



# SOFTWARE DEVELOPMENT STANDARD FOR SPACECRAFT

Mar. 23, 2016

Japan Aerospace Exploration Agency

This is an English translation of JERG-2-610A. Whenever there is anything ambiguous in this document, the original document (the Japanese version) shall be used to clarify the intent of the requirement.

Disclaimer

The information contained herein is for general informational purposes only. JAXA makes no warranty, express or implied, including as to the accuracy, usefulness or timeliness of any information herein. JAXA will not be liable for any losses relating to the use of the information.

Published by  
Japan Aerospace Exploration Agency  
Safety and Mission Assurance Department  
2-1-1 Sengen Tsukuba-shi, Ibaraki 305-8505, Japan

## Table of Contents

<b>1</b>	<b>Scope .....</b>	<b>1</b>
<b>2</b>	<b>References .....</b>	<b>1</b>
<b>2.1</b>	<b>Document for compliance .....</b>	<b>1</b>
<b>2.2</b>	<b>Informative references .....</b>	<b>1</b>
<b>3</b>	<b>Terms, definitions and abbreviated terms .....</b>	<b>1</b>
<b>3.1</b>	<b>Terms redefined in this standard.....</b>	<b>1</b>
<b>3.2</b>	<b>Terms newly defined in this standard .....</b>	<b>2</b>
<b>3.3</b>	<b>Terms defined in JERG-0-049A .....</b>	<b>2</b>
<b>4</b>	<b>Organization of this standard .....</b>	<b>6</b>
<b>5</b>	<b>Primary life cycle processes.....</b>	<b>8</b>
<b>5.1</b>	<b>Not used.....</b>	<b>8</b>
<b>5.2</b>	<b>Not used.....</b>	<b>8</b>
<b>5.3</b>	<b>Development process .....</b>	<b>8</b>
<b>5.3.1</b>	<b>Process implementation .....</b>	<b>9</b>
<b>5.3.2</b>	<b>Items to be applied to all processes.....</b>	<b>9</b>
<b>5.3.3</b>	<b>Computer system requirements analysis.....</b>	<b>10</b>
<b>5.3.4</b>	<b>Computer system architectural design.....</b>	<b>10</b>
<b>5.3.5</b>	<b>Software requirements analysis .....</b>	<b>11</b>
<b>5.3.6</b>	<b>Software design .....</b>	<b>13</b>
<b>5.3.7</b>	<b>Not used.....</b>	<b>14</b>
<b>5.3.8</b>	<b>Software coding and testing .....</b>	<b>14</b>
<b>5.3.9</b>	<b>Not used.....</b>	<b>16</b>
<b>5.3.10</b>	<b>Software integration.....</b>	<b>16</b>
<b>5.3.11</b>	<b>Software integration test .....</b>	<b>16</b>
<b>5.3.12</b>	<b>Software installation into target platforms (embedding) .....</b>	<b>18</b>
<b>5.3.13</b>	<b>Computer system integration and computer system integration test .....</b>	<b>19</b>
<b>5.3.14</b>	<b>Supply and introduction of software product .....</b>	<b>21</b>
<b>5.3.15</b>	<b>Software acceptance.....</b>	<b>21</b>
<b>5.4</b>	<b>Operational process .....</b>	<b>21</b>
<b>5.4.1</b>	<b>Process implementation .....</b>	<b>21</b>
<b>5.4.2</b>	<b>Operational testing .....</b>	<b>22</b>
<b>5.4.3</b>	<b>Operation of software or computer system.....</b>	<b>22</b>
<b>5.4.4</b>	<b>Operator and user support.....</b>	<b>22</b>
<b>5.5</b>	<b>Maintenance process .....</b>	<b>22</b>
<b>5.5.1</b>	<b>Process implementation .....</b>	<b>23</b>
<b>5.5.2</b>	<b>Problem identification and modification analysis .....</b>	<b>23</b>
<b>5.5.3</b>	<b>Modification implementation.....</b>	<b>23</b>
<b>5.5.4</b>	<b>Software reprogramming.....</b>	<b>23</b>

5.5.5	Migration.....	24
5.5.6	Software retirement.....	24
6	Supporting life cycle process .....	25
6.1	Documentation process.....	25
6.1.1	Process implementation .....	25
6.1.2	Development .....	25
6.1.3	Document production .....	26
6.1.4	Document maintenance .....	26
6.2	Configuration management process.....	26
6.2.1	Process implementation .....	26
6.2.2	Configuration identification.....	26
6.2.3	Configuration control.....	27
6.2.4	Configuration management accounting and status report.....	28
6.2.5	Configuration evaluation .....	28
6.2.6	Release management.....	28
6.3	Quality assurance process.....	28
6.3.1	Process implementation .....	28
6.3.2	Implementation of quality assurance activities .....	29
6.3.3	Products quality assurance.....	29
6.3.4	Process assurance .....	29
6.3.5	Assurance of quality system.....	30
6.3.6	Perusal by related parties.....	30
6.4	Verification process .....	30
6.4.1	Process implementation .....	31
6.4.2	Verification .....	31
6.5	Validation process.....	33
6.5.1	Process implementation .....	33
6.5.2	Validation.....	33
6.6	Joint review process .....	33
6.6.1	Process implementation .....	33
6.6.2	Software development management reviews .....	34
6.6.3	Technical reviews .....	34
6.7	Assessment process.....	35
6.7.1	Process implementation .....	35
6.7.2	Assessment implementation.....	36
6.8	Problem resolution process.....	36
6.8.1	Process implementation .....	36
6.8.2	Problem resolution .....	36
6.8.3	Problem trend analysis .....	38

**6.8.4 Corrective and preventive action.....38**

## 1 Scope

This standard applies to the activities relating to a computer system (each subsystem, component, mission sensor system, and so on) boarded on spacecraft and the development, operation and maintenance of software, and also necessary system development activities and related support activities. When this standard is applied, embodying and tailoring may be performed in accordance with the characteristics of specific projects and other factors.

## 2 References

### 2.1 Document for compliance

- (1) JERG-0-049A Software Development Standard

### 2.2 Informative references

- (1) ISO/IEC 12207:1995 Information technology - Software life cycle processes
- (2) JIS X0160:1996 Software life cycle processes
- (3) ISO/IEC 14764:2006 Software Life Cycle Processes - Maintenance
- (4) JIS X0161:2008 Software Engineering - Software life cycle processes - Maintenance
- (5) ISO/IEC 15504-2:2003 Information technology - Process assessment - Part 2: Performing an assessment
- (6) Software Life Cycle Processes- Japan Common Frame 2007 (Japanese) (Copyright IPA/SEC 2009)

## 3 Terms, definitions and abbreviated terms

The definitions of terms and abbreviated terms used in this standard shall basically correspond to JERG-0-049A. Terms redefined into the details and newly defined are shown below.

### 3.1 Terms redefined in this standard

Term	Definition
Operation	Operation is an action to carry out missions for the achievement of a purpose by means of an appropriate computer system. An operation utilizes the computer system from beginning to end, and includes monitoring and file maintenance functions, and so on.
Verification	<p>Verification is a process. It uses objective evidence to confirm that specified requirements have been met. Whenever specified requirements have been met, a verified status is achieved. [JERG-0-049A, ISO9000].</p> <p>All activities to verify that the software satisfies the requirements and design specifications. Beside tests as the verification procedure, these include simulation, walk-through, document review such as source-code review, inspection, demonstration, and so on.</p>

### 3.2 Terms newly defined in this standard

Term	Definition
SOOH	Spacecraft Orbital Operations Handbook: SOOH SOOH provides the necessary information for the operation of the spacecraft. It is the source to develop SOP, to set the database for tracking and control, and during actual operations, it may be referred to confirm operational procedures, planning countermeasures for anomalies, and planning and setting new operational plans and procedures. (by JERG-2-012 Guidelines for Spacecraft Orbital Operation Handbook (SOOH))
SOP	Spacecraft Operations Procedure: SOP Usually it is prepared for each independent functional unit, to properly utilize the functions of the tracking and control system enough, and in the operation phase, it is used by a single unit or by combined multiple units. (by JERG-2-012 Guidelines for Spacecraft Orbital Operation Handbook (SOOH))

### 3.3 Terms defined in JERG-0-049A

Term	Definition
Acceptance Inspection and Acceptance Testing	Acceptance inspection and acceptance testing are actions to evaluate the compliance with requirements at the time of acquiring software products. Inspection is an action to confirm the compliance of a product, in accordance with evaluation criteria based on either requirements specifications or a predetermined value, by visually checking quantities and test results. Test comprises analysis, evaluation and checking of the functionality and capabilities of the software in order to obtain evaluation results and data required for use in inspection.
Activity	Activity is a component of a process and a set of strong correlative tasks.
Assessment	Assessment is a process to evaluate the strengths and the weaknesses of a subject process and to identify opportunities to improve the process for some predetermined purpose.
Computer system	Computer system is the entire system consisting of sets of software, platforms, and hardware, including the platform and hardware that are able to execute the targeted software for development. Although the definition of what a computer system contains is arbitrary, the definition shall be unique for software products that are subjected for development. The definition of what is contained in a computer system may be determined by the format which is defined at the boundary where it is not included in a computer system, or by the format which is defined as items included inside the boundary. In addition, even if the format is defined within the boundary, it is assumed that the definition of items inside the boundary shall be included as part of the computer system design. As a computer system may be defined in various ways, from a one-chip microcomputer to multiple general-purpose computers connected to a network.

Term	Definition
Configuration management	<p>Configuration management is an action to define the configuration items, i.e. computer systems or projects, to record changes of content and to manage such aspects as their storage, handling, and distribution.</p> <p>If the software consists of multiple modules, then not only must the software version be managed, but the version of each software module must also be managed. In addition, configuration management items require software consistent modules, requirements specifications, operation manuals, and so on.</p>
COTS	COTS is the abbreviation of Commercial Off-The-Shelf. It has already been developed and is available in the commercial marketplace.
Identifier	<p>Identifier is a short line of numbers, letters, and symbols, which is appended to each item of output and input to enable each item to be identified and classified by item type. Identifiers shall consist of not only a set of numbers, but also a set of letters and symbols. In addition, it is not always necessary identifiers be serial numbers.</p> <p>Appending a unique identifier to each item is convenient for requirements management and for traceability.</p>
Independent Verification and Validation: IV&V	IV&V are the verification and the validation that are performed by the organization independent of the software development organization. With regard to independence, financial, technical and management viewpoints shall be considered.
Input	Input is information needed to implement an activity.
Integrity	<p>Integrity is defined as the following properties in this standard:</p> <p>(1) Software component is complete with no deficiencies.</p> <p>(2) Software component is at an appropriate version.</p>
Non-functional requirements	Non-functional requirements are all requirements except functional requirements, such as performance, safety, and reliability.
Output	Output is information that is transformed from input by performing activities.
Process	Process is a set of interrelated or interacting activities to transform input to output.
Project	Project is a time-limited endeavor to be implemented by means of specified resources and a temporary organization, with the purpose of fulfilling the project's mission.
Requirement	Requirement is one of a set of functions and performance targets requested for computer system or software and they may be also included such as not embodied and not detailed enough, or ambiguity in expression and vague expectations.



Term	Definition
Requirements specification	<p>Requirements specification is defined as the description of functions and performance required for computer system or software, embodied and specifically defined, and which also consider feasibility.</p> <p>In principle, the specified requirements specifications shall be verifiable as both feasible and mutually consistent.</p> <p>However, on the characteristic of adopted development process and required functions and performance, if the requirements specifications representative format is not a feasible verification format, the verification of feasibility of the requirements specification shall be complemented by the following methods:</p> <p>(1) It shall include the planning of agreement procedure with computer system users that the requirements specifications are satisfactory, and software verification plan shall include the planning of agreement procedure.</p> <p>(2) It shall include the test specifications enough to verify the requirements in the verification plan.</p> <p>In principle, any restrictions, laws, rules, and a project policy shall be included in the requirements specifications.</p>
Risk	<p>Risk is defined as the degree of danger attaching to a system's safety and surrounding projects. It includes assessment of undesirable outcomes which may occur as a result.</p>
Software	<p>Software is a set of computer system configuration items comprising commands and data, which are executed or processed by a CPU to fulfill functions and capabilities defined in the software requirements specifications. If it is a set of commands and data which are implemented or managed by a CPU, it is categorized as software, and the software development process standards apply to it. However, for the driver of firmware, OS, and middleware, appropriate development processes are applied, in accordance with the characteristics and therefore may be removed from the software development process standard. For example, hardware and its integrated driver development shall be considered as such a case.</p>
Software life cycle	<p>Software life cycle is the period from the beginning of the requirements analysis phase until the termination of use of the software.</p>
Software products	<p>Software products are the set of software, source code, and related documentation.</p>
Software test specifications	<p>Software test specifications are defined as the descriptions of test conditions and expected results, expressed unambiguously, to prove that software meets the requirements specifications.</p> <p>If the requirements specifications are represented in a verifiable format, they may be treated as appropriate software test specifications.</p>
Software under test	<p>Software under test is software that is being subjected to testing and inspection.</p>
Software user's manual	<p>Software user's manual is the set of information a user needs in order to use software. It includes operation unit manuals, computer system operation manuals, and work operation manuals.</p>
Software verification plan	<p>Software verification plan is a documentation of the scope, content, method, environment (such as test equipment) and schedule pertaining to the verification of software development.</p> <p>A validation plan may be included.</p>

Term	Definition
Tailoring	Tailoring is defined as the activity to change processes defined in this standard in order to meet the project's particular characteristics and to establish an appropriate framework for each system development project.
Tasks	Tasks are components of activities corresponding to each stage of work.
Test plannability	Test plannability is defined as the aspect of test specification descriptions that indicates the possibility of testing and planning using the appropriate development phases and test environment for target test items.
The stability (maturity) of software requirements specifications	The stability (maturity) of software requirements specification is defined as the index which shows the possibility of specification change is small because of the software requirements specification be extracted and analyzed sufficiently. The definition of the index, and how it is evaluated, are arbitrary. Generally, provision for essential or refined changes to software requirements specifications affect process cost, delivery date and quality which let the software requirements specifications inputs. It is hoped that the index which is used to evaluate this shall be selected based upon the stability and maturity of software requirements specifications.
Traceability	Traceability is defined as the property which shows the correspondence with higher level documents.
Validation	Validation is a process. It uses objective evidence to confirm that the requirements which define an intended use or application have been met. Whenever all requirements have been met, a validated status is achieved. The process of validation can be carried out under realistic use conditions or within a simulated use environment. [ISO9000]

#### 4 Organization of this standard

This standard categorizes the software life cycle into three primary life cycle processes and eight supporting life cycle processes, and defines these processes. The definition of these processes is shown in Figure 4.1 and Table 4.1.

The primary life cycle processes are processes in a software life cycle directly related to the development of target software, and consist of processes implemented during development, operation, or maintenance. The supporting life cycle processes are processes that indirectly affect the software life cycle process, with reference specifically to the development of object software, and act to support a primary life cycle process and are called by other processes, as necessary.

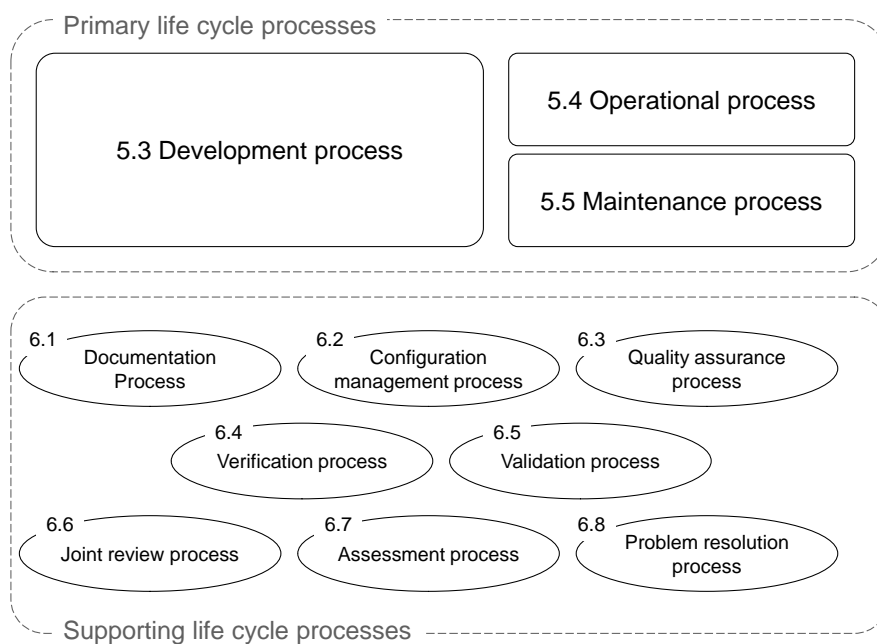


Figure 4.1 - Structure of the standard

The processes may be implemented in an order different from their clause number order in this document. Also, note that there will be cases where identical activities are described in multiple processes. For example, activities relating to software verification may be described as part of the development process, which is a primary life cycle process. These are also activities relating to the verification process, which is a supporting life cycle process.

The classification of processes is better understood as a classification according to the different viewpoints. This standard aims to define processes from various viewpoints, so that all of the required contents is covered completely (duplication is allowed). Therefore, this standard assumes that it will be applied after appropriate concretizing and tailoring have been performed regarding the relevant processes.

Table 4.1 Process list

Process		Description
Primary life cycle processes	5.1 Not used	
	5.2 Not used	
	5.3 Development	Process to be implemented from the viewpoint of development. Requirements analysis, design, coding and testing, installation on target platforms (embedding), supply, introduction, acceptance, and so on.
	5.4 Operation	Process to be implemented from the viewpoint of operation. Operation in this standard is used for spacecraft operation support by spacecraft operator and user,
	5.5 Maintenance	Process to be implemented from the viewpoint of maintenance. Drafting plans and rules for maintenance, problem identification, modification, retirement, and so on.
Supporting life cycle processes	6.1 Documentation	Process regarding the record of the outcomes of individual processes.
	6.2 Configuration management	Process regarding the management of software and documents.
	6.3 Quality assurance	Process regarding the confirmation that the process meets the requirements of this standard and processes are managed according to the plan.
	6.4 Verification	Process regarding the confirmation that specified requirements have been fulfilled, based on the provision of objective evidence.
	6.5 Validation	Process regarding the confirmation that the requirements for a specific intended use or application have been fulfilled, based on the provision of objective evidence.
	6.6 Joint review	Process regarding a joint review means it is conducted by multiple personnel with different viewpoints.
	6.7 Assessment	Process regarding the confirmation of the executing status and identifying the items which need to be improved.
	6.8 Problem resolution	Process regarding resolving problems which occur during implementation.

## **5 Primary life cycle processes**

This clause defines the following primary life cycle processes:

- (1) Not used
- (2) Not used
- (3) Development process
- (4) Operation process
- (5) Maintenance process

### **5.1 Not used**

### **5.2 Not used**

### **5.3 Development process**

The development process is a collective process of defined activities, inputs, and outputs comprising the following processes:

- (1) computer system requirements analysis process;
- (2) computer system architectural design process;
- (3) software requirements analysis process;
- (4) software design process;
- (5) software coding and testing process;
- (6) software integration process;
- (7) software integration test process;
- (8) software installation into target platforms (embedding) process;
- (9) computer system integration and computer system integration test process;
- (10) supply and introduction of software product(\*1) process;
- (11) software acceptance(\*2) process.

These activities may be implemented in a different order from what is described in this document. However, the overall configuration of the processes shall be defined as well as the management method of the entire process, so that appropriate process management is performed.

\*1: In this standard, supply and introduction of the software product process is not applied.

\*2: In this standard, the software acceptance process is not applied.

### 5.3.1 Process implementation

When software development is started, activities that meet the following requirements shall be performed:

- (1) The software development plan including the following information shall be established to cover:
  - (a) Scope of computer (\*1) system
  - (b) Identification of target software
  - (c) Definition of the identification of software development processes and their relationship
  - (d) Definition of software development processes and activities (\*2)
  - (e) Activities in each individual development process and their implementation management plan
  - (f) Review plan
  - (g) Preparation of development related documents' structure, and relationships of input and output in each development process
  - (h) Documentation plan, including development department and schedule
  - (i) Appropriate work allocations and methods for work plans and progress management for each work including initial development schedule
  - (j) Environment to be used for software development and verification (simulator, real hardware, test environment, source code analysis tool, and so on)
  - (k) Management plan for COTS, reused software and development products by external entities (ordering products to a contract company and so on). The following shall be included:
    - (i) Identification of COTS, reused software items, and products developed by external entities
    - (ii) Definition of the quality assurance process regarding COTS, reused software items, and products developed by external entities
  - (l) Evaluation plan of appropriateness with computer system

\*1: Each subsystem, component, mission sensor system, and so on of spacecraft may be considered.

\*2: It is desirable to take account of the software life cycle. Activities related to computer system integrated software may be included, as necessary.

- (2) Software development plan shall be documented.

#### 5.3.1.1 Output

- (1) Software development plan

### 5.3.2 Items to be applied to all processes

Activities that meet the following requirements shall be performed throughout the entire development process:

- (1) The software development plan shall be updated and managed in accordance with the development status.
- (2) The progress of software development shall be monitored. It shall be reported to

administrators as necessary.

#### **5.3.2.1 Input**

- (1) Software development plan

#### **5.3.2.2 Output**

- (1) Software development plan (updated)
- (2) Software development progress report

### **5.3.3 Computer system requirements analysis**

#### **5.3.3.1 Activity**

Activities that meet the following requirements shall be performed with respect to the computer system requirements analysis:

- (1) Requirements extraction

The requirements for computer system to be developed, the operational concept shall be analyzed, and operational scenarios shall be documented.

- (2) Requirements specification development

Feasibility and consistency shall be confirmed, based on the operational scenarios, and the requirements specifications for computer system shall be defined.

The rationale for the requirements specifications for the computer system shall be clarified, and the traceability of higher level requirements for the computer system shall be evaluated.

#### **5.3.3.2 Input**

- (1) Requirements for the computer system
- (2) Operational concept

#### **5.3.3.3 Output**

- (1) Operational scenarios
- (2) Requirements specifications for the computer system
- (3) Evaluation results of the traceability of the requirements specifications and requirements for the computer system

### **5.3.4 Computer system architectural design**

#### **5.3.4.1 Activity**

Activities that meet the following requirements shall be performed for the computer system architectural design:

- (1) Computer system architecture shall be designed based on the requirements specifications for the computer system and the operational scenarios. Configuration items and their various categories (hardware, firmware, software, and operational) shall be clarified.
- (2) Requirements pertaining to the requirements specifications for the computer system shall be allocated among the individual configuration items of the system.
- (3) Feasibility of software items in fulfilling their allocated requirements shall be evaluated.
- (4) Rationale for the design and preconditions (e.g. operational assumptions) for the

computer system architectural design specifications shall be identified, and an appropriate evaluation shall be performed.

- (5) Traceability of the computer system architectural design specifications relative to higher level requirements, such as the requirements specifications for the computer system, shall be evaluated.
- (6) Interface requirements for the software shall be extracted.

#### **5.3.4.2 Input**

- (1) Operational scenarios
- (2) Requirements specifications for the computer system

#### **5.3.4.3 Output**

- (1) Computer system architectural design specifications
- (2) Software Requirements
- (3) Interface requirements, including requirements for telemetry and command
- (4) Evaluation result of traceability relative to computer system architectural design specifications and requirements specifications for computer system

### **5.3.5 Software requirements analysis**

#### **5.3.5.1 Activity**

The following activities shall be performed for the software requirements analysis:

- (1) Software requirements specifications shall be developed, based upon the analysis of the computer system architectural design specifications, interface requirements, and software requirements including non-functional requirements.
- (2) Identifiers shall be included in the individual software requirements specifications.
- (3) Specifications for data and databases to be handled by the software shall be included in the software requirements specifications.
- (4) Specifications for failure detection and handling functions shall be included in the software requirements specifications.
- (5) Interface requirements shall be analyzed, and interface specifications shall then be developed. Agreement with the relevant parties regarding the interface specifications shall be made based on a common understanding and interpretation of the contents.
- (6) Traceability and consistency of the software requirements specifications relative to the computer system architectural design specifications and interface requirements shall be analyzed and documented.
- (7) Rationale of individual requirements of the software requirements specifications shall be clarified, and their feasibility shall be evaluated.
- (8) If COTS or reused software is used, compliance with the software requirements specifications and its applicability with the computer system architectural design specifications shall be analyzed.
- (9) Operational assumptions and constraints regarding software requirements specifications shall be extracted.
- (10) Verifiability of the individual requirements of the software requirements and interface specifications shall be evaluated, and a software verification plan, including the



validation method, shall be established.

- (11) Software verification coverage pertaining to software function, performance, and operational scenarios in the verification plan shall be evaluated, and test plannability regarding the software requirements specifications and interface specifications shall be evaluated.
- (12) With regard to the software verification plan, whether the test is affected by the behavioral difference between the test environment and the real hardware, or whether verification is performed by review, analysis and so on, without testing, the evaluation that shows the adequate identification and verification methods shall be included.

#### **5.3.5.2 Measurement**

In terms of evaluating stability and quality levels of software requirements specifications, the following measurement shall be performed for software requirements analysis:

- (1) The definition of the data to be collected for evaluating the stability (maturity) of the software requirements specifications and their evaluation methods shall be defined.
- (2) Collection and evaluation of data defined in (1) above shall be planned.
- (3) Collection and evaluation of data defined in (1) above shall be performed, and the results recorded.

#### **5.3.5.3 Input**

- (1) Computer system architectural design specifications
- (2) Software requirements
- (3) Interface requirements
- (4) Operational scenarios

#### **5.3.5.4 Output**

- (1) Software requirements specifications
- (2) Interface specifications
- (3) Software requirements specification traceability and consistency evaluation results
- (4) Software requirements specification rationale and their feasibility evaluation results
- (5) Applicability evaluation results of COTS or reused software
- (6) Operational assumptions and constraints
- (7) Software verification plan, including validation plan
- (8) Verification completeness and test plannability for the software verification plan evaluation results
- (9) Software requirements stability (maturity) evaluation results

#### **5.3.5.5 Review**

For each output, a software requirements review shall be performed. Items to be reviewed shall be chosen based on 5.3.5.4 and shall be defined in a plan document such as the software development plan document. Appropriate follow-up of action items shall be performed in accordance with due dates, follow-up status, degree of influence, and so on. If a review is performed, it shall be documented in a technical review record after its completion. In addition, quantitative data such as the reviewers' positions, their review time, the number of questions, and their comments shall be recorded, and the quality of the review shall be evaluated.

### 5.3.6 Software design

In this standard, software architectural design and detailed design are not separated especially. However, in an actual development, the software architectural design and the detailed design phases may be separated as necessary.

#### 5.3.6.1 Activity

Activities that meet the following requirements shall be performed for the software design:

##### **Software architectural design**

- (1) Functional decomposition and module partitioning shall be performed based on the software requirements specifications and the relationships between the modules comprising the functions shall be clarified, so that an appropriate software architectural design is performed.
- (2) Software architectural design shall include the design and distribution of non-functional requirements (processing time requirements, requirements of resources such as memory, and so on) and shall be defined as software requirements specifications.
- (3) Interface specifications shall be detailed in accordance with the decomposition of the software functions and modules. Agreement with the relevant parties as to the interface specifications shall be arrived at based on a common understanding and common interpretation of the contents.

##### **Software detailed design**

- (4) Each individual module shall be designed in accordance with the decomposition of functions and modules, and a software detailed design shall be performed.
- (5) Interface specifications shall be detailed in accordance with the design of the module. Agreement with the relevant parties on the interface specifications shall be made based on a common understanding and interpretation of the contents.

##### **Common to software architectural and detailed designs**

- (6) Traceability and consistency of the software design with the software requirements specifications, interface specifications, and with the necessary related documents shall be analyzed and documented.
- (7) As necessary, with regard to the individual software design, the design rationale shall be clarified and its feasibility evaluated.
- (8) If COTS or reused software is used, its appropriateness with the software design shall be analyzed.
- (9) Operational assumptions and constraints regarding the software design shall be identified.
- (10) Software test plans and specifications shall be established in accordance with the software verification plan.
  - (a) For software test specifications, the following shall be considered:
    - (i) Operational scenarios
    - (ii) Interface specifications
    - (iii) Maximum loads for assumed scenarios
    - (iv) Coverage for software requirements specifications and software design specifications
    - (v) Anomalies, such as exceptions and failures

- (vi) Appropriateness with COTS or reused software items with computer system
- (11) If new operational assumptions and constraints arise or are identified, they shall be updated.

#### **5.3.6.2 Measurement**

The following measurement shall be performed for software design in order to evaluate the progress risk of the software design:

- (1) Definition of the data to be collected and of the evaluation methods for progress management and risk evaluation of software design shall be defined.
- (2) Collection and evaluation of data defined in (1) above shall be planned.
- (3) Collection and evaluation of data defined in (1) above shall be performed, and the results recorded.

#### **5.3.6.3 Input**

- (1) Software requirements specifications
- (2) Interface specifications
- (3) Operational assumptions and constraints
- (4) Applicability evaluation results of COTS or reused software
- (5) Software verification plan

#### **5.3.6.4 Output**

- (1) Software architectural and detailed design specifications
- (2) Interface specifications (updated)
- (3) Software design traceability and consistency evaluation results
- (4) Software design rationales and their feasibility evaluation results
- (5) Applicability evaluation results of COTS or reused software (updated)
- (6) Operational assumptions and constraints (updated)
- (7) Software test plan
- (8) Software test specifications

#### **5.3.6.5 Review**

For each output, a software design review shall be performed. Items to be reviewed shall be chosen based on 5.3.6.4 and shall be defined in a plan document such as the software development plan document. Appropriate follow-up on action items shall be performed in accordance with due dates, follow-up status, degree of influence, and so on. If a review is performed, it shall be documented in a technical review record after its completion. In addition, quantitative data such as the reviewers' positions, their review time, the number of questions, and their comments shall be recorded, and the quality of the review shall be evaluated.

#### **5.3.7 Not used**

#### **5.3.8 Software coding and testing**

##### **5.3.8.1 Activity**

Activities that meet the following shall be performed for the software coding and testing:

- (1) Preparation

- (a) Rules for coding shall be defined as a coding standard.
  - (b) Definitive implementation guidelines for error handling shall be considered.
  - (c) For unit testing, criteria shall be established to determine pass or failure per software design specifications.
  - (d) For unit testing, criteria for the test coverage for source code shall be established. The branch of source code shall be well covered at least.
- (2) Coding and unit testing
- (a) Source code shall be developed based on software design specifications and interface specifications.
  - (b) Source code shall be developed based on the defined coding standard.
  - (c) Unit testing specifications shall be developed in accordance with the software verification plan, the software test plan and the acceptance criteria defined in (1) (c) above.
  - (d) Unit testing shall be performed in accordance with the unit testing specifications, and the test results shall be recorded in a format that it allows determination of pass or failure.
  - (e) For unit testing, the test shall be performed so that the criteria of the test coverage for source code are satisfied.
  - (f) Static analysis shall be performed with a code checking tool or equivalent and the source code quality shall be evaluated.
  - (g) The traceability of the source code and software design specifications shall be analyzed and the results shall be recorded.
  - (h) The operational assumptions and constraints identified during software coding and testing shall be identified.

#### **5.3.8.2 Measurement**

The following measurement shall be performed for software coding and unit test in order to evaluate the quality:

- (1) The definition of data to be collected for evaluating source code quality, and the evaluation method, shall be defined.
- (2) Collection and evaluation of data defined in (1) above shall be planned.
- (3) Collection and evaluation of data defined in (1) above shall be performed and the results recorded.
- (4) Results of the evaluation in (3) above shall be reported periodically, or for each milestone.

#### **5.3.8.3 Input**

- (1) Software design specifications
- (2) Interface specifications
- (3) Operational assumptions and constraints
- (4) Software verification plan

#### **5.3.8.4 Output**

- (1) Source code
- (2) Operational assumptions and constraints (updated)
- (3) Unit testing specifications

- (4) Unit testing record
- (5) Traceability analysis record
- (6) Source code quality evaluation results

#### **5.3.8.5 Review**

For each output, a software coding and testing review shall be performed. Appropriate follow-up shall be performed with regard to action items in accordance with due dates, follow-up status, degree of influence, and so on. If a review is performed, it shall be documented in a technical review record after its completion. In addition, quantitative data such as the reviewers' positions, their review time, the number of questions, and their comments shall be recorded, and the quality of the review shall be evaluated.

#### **5.3.9 Not used**

#### **5.3.10 Software integration**

##### **5.3.10.1 Activity**

For software integration, activities that meet the following items shall be conducted:

- (1) Software shall be integrated and baseline shall be determined after software integration.
- (2) Problems (such as interface problems and so on) found during software integration shall be recorded.

##### **5.3.10.2 Measurement**

The following measurement shall be performed for software integration for quality evaluation:

- (1) For problems collected during software integration, the definition of data to be collected for evaluating software products quality and its evaluation method shall be defined.
- (2) Collection and evaluation of data defined in (1) above shall be planned.
- (3) Collection and evaluation of data defined in (1) above shall be performed and the results recorded.
- (4) Results of evaluation in (3) above shall be reported periodically, or at every milestone.

##### **5.3.10.3 Input**

- (1) Source code

##### **5.3.10.4 Output**

- (1) Software
- (2) Record of problems collected during software integration
- (3) Evaluation results of collected data regarding software quality

#### **5.3.11 Software integration test**

##### **5.3.11.1 Activity**

Activities that meet the following shall be performed for the software integration test:

- (1) Test preparation
  - (a) As the result of software coding, testing and software integration, the software test specifications shall be updated as necessary.

- (b) For software test, the following shall be considered:
  - (i) Operational scenarios
  - (ii) Interface specifications
  - (iii) Maximum loads for assumed scenarios
  - (iv) Coverage for software requirements specifications and software design specifications
  - (v) Anomalies, such as exceptions and failures
  - (vi) Appropriateness with COTS or reused software items with computer system
- (c) Software integration test procedures shall be documented in accordance with the software verification plan, the software test plan, and the software test specifications.
- (d) Criteria to determine pass or failure shall be established.
- (e) In cases where any operational assumptions and constraints are found to be necessary during the preparation of the software integration test procedure, the operational assumptions and constraints shall be updated.

(2) Implementation of tests

- (a) The tests shall be performed in accordance with the software (integration) test procedure.
- (b) During the software integration test, a joint review of the test results shall be performed as necessary, and a decision shall be made on whether the test shall be continued or not.
- (c) With regard to the software integration test, information about the test environment, test data, configuration and version of the software under test shall be recorded to ensure the reproducibility of test conditions.
- (d) Test results shall be recorded and stored appropriately, and they shall be presented as necessary.
- (e) If software or software integration test specifications need to be revised during a software integration test, the effectiveness of the various tests performed after software coding and testing shall be evaluated, and tests performed after software coding and testing shall be performed again as necessary.

### 5.3.11.2 Measurement

For software integration tests, the following measurement shall be performed in order to evaluate the quality:

- (1) Collection of quality metrics data
  - (a) Problems found during software integration tests shall be recorded, together with related information such as test cases.
- (2) Quality index data setting
 

If quality metrics other than (1) above are set, collected, and evaluated, the following shall be implemented:

  - (a) Metrics for quality evaluation during tests shall be set
  - (b) Identified data shall be collected
  - (c) Analysis evaluation method for the identified data shall be defined

- (d) Analysis and evaluation of identified data shall be performed
- (3) Result of data analysis and evaluation shall be reported periodically or for each milestone

#### **5.3.11.3 Input**

- (1) Interface specifications
- (2) Software requirements specifications
- (3) Software design specifications
- (4) Software verification plan
- (5) Software test plan
- (6) Software test specifications
- (7) Operational assumptions and constraints
- (8) Source code (integrated)
- (9) Software (integrated)
- (10) Operational scenarios
- (11) Applicability evaluation results of COTS or reused software

#### **5.3.11.4 Output**

- (1) Software integration test procedure
- (2) Software integration test record, including pass or failure judgment results
- (3) Operational assumptions and constraints (updated)
- (4) Source code (tested)
- (5) Software (tested)
- (6) Software test specifications (updated)

#### **5.3.11.5 Review**

For each output, a software test review shall be performed. Items to be reviewed shall be chosen based on 5.3.11.4 and shall be defined in a plan document, such as the software development plan. Appropriate follow-up on action items shall be performed in accordance with due dates, follow-up status, degree of influence, and so on. If a review is performed, it shall be documented in a technical review record after its completion. In addition, quantitative data, such as the reviewers' positions, their review time, the number of questions, and their comments shall be recorded, and the quality of the review shall be evaluated.

### **5.3.12 Software installation into target platforms (embedding)**

#### **5.3.12.1 Activity**

Activities that meet the following shall be performed for the software installation into target platforms (embedding):

- (1) Software shall be prepared from the repository of the software development environment and prepared in a form that allows installation (embedding) into the target platform, and the configuration management information (filename, version information, and so on) of the software shall be acquired.  
Installation into a target platform means embedding software into the targeted computer, including writing into ROM.
- (2) Configuration management information of software to be released shall be prepared, to

include the installation (embedding) procedure into the target platform, the installation (embedding) result check procedure, and the operational assumptions and constraints.

- (3) Software shall be installed into the target platform in accordance with the installation (embedding) procedure. However, this can be omitted if the software has already been installed into the target platform.
- (4) A check (\*) that the software has been properly installed (embedding) shall be performed, in accordance with the results check procedure.

\*: For example, the original data after installation and the data on the target platform shall be compared by a file dump, and shall be confirmed by a comparison of their consistency.

#### **5.3.12.2 Input**

- (1) Software
- (2) Operational assumption and constraints

#### **5.3.12.3 Output**

- (1) Software prepared in a form that allows installation (embedding) into the target platform
- (2) Configuration management information
- (3) Installation (embedding) procedure
- (4) Computer system installed (embedded) software
- (5) Installation (embedding) results check procedure
- (6) Operational assumption and constraints (updated)

### **5.3.13 Computer system integration and computer system integration test**

#### **5.3.13.1 Activity**

Activities that meet the following shall be performed for the computer system integration and computer system integration test:

- (1) Test preparation
  - (a) Computer system integration shall be performed.
  - (b) Computer system integration test specifications shall be documented. With regard to the computer system integration test specifications, the following viewpoints shall be considered:
    - (i) Operational scenarios
    - (ii) Maximum load for assumed scenarios
    - (iii) Coverage regarding requirements specifications for a computer system, computer system architectural design, and software requirements specifications
    - (iv) Anomalies, such as exceptions and failures
    - (v) Appropriateness with COTS or reused software items with the computer system
  - (c) Computer system integration test procedure shall be developed in accordance with a software verification plan.
  - (d) Problems found during test preparation (test procedure checks, and so on) shall be



- recorded and managed.
- (e) In cases where operational assumptions and constraints are found during the preparation of computer system integration test specifications and a computer system integration test procedure, the operational assumptions and constraints shall be updated.
- (2) Implementation of tests
- (a) The tests shall be performed in accordance with computer system integration test procedure.
  - (b) As necessary, based on the operational scenarios, tests shall be performed by using tools such as simulators, and the verification coverage regarding operational scenarios shall be checked.
  - (c) With regard to the computer system integration test, information about the test environment, test data, configuration, and version of the software under test shall be recorded to ensure the reproducibility of test conditions.
  - (d) Test results shall be recorded and stored appropriately, and they shall be presented as necessary.
  - (e) When software or computer system integration test specifications need to be revised during computer system integration tests, the effectiveness of the tests shall be evaluated, and tests performed after software coding and testing shall be performed again as necessary.
  - (f) During computer system integration test, a joint review of the test results shall be performed as necessary, and decision shall be made as to whether the test should be continued or not.
- (3) Regarding software to be released, the configuration management information operational assumptions and constraints shall be prepared.

#### **5.3.13.2 Measurement**

The following measurement activities shall be conducted for computer system integration and computer system integration tests in order to evaluate the quality:

- (1) Collection of quality metrics data
  - (a) Problems found during the computer system integration tests shall be recorded, together with related information such as information about test cases.
  - (b) Problems found during test preparation (test procedure checks, and so on.) shall be recorded together with related information such as information about test cases.
- (2) Quality metrics data definition
 

If a quality metrics other than (1) above is defined, collected, and evaluated, the following shall be implemented:

  - (a) The metrics for quality evaluation during computer system integration tests shall be defined.
  - (b) Identified data shall be collected.
  - (c) The analysis evaluation method for the identified data shall be defined.
  - (d) Analysis and evaluation of identified data shall be performed.
- (3) The result of data analysis and evaluation shall be reported periodically, or for each milestone.

**5.3.13.3 Input**

- (1) Operational scenarios
- (2) Requirements specifications for computer system
- (3) Computer system architectural design specifications
- (4) Software requirements specifications
- (5) Software verification plan
- (6) Operational assumptions and constraints
- (7) Software in a form that allows installation (embedding) into the target platform
- (8) Configuration management information
- (9) Applicability evaluation results of COTS or reused software

**5.3.13.4 Output**

- (1) Computer system integration test specifications
- (2) Computer system integration test procedure
- (3) Computer system integration test record, including pass or failure judgment results
- (4) Operational assumptions and constraints (updated)
- (5) Software products
- (6) Configuration management information (updated)

**5.3.14 Supply and introduction of software product**

Not applicable.

**5.3.15 Software acceptance**

Not applicable.

**5.4 Operational process**

In this process, the activity to perform operational support for the operator and user, before and after the launch of spacecraft, is defined regarding the computer system (where the program has been installed) integrated into the spacecraft.

- (1) Process implementation
- (2) Operational testing
- (3) Operation of software or computer system
- (4) Operator and user support

**5.4.1 Process implementation**

A plan (operational plan) for implementing this process shall be documented and executed.

**5.4.1.1 Establishment of a procedure regarding problem report and request for modification**

A procedure regarding problem report and request for modification of software under spacecraft operation shall be established.

**5.4.1.2 Establishment of procedure regarding software reprogramming**

The procedure regarding software reprogramming for spacecraft in operation shall be established.

**5.4.2 Operational testing**

Not applicable.

**5.4.3 Operation of software or computer system**

Not applicable.

**5.4.4 Operator and user support****5.4.4.1 Support before the start of a spacecraft operation**

- (1) The following contents shall be provided to the persons in charge of a computer system or subsystem, and shall be handled as the source data for SOOH and SOP:
  - (a) Command information that software receives, such as a command list, a command format, the data pattern of each command (fixed and variable value), and command transmission procedures that are necessary for operation
  - (b) Telemetry information that software outputs, such as a telemetry list, telemetry format, and frequency of occurrence of telemetry
  - (c) Interrelationships between transmission commands and related telemetry
  - (d) Interface between the on-orbit reprogramming files and the ground software, necessary for reprogramming and comparison of spacecraft software.
  - (e) Memory map and procedure necessary for reprogramming and comparison of software
  - (f) Operational constraints identified from the software design to the testing

**5.4.4.2 Support after start of spacecraft operation**

- (1) Support for the requirements by operators and users shall be performed. These requirements and the response status shall be recorded and followed.
- (2) The requirements by operators and users shall be delivered to the maintenance process (refer to 5.5) for the purpose of problem resolution, as necessary. These requirements shall be clarified and planned, and the action to be performed shall be reported to the requester. Every solution shall be monitored until it reaches its conclusion.
- (3) Before the final solution of the reported problem, if any workaround may exist, the option of whether the workaround is selected or not shall be provided to the originator of the problem report.
- (4) The permanent modification and functional addition for software operating on-orbit shall be performed by software reprogramming according to the maintenance process (refer to 5.5).

**5.5 Maintenance process**

This process is performed in cases where the software and related documents must be modified for the purpose of correction and improvement, or to apply partial modifications called patches, of onboard software or additional modification of software for the reason of functional addition, extension, and so on. This process is performed for onboard software as necessary, after completion of software tests or spacecraft system tests, and for the delivery of the modified software to the spacecraft system to end spacecraft operations. Software reprogramming during spacecraft operation is also included in this process.

This process terminates at the end of the spacecraft operation, that is, through retirement of the computer system onboard the spacecraft.

This process consists of the following activities:

- (1) Process implementation
- (2) Problem identification and modification analysis
- (3) Modification implementation
- (4) Software reprogramming
- (5) Migration
- (6) Software retirement

#### **5.5.1 Process implementation**

The following shall be performed for the process implementation:

- (1) The maintenance plan for implementation of the maintenance process shall be documented and implemented. The following shall be included in the plan:
  - (a) Maintenance management method for developed software products
  - (b) The structure and the related documentation
  - (c) Data, including the management information necessary for maintenance
  - (d) Record
  - (e) Maintenance environment (environment of development phase, and so on)
  - (f) Actions for maintaining other related processes appropriately (problem resolution process (refer to 6.8), configuration management process (refer to 6.2), and so on)
- (2) Configuration management during on-orbit operation shall be performed according to the configuration management process (refer to 6.2).
- (3) Quality assurance shall be planned according to the quality assurance process (refer to 6.3).

#### **5.5.2 Problem identification and modification analysis**

- (1) In accordance with the plans, the analysis of the requested contents and correspondence shall be evaluated.
- (2) If software modification and reprogramming are performed, prior agreement for the modification plan and software reprogramming procedure shall be obtained.

#### **5.5.3 Modification implementation**

If a modification is needed, it shall be implemented in accordance with the agreed modification implementation plan. The corresponding process of this standard shall be implemented again, as necessary.

#### **5.5.4 Software reprogramming**

##### **5.5.4.1 Software reprogramming and verification procedure**

Reprogramming from the currently operating software on spacecraft to the modified software or previously installed software shall be performed according to the following procedure:

- (1) The procedure for software reprogramming shall be modified, as necessary.
- (2) The software version after software reprogramming shall be defined.
- (3) The version of software and its parameters before reprogramming shall be defined and recorded.

- (4) The procedure used for software reprogramming and verification shall be documented and software reprogramming shall be executed. The recovery procedure to be applied in case the software reprogramming was not performed successfully shall be included in the procedure.
- (5) Completion of software reprogramming shall be notified to all concerned parties.

**5.5.4.2 Implementation of quality assurance**

The quality assurance activity planned in 5.5.1 shall be performed.

**5.5.5 Migration**

Not applicable.

**5.5.6 Software retirement**

Not applicable.

## **6 Supporting life cycle process**

This clause defines the following supporting life cycle processes:

- (1) Documentation process
- (2) Configuration management process
- (3) Quality assurance process
- (4) Verification process
- (5) Validation process
- (6) Joint review process
- (7) Assessment process
- (8) Problem resolution process

The activities and tasks in a supporting process are the responsibility of the organization performing that process. This organization ensures that the process is in existence and functional.

### **6.1 Documentation process**

The purpose of this process is to define the management plan of the documents necessary for all persons concerned who develop and utilize software or computer system, and to perform document production and distribution and so on according to the management plan.

This process consists of the following activities:

- (1) Process implementation
- (2) Development
- (3) Document production
- (4) Document maintenance

#### **6.1.1 Process implementation**

A management plan for developed and received documents in all processes required by this standard shall be established.

The following information shall be included in the management plan:

- (1) Type of document
- (2) Purpose of the document
- (3) Plan for issue date and acquisition date of the document
- (4) Relevant departments
- (5) Procedures and responsibilities regarding the input information including received documents: development (development, inspection, and approval), production (production and storing), and maintenance (maintenance, storing and original copy disposal).
- (6) The form (content, format, and so on) appropriate for such documents shall be decided.

#### **6.1.2 Development**

- (1) Each document shall be developed according to the separately defined form, style, and technical contents.
- (2) The appropriateness of information sources for the developed documents shall be confirmed.

- (3) The document shall be confirmed and corrected in accordance with the standard format.
- (4) The document shall be approved for its appropriateness by authorized personnel prior to production.

### **6.1.3 Document production**

- (1) The document shall be issued to the suitable parties for distribution according to the management plan.
- (2) The produced document shall be stored according to the related standard.
- (3) At the production, it shall be considered so that the latest edition is surely utilized as necessary.

### **6.1.4 Document maintenance**

The plan for document maintenance shall be established, and the work such as the storing, revision, and disposal of documents shall be performed according to the maintenance plan, so that the necessary related documents are maintained after the development completion of software products.

## **6.2 Configuration management process**

The configuration management process is a process that applies to the following administrative and technical procedures through the software life cycle.

This process consists of the following activities:

- (1) Process implementation
- (2) Configuration identification
- (3) Configuration control
- (4) Configuration management accounting and status report
- (5) Configuration evaluation
- (6) Release management

### **6.2.1 Process implementation**

A configuration management plan shall be developed. The following information shall be included in the plan:

- (1) Configuration management activities and procedures to perform activities (standards to be applied, tools, and so on)
- (2) Milestones for configuration management, such as time to start and period of configuration management activities which are applied to management system
- (3) System to store and maintain information related to configuration management
- (4) All departments (roles and organization) involved in the implementation of configuration management; contents of configuration management activities performed by each department, mutual relationships, role sharing, and cooperative relationships among departments

### **6.2.2 Configuration identification**

- (1) Identification of configuration management items  
Configuration management items shall be identified from the products generated through the software development during the project.

## (2) Establishment of baseline

The configuration baseline shall be established in accordance with its purpose (\*). The configuration items and versions included in the baseline shall be clarified.

\*: The following may be items for this purpose.

- (a) Identification of the configuration for design review baseline
- (b) Identification of the configuration after software integration tests
- (c) Identification of the configuration at the end of checkout at launch

## (3) Version control system

Units for version control, a version control system, and a procedure shall be established. By using the system and procedure, the history of configuration management activity shall be recorded and shall be traceable.

**6.2.3 Configuration control****6.2.3.1 Management for configuration management items**

For the configuration management items, the history of the revision shall be recorded and controlled, based on the control system established in 6.2.2(3).

**6.2.3.2 Management for a set of configuration items**

A set of configuration items (\*) shall be confirmed so that configuration items are included neither too much nor too little and are the appropriate version.

The contents shall be recorded and the history shall be managed.

\*: Combination of items selected according to their purpose from configuration management items. This sub-clause shall be applied to the baseline control.

**6.2.3.3 Change request control**

## (1) Identification and recording of change requests

Change requests for configuration management items shall be identified. And the relationship to other change requests shall be evaluated if other change requests exist. Change requests, rationale of change, analysis results, modification status, and the status of change request shall be recorded.

## (2) Analysis and evaluation of the changes

For the change requests, technical influence, cost, and merit and demerit of the change shall be analyzed and evaluated.

The verification and validation activities performed at the change shall be analyzed.

## (3) Approval or disapproval of change requests

Approval or disapproval of change requests shall be determined by review.

The result of the review shall be recorded so that the reasons for the change and its approval or disapproval are traceable.

## (4) Confirmation of modification and verification of software items

The completion of modification and verification of software items shall be confirmed based on the approved change request.



#### **6.2.4 Configuration management accounting and status report**

Configuration management accounting and status reports that show the status and history of configuration management items shall be prepared for the purpose of configuration evaluation and so on.

#### **6.2.5 Configuration evaluation**

The baseline of a set of configuration items shall be evaluated whether configuration items are functionally and physically neither too much nor too little and have appropriate versions.

#### **6.2.6 Release management**

##### **6.2.6.1 Implementation of release management**

The release of software products and computer system shall be managed in accordance with the decided procedures.

And, the management for a set of configuration items, applied with the sub-clause 6.2.3.2, shall be performed for each release (\*1).

The original and the master copy of the configuration management items shall be maintained through the life cycle of software products or computer system.

And, the source code and documents shall be handled and stored according to the policy of the organization (\*2) and the policy defined in accordance with the type of information included in the source code and documents.

\*1: Release means distribution of the official version, delivery to system and delivery to testing.

\*2: In "the policy of the organization" mentioned here, company regulations and project rules are included.

### **6.3 Quality assurance process**

The quality assurance process is a process for providing adequate assurance that the process activities and outputs, of the software or computer system life cycle and the spacecraft life cycle, adhere to their established plans which are based on this standard or by tailoring its results.

This process consists of the following activities:

- (1) Process implementation
- (2) Implementation of quality assurance activities
- (3) Products quality assurance
- (4) Process assurance
- (5) Assurance of quality system
- (6) Perusal by related parties

#### **6.3.1 Process implementation**

##### **6.3.1.1 Appointment of the responsible person for quality assurance**

The responsible person who is invested with the responsibilities and authorities independent of the development organization shall be appointed. The responsible person shall be responsible for the management of quality assurance activities and the activities for the

check of status and improvement.

#### **6.3.1.2 Quality assurance activity plan**

The quality assurance activity plan shall be prepared. The quality assurance activity plan shall include the following:

- (1) Identification of the system to be applied (\*)
- \*: COTS, reused software items, and outsourcing development products shall be included.
- (2) Standards to implement this development standard
- (3) Activities, and related procedures and tools for the quality assurance implementation
- (4) Classification of problem handling and the problem resolution process (refer to 6.8) and its rules
- (5) Assignment of activities to each organization and the relationship between each organization for the purpose of quality assurance
- (6) Responsible person and the scope of responsibility for each organization described in (5) above
- (7) Schedules
- (8) Procedures for review
- (9) Procedures for the work such as identification, collection, filing, maintenance, disposition and disposal of quality assurance activity records
- (10) Education and training regarding quality assurance
- (11) Requirements and activities regarding quality assurance for purchase management and suppliers
- (12) Procedures for release
- (13) Relationships to the verification process (refer to 6.4), the validation process (refer to 6.5), the joint review process (refer to 6.6), the assessment process (refer to 6.7), and the problem resolution process (refer to 6.8), and selected activities and tasks.

#### **6.3.1.3 Agreement on quality assurance activities plan**

The quality assurance activities plan shall be agreed with the acquirer.

#### **6.3.2 Implementation of quality assurance activities**

Quality assurance activity shall be performed in accordance with the quality assurance activity plan.

#### **6.3.3 Products quality assurance**

- (1) Software, or computer system, and related documentation shall be developed without insufficiency with the plans.
- (2) Software or computer system shall have fully met the requirements of, and is acceptable to, the acquirer as preparation for release.
- (3) Products quality shall be assured by IV&V as necessary.

#### **6.3.4 Process assurance**

- (1) It shall be assured that the software development plans, operational plans, and processes defined by maintenance plans, comply with this standard.
- (2) It shall be assured that the software developments perform in accordance with the

processes defined in a development plan, an operational plan, or a maintenance plan.

### **6.3.5 Assurance of quality system**

It shall be assured that the quality system contains the quality management tasks listed below.

#### **6.3.5.1 Education and training**

Not applicable.

#### **6.3.5.2 Purchase management and supplier management**

(1) Purchase management

The reliability and quality of purchased items (COTS included) shall meet the software quality assurance requirements of the organization for the developing software product.

(2) Suppliers and purchase vendor selection

Based on the capability evaluation and selection record regarding suppliers and purchase vendor maintained in an organization, suppliers and purchase partners shall be selected.

#### **6.3.5.3 Management of items supplied by acquirer**

Procedures for inspection at the time of accepting items supplied or lent from the acquirer, and procedures for their storage and maintenance management, shall be established and followed.

#### **6.3.5.4 Management of existing software items (reused software items)**

With regard to existing software items (reused software items), the following shall be included in the management items.

(1) Any benefits gained by using existing software items

(2) Evaluation items and levels that allow the use of existing software items:

- (a) Appropriateness with existing software with regard to the developing software
- (b) Traceability relative to requirements applied to development software items
- (c) Risk obtained from the information such as the past performance of product, and so on, of software items to be used
- (d) Associated documents, which are obtainable and usable
- (e) Introduction, preparation, training, and constraints
- (f) Identification of versions and other details, and the configuration management method

#### **6.3.5.5 Handling, storing, and labeling**

Not applicable.

### **6.3.6 Perusal by related parties**

Records of quality assurance activities shall be available for perusal by related parties based on any agreements.

## **6.4 Verification process**

The verification process consists of the following activities. This process may include the

way such as review, test, and so on as the methods for verification.

- (1) Process implementation
- (2) Verification

#### **6.4.1 Process implementation**

- (1) An organization responsible for performing the verification shall be selected. This organization shall be provided with its independence and authority to perform the verification process.
- (2) The target for verification shall be determined, and appropriate tasks defined in 6.4.2 below shall be selected according to their degree of importance.
- (3) Based upon the selected tasks, a verification plan shall be documented.

#### **6.4.2 Verification**

This activity consists of the following tasks. In case of execution, the verification shall be performed considering the viewpoints (6.4.2.1 through 6.4.2.6) described for each task:

- (1) Process verification
- (2) Requirements verification
- (3) Design verification
- (4) Source code verification
- (5) Integration verification
- (6) Documentation verification

##### **6.4.2.1 Process verification**

The following shall be considered:

- (1) Adequacy of the processes selected for the project
- (2) Planning of the processes is adequate
- (3) Applicable criteria and environment for the project's processes in place
- (4) Adequate levels of competent staff allocated to the processes
- (5) Proper execution of the processes

##### **6.4.2.2 Requirements verification**

The following viewpoints shall be considered:

- (1) Requirements for the computer system are properly reflected in the computer system requirements specifications. The requirements specifications for the computer system are consistent, feasible, and verifiable.
- (2) Requirements specifications for the computer system are properly allocated to hardware, software, and operation.
- (3) Software requirements specifications are consistent, feasible, and verifiable. And the requirements specifications for the computer system are properly reflected in the software requirements specifications.
- (4) Software requirements specifications satisfy the standards applied to software items.
- (5) Concerning to matters to be especially taken care such as safety and security and so on, it is able to show the proper method that it satisfies the higher level requirements and the standards applied to items.

**6.4.2.3 Design verification**

The following viewpoints shall be considered:

- (1) The design meets requirements specifications, and has the traceability for requirements specifications.
- (2) Designed properly with respect to data interface, timing, computer resource (memory capacity, processing speed, and so on), logic design, processing sequence and processing contents (especially initialization, termination, exception handling and so on).
- (3) The characteristics such as portability, modifiability and ease of problem resolution have been covered.
- (4) Software design satisfies the requirements of the standards applied to software items.
- (5) Concerning to matters to be especially taken care such as safety and security and so on, it is able to show the proper method that it satisfies the higher level requirements and the standards applied to items.

**6.4.2.4 Source code verification**

The following viewpoint shall be considered:

- (1) Source code meets design, and has the traceability for design.
- (2) Implemented properly with respect to data interface, timing, computer resources (memory capacity, processing speed, and so on.), logic design, processing sequence and processing contents (especially initialization, termination, exception handling and so on).
- (3) The characteristics such as portability, modifiability and ease of problem resolution have been covered.
- (4) Concerning to matters to be especially taken care such as safety and security and so on, it is able to show the proper method that it satisfies the higher level requirements and the standards applied to items.
- (5) Source code shall conform to e.g. coding standards.

**6.4.2.5 Integration verification**

The following viewpoint shall be considered:

- (1) The configuration of the software modules and data is in proper and correct versions.
- (2) The software modules and data are completely and correctly integrated.
- (3) The components of the computer system are completely and correctly integrated.
- (4) The integration of software and the integration of the computer system have been performed in accordance with plan.

**6.4.2.6 Documentation verification**

The following shall be considered:

- (1) Documentation activity is performed properly in accordance with the documentation process (refer to 6.1).
- (2) Documentation activity is performed with adequate timing
- (3) Configuration management of the documentation is performed in accordance with the defined procedures.

## 6.5 Validation process

Validation is a process to confirm that the software or computer system fulfills the purpose of the intended use.

This process consists of the following activities:

- (1) Process implementation
- (2) Validation

### 6.5.1 Process implementation

- (1) An organization responsible for performing the validation shall be selected. This organization shall be provided with its independence and authority to perform the validation process.
- (2) A validation plan shall be developed. The plan shall include the following:
  - (a) Items subject to validation
  - (b) Validation tasks to be performed
  - (c) Resources, responsibilities, and schedule for validation
  - (d) Procedures for distributing validation reports

### 6.5.2 Validation

In accordance with the validation plan, the validation shall be performed considering the following viewpoints:

- (1) Software or computer system shall satisfy the assumed method and the use of the software. Software or computer system shall satisfy the requirements specifications.
- (2) Regarding the validation for the software or computer system performed as appropriate in the target environment, the difference between the target environment and the simulated environment shall be evaluated, if simulated environment is used,

\*: As the validation, the method except test (analysis, modeling and simulation and so on.) may be adopted.

## 6.6 Joint review process

The joint review process is a process to optionally evaluate the action status and products of software development.

At a joint review, both management (progress status and so on.) and technical levels of software development are reviewed.

This process may be applied to all reviews.

Joint review process consists of the following activities:

- (1) Process implementation
- (2) Software development management reviews
- (3) Technical reviews

### 6.6.1 Process implementation

When the software development is started, reviews corresponding to the project milestones including necessary resources (budget, personnel, location, facilities and tools, and so on.) shall be planned.

- (1) Periodical reviews shall be held at predetermined milestones, as specified in the project plans.

It is recommended that ad hoc reviews shall be called when deemed necessary by either reviewing party or reviewed party.

- (2) All resources required to perform the reviews shall be agreed on between the parties. These resources include personnel, location, facilities, hardware, software, and tools.
- (3) Both reviewing party and reviewed party concerned shall agree on the following at each review:
  - (a) Matters to be reviewed
  - (b) Review scope and subjects
  - (c) Review procedure
  - (d) Entry and exit criteria for the review
- (4) Reviews shall be performed according to the plan. Ad hoc reviews shall be conducted as necessary.
- (5) Problems identified in the review shall be recorded.
- (6) Both reviewing party and reviewed party shall agree with the related persons regarding following and shall distribute the results:
  - (a) Review results
  - (b) Assignment of responsibilities for the handling of identified problems
  - (c) Review close-out criteria

## **6.6.2 Software development management reviews**

### **6.6.2.1 Evaluation of software development status**

The development status shall be evaluated by comparing the current status to the software development plan and shall be evaluated by following viewpoints:

- (1) Development is progressing according to the plan.
- (2) Resources necessary for implementing the software development are allocated properly and managed.
- (3) The gap with the plan is monitored, and the necessity of software development direction change or implementing an alternate plan is determined.
- (4) Any risk that may jeopardize the success of the project is identified, evaluated, and managed.

### **6.6.3 Technical reviews**

Technical reviews shall be undertaken for software items from a technical viewpoint, and these reviews shall clarify the risks involved with the implementation of software and computer system that meet requirements specifications and standards.

This activity consists of the following tasks:

#### **6.6.3.1 Review**

- (1) The following shall be implemented for review preparation:
  - (a) Reviewers shall be selected for review implementation. Reviewers shall include appropriate personnel, consisting not only of parties to the work but also personnel who have enough sufficient knowledge of the area under review, as well as project parties which include interface designers and the originator of the requirements.
  - (b) The review subjects and purposes shall be clarified, and materials shall be

completed in advance.

- (c) The review purposes, subjects and reviewers shall be clarified, and documented.
- (2) Software items shall be evaluated in a step-by-step approach as developments progress.
- (3) Technical actions regarding software shall be evaluated.
- (4) Review reports shall be developed after the review is completed. Also, quantitative data, such as the evaluation time, and the number of questions and comments at the review, shall be recorded, and a quality evaluation for the review shall be implemented.

#### **6.6.3.2 Walk-through**

A walk-through, including a peer review and so on, is an action to improve quality by detecting and removing errors early in the design and development. A walk-through shall be implemented, and performed mainly by the persons in charge, as necessary:

- (1) As preparation for the walk-through, the reviewed party shall provide the relevant documents and codes, including those still in development.
- (2) Parties necessary for walk-through (such as persons in charge of higher level process, persons in charge of lower process, and so on) shall check the documents and codes.
- (3) Questions and problems found through walk-through shall be recorded and followed up until the resolution is completed, and mutually agreed.

#### **6.6.3.3 Evaluation of development status**

Software products or computer system shall be evaluated according to the following viewpoints:

- (1) Software products or computer system are complete.
- (2) Software products or computer system conform to standards and specifications.
- (3) Changes to software products or computer system are implemented properly.
- (4) Preparation to switch to the next action (preparation for the next process transition) is ready.
- (5) Development, operation or maintenance are performed in accordance with the planning, schedule, standards, and guidelines of the project.

### **6.7 Assessment process**

The assessment process is a process for checking the process implementation status and identifying the items to be improved.

This process consists of the following activities:

- (1) Process implementation
- (2) Assessment implementation

#### **6.7.1 Process implementation**

The following directions for project designated personnel who are responsible for performing an assessment (hereafter, referred to as the "sponsor"), shall meet the following tasks and shall be planned and agreed with sponsors:

- (1) It shall be evaluated whether the software development process to be implemented meets the requirements of this standard. The strengths and the weaknesses of the process shall be identified.



- (2) Personnel who are well informed of assessment methods and relevant standards shall be selected as assessors.
- (3) Assessment results, including improvement offers on items needing improvement, are reported in documents to sponsors.

### **6.7.2 Assessment implementation**

Based on the assessment plan, the assessment shall be implemented.

### **6.8 Problem resolution process**

The problem resolution process is a process for analyzing and resolving any problems that are detected during the development process (refer to 5.3), the operation process (refer to 5.4), the maintenance process (refer to 5.5), or other processes.

In this sub-clause, the problem items defined in the quality assurance process (refer to 6.3) shall be dealt with. Correction and improvement shall be included in the type of modifications required to solve a problem.

This process consists of the following activities:

- (1) Process implementation
- (2) Problem resolution
- (3) Problem trend analysis
- (4) Corrective and preventive action

#### **6.8.1 Process implementation**

##### **6.8.1.1 Problem resolution plan**

Based on the quality assurance activity plan, the problem resolution process shall be established to handle all problems encountered in the software or the computer system.

According to the criteria of the problem, the following recording and countermeasure process shall be decided.

- (1) Problem management method (method of acceptance and recording, reporting, and method of maintenance (including storing period))
- (2) Analysis of the cause of the problem and the resolution policy
- (3) Follow-up policy for problem resolution status
- (4) Policy for corrective and preventive action

##### **6.8.2 Problem resolution**

Problems shall be resolved in accordance with the problem resolution plan. Particularly in cases where a problem requires the modification of design specifications or source code, it shall be confirmed that the necessary items among the following activities based on the management level are included in the problem resolution plan, and appropriate problem resolution activities shall be performed.

###### **6.8.2.1 Acceptance and record of problems**

- (1) All problems detected in the primary life cycle process and the support life cycle process shall be reported quickly after recording them for each process they affect, and the problems shall be accepted into the problem resolution process as quickly as possible.
- (2) Recording of the problem shall be started.

**6.8.2.2 Analysis**

- (1) In accordance with a quality index, the problem shall be classified.  
Based on the classified results, the following shall be conducted, as necessary.
- (2) The reappearance of the problem shall be confirmed using the maintenance environment, or verified using source code or software design specifications on a desk work.
- (3) The problems shall be analyzed and the causes and the areas of influence shall be identified.
- (4) Results of the problem analysis shall be recorded.

**6.8.2.3 Consideration for problem resolution**

The following methods of problem resolution shall be considered:

- (1) The determination as to the necessity of the modification (judgment whether the modification should be performed considering the effect of the modification, scale of the work, degree of fatality and emergency, and so on.) shall be performed.
- (2) The method of modification shall be investigated based on the analysis. It is desirable that multiple candidates for the modification plan are investigated and the modification method is selectable from among the candidates.
- (3) The modification method shall be recorded, and its correspondence to the problem report, the modification request and analysis shall be managed so that they are easy to understand.
- (4) Notes for the operation shall be documented.

**6.8.2.4 Determination of problem resolution plan**

Plans shall be documented and agreed with the parties:

- (1) The problem resolution plan shall be determined and agreed with the related parties.  
The plan for the problem resolution shall be drafted, as necessary.
- (2) In case the problem occurred after software release, as a temporary measure, if there is a way of mitigating the problem, the problem resolution plan shall be determined and agreed with related parties. The plan for the temporary problem resolution shall be drafted as necessary.

**6.8.2.5 Problem resolution implementation**

In accordance with the agreed plans, problem resolution shall be implemented, and related parties shall be notified.

**6.8.2.6 Problem report**

A record of the problem shall be properly reported.

**6.8.2.7 Recording and monitoring**

A set of problem resolution actions and the status shall be recorded, monitored, and managed. In principle, after the step has been implemented, monitoring shall continue until it is determined that the problems are resolved, and no new problem has occurred.

**6.8.2.8 Maintenance of records**

A record of the problem shall be stored for the period based on a plan.

**6.8.3 Problem trend analysis**

Problem trend analysis shall ideally be performed.

**6.8.4 Corrective and preventive action**

For the serious problems (an occurrence of similar problems and so on), the action to prevent from the reoccurrence, shall be taken, as necessary. And roll out of the proper information shall be implemented to the related or similar work process and products, and related department.