chosen shall be collected, stored, analysed and reported on a regular basis.

4.1.10

The metrics shall be analysed against target values or quality objectives and remedial actions taken, if necessary, to ensure achievement of quality goals..

REQUIRED OUTPUT: *Report of the analysis and remedial actions in the software product assurance report*

Numerical accuracy

4.1.11

For software in which numerical accuracy has a functional importance (e.g. for an attitude and orbit control subsystem) specific rules on design and code shall be defined to ensure that the appropriate level of accuracy will be obtained.

4.1.12

Numerical errors shall be estimated and checked from the design phase to the end of development.

REQUIRED OUTPUT: *Result of studies on numerical errors presented at each milestone of the software development.*

Analysis of software behaviour

4.1.13

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

4.1.14

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

– descriptive statistics (e.g. the number of modules at each level of complexity);

– trend analysis.

4.1.15

These data shall be used to determine when further actions are necessary to improve the software.

REQUIRED OUTPUT: *Records of data collection, analysis and results and actions for improvement*

4.2 Product Quality Requirements

4.2.1 Software Requirements Specification

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

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

c. Requirements shall be stated sufficiently precisely to allow verification and validation, where appropriate, in measurable terms (e.g. with the use of metrics).

d. (RE) For each requirement the method for verification/validation shall be specified.

e. (RE) Each requirements shall have a unique identifier to support the verification and validation activities.

4.2.2 Design and Related Documentation

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

b. (RE) The design description shall as a minimum cover hierarchy, dependency and interfaces for the software components.

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

d. (RE) The design description shall document the process, data and control aspects of the product.

e. If the software is intended to be re–used in a future development, the supplier shall take this objective into account in the way the documentation is organised, by ensuring easy access to the following information for each re–usable function:

- description of the function from the external point of view;
- external interfaces;

- implementation;
- validation.

f. For software with a long planned lifetime over which it is planned to be maintained, the design of the software shall minimise dependency on the operating system and the hardware, in order to aid portability.

REQUIRED OUTPUT: *Product quality requirements reflected in coding and design standards*

4.2.3 Code

Further requirements on code beyond those specified in sub–clause 3.3.3 (e.g. a customer-specific coding standard, permitted language subsets) should be inserted here during tailoring.

4.3 Supporting Documentation

4.3.1 Test Documentation

a. Detailed test planning documentation (designs, case specifications, procedures, expected results, etc) shall be consistent with the strategy defined in the test plan (see sub–clause 3.3.4).

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

c. (RE) The criteria for the completion of the test and any contingency steps shall be specified.

d. (RE) Test cases, procedures and expected results shall be specified.

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

REQUIRED OUTPUT: *Detailed test planning documentation*

f. (RE) Test results shall be recorded and documented in reports as defined in the applicable project standards.

REQUIRED OUTPUT: *Test reports*

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

REQUIRED OUTPUT: *Verification reports*

4.3.2 Reports and Analysis

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

4.4 Purchased Software and Standard Hardware

Purchased software

4.4.1

The choice of purchased software (including commercial off the shelf COTS software) shall take into account constraints associated with development and future use.

The following aspects should be considered:

- the assessment of the product with respect to quality requirements and objectives;
- the available support documentation;
- the acceptance and warranty conditions;

- the conditions of installation, preparation, training and use;
- the maintenance conditions, including possibility of evolutions (portability by example);
- copyright constraints.

4.4.2

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

b. The software component list shall include specification of, at least:
   - ordering criteria (versions, options, extensions, etc.);
   - 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.

REQUIRED OUTPUT: *Software component list at first milestone*

4.4.3

All the purchased software which will be used in the operational system shall be identified and registered by configuration management.

Assurance activity:

4.4.4

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

4.4.5

A receiving inspection report (including identification of detected problems) shall be generated.

It could be necessary to specify supplementary specific tests associated with the service environment.

Standard hardware

4.4.6

The subcontracting and procurement of hardware shall be carried out according to the requirements of ECSS–Q–20 clause 5.

4.4.7

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

4.4.8

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

– performance;
– availability;
– possibility of changes;
– compatibility;
– maintenance and durability.

4.4.9

The ease of maintenance and maintenance policy proposed by the manufacturer, including any substitutions, shall be comparable with the specified service life and operational constraints.

4.4.10

Any constraints regarding exportability shall also be taken into account.

4.4.11

The development computer equipment shall be selected according to the following criteria:

– 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.

4.4.12

Descriptions of the choices of both ground and development computer equipment shall be described or referenced in the software product assurance plan.

REQUIRED OUTPUT: *Descriptions of choices of ground and development equipment*

4.5 Re-used Software

Re-used software includes any software developed outside the contract to which this standard is applicable, by:

- the supplier;
- another supplier;
- the customer.

a. Re-used software shall be subject to the same requirements as purchased software, as set out in clause 4.4 of this standard.

4.6 Firmware

4.6.1

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

REQUIRED OUTPUT: *Procedures described or referenced in the software product assurance plan*

4.6.2

The supplier shall document the tests required to validate the software embedded into the firmware device.

4.6.3

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

Assurance activity:

4.6.4

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