



SOFTWARE DEVELOPMENT STANDARD FOR GROUND SYSTEM

Mar. 23, 2016

Japan Aerospace Exploration Agency

This is an English translation of JERG-3-003A. 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

1	Scope	1
2	References	1
2.1	Document for compliance	1
2.2	Informative references	1
3	Terms, definitions and abbreviated terms	1
3.1	Terms redefined in this standard.....	1
3.2	Terms newly defined in this standard	1
3.3	Terms defined in JERG-0-049.....	2
4	Organization of this standard	6
5	Primary life cycle processes.....	8
5.1	Not used.....	8
5.2	Not used.....	8
5.3	Development process	8
5.3.1	Process implementation	9
5.3.2	Items to be applied to all processes.....	9
5.3.3	System requirements analysis.....	10
5.3.4	System architectural design.....	10
5.3.5	Software requirements analysis	11
5.3.6	Software design	13
5.3.7	Not used.....	16
5.3.8	Software coding and testing	16
5.3.9	Not used.....	17
5.3.10	Software integration.....	17
5.3.11	Software integration test	18
5.3.12	Software installation into target machine.....	20
5.3.13	System integration and system integration test.....	21
5.3.14	Supply and introduction of software products	22
5.3.15	Software acceptance	23
5.4	Operational process	24
5.4.1	Process implementation	24
5.4.2	Operational testing	24
5.4.3	Operation	24
5.4.4	User support	25
5.5	Maintenance process	25
5.5.1	Process implementation	26
5.5.2	Problem identification and modification analysis	26
5.5.3	Modification implementation.....	26
5.5.4	Software reprogramming.....	26

5.5.5	Migration.....	26
5.5.6	Software retirement and replacement.....	27
6	Supporting life cycle process	28
6.1	Documentation process.....	28
6.1.1	Process implementation	28
6.1.2	Development	28
6.1.3	Production	29
6.1.4	Revision	29
6.2	Configuration management process.....	29
6.2.1	Process implementation	29
6.2.2	Configuration identification.....	30
6.2.3	Configuration change control	30
6.2.4	Record of configuration change status	31
6.2.5	Evaluation of configuration change status	31
6.2.6	Release management and delivery.....	31
6.3	Quality assurance process.....	31
6.3.1	Process implementation	32
6.3.2	Product quality assurance.....	32
6.3.3	Process assurance.....	32
6.3.4	Assurance of quality system.....	33
6.4	Verification process	34
6.4.1	Process implementation	34
6.4.2	Verification	34
6.5	Validation process.....	35
6.5.1	Process implementation	36
6.5.2	Validation	36
6.6	Joint review process	36
6.6.1	Process implementation	36
6.6.2	Project management reviews	37
6.6.3	Technical reviews	37
6.7	Assessment process.....	37
6.8	Problem resolution process.....	38
6.8.1	Process implementation	38
6.8.2	Problem identification.....	38
6.8.3	Problem analysis, improvement and resolution	39
6.8.4	Improvement and resolution status monitoring	39
6.8.5	Notification to concerned parties	39
	Appendix	40

1 Scope

This standard applies to activities relating to the development, operation, and maintenance of software for ground systems and the activities needed for system development and related support. When this standard is applied, embodying and tailoring may be performed in accordance with the characteristics of specific projects and other factors.

In principle, this standard does not define the categories of personnel who implement processes. These personnel shall be defined by software development standard (JERG-0-049A) that is the document for compliance or individual contracts or other agreements to which this standard is applied.

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) ISO 9000: 2000 Quality management system, Fundamentals and vocabulary
- (7) Software Life Cycle Processes- Japan Common Frame 2007 (Japanese) (Copyright IPA/SEC 2009)

3 Terms, definitions and abbreviated terms

3.1 Terms redefined in this standard

System	System is the entire system consisting of sets of software, platforms, network, and hardware, including the platform and hardware that are able to execute the target software for development. Although the definition of what a system contains is arbitrary, the definition shall be unique for software products that are subjected for development. As a system may be defined in various ways, from a one-chip microcomputer to multiple general-purpose connected to a network.
--------	---

3.2 Terms newly defined in this standard

External interface, Internal interface	Basically, external interface is interface between the outside of system and the developing system, and internal interface is interface inside of the developing system. The definitions may be applied such as that the inside of system or subsystem may be called as internal interface, and the outside of system or subsystem may be called as external interface by using under the uniform regulation.
Operator, User	Operator: An individual or organization who operates the system. User: An individual or organization who uses the operational system to perform a specific function.

Quality system	Quality system is a set of organizational structures, procedures, processes, and management resources to perform quality management.
Regression test	Regression test (Recursive test) is re-testing to verify whether a software change causes unintended results.
Requirement	Requirement is one of a set of functions and performance targets requested for system or software and they may be also included such as not embodied and not detailed enough, or ambiguity in expression and vague expectations.
Target machine	Target machine is a computer that the software is installed into and executes the software.
Test specification	Software test specifications are defined as the descriptions of test conditions and expected results, expressed unambiguously, to prove that software meets the requirements specifications. "The unit testing specifications" used in the process of software coding and testing, "The software integration test specifications" used in the process of software integration test, and "The system integration test specifications" used in the process of software integration test are treated as test specifications.
Use case	Use case describes the interaction between system, and operator or user in the view point of operator and user. "Use and operational concept" and "use and operational scenarios" shall be considered.

3.3 Terms defined in JERG-0-049

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 implement status to identify opportunities to improve the process for some predetermined purpose.
Configuration management	Configuration management is an action to define the configuration items, i.e., systems or projects, to record changes of content, and to manage such aspects as their storage, handling, and distribution. For example, if the software consists of multiple modules, then not only must the software version be managed, but also each version of the module software must be managed. In addition, configuration management items require software consistent modules, requirements specifications, operation manuals, and so on.
COTS	COTS is the abbreviation of Commercial Off-The-Shelf. It has already been developed and is available in the commercial marketplace.

Term	Definition
Identifier	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. Appending a unique identifier to each item is convenient for requirements management and for traceability.
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	Integrity is defined as the following properties in this standard: (1) Software component is complete with no deficiencies. (2) Software component is at an appropriate version.
Non-functional requirements	Non-functional requirements are all requirements except functional requirements, such as reliability, usability, efficiency, maintainability, portability, and so on.
Operation	Operation is an action to carry out missions for the achievement of a purpose by means of an appropriate system. An operation utilizes the system from beginning to end, and includes monitoring and maintenance functions, and so on.
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. It includes status change, output, evaluation of work, and so on.
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.
Requirements specification	Requirements specification is defined as the description of functions and performance required for system or software, embodied and specifically defined, and which also consider feasibility. In principle, the specified requirements specifications shall be verifiable as both feasible and mutually consistent. 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 specifications shall be complemented by the following methods: (1) The planning process shall include agreement with system users that the requirements specifications are satisfactory, and shall also include a software verification plan. (2) It shall include the test specifications to verify the requirements in the verification plan. In principle, any restrictions, laws, rules, and a project policy shall be included in the requirements specifications.
Risk	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.

Term	Definition
Software	Software is a set of 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. 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 on a case-by-case basis.
Software life cycle	Software life cycle is the period from the beginning of the requirements analysis phase until the termination of use of the software.
Software products	Software products are the set of software, source code, and related documentation.
Software under test	Software under test is software that is being subjected to testing and inspection.
Software user's manual	Software user's manual is the set of information a user needs in order to use software. It includes operation unit manuals, system operation manuals, and work operation manuals.
Software verification plan	Software verification plan is a documentation of the scope, content, method, environment (such as test equipment) and schedule pertaining to the verification and the validation of software development. A validation plan may be included.
System	System is the entire system consisting of sets of software, platforms, network, and hardware, including the platform and hardware that are able to execute the target software for development. Although the definition of what a system contains is arbitrary, the definition shall be unique for software products that are subjected for development. As a system may be defined in various ways, from a one-chip microcomputer to multiple general-purpose connected to a network.
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 process and test environment for target test items.
The stability 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.

Term	Definition
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]
Verification	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. [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 ground system, 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 ground system, and act to support a primary life cycle process and are called by other processes, as necessary.

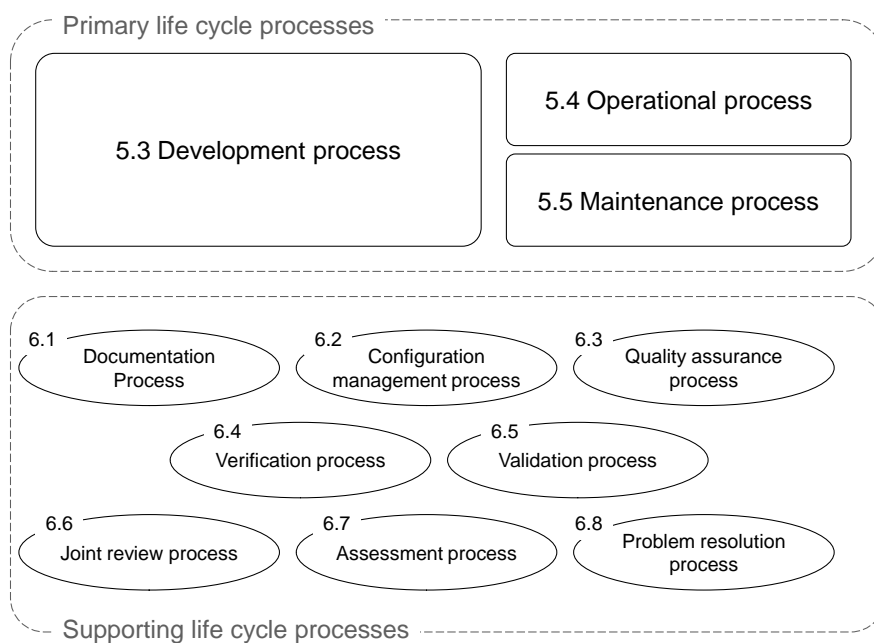


Figure 4.1 - Structure of the standard

The processes and each activity may be implemented in an order different from the clause number order in this standard. Also note that there are cases where identical activities have been described in multiple processes. For example, activities regarding software verification have been described for the development process, which is one of the primary life cycle processes. These are also activities regarding the verification process, which is one of the supporting life cycle processes.

The classification of processes will be better understood if it is considered to be classified according to the difference of viewpoints. This standard aims to define processes from various viewpoints so that all of the required contents are covered completely (duplication is allowed). Therefore, this standard assumes that it will be applied after appropriate concretizing and tailoring are 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 into target machine, supply, introduction, acceptance, and so on.
	5.4 Operation	Process to be implemented from the viewpoint of operation. Drafting plans and rules for operations, operational testing, operation, user support, and so on.
	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 assurance that the product quality, the process and the quality system have been performed, according to the plan.
	6.4 Verification	Process regarding the confirmation that the requirements for a specific intended use or application have been fulfilled, based on the provision of objective evidence.
	6.5 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 against the requirements of this standard.
	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 for the following processes:

- (1) system requirements analysis process;
- (2) 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 tests process;
- (8) software installation into the target machine process;
- (9) system integration and system integration tests process;
- (10) supply and introduction of software products process;
- (11) software acceptance process.

These processes and 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.

5.3.1 Process implementation

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

- (1) A software development plan including the following information shall be established:
 - (a) Scope of the system and sub-systems for development
 - (b) Definition of the development process, including activities for the system that the software is installed into (the target machine)
 - (c) Activities in each development process and their management plan, including the plan for the quality assurance process
 - (d) Preparation of development related documents' structures and relationships of input and output in each development process
 - (e) Appropriate work allocations and methods for work plans and progress management for each task, including development of a schedule and the organizations in charge
 - (f) Environment to be used for software development and verification (simulator, test environment, and so on)
 - (g) The management plan shall include the followings, if COTS and reused software, including the software owned by JAXA or software supplier, are used:
 - (i) Identification of COTS items and reused software items
 - (ii) Definition of the quality assurance process regarding COTS items and reused software items:
 - Regarding COTS items, the past usage record and the functions to be used shall be confirmed in advance
 - Regarding reused software items, the availability of design and verification information for the appropriateness with the usage condition and specifications shall be confirmed
 - (h) Compatibility evaluation plan with other parts of the target system
- (2) The software development plan shall be documented and it shall be established for configuration management.

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.

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 System requirements analysis

5.3.3.1 Activity

Activities that meet the following requirements shall be performed for the system requirements analysis:

(1) Requirements extraction

The requirements for the system to be developed and its operational concept shall be analyzed. Use and operational configuration shall be investigated and the functional requirements of the target system, especially non-functional requirements such as performance, quality, and operational scenarios shall be extracted.

(2) Requirements specifications development

Feasibility and consistency shall be confirmed based on the extracted requirements, and requirements specifications for the target system and external interface shall be defined and documented. The requirements specifications shall be mutually agreed with the contact organization of the external system, including interpretation of the contents.

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

5.3.3.2 Input

(1) Requirements for target system

(2) Operational concept

5.3.3.3 Output

(1) Operational scenarios

(2) Requirements specifications for the target system

(3) External interface specifications (may be included in (2) above)

(4) Evaluation results of traceability for higher level requirements

5.3.4 System architectural design

5.3.4.1 Activity

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

(1) System architecture for the target system shall be designed based on the operational scenarios, requirements specifications for the system, and external interface specifications. The configuration items (system and sub-system), development items (new, COTS, modified, or reused software), and classification (hardware, firmware, software, and operation) shall be identified, and the feasibility of the system shall be evaluated. Design results shall be documented as the design specifications of the target system and as the internal interface specifications for configuration items.

(2) The requirements pertaining to the requirements specifications for a target system shall be allocated among the individual configuration items of the system. All requirements extracted from the system requirements for the target system shall be allocated to the configuration items clarified in (1) above.

(3) Operational scenarios shall be analyzed and allocated to each system configuration

item as was allocated in (1) above.

- (4) Along with summarizing the design results from (1) through (3) above, the target system architectural design specifications, including the rationale of the design, operational assumptions, and constraints shall be identified and appropriate evaluations shall be performed regarding the system architectural design specifications.
- (5) When developing the target system requirements specifications, requirements for the target system such as higher level requirements and the traceability of operational scenarios shall be evaluated.
- (6) Regarding (1) above, the requirements allocated to software shall be extracted and summarized as software requirements.
- (7) According to the progress of design, the requirements and constraints regarding target machines and related equipment (network, uninterruptible power supply, and so on) shall be summarized. In some cases, this may be performed in the software design process.

5.3.4.2 Input

- (1) Operational scenarios
- (2) Requirements specifications for the target system
- (3) External interface specifications (may be included in (2) above)

5.3.4.3 Output

- (1) Target system architectural design specifications
- (2) Internal interface specifications (may be included in (1) above)
- (3) Software requirements
- (4) Evaluation results of traceability for higher level requirements
- (5) Operational assumptions and constraints
- (6) Constraints of the target machine

5.3.5 Software requirements analysis

5.3.5.1 Activity

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

- (1) Software requirements specifications, especially including non-functional requirements such as performance and reliability requirements, shall be developed based upon the analysis of the target system architectural design specifications, internal interface specifications, software requirements, and constraints of the target machine.
 - (a) Identifiers shall be appended to the individual requirements that make up the software requirements specification.
 - (b) Software requirements specifications shall include specifications for data handled by software and requirements for databases (expected number of data items, data size, retrieval speed, storage speed, and so on).
 - (c) Software requirements specifications shall include the specifications for anomaly detection, and handling function requirements.

Examples of anomaly handling function requirements:

Common requirements for anomaly handling such as log records, the handling requirements for an expected critical anomaly, the handling requirements for an unexpected anomaly, the handling requirements for an anomaly related to the interface, and so on.

- (d) As necessary, software requirements shall include the analysis results of security requirements.
- (e) Software requirements shall include the analysis result of use cases.
- (2) Based upon the analysis of the specifications such as internal interface specifications, and so on, the internal interface specifications with the components of a system such as other software (including COTS, modification or reuse of existing software), hardware, and humans shall be detailed.
Agreement with the relevant parties regarding the detailed internal specifications shall be reached based on a common understanding and common interpretation of the contents.
- (3) Traceability and consistency of the software requirements specifications relative to target system architectural design specifications, internal interface specifications, and software shall be evaluated.
- (4) Rationale of software requirements specifications shall be clarified and their feasibility shall be evaluated, as necessary.
- (5) If COTS or reused software is used, the conformance with the software requirements specifications (identification and validation of requirements realized by COTS or reused software, and so on) shall be evaluated.
- (6) Operational assumptions and constraints regarding software requirements specifications shall be extracted and updated.
- (7) The verifiability of the individual requirements of the software requirements and internal interface specifications shall be evaluated, and a software verification plan including the validation methods shall be established.
- (8) The software verification coverage pertaining to software requirements and operational scenarios in the verification plan shall be evaluated, and test plannability regarding software requirements specifications and internal interface specifications shall be evaluated.
- (9) With regard to the software verification plan, for planning purposes, whether the test is affected by the behavioral difference from the real operational environment, 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 in the software verification plan.
- (10) Necessary requirements regarding the target machine and related facilities shall be reviewed and the results shall be reflected in these requirements and constraints.
- (11) The results shall be reflected onto the operational scenarios, as necessary.

5.3.5.2 Measurement

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

- (1) The definition of data to be collected for evaluating the stability (maturity) of 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 shall be recorded.

5.3.5.3 Input

- (1) Target system architectural design specifications
- (2) Internal interface specifications (may be included in (1) above)
- (3) Software requirements
- (4) Operational assumptions and constraints
- (5) Constraints of the target machine
- (6) Operational scenarios

5.3.5.4 Output

- (1) Software requirements specifications
- (2) Internal interface specifications (updated)
- (3) Software requirements specifications traceability and consistency evaluation results
- (4) Software requirements specifications rationale and feasibility evaluation results
- (5) Applicability evaluation results of COTS or reused software
- (6) Operational assumptions and constraints (updated)
- (7) Software verification plan
- (8) Verification completeness and test plannability for the software verification plan evaluation results
- (9) Software requirements specifications stability (maturity) evaluation results
- (10) Operational scenarios (updated)
- (11) Constraints of the target machine (updated)

5.3.5.5 Review

For each output, a software requirements specifications review shall be performed. Items to be reviewed shall be chosen based on 5.3.5.4 and shall be defined in the plan such as the software development plan. 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, the software architectural design and detailed design are not considered to be strictly separate. However, in an actual development process, the software architectural design and the detailed design 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, and software design specifications shall be developed.

- (2) Software design specifications shall include the design and distribution of non-functional requirements (processing time requirements, requirements of resources such as memory, and so on) and be defined as software requirements specifications.
- (3) Internal interface specifications shall be detailed in accordance with the decomposition of the software functions and modules. Agreement with the relevant parties as to the internal interface specifications shall be arrived at based on a common understanding and common interpretation of the contents.
- (4) Software design specifications shall include screen and HMI (Human Machine Interface) design. Agreement shall be reached with the relevant users as to screen and HMI design, especially the specifications such as screen transition, response speed, and so on.
- (5) Use cases shall be analyzed and the results shall be reflected into the software design specifications and operational scenarios, as necessary.
- (6) Database design shall be included, as necessary.

Software detailed design

- (7) Each individual module shall be designed in accordance with the decomposition of functions and modules, and a software detailed design shall be performed.
- (8) Internal interface specifications shall be detailed in accordance with the design of the module. Agreement with the relevant parties as to the internal interface specifications shall be arrived at based on a common understanding and common interpretation of the contents.

Common to software architectural and detailed designs

- (9) Traceability and consistency of software design with software requirements specifications, internal interface specifications, and necessary related documents shall be evaluated for each module.
- (10) As necessary, with regard to the individual software design, the design rationale shall be clarified and its feasibility evaluated.
- (11) If COTS or reused software is used, its appropriateness with the software design shall be analyzed.
- (12) Operational assumptions and constraints regarding software design shall be identified.
- (13) Software integration test plan and specifications shall be established in accordance with the software verification plan. Evaluation criteria shall be included in software integration test specifications.
 - (a) For software integration test specifications, the following points shall be considered:
 - (i) Operational assumptions and constraints
 - (ii) Maximum load for assumed scenarios
 - (iii) Coverage for software requirements specifications, software design specifications, and internal interface specifications
 - (iv) Anomaly such as exceptions and failures
 - (v) Appropriateness of COTS or reused software items with target system
 - (vi) Appropriateness with screen design and HMI design

- (vii) Appropriateness with security requirements, as necessary
- (14) Necessary requirements regarding the target machine and related facilities shall be reviewed and the results shall be reflected in these requirements and constraints.
- (15) Design features that meet any security requirements shall be performed as necessary.
- (16) 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) Internal interface specifications
- (3) Operational assumptions and constraints
- (4) Applicability evaluation results of COTS or reused software
- (5) Software verification plan
- (6) Operational scenarios

5.3.6.4 Output

- (1) Software design specifications (may be divided into software architectural design specifications and detailed design specifications)
- (2) Internal interface specifications (updated)
- (3) Software design traceability and consistency evaluation results
- (4) Software design rationale and feasibility evaluation results
- (5) Applicability evaluation results of COTS or reused software (updated)
- (6) Operational assumptions and constraints (updated)
- (7) Software integration test plan
- (8) Software integration test specifications
- (9) Operational scenarios (updated)

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 the plan 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.

5.3.7 Not used

5.3.8 Software coding and testing

5.3.8.1 Activity

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

- (1) Rules for coding shall be defined as a coding standard.
- (2) Definitive implementation guidelines for error handling shall be considered.
- (3) Source code shall be developed based on software design specifications and internal interface specifications.
- (4) Source code shall be developed based on the defined coding standard.
- (5) Unit load modules shall be made from source code.
- (6) Unit testing specifications shall be developed and unit testing shall be performed and the results shall be recorded in accordance with the specifications. For unit testing, the test coverage and condition branches shall be considered. Confirmation by the unit testing with combining modules may be performed.
- (7) Based on the software verification plan, the source quality evaluation shall be performed by static analysis using the tool such as a source code checking tool and so on, or by source code review.
- (8) Traceability and consistency of the source code and software design specifications shall be analyzed.
- (9) If new operational assumptions and constraints arise or are identified, they shall be updated.

5.3.8.2 Measurement

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

- (1) The definition of data to be collected for evaluating source code 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 shall be 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) Internal interface specifications
- (3) Operational assumptions and constraints
- (4) Software verification plan

5.3.8.4 Output

- (1) Source code (after unit testing)
- (2) Unit load module (after unit testing)
- (3) Operational assumptions and constraints (updated)

- (4) Unit testing specifications
- (5) Unit testing record
- (6) Source code quality evaluation results
- (7) Source code and software design requirements traceability and consistency 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

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

- (1) Software shall be integrated by using source code (after unit testing) or by unit load modules (after unit testing).
- (2) A baseline after software integration shall be established according to the configuration management process (refer to 6.2).
- (3) Debug information (e.g., compile failure, link failure, problems regarding the test environment, and so on) collected during software integration shall be recorded. Based on the debug information, if review and modification for the source code, software, or test environment is required, the related processes such as the problem resolution process (refer to 6.8) and configuration management process (refer to 6.2), and so on shall be performed.

5.3.10.2 Measurement

The following measurement shall be performed for the software integration to evaluate software quality:

- (1) For debug information collected during software integration, the definition of data to be collected for evaluating software products quality and its evaluation method shall be defined. For example, warning messages and levels displayed during software linkage and so on, shall be considered as the definition of collected data.
Unacceptable messages and software execution status during software integration which have influence on software quality shall be considered as part of the evaluation method.
- (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 shall be recorded.
- (4) The evaluation results in (3) above shall be managed periodically or at every milestone,

during the software integration.

5.3.10.3 Input

- (1) Source code (after unit testing)
- (2) Unit load module (after unit testing)

5.3.10.4 Output

- (1) Source code (after software integration)
- (2) Software (after software integration)
- (3) Debug record

5.3.11 Software integration test

5.3.11.1 Activity

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

- (1) Test preparation
 - (a) As the result of software coding, testing and software integration, the software integration test specifications shall be updated as necessary.
 - (b) For software integration test, the following points shall be considered:
 - (i) Operational assumptions and constraints
 - (ii) Maximum load for assumed scenarios
 - (iii) Coverage for software requirements specifications, software design specifications, and internal interface specifications
 - (iv) Anomaly such as exceptions and failures
 - (v) Appropriateness of COTS or reused software items with target system
 - (vi) Appropriateness with screen design and HMI design
 - (vii) Appropriateness with security requirements, as necessary
 - (c) Software integration test procedures shall be documented in accordance with the software verification plan, the software integration test plan, and the software integration test specifications.
 - (d) According to the software integration test procedure, test equipments or testing data necessary for software integration testing shall be prepared.
- (2) Implementation of tests
 - (a) The tests shall be performed in accordance with the software integration test procedure.
 - (b) During the software integration test, the intermediate joint review of the test results shall be performed as necessary, and whether the test should be continued or not shall be determined.
 - (c) With regard to the software after software integration, information regarding the test environment, test data, configuration, and version of the software under test shall be recorded to ensure the reproducibility of test conditions.
 - (d) A test result of pass or failure for test criteria shall be recorded and stored appropriately, and shall be presented as required.
 - (e) If any problem occurred during the software integration test, action shall be

performed in accordance with the problem solving procedure (refer to 6.8). Analysis and evaluation regarding the modification requirements of software or software integration test specifications shall be performed as necessary. The effectiveness of the tests performed after software coding and testing shall be evaluated, and those tests shall be performed again, as necessary.

- (f) Based on the results of the software integration test, the information shall be updated, if new operational assumptions and constraints items are identified.

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) The results of data analysis and evaluation shall be managed and reported periodically, or for each milestone

5.3.11.3 Input

- (1) Internal interface specifications
- (2) Software requirements specifications
- (3) Software design specifications
- (4) Software verification plan
- (5) Software integration test plan
- (6) Software integration test specifications
- (7) Operational assumptions and constraints
- (8) Software (after software integration)

5.3.11.4 Output

- (1) Software integration test procedure
- (2) Software integration test record, including pass or failure results
- (3) Operational assumptions and constraints (updated)
- (4) Software (after software integration test)
- (5) Software integration test specifications (updated)

5.3.11.5 Review

For each output, a software integration test review shall be performed. Items to be reviewed shall be chosen based on 5.3.11.4 and shall be defined in the relevant plan, 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 machine

5.3.12.1 Activity

Activities that meet the following requirements shall be performed for software installation into the target machine:

- (1) Software shall be prepared in a form that allows installation into the target machine, and the configuration management information (filename, version information, and so on.) of the software shall be acquired.
- (2) According to configuration information after software integration, released items including the existing products shall be clarified.
- (3) Released software (in a form that allows installation into the target machine) shall be prepared and the configuration management of the software shall be performed.
- (4) The configuration management information of the software to be released and the installation procedure into the target machine shall be documented. The installation procedure shall include the confirmation before installation, installation methods, and the confirmation after installation.

Example of the confirmation items:

- (a) Status and behavior of the target machine
- (b) Software to be released
- (c) COTS items
- (d) Environment variables
- (e) Constraints during installation (including working hours, new or updated, security, and so on)
- (f) Network status
- (g) Constraints of the target machine
- (5) According to the installation procedure, the confirmation before installation, installation (this can be omitted if software has already been installed into the target machine) and the confirmation after installation shall be performed according to the installation procedure and the confirmation results of installation shall be recorded.

5.3.12.2 Input

- (1) Software (after software integration test)
- (2) Operational assumptions and constraints
- (3) Constraints of target machine

5.3.12.3 Output

- (1) Software prepared in a form that allows installation into the target machine
- (2) Installation procedure
- (3) Constraints when the software is installed into the target machine
- (4) Target machine installed software
- (5) Confirmation results of installation

- (6) Configuration management information

5.3.13 System integration and system integration test

5.3.13.1 Activity

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

- (1) Test preparation
 - (a) For system integration, the test environment that is equivalent to the operational environment shall be constructed for the target machine with software installed, considering network and interface with other systems.
 - (b) System integration test specifications and procedure (including pass or failure criteria) shall be documented in accordance with the software verification plan. With regard to the system integration test specifications, the following viewpoints shall be considered:
 - (i) Operational scenarios
 - (ii) Maximum load
 - (iii) Coverage for target system requirements specifications and target system architectural design specifications
 - (iv) Anomalies such as exceptions and failures
 - (v) Appropriateness of COTS or reused software items with the target system
 - (vi) Compliance with security requirements
 - (vii) Items related to external and internal interface specifications
 - (c) Problems found during test preparation (test procedure checks, and so on) shall be recorded and managed.
 - (d) Operational assumptions shall be clarified in the test preparation phase.
- (2) Implementation of tests
 - (a) The tests shall be performed in accordance with the system integration test procedure. The intermediate joint review of the test results shall be performed, as necessary, and judgment shall be made on whether the test should be continued or not.
 - (b) As necessary, tests based on the operational scenarios shall be performed by using tools such as simulators, and the verification coverage regarding operational scenarios shall be confirmed.
 - (c) With regard to the system integration test, information about software, test environment, test data, configuration, and version of the software under test shall be recorded to ensure the reproducibility of test conditions.
 - (d) System Integration test results shall be recorded and stored appropriately.
 - (e) When software or system integration test specifications need to be revised during the system integration test, their effectiveness shall be evaluated and the tests performed after the software coding and testing shall be performed again, as necessary.
 - (f) As the result of the system integration test, the information shall be updated if new operational assumptions and constraints items are identified.

5.3.13.2 Measurement

The following measurement shall be performed for system integration and the system integration test to evaluate the quality:

- (1) Collection of quality metrics data
 - (a) Problems found during system integration test shall be recorded together with related information such as information about test cases.
 - (b) Problems found during system integration test preparation (test procedure checks, and so on) shall be recorded together with related information such as information about the 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 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 managed and reported periodically, or for each milestone.

5.3.13.3 Input

- (1) Target system architectural design specifications
- (2) Software requirements specifications
- (3) Software design specifications
- (4) External interface specifications
- (5) Internal interface specifications
- (6) Software verification plan
- (7) Operational assumptions and constraints
- (8) Operational scenarios
- (9) Requirements specifications for target system

5.3.13.4 Output

- (1) System integration test specifications
- (2) System integration test procedure, including pass or failure criteria
- (3) System integration test record, including pass or failure results
- (4) Operational assumptions and constraints (updated)

5.3.14 Supply and introduction of software products**5.3.14.1 Activity**

Activities that meet the following requirements shall be performed for software supply and introduction of software products:

- (1) Software user's manual including installation manual and operational constraints shall be documented.
- (2) Software is ready to be supplied shall be confirmed by the review of software integration test completion and so on, and confirmation results shall be recorded.

- (3) Introduction procedure of software shall be documented. The introduction plan of the software, including the replacement of existing systems and temporary parallel operation shall be documented, as necessary.
- (4) Software products shall be introduced according to the introduction plan and procedure. And introduction results shall be recorded.
- (5) Identified maintenance requirements (preparation of spare parts, collection of updated information, recovery procedure, and so on) shall be arranged and transferred to the Maintenance Process (refer to 5.5).
- (6) If the configuration management information and installation procedure are newly added or identified, the configuration management information and installation procedure shall be updated.

5.3.14.2 Input

- (1) Software requirements specifications
- (2) Software design specifications
- (3) Software (in a form that allows installation into the target machine)
- (4) Installation procedure

5.3.14.3 Output

- (1) Software user's manual, including installation manual and operational constraints
- (2) Software products (supply format)
- (3) Confirmation results that software is ready to be supplied
- (4) Introduction plan
- (5) Introduction procedure
- (6) Record of introduction results
- (7) Maintenance requirements
- (8) Installation procedure (updated)

5.3.15 Software acceptance

5.3.15.1 Activity

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

- (1) With regard to acceptance inspection and acceptance testing, plans shall be established, subsequently, specifications and procedures shall be documented. If this is to be substituted by tests performed by the suppliers, the acquirers' approval shall be needed regarding the content of those tests.
- (2) Acceptance inspection and acceptance testing shall be performed in accordance with the plans, the specifications and the procedures described in (1). Records of the acceptance inspection and acceptance testing shall be maintained.
- (3) A review shall be performed regarding the record of confirmation that the software to be acquired is ready to be supplied.

5.3.15.2 Input

- (1) Software user's manual

- (2) Software products (supply format)
- (3) Record of confirmation that the software is ready to be supplied

5.3.15.3 Output

- (1) Acceptance inspection and testing plan
- (2) Acceptance inspection and testing specifications
- (3) Acceptance inspection and testing procedure
- (4) Acceptance inspection and testing record
- (5) Review of confirmation log indicating that the software is ready to be supplied

5.4 Operational process

The following operations and user support shall be performed regarding the software or the system:

- (1) Process implementation
- (2) Operational testing
- (3) Operation
- (4) User support

5.4.1 Process implementation

A plan for implementing this process including the organization responsible, shall be documented. As necessary, training and education plans for operators and users shall be documented. The following rules (operation standards and procedures) shall be established:

- (1) Problem management procedure for operation
- (2) Work procedure regarding the operation and user support of the software or the system

5.4.1.1 Establishment of problem management for the operation

In accordance with the problem resolution process (refer to 6.8), a problem report handling procedure shall be developed e.g. for receiving problem reports, recording, resolving, tracking problems, and notice of the status.

5.4.1.2 Establishment of operational procedures for the operation and user support

The operational procedure in order to operate software or system under operational environment, the test procedure, the procedure of transferring the problem report and modification request for the maintenance process (refer to 5.5), the procedures for releasing the software or system for operational use and the procedures for user support shall be established.

5.4.2 Operational testing

For each release of software or system, operational testing, including temporary and test operations shall be performed in accordance with the plan. The software or system shall satisfy the specified criteria to be released for operational use.

5.4.3 Operation

Activities that meet the following requirements shall be performed for the software or system operation:

- (1) Software shall be operated under the intended environment according to the software

user's manual and standard (operational standard and procedure).

- (2) Data related to problems of the software or system shall be collected and recorded. Based on monitoring the log and performance data, preventive actions shall be applied if signs of failure are found.
- (3) Based on the monitoring of problem reports, version-up information, and maintenance service information of COTS, the necessity of a software update shall be analyzed.
- (4) If any problem is identified during operation, it shall be resolved in accordance with the problem resolution process (6.8). If necessary, it shall be transferred to the related process such as the maintenance process (5.5) and so on. Problem resolution status shall be monitored until the problem is resolved.
- (5) If the security requirements such as security standards or policies and so on are applied, these shall be complied with. The information such as operation log data shall be monitored to find any illegal usage or modification.
- (6) Backup of the information such as software, data, setting, and so on, shall be planned and performed considering the operational conditions such as the target, the degree of importance, timing, and constraints during storing. Backup shall be planned so that backup will not disturb software usage and users shall be notified.

5.4.4 User support

Activities that meet the following requirements shall be performed for user support:

- (1) Support service shall be provided to the users. These request and subsequent actions taken for support shall be recorded and managed to provide support adequately.
- (2) If a problem is identified in the user support, it shall be resolved in accordance with the problem resolving process (refer to 6.8). As necessary, it shall be transferred to the related process such as the maintenance process (refer to 5.5) and so on. For problem resolution, the clarification of the problem (problem to be resolved), options for resolution, and a resolution plan, including the determination of the resolution method and schedule of completion of resolution, shall be reported to the users who require support for the problem. Problem resolution status shall be monitored until the problem is resolved.
- (3) If there is a temporary work-around for an identified problem, it shall be provided to users.
- (4) Education for operator and users shall be provided, as necessary.

5.5 Maintenance process

The following activities shall be performed for the modification of maintenance software products, including source code and related documents and COTS:

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

5.5.1 Process implementation

The plan and procedure for the maintenance process shall be documented. The maintenance process shall be started from the supply and introduction of software products process, not after the operation start:

- (1) Receipt, record, resolution, and trace of the problem report or modification request, and notification of the status shall be performed in accordance with the problem resolution process (refer to 6.8).
- (2) It shall be performed in accordance with the configuration management process (refer to 6.2) to manage modification of the current system.

5.5.2 Problem identification and modification analysis

Activities that meet the following requirements shall be performed for problem identification and modification analysis:

- (1) Modification shall be analyzed according to the following viewpoints:
 - (a) Modification objectives and categories (correction, improvement, prevention, and appropriateness to new environment, and so on)
 - (b) Scope (scale of modification, necessary cost, working hours for modification)
 - (c) Impact (performance, influence on safety and security, and so on)
- (2) Modification method shall be investigated based on the analysis. It is recommended that multiple modification methods are investigated so that it is possible to select the best modification method among them.
- (3) Reason of modification, analysis result of modification, and method of modification shall be recorded.
- (4) The selected modification shall be approved.

5.5.3 Modification implementation

Activities that meet the following requirements shall be performed for the modification implementation:

- (1) Software products to be modified and their version shall be determined and modification shall be performed.
- (2) Tests of the modified software products or tests of the system shall be performed and the test results shall be recorded. Based on the analysis of the modification and the scope of area affected by modification, a test for non-modification area may be limited to the area affected by modification.
- (3) Joint review shall be performed with organizations that have authority to approve the modification and it shall be confirmed that software or system modification is performed properly with no deficiencies and the modification shall be approved.
- (4) Modification shall be performed in accordance with the corresponding process such as the development process (refer to 5.3) and so on, as necessary.

5.5.4 Software reprogramming

Not applicable.

5.5.5 Migration

This process is an activity for migrating software products to a new environment.

5.5.5.1 Not used**5.5.5.2 Development and execution of migration plan**

A migration plan shall be developed and executed. In the plan, the following viewpoints and the viewpoints of documenting the introduction plan in 5.3.14 shall be considered:

- (1) Requirements analysis and definition for migration
- (2) Migration tools
- (3) Conversion of software product and data
- (4) Migration execution (parallel operation, as necessary)
- (5) Migration verification
- (6) Support for old environments in the future

5.5.5.3 Notification to users

The migration plan and contents of implementation shall be notified to users. Notifications shall include the following viewpoints:

- (1) Statement of why the old environment is no longer to be supported
- (2) Description of the new environment, with its date of availability
- (3) Description of other support options available, if any, once support for the old environment has been removed

5.5.5.4 Storage of old environments

The record such as documentation, logs, and other items that relate to old environments shall ideally be retained.

5.5.6 Software retirement and replacement

Activities that meet the following requirements shall be performed for software retirement and replacement:

- (1) Retirement plan to remove active support by an institution engaged in operation and maintenance shall be developed. Users shall be included to the development of the plan.
- (2) Users shall be given notification of the retirement plans and activities. Notifications shall include the following:
 - (a) Description of the replacement or upgrade of software or system with its date of availability
 - (b) Statement of why the software or system is no longer to be supported
 - (c) Description of other support options available, once support has been removed
- (3) Parallel operation of the retiring and the new software or system shall be performed to provide a smooth transition to new software or replacement to a new system. It is recommended that user training shall be provided during this period.
- (4) When the scheduled retirement arrives, notification shall be sent to all concerned parties. It is recommended that all associated development documentation, logs, and code shall be archived.
- (5) Data used by, or associated with, the retired or retiring software or system shall be accessible.

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 documentation process is a process for development,, production and revision, whose targets are documents produced throughout a software or system life cycle. Documents required for configuration management follow the configuration management process (refer to 6.2).

This process consists of the following activities:

- (1) Process implementation
- (2) Development
- (3) Production
- (4) Revision

6.1.1 Process implementation

A plan including the following shall be established as a documentation plan:

- (1) Documents developed through the software and system life cycle, and the documents schedule for development shall be clarified.
- (2) Procedures of the development of documents (development, inspection, and approval), production (production, distribution, and storing), and revision (revision and disposal) shall be decided.
- (3) Documentation standards and procedures applied to documentation shall be prepared.

6.1.2 Development

Documents shall be developed in accordance with the defined procedure. The following shall be considered:

- (1) Documents shall be developed in accordance with the rules such as documentation standards and procedures
- (2) Document number (Identifier) shall be appended to design documents
- (3) Whether source documents used for documentation are formally delivered shall be confirmed
- (4) Whether documents are implemented in accordance with the defined procedures and

documentation standards and whether technical content is adequate, shall be reviewed

- (5) Developed documents shall be produced after approval in accordance with a predetermined procedure

6.1.3 Production

Documents shall be produced in accordance with the defined procedures. The following shall be considered:

- (1) Documents shall be produced at adequate schedule according to the documentation plan. Suitable media such as paper, electronic, or others shall be used for documents. Documents shall be distributed to the predetermined parties after documents production.
- (2) The original documents shall be stored in accordance with their requirements (security management, backup, and so on).

6.1.4 Revision

Documents shall be revised in accordance with the defined procedure. The following shall be considered:

- (1) Documents must be reproduced in a manner that a unique revision identifier is assigned to the documents for identification.
- (2) The information such as reasons for revision shall be included, and changed parts shall be specified.
- (3) Revised documents shall be distributed to the predetermined parties.
- (4) In the case that an old version document is stored for the management of history and so on, an identifier indicating the document is outdated shall be assigned to the document to prevent any mistakes.
- (5) Revised documents shall be approved according to the defined procedure.

6.2 Configuration management process

The configuration management process is a process to administrate and ensure that the software configuration is correct and software changes are properly performed throughout the software life cycle.

This process consists of the following activities:

- (1) Process implementation
- (2) Configuration identification
- (3) Configuration change control
- (4) Record of configuration change status
- (5) Evaluation of configuration change status
- (6) Release management and delivery

6.2.1 Process implementation

For process implementation, the configuration management plan including the following shall be developed:

- (1) Configuration management activities

The objectives and contents regarding all activities performed as configuration management shall be clarified.

Examples of configuration management activities:

Technical change, deviation, waiver, and so on

(2) Procedures and timing for performing activities

Procedures for performing each activity shall be clarified by using documents and figures. Also, timing for performing activities (frequencies, milestones, conditions, and so on) shall be clarified.

(3) Responsible organization and concerned parties for performing activities

The responsible organization and concerned parties for performing activities shall be clarified. Also, activities of each organization shall be clarified and which activities are allocated to each organization shall be confirmed regarding all activities.

6.2.2 Configuration identification

For the configuration identification, the following shall be performed:

(1) Decision of configuration management items and units

Configuration management items and unit shall be decided for software products. An identifier (name or number) shall be appended to distinguish each management items and units combination.

Examples of configuration items:

Software, Source code, Load module, Data (Make file, Linkage map, Data file including Environment variables, Database), References (Requirements specifications, Design specifications), Media with electronic data, and so on.

(2) Definition of the version control system

A version system shall be defined to clarify change history of configuration management items and units.

(3) Establishment of baseline

Baseline shall be decided from the configuration management items and units defined in (1) and (2) above. Schedule of baseline establishment shall be clarified.

6.2.3 Configuration change control

The following shall be performed for the configuration change control:

(1) Identification and recording of change requests

To identify, retrieve and trace configuration management items, for each review, testing, delivery, modification, and so on, the configuration and version of the target software product shall be identified and its record with schedule shall be managed. Especially the records related to modification reasons, modification targets, and approval personnel shall be clarified so changes can be traceable.

(2) Analysis and evaluation of changes

Scope and degree of the impact (period, cost, feasibility, and so on) of the change shall be analyzed.

Validity of the analysis results above shall be evaluated by concerned parties. The analysis shall be performed in accordance with the problem resolution process (refer to 6.8), as necessary.

(3) Approval or disapproval of change requests

Based on the evaluation of the analysis results, approval or disapproval of change

requests shall be decided. The decision of approval or disapproval shall be decided by the responsible persons for identification and change of configuration management items.

(4) Implementation, verification, and release of modified software items

For modifying software products, the baselines of items in all the scope of impact identified by analysis shall be modified, and shall be released after verification, including operational verification, to verify the modification is appropriately performed.

Regarding items that handle safety or security critical functions, all access to the configuration management shall be managed and audited.

6.2.4 Record of configuration change status

Management records that show the status of configuration management items and the history of configuration management shall be prepared.

6.2.5 Evaluation of configuration change status

To ensure the functional and the physical completeness of the configuration items against their requirements, modification during the software life cycle shall be evaluated regarding the configuration status of software items, according to the following viewpoints:

- (1) Incorrect items are not included
- (2) Unintended items are not released
- (3) Documents are revised completely
- (4) Source code is produced based on the requirements of the baseline specifications
- (5) Necessary verification (regression test and so on) is performed

6.2.6 Release management and delivery

The following shall be performed for the release management and delivery:

- (1) Release and delivery procedures for the software products shall be established in advance and shall be managed in accordance with procedure.
- (2) It shall be confirmed that software products are not infected by viruses.
- (3) Released and delivered source code, and deliverable documents shall be maintained throughout the software or system life cycle.
- (4) For storing, packaging and delivery, source code and documentation that contain safety or security critical functions shall be handled in accordance with the policies of organizations.

6.3 Quality assurance process

The quality assurance process is a process for providing adequate assurance that activities and outputs, are performing, in accordance with their established plans that are based on this standard or by tailoring its results, during the life cycle of software or system.

This process consists of the following:

- (1) Process implementation
- (2) Product quality assurance
- (3) Process assurance
- (4) Assurance of quality system

6.3.1 Process implementation

This activity consists of the following tasks:

6.3.1.1 Organization

The management organization of the quality assurance process shall be clarified, and the implementation status and validity of quality assurance process shall be managed and reported. The management organization of the quality assurance process shall be authorized to maintain organizational freedom and authority to assess the problem and offer a resolution.

In addition, the person who is invested with all the responsibilities and authority for quality assurance process shall be independent of the development organization.

6.3.1.2 Quality assurance activity plan

For process implementation, the quality assurance activity plan including the following shall be established:

- (1) Identification of the system to be applied
- (2) Resources, quality standards, methodologies, procedures, and tools needed for performing the quality assurance process (including identification of all documents)
- (3) Selected activities and tasks from the processes, such as 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)
- (4) Organization, structure, responsibilities, education, and training in place to support the quality assurance activities
- (5) Schedules
- (6) Procedures for quality assurance activities such as joint reviews, and so on
- (7) Procedures for the work such as identification, collection, filing, maintenance, disposition and disposal of quality assurance activity records
- (8) Requirements and process regarding quality assurance for purchase management and supplier
- (9) Management for existing software items (COTS items or reused software items)
- (10) Procedures for the delivery

6.3.2 Product quality assurance

The following shall be performed for the products quality assurance:

- (1) It shall be assured that software products, or a system, and related documents are developed without insufficiency with the plans.
- (2) It shall be assured that the software products or system has fully met the requirements of, and is acceptable to, the acquirer as preparation for delivery.
- (3) Products quality shall be assured by IV&V as necessary.

6.3.3 Process assurance

The following shall be performed for the 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.

6.3.4 Assurance of quality system

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

This activity consists of the following tasks:

6.3.4.1 Education and training

All techniques, abilities, and qualifications needed for personnel engaged in development, maintenance and operation work with the software products or system shall be identified, and education and training shall be conducted.

6.3.4.2 Purchase management and supplier management

The following shall be performed for 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.

6.3.4.3 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.

6.3.4.4 Management of existing software items (COTS or reused software items)

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

(1) Any benefits of using existing software items

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

- (a) Appropriateness of existing software with regard to the developing software
- (b) Traceability relative to requirements applied to development software items
- (c) Risk obtained from the past performance of product, and so on, of software items to be used
- (d) Acceptance and assurance conditions
- (e) Associated documents, which are obtainable and usable
- (f) Introduction, preparation, training, and constraints
- (g) Identification of the software items such as versions and other details, and the configuration management method
- (h) Maintenance and future support
- (i) The rights such as intellectual property rights

6.3.4.5 Handling, storing, and labeling

To ensure appropriate handling and storing of software or system, requirements which include the following items, shall be documented, and the products shall be released in accordance with that document:

- (1) Media shall have labels (names, identifiers, and so on) so that stored software can be identified.
- (2) Identification of software shall be checked when reading software from media.

6.4 Verification process

The verification process is a process for confirming that the specified requirements are fulfilled. This process may include the checking works such as test, review, analysis, and so on.

This process consists of the following activities:

- (1) Process Implementation
- (2) Verification

6.4.1 Process implementation

For process implementation, activity including the following shall be established:

- (1) The degree of independence and authority level for the verification organization shall be decided and assured.
- (2) Task and software items which are targets for verification shall be decided and appropriate tasks shall be selected from the tasks defined in 6.4.2 according to their degree of importance.
- (3) Based upon the selected tasks, a verification plan shall be documented. Task, organization, and schedule applied to work and software items that are targets for verification shall be defined in the verification plan.

6.4.2 Verification

Verification shall be performed according to the verification plan. Identified problem shall be transferred to the problem resolution process (refer to 6.8). This activity consists of the following tasks:

6.4.2.1 Process verification

The process shall be verified considering followings:

- (1) Processes selected for the project shall be adequate
- (2) Planning of the processes in the software development plan shall be adequate
- (3) Adequate standards and environments shall be prepared for processes
- (4) An adequate number and level of competent staff shall be allocated to perform the processes
- (5) The processes shall be performed properly in accordance with the software development plan

6.4.2.2 Requirements verification

The requirements shall be verified considering followings:

- (1) The requirements shall be consistent, feasible, and verifiable.
- (2) Requirements for software items shall be appropriately allocated (not including requirements for hardware items and operation).
- (3) The higher level requirements and the standards applied to items shall be satisfied.
- (4) Concerning to the requirement to be especially taken care such as security and so on, it shall be able to show with the proper method that it satisfies the requirements.

6.4.2.3 Design verification

The design shall be verified considering followings:

- (1) The design shall meet requirements, and shall be traceable to requirements.
- (2) Shall be designed 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 shall have been covered.
- (4) Concerning to the design to be especially taken care such as security and so on, it shall be able to show the proper method that it satisfies the requirements and the standards applied to items.

6.4.2.4 Source code verification

The source code shall be verified considering followings:

- (1) Source code shall meet design, and shall be traceable to the design.
- (2) Shall be implemented properly with respect to data interface, timing, computer resources (especially 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 shall have been covered.
- (4) Concerning to the design to be especially taken care such as safety and security and so on, it shall be able to show the proper method that it satisfies the requirements and the standards applied to items.
- (5) Source code shall conform to defined coding standards and so on.

6.4.2.5 Integration verification

The integration shall be verified considering followings:

- (1) The configuration of the software items and data shall be in the proper and correct version.
- (2) The configuration of the software items and data shall have been completely and correctly integrated into the software with no deficiencies.
- (3) The software integration shall have been performed in accordance with the plan.

6.4.2.6 Documentation verification

The documentation shall be verified considering followings:

- (1) The documentation shall be adequate and consistent.
- (2) The documentation shall be planed and the documents shall be produced according to the appropriate schedule.
- (3) The documents required configuration management, shall conform to configuration management process (refer to 6.2).

6.5 Validation process

Validation is a process to confirm that the software or 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

For process implementation, activity including following shall be established:

- (1) The necessity of validation shall be decided.
- (2) The degree of independence and authority of the organization to perform validation shall be decided and assured.
- (3) 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

The validation shall be performed according to the validation plan. Problems identified by validation shall be transferred to the problem resolution process (refer to 6.8):

- (1) Test requirements and test cases shall be selected for validation, and test specifications shall be prepared.
- (2) Test requirements and test cases shall be ensured so that they correspond to the method and use of the software or system.
- (3) Tests shall be performed to include the following view points, as necessary:
 - (a) Testing with load, boundary, and illegal data
 - (b) Fault testing (functional degradation in failure, operation with an anomaly, and so on)
 - (c) Testing that representative users can successfully achieve their intended tasks
- (4) Validation shall be performed so that the software or system satisfies its intended use.
- (5) Tests shall be performed as appropriate in target environment. When a simulated instead of a real environment is used, the difference between them shall be evaluated.

6.6 Joint review process

Joint review process is confirmation task to ensure that a common understanding of progress toward agreed objectives and outcome satisfies stakeholders. The joint review process consists of the following activities:

- (1) Process implementation
- (2) Project management reviews
- (3) Technical reviews

6.6.1 Process implementation

For process implementation, activity including following shall be established:

- (1) Periodical reviews shall be held at predetermined milestones as specified in the plan such as software development plan.
It is recommended that ad hoc reviews shall be conducted when deemed necessary by either reviewing party or reviewed party.
- (2) All resources required to perform the reviews shall be agreed on between all the parties.

These resources include personnel, location, facilities, hardware, software products, and tools.

- (3) It is recommended that all the parties concerned shall agree on the following at each review:
 - (a) Matters to be reviewed
 - (b) Review scope and viewpoint
 - (c) Review method
 - (d) Entry and exit criteria for the review
- (4) Problems detected during the reviews shall be recorded and transferred to the problem resolution process (refer to 6.8), as necessary.
- (5) The review results shall be documented and distributed to concerned parties.
- (6) All the parties concerned shall agree on the outcome of the review and any action item responsibilities and closure criteria.

6.6.2 Project management reviews

The software project status shall be evaluated against the applicable project plans, and risks to the project's complete development in accordance with the plan shall be evaluated and managed. If a project delay is detected and it is difficult to complete the development according to the plan, a change of plan including a revision to the software specifications and the schedule shall be considered.

6.6.3 Technical reviews

For technical reviews, activities that meet the following requirements shall be performed:

- (1) Software items shall be evaluated from technical viewpoints, and risks regarding the implementation of software that meets the requirements specifications and standards shall be clarified. In case there is any difficulty to meet the requirements specifications and standards, readjustment of the method to apply the software specifications and the standards including the software requirements specifications changes or hardware shall be investigated.
- (2) Review reports shall be developed after each technical review is completed. Also, reviewers responsibility and 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.

6.7 Assessment process

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

This process consists of the following activities:

- (1) Following the directions by personnel who are responsible for performing an assessment (hereafter, referred to as the "sponsor"), assessment shall be planned, and the procedure shall be documented, and assessment shall be performed according to the plan.
- (2) The following shall be clarified for assessment implementation:
 - (a) Objectives and goals of the assessment
 - (b) Business objectives and goals for improvement

- (c) Assessment model and procedure
- (d) Scope of assessment and target process
- (3) The implementation status of the target process against this standards or its requirements shall be confirmed and shall be evaluated based on the development implementation status and the result of software development. The items to be improved shall be identified.
- (4) Assessment shall be performed by assessment team (consists of leader and assessor) including personnel who are well familiar with an assessment model, assessment methods and standards to apply.
- (5) Assessment results, including improvement offers on items needing improvement, are reported in documents to sponsors.

6.8 Problem resolution process

The problem resolution process is a process for analyzing all the problems found during the software life cycle process and for improving the software and for solving problems with the software.

The objective of this process is to provide applicable techniques for analyzing, improving, solving, and identifying trends.

This process consists of the following activities:

- (1) Process implementation
- (2) Problem identification
- (3) Problem analysis, improvement and resolution
- (4) Improvement and resolution status monitoring
- (5) Notification to concerned personnel

6.8.1 Process implementation

A procedure to implement this process shall be prepared.

- (1) A procedure including the following shall be established:
 - (a) Found problems shall be included in this process
 - (b) Existence of the problem shall be notified to concerned parties
 - (c) Resolution plan shall be decided by analyzing the problems and identifying the causes of the problem. Confirmation method after implementation shall be included in resolution plan
 - (d) The resolution plan shall be performed with appropriate work and technical direction
 - (e) Status shall be traced and records shall be documented, maintained, and notified to concerned parties
- (2) Responsibility and authority of each organization and internal organization shall be decided for the procedure of performing this process.
- (3) Criteria of classification such as type of problem, degree of importance and urgency shall be defined.
- (4) Record and report format through the identification of the problem to the completion of the resolution shall be defined.

6.8.2 Problem identification

Problems found during each process shall be handed over to this process and shall be

identified and recorded according to prepared management procedure.

6.8.3 Problem analysis, improvement and resolution

Regarding the identified problem, evaluation of the degree of necessity of improvement, resolution, and the analysis of root causes of the problem shall be performed, and then a feasible improvement and resolution plan shall be established depending on the degree of necessity.

- (1) In advance of finalizing the resolution plan, investigation and analysis to decide the appropriate resolution plan shall be performed with clarification of the causes and problem mechanisms and evaluation of its influence on product quality.

Root cause of the problem shall be clarified and reflected to the correction plan, according to the analysis of the background of the cause corresponding to the degree of importance.

- (2) Organization shall be established with members appropriate to the degree of characteristics and importance of the identification. Whether to perform the temporary plan shall be determined, as necessary. These shall be recorded, and shall be provided to the organizations that will participate in the resolution plan.

- (3) If technical changes become necessary for problem resolution, these shall be performed according to the configuration management process (refer to 6.2).

Analysis such as problem trend analysis based on numbers of identified problems shall be performed and preventive improvement and a resolution plan shall be established for expected problems if possible.

6.8.4 Improvement and resolution status monitoring

Regarding the improvement and resolution plan, status monitoring shall be performed according the method determined at problem analysis such as whether the finalized corrective action and preventive measures are continuously implemented and whether expected results are obtained or not, and so on.

6.8.5 Notification to concerned parties

Regarding such as identified problem, analysis result, information such as improvement and resolution plan, improvement and resolution status shall be written in predefined format and notified to the concerned parties periodically.

Problem reports shall be written in a form that can be used for problem identification, study, analysis, investigation of cause, resolution, and investigation of trend across the problems.

If problem management by using such as an inventory table is acceptable, a simplified format may be used in a scope it does not impact collection of quality metrics. And critical quality problems shall be quickly notified to acquire.

Appendix

Appendix I Development Process (Example)

Example of development process flow and primary output of waterfall model are shown.

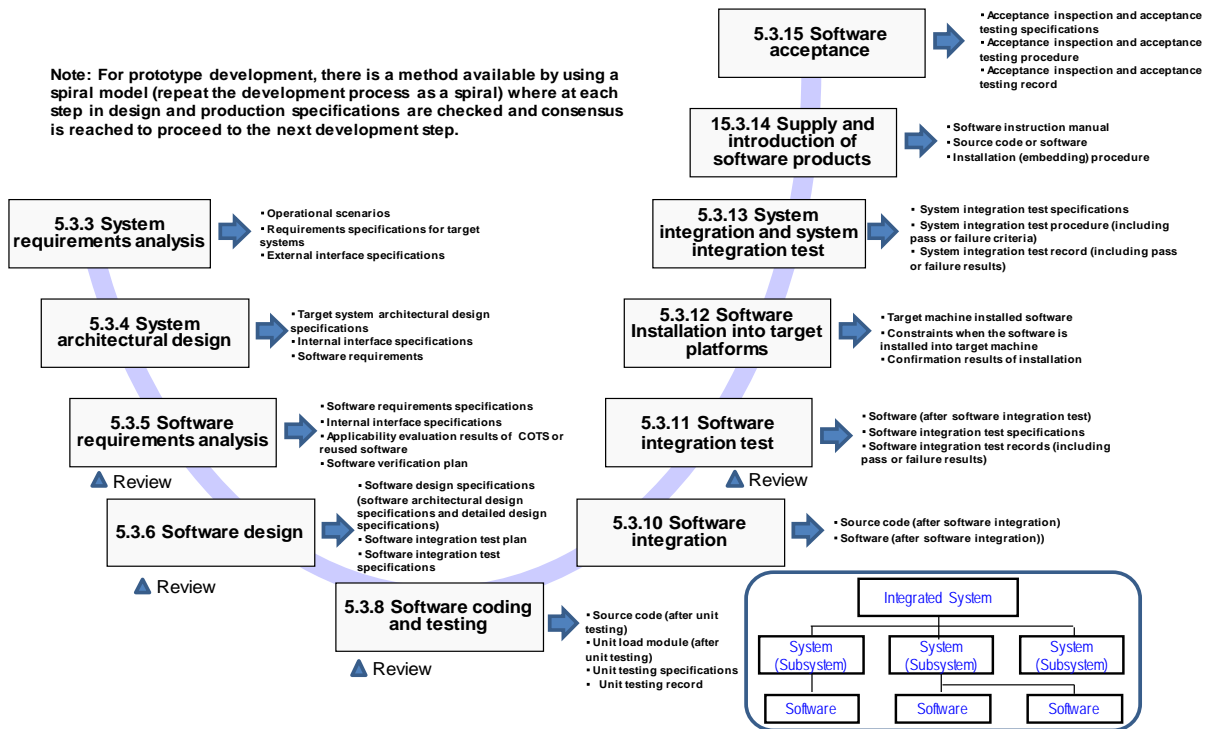


Figure I-1 Example of the development process of a ground system (waterfall model)

Table I-1 Primary inputs and outputs in each development process

Development Process		Input(primary)	Output(primary)
5.3.3	System requirements analysis	<ul style="list-style-type: none"> • Requirements for target system • Operational concept 	<ul style="list-style-type: none"> • Operational scenarios • Requirements specifications for target system • External interface specifications
5.3.4	System architectural design	<ul style="list-style-type: none"> • Operational scenarios • Requirements specifications for the target system • External interface specifications 	<ul style="list-style-type: none"> • Target system architectural design specifications • Internal interface specifications • Software requirements
5.3.5	Software requirements analysis	<ul style="list-style-type: none"> • Target system architectural design specifications • Internal interface specifications • Software requirements • Operational scenarios • Operational assumptions and constraints 	<ul style="list-style-type: none"> • Software requirements specifications • Internal interface specifications (updated) • Applicability evaluation results of COTS or reused software • Software verification plan
5.3.6	Software design	<ul style="list-style-type: none"> • Software requirements specifications • Applicability evaluation results of COTS or reused software • Software verification plan • Internal interface specifications • Operational scenarios 	<ul style="list-style-type: none"> • Software design specifications (software architectural design specifications and detailed design specifications) • Software integration test plan • Software integration test specifications
5.3.8	Software coding and testing	<ul style="list-style-type: none"> • Software design specifications • Internal interface specifications • Operational assumptions and constraints 	<ul style="list-style-type: none"> • Source code, unit load module (after unit testing) • Unit testing specifications • Unit testing record
5.3.10	Software integration	<ul style="list-style-type: none"> • Source code, unit load module (after unit testing) 	<ul style="list-style-type: none"> • Source code (after software integration) • Software (after software integration) • Debug record
5.3.11	Software integration tests	<ul style="list-style-type: none"> • Internal interface specifications • Software design specifications • Software requirements specifications • Operational assumptions and constraints • Software integration test plan • Software integration test specifications 	<ul style="list-style-type: none"> • Software (after software integration test) • Software integration test specifications • Software integration test record, including pass or failure result
5.3.12	Software installation into target machine	<ul style="list-style-type: none"> • Software (after software integration test) • Operational assumptions and constraints • Constraints of target machine 	<ul style="list-style-type: none"> • Target machine installed software • Constraints when the software is installed into the target machine • Confirmation results of installation
5.3.13	System integration and system integration tests	<ul style="list-style-type: none"> • Target system architectural design specifications • Internal interface specifications • External interface specifications • Software requirements specifications • Operational scenarios • Software design specifications 	<ul style="list-style-type: none"> • System integration test specifications • System integration test procedure, including pass or failure criteria • System integration test record, including pass or failure results
5.3.14	Supply and introduction of software products	<ul style="list-style-type: none"> • Software requirements specifications • Installation procedure • Software design specifications • Software (in a form that allows installation into the target machine) 	<ul style="list-style-type: none"> • Software products (supply format) • Record of introduction results • Software user's manual (including installation manual and operational constraints)
5.3.15	Software acceptance	<ul style="list-style-type: none"> • Software products (supply format) • Software user's manual 	<ul style="list-style-type: none"> • Acceptance inspection and acceptance testing specifications • Acceptance inspection and acceptance testing procedure • Acceptance inspection and acceptance testing record