SOFTWARE DEVELOPMENT STANDARD

Mar. 23 , 2016

Japan Aerospace Exploration Agency

This is an English translation of JERG-0-049A.
If there is anything ambiguous in this document, the original document (the Japanese version) shall be used to clarification.

Table of Contents

1 Scope

This standard applies to the activities relating to the development, operation, and maintenance of software for satellites, probe, launch vehicle and 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 document does not define the categories of personnel who implement processes. These personnel shall be defined by individual contracts or other agreements to which this standard is applied.

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

3 Terms, definitions and abbreviated terms

Term	Definition
Acceptance Inspection and Acceptance Testing	Acceptance inspection and acceptance testing are actions to evaluate compatibility with requirements at the time of acquiring software products. Inspection is an action to confirm the compatibility of a product, in accordance with evaluation criteria based on either requirements specifications or a predetermined value, by visually checking quantities and test results. Test comprises analysis, evaluation and checking of the functionality and capabilities of the software in order to obtain evaluation results and data required for use in inspection.
Activity	Activity is a component of a process and a set of strong correlative tasks.
Assessment	Assessment is a process to evaluate the strengths and the weaknesses of a subject process and to identify opportunities to improve the process for some predetermined purpose.
Computer system	Computer system is the entire system consisting of sets of software, platforms, and hardware, including the platform and hardware that are able to execute the targeted software for development. Although the definition of what a computer system contains is arbitrary, the definition shall be unique for software products that are subjected for development. As a computer system may be defined in various ways, from a one-chip microcomputer to multiple general-purpose computers connected to a network.
Configuration management	Configuration management is an action to define the configuration items, i.e. computer systems or projects, to record changes of content and to manage such aspects as their storage, handling, and distribution. If the software consists of multiple modules, then not only must the software version be managed, but the version of each software module must also be managed. In addition, configuration management items require software consistent modules, requirements specifications, operation manuals, and so on.

Term	Definition
COTS	COTS is the abbreviation of Commercial Off-The-Shelf. It has already been developed and is available in the commercial marketplace.
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 performance, safety, and reliability.
Operation	Operation is an action to carry out missions for the achievement of a purpose by means of an appropriate computer system. An operation utilizes the computer system from beginning to end, and includes monitoring and 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.
Project	Project is a time-limited endeavor to be implemented by means of specified resources and a temporary organization, with the purpose of fulfilling the project's mission.
Requirement	Requirement is one of a set of functions and performance targets requested for computer system or software and they may be also included such as not embodied and not detailed enough, or ambiguity in expression and vague expectations.
Requirements specification	Requirements specification is defined as the description of functions and performance required for computer system or software, embodied and specifically defined, and which also consider feasibility. 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 specification shall be complemented by the following methods: (1) It shall include the planning of agreement procedure with computer system users that the requirements specifications are satisfactory, and software verification plan shall include the planning of agreement procedure. (2) It shall include the test specifications enough 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.

Term	Definition
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.
Software	Software is a set of computer system configuration items comprising commands and data, which are executed or processed by a CPU to fulfill functions and capabilities defined in the software requirements specifications. If it is a set of commands and data which are implemented or managed by a CPU, it is categorized as software, and the software development process standards apply to it. However, for the driver of firmware, OS, and middleware, appropriate development processes are applied, in accordance with the characteristics and therefore may be removed from the software development process standard. For example, hardware and its integrated driver development shall be considered as such a case.
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 test specifications	Software test specifications are defined as the descriptions of test conditions and expected results, expressed unambiguously, to prove that software meets the requirements specifications. If the requirements specifications are represented in a verifiable format, they may be treated as appropriate software test specifications.
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, computer 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 of software development. A validation plan may be included.
Tailoring	Tailoring is defined as the activity to change processes defined in this standard in order to meet the project's particular characteristics and to establish an appropriate framework for each system development project.
Tasks	Tasks are components of activities corresponding to each stage of work.
Test plannability	Test plannability is defined as the aspect of test specification descriptions that indicates the possibility of testing and planning using the appropriate development phases and test environment for target test items.
The stability (maturity) of software requirements specifications	The stability (maturity) of software requirements specification is defined as the index which shows the possibility of specification change is small because of the software requirements specification be extracted and analyzed sufficiently. The definition of the index, and how it is evaluated, are arbitrary. Generally, provision for essential or refined changes to software requirements specifications affect process cost, delivery date and quality which let the software requirements specifications inputs. It is hoped that the index which is used to evaluate this shall be selected based upon the stability and maturity of software requirements specifications.
Traceability	Traceability is defined as the property which shows the correspondence with higher level documents.

Term	Definition
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 target software, and consist of processes implemented during development, operation, or maintenance. The supporting life cycle processes are processes that indirectly affect the software life cycle process, with reference specifically to the development of object software, and act to support a primary life cycle process and are called by other processes, as necessary.
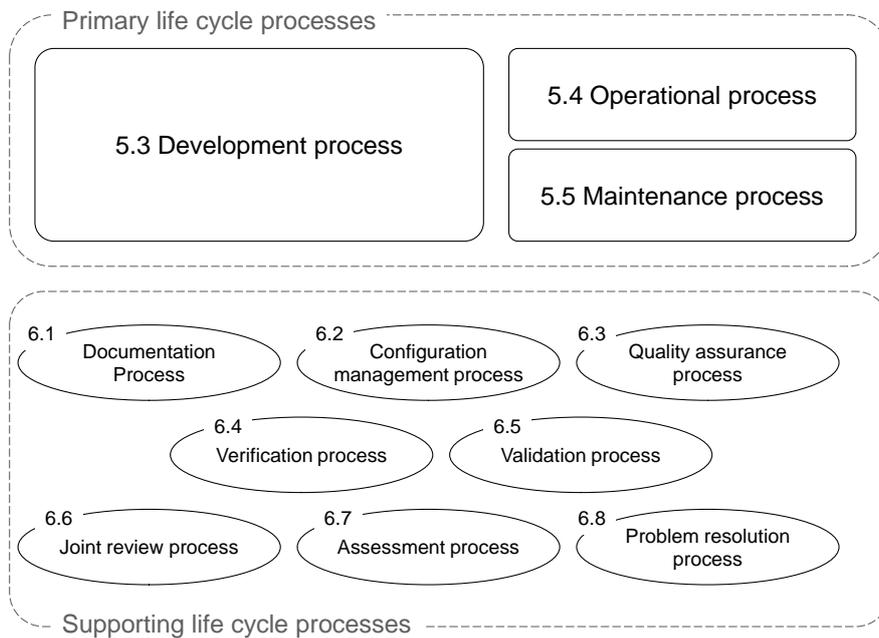
Figure 4.1 - Structure of the standard

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

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

Table 4.1 Process list

Process			Description
Primary life cycle processes	5.1	Not used	
	5.2	Not used	
	5.3	Development	Process to be implemented from the viewpoint of development. Requirements analysis, design, coding and testing, installation on target platforms (embedding), supply, introduction, acceptance, and so on.
	5.4	Operation	Process to be implemented from the viewpoint of operation. 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 confirmation that the process meets the requirements of this standard and processes are managed according to the plan.
	6.4	Verification	Process regarding the confirmation that specified requirements have been fulfilled, based on the provision of objective evidence.
	6.5	Validation	Process regarding the confirmation that the requirements for a specific intended use or application have been fulfilled, based on the provision of objective evidence.
	6.6	Joint review	Process regarding a joint review means it is conducted by multiple personnel with different viewpoints.
	6.7	Assessment	Process regarding the confirmation of the executing status and identifying the items which need to be improved.
	6.8	Problem resolution	Process regarding resolving problems which occur during implementation.

5　Primary life cycle processes

This clause defines the following primary life cycle processes:

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

5.1　Not used

5.2　Not used

5.3　Development process

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

(1) computer system requirements analysis process;
(2) computer system architectural design process;
(3) software requirements analysis process;
(4) software design process;
(5) software coding and testing process;
(6) software integration process;
(7) software integration test process;
(8) software installation into target platforms (embedding) process;
(9) computer system integration and computer system integration test process;
(10) supply and introduction of software product 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) The software development plan including the following information shall be established to cover:

    (a) Scope of computer system

    (b) Identification of target software

    (c) Definition of the identification of software development processes and their relationship (operation process, maintenance process, and so on are considered)

    (d) Definition of software development processes and activities*

        * Including activities regarding computer system with software installed.

    (e) Activities in each individual development process and their implementation management plan

    (f) Review plan

    (g) Preparation of development related documents' structure, and relationships of input and output in each development process

    (h) Documentation plan, including development department and schedule

    (i) Appropriate work allocations and methods for work plans (schedules, milestones), and progress management for each work

    (j) Environment to be used for software development and verification (simulator, real hardware, test environment, and so on)

    (k) Management plan for COTS and reused software. The following shall be included:

        (i) Identification of COTS items and reused software items

        (ii) Definition of the quality assurance process regarding COTS items and reused software items

    (l) Evaluation plan of compatibility with computer system

(2) Software development plan shall be documented

5.3.1.1 Output

(1) Software development plan

5.3.2 Items to be applied to all processes

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

(1) The software development plan shall be updated and managed in accordance with the development status.

(2) The progress of software development shall be monitored. It shall be reported to administrators as necessary.

5.3.2.1 Input

(1) Software development plan

5.3.2.2 Output

(1) Software development plan (updated)

(2) Software development progress report

5.3.3 Computer system requirements analysis

5.3.3.1 Activity

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

(1) Requirements extraction

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

(2) Requirements specification development

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

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

5.3.3.2 Input

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

5.3.3.3 Output

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

5.3.4 Computer system architectural design

5.3.4.1 Activity

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

(1) Computer system architecture shall be designed based on the requirements specifications for the computer system and the operational scenarios. Configuration items and their various categories (hardware, firmware, software, and operational) shall be clarified.

(2) Requirements pertaining to the requirements specifications for the computer system shall be allocated among the individual configuration items of the system.

(3) Feasibility of software items in fulfilling their allocated requirements shall be evaluated.

(4) Rationale for the design and preconditions (e.g. operational assumptions) for the computer system architectural design specifications shall be identified, and an appropriate evaluation shall be performed.

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

(6) Interface requirements for the software shall be extracted.

5.3.4.2 Input

(1) Operational scenarios

(2) Requirements specifications for the computer system

5.3.4.3 Output

(1) Computer system architectural design specifications

(2) Requirements for the software, including operational scenarios after the analysis

(3) Interface requirements

(4) Evaluation results of traceability relative to architectural design specifications with the computer system and requirements specifications for the computer system

5.3.5 Software requirements analysis

5.3.5.1 Activity

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

(1) Software requirements specifications shall be developed, based upon the analysis of the computer system architectural design specifications, interface requirements, and requirements for software including non-functional requirements.

(2) Identifiers shall be included in the individual software requirements specifications.

(3) Specifications for data and databases to be handled by the software shall be included in the software requirements specifications.

(4) Specifications for failure detection and handling functions shall be included in the software requirements specifications.

(5) Interface requirements shall be analyzed, and interface specifications shall then be developed. Agreement with the relevant parties regarding the interface specifications shall be made based on a common understanding and interpretation of the contents.

(6) Traceability and consistency of the software requirements specifications relative to the computer system architectural design specifications and interface requirements shall be analyzed and documented.

(7) Rationale of individual requirements of the software requirements specifications shall be clarified, and their feasibility shall be evaluated.

(8) If COTS or reused software is used, compliance with the software requirements specifications and its applicability with the computer system architectural design specifications shall be analyzed.

(9) Operational assumptions and constraints regarding software requirements specifications shall be extracted.

(10) Verifiability of the individual requirements of the software requirements and interface specifications shall be evaluated, and a software verification plan, including the validation method, shall be established.

(11) Software verification coverage pertaining to software function, performance, and operational scenarios in the verification plan shall be evaluated, and test plannability regarding the software requirements specifications and interface specifications shall be evaluated.

(12) With regard to the software verification plan, whether the test is affected by the behavioral difference between the test environment and the real hardware, or whether verification is performed by review, analysis and so on, without testing, the evaluation that shows the adequate identification and verification methods shall be included.

5.3.5.2 Measurement

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

(1) The definition of the data to be collected for evaluating the stability (maturity) of the software requirements specifications and their evaluation methods shall be defined. This measurement result is used as input for the project management, and is also referenced data for the post-project.

(2) Collection and evaluation of data defined in (1) above shall be planned.

(3) Collection and evaluation of data defined in (1) above shall be performed, and the results recorded.

5.3.5.3 Input

(1) Computer system architectural design specifications

(2) Requirements for software, including operational scenarios after the analysis

(3) Interface requirements

5.3.5.4 Output

(1) Software requirements specifications

(2) Interface specifications

(3) Software requirements specification traceability and consistency evaluation results

(4) Software requirements specification rationale and their feasibility evaluation results

(5) COTS or reused software applicability evaluation results

(6) Operational assumptions and constraints

(7) Software verification plan, including validation plan

(8) Verification completeness and test plannability for the software verification plan evaluation results

(9) Software requirements stability (maturity) evaluation results

5.3.5.5 Review

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

5.3.6 Software design

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

5.3.6.1 Activity

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

Software architectural design

(1) Functional decomposition and module partitioning shall be performed based on the software requirements specifications and the relationships between the modules comprising the functions shall be clarified, so that an appropriate software architectural design is performed.

(2) Software architectural design shall include the design and distribution of non-functional requirements (processing time requirements, requirements of resources such as memory, and so on) and be defined as software requirements.

(3) Interface specifications shall be detailed in accordance with the decomposition of the software functions and modules. Agreement with the relevant parties as to the interface specifications shall be arrived at based on a common understanding and common interpretation of the contents.

Software detailed design

(4) Each individual module shall be designed in accordance with the decomposition of functions and modules, and a software detailed design shall be performed.

(5) Interface specifications shall be detailed in accordance with the design of the module. Agreement with the relevant parties on the interface specifications shall be made based on a common understanding and interpretation of the contents.

Common to software architectural and detailed designs

(6) Traceability and consistency of the software design with the software requirements specifications, interface specifications, and with the necessary related documents shall be analyzed and documented.

(7) As necessary, with regard to the individual software design, the design rationale shall be clarified and its feasibility evaluated.

(8) If COTS or reused software is used, its applicability with the software design shall be analyzed.

(9) Operational assumptions and constraints regarding the software design shall be identified.

(10) Software test plans and specifications shall be established in accordance with the software verification plan.

   (a) For software test specifications, the following shall be considered:
      (i) Operational scenarios
      (ii) Interface specifications
      (iii) Maximum load for assumed scenarios
      (iv) Coverage for software requirements specifications and software design specifications
      (v) Anomalies, such as exceptions and failures
      (vi) Applicability of COTS or reused software items with computer system

(11) If new operational assumptions and constraints arise or are identified, they shall be updated.

5.3.6.2 Measurement

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

(1) Definition of the data to be collected and of the evaluation methods for progress management and risk evaluation of software design shall be defined.

(2) Collection and evaluation of data defined in (1) above shall be planned.

(3) Collection and evaluation of data defined in (1) above shall be performed, and the results recorded.

5.3.6.3　Input
(1)　Software requirements specifications

(2)　Interface specifications

(3)　Operational assumptions and constraints

(4)　COTS or reused software applicability evaluation results

(5)　Software verification plan

5.3.6.4　Output
(1)　Software architectural and detailed design specifications

(2)　Interface specifications (updated)

(3)　Software design traceability and consistency evaluation results

(4)　Software design rationales and their feasibility evaluation results

(5)　COTS or reused software applicability evaluation results (updated)

(6)　Operational assumptions and constraints (updated)

(7)　Software test plan

(8)　Software test specifications

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

5.3.7　Not used

5.3.8　Software coding and testing

5.3.8.1　Activity
Activities that meet the following shall be performed for the software coding and testing:
(1)　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 the software design specifications and interface specifications.

(4)　Source code shall be developed based on the defined coding standard.

(5)　Unit testing specifications shall be developed in accordance with a software verification plan and a software test plan.

(6)　Unit testing shall be performed in accordance with the unit testing specifications, and the test results shall be recorded in a format that it allows determination of pass or failure.

(7)　For unit testing, the criteria for the test coverage for source code shall be established and the test shall be performed so that these criteria are satisfied. The branch of source code shall be well covered at least.

(8)　Static analysis shall be performed with a source code checking tool or equivalent, and the source quality shall be evaluated.

(9) Traceability of the source code and software design specifications shall be analyzed and the results shall be recorded.

(10) 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 software coding and unit test in order to evaluate the quality:

(1) The definition of data to be collected for evaluating source code quality, and the evaluation method, shall be defined.

(2) Collection and evaluation of data defined in (1) above shall be planned.

(3) Collection and evaluation of data defined in (1) above shall be performed and the results recorded.

(4) Results of the evaluation in (3) above shall be reported periodically, or for each milestone.

5.3.8.3 Input

(1) Software design specifications

(2) Interface specifications

(3) Operation assumptions and constraints

(4) Software verification plan

5.3.8.4 Output

(1) Source code

(2) Operation assumptions and constraints (updated)

(3) Unit testing specifications

(4) Unit testing record

(5) Traceability analysis record

(6) Source code quality evaluation results

5.3.8.5 Review

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

5.3.9 Not used

5.3.10 Software integration

5.3.10.1 Activity

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

(1) Software shall be integrated and the baseline shall be determined after software integration.

(2) Debug information collected during software integration shall be recorded and, as necessary, related processes such as the problem resolution (refer to 6.8) and configuration management (refer to 6.2) shall be implemented.

5.3.10.2 Measurement

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

(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.

(2) Collection and evaluation of data defined in (1) above shall be planned.

(3) Collection and evaluation of data defined in (1) above shall be performed and the results recorded.

(4) Results of evaluation in (3) above shall be reported periodically, or at every milestone.

5.3.10.3 Input
(1) Source code (unit)

5.3.10.4 Output
(1) Source code (integrated)
(2) Software (integrated)

5.3.11 Software integration test

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

(1) Test preparation

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

  (b) For software test, the following shall be considered:

    (i) Operational scenarios

    (ii) Interface specifications

    (iii) Maximum load for assumed scenarios

    (iv) Coverage for software requirements specifications and software design specifications

    (v) Anomalies, such as exceptions and failures

    (vi) Applicability of COTS or reused software items with computer system

  (c) Software integration test procedures shall be documented in accordance with the software verification plan, the software test plan, and the software test specifications.

  (d) If new operational assumptions and constraints arise or are identified, they shall be updated.

  (e) Test specifications and procedures shall be checked.

(2) Implementation of tests

  (a) The tests shall be performed in accordance with the software integration test procedure.

  (b) During the software integration test, quick reviews of the test results shall be performed as necessary, and judgment shall be made about whether the test shall be continued or not.

  (c) With regard to the software integration test, information about the test environment, test data, configuration and version of the software under test shall be recorded to ensure the reproducibility of test conditions.

  (d) Test results shall be recorded and stored appropriately, and they shall be presented as necessary.

  (e) If software or software integration test specifications need to be revised during a software integration test, the effectiveness of the activities such as the joint reviews, verifications, and validations performed formerly shall be evaluated and tests shall be performed again as

15

necessary.

5.3.11.2 Measurement

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

(1) Collection of quality metrics data

    (a) Problems found during software integration tests shall be recorded, together with related information such as test cases.

(2) Quality index data setting

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

    (a) Metrics for quality evaluation during tests shall be set

    (b) Identified data shall be collected

    (c) Analysis evaluation method for the identified data shall be defined

    (d) Analysis and evaluation of identified data shall be performed

(3) Result of data analysis and evaluation shall be reported periodically or for each milestone

5.3.11.3 Input

(1) Interface specifications
(2) Software requirements specifications
(3) Software design specifications
(4) Software verification plan
(5) Software test plan
(6) Software test specifications
(7) Operation assumptions and constraints
(8) Source code (integrated)
(9) Software (integrated)
(10) Operational scenarios

5.3.11.4 Output

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

5.3.11.5 Review

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

5.3.12 Software installation into target platforms (embedding)

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

(1) Software shall be prepared in a form that allows installation (embedding) into the target platform, and the configuration management information (filename, version information, and so on) of the software shall be acquired.

(2) Configuration management information of software to be released shall be prepared, to include the installation (embedding) procedure into the target platform, the installation (embedding) result check procedure, and the operational assumptions and constraints.

(3) Software shall be installed into the target platform in accordance with the installation (embedding) procedure. However, this can be omitted if the software has already been installed into the target platform.

(4) A check that the software has been properly installed (embedding) shall be performed, in accordance with the results check procedure.

5.3.12.2 Input
(1) Software
(2) Operation assumption and constraints

5.3.12.3 Output
(1) Software prepared in a form that allows installation (embedding) into the target platform
(2) Configuration management information
(3) Installation (embedding) procedure
(4) Computer system installed (embedding) software
(5) Installation (embedding) result check procedure

5.3.13 Computer system integration and computer system integration test

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

(1) Test preparation
  (a) Computer system integration shall be performed.
  (b) Computer system integration test specifications and procedures shall be documented in accordance with the software verification plan.
  (c) With regard to the computer system integration test specifications, the following viewpoints shall be considered:
    (i) Operational scenarios
    (ii) Software requirements specifications
    (iii) Maximum load
    (iv) Coverage regarding requirements specifications for a computer system and computer system architectural design specifications
    (v) Anomalies, such as exceptions and failures

(vi) Applicability of COTS or reused software items with the computer system

(d) Problems found during test preparation (test procedure checks, and so on) shall be recorded and managed.

(2) Implementation of tests

(a) The tests shall be performed in accordance with the computer system integration test procedure.

(b) As necessary, based on the operational scenarios, tests shall be performed by using tools such as simulators, and the verification coverage regarding operational scenarios shall be checked.

(c) With regard to the computer system integration test, information about the test environment, test data, configuration, and version of the software under test shall be recorded to ensure the reproducibility of test conditions.

(d) Test results shall be recorded and stored appropriately.

(e) When software or computer system integration test specifications need to be revised during a computer system integration test, the performance effectiveness of the activities such as the joint reviews, verification, validation, tests, and others shall be evaluated and performed again, if necessary.

5.3.13.2 Measurement

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

(1) Collection of quality metrics data

(a) Problems found during the computer system integration tests shall be recorded, together with related information such as information about test cases.

(b) Problems found during the test preparation (test procedure checks, and so on) shall be recorded, together with related information such as information about test cases.

(2) Quality metrics data definition

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

(a) The metrics for quality evaluation during computer system integration tests shall be defined.

(b) Identified data shall be collected.

(c) The analysis evaluation method for the identified data shall be defined.

(d) Analysis and evaluation of identified data shall be performed.

(3) The result of data analysis and evaluation shall be reported periodically, or for each milestone.

5.3.13.3 Input

(1) Requirements specifications for computer system

(2) Computer system architectural design specifications

(3) Software requirements specifications

(4) Software verification plan

(5) Operational assumptions and constraints

(6) Operational scenarios

(7) Computer system installed software

5.3.13.4 Output

(1) Computer system integration test specifications

(2) Computer system integration test procedure

(3) Computer system integration test record, including pass or failure judgment results

(4) Operational assumptions and constraints (updated)

5.3.14 Supply and introduction of software product

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

(1) Documentation of manuals

　　Software user's manuals shall be documented.

(2) Preparation for supply

　　The software that is ready to be supplied shall be confirmed, and the confirmation results shall be recorded.

(3) Establishment of an introduction plan

　　The introduction of software shall be planned, including replacement of the existing system and temporary parallel operation, and a procedure shall be documented. For the introduction plan and procedure, not only installation (embedding) of the software into the target platform, but also work for introducing computer system into the actual operational environment, shall be considered.

(4) Implementation of introduction and recording of results

　　Software shall be installed (embedded) into the target platform in accordance with the introduction plan and procedure, and the computer system shall be introduced to the actual operational environment. The results of the introduction shall be recorded.

5.3.14.2 Input
(1) Software requirements specifications

(2) Software design specifications

(3) Source code or software

(4) Installation (embedding) procedure

5.3.14.3 Output
(1) Software user's manual

(2) Source code or software

(3) Confirmation results that software is ready to be supplied

(4) Installation (embedding) procedure

(5) Introduction plan

(6) Introduction procedure

(7) Record of introduction results

5.3.15 Software acceptance

5.3.15.1 Activity
 Activities that meet the following 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 is 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 confirmation of the results that the software to be acquired is ready to be supplied.

5.3.15.2   Input
(1) Software user's manual
(2) Source code or software
(3) Configuration management information
(4) Record of confirmation that software is ready to be supplied
(5) Installation (embedding) procedure

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

5.4   Operational process
The following operations and operational support for users shall be performed regarding the software or the computer system:
(1) Process implementation
(2) Operational testing
(3) Operation of the software and the computer system
(4) User support

5.4.1   Process implementation
A plan for implementing this process shall be documented and executed. Moreover, a standard for the operation shall be established.

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 software or the computer system operation and user support
The procedures for entering the test procedure and modification request, for the software or computer system in the operational environment, to the maintenance process (refer to 5.5), and the procedures for releasing the software or computer system for operational use, shall be established.

5.4.2   Operational testing
For each release of the software or the computer system, operational testing shall be performed. After

the software or the computer system satisfies the specified criteria, it shall be released for operational use.

5.4.3 Operation of software or computer system
The software or computer system shall be operated in its intended environment according to software user's manuals.

5.4.4 User support
The following activities shall be performed for the user support:
(1) Support service shall be provided to the users. These requests 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 procedure established in 5.4.1.1.
(3) If there is a temporary work-around for identified problem, it shall be provided to users.

5.5 Maintenance process
The maintenance process is defined when the modification and maintenance of the software products, including design documents, requires a change of development environments. The following activities shall be performed:
(1) Process implementation
(2) Problem identification and modification analysis
(3) Modification implementation
(4) Software reprogramming
(5) Migration
(6) Software retirement

5.5.1 Process implementation
The maintenance plan for implementation of the maintenance process shall be documented and implemented. The following shall be included in the plan:
(1) Maintenance management method for developed software products
(2) The structure and related documentation
(3) Data, including the management information necessary for maintenance
(4) Record
(5) Maintenance environment (environment of development phase, and so on)
(6) Actions for maintaining other related processes appropriately (problem resolution process (refer to 6.8), configuration management process (refer to 6.2), and so on)

5.5.2 Problem identification and modification analysis
(1) In accordance with the plans, the analysis of the requested contents regarding maintenance or modification of the software product, and correspondence shall be evaluated.
(2) If the software modification and reprogramming are implemented, agreement for the modification plan shall be obtained.

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

5.5.4 Software reprogramming
   This standard shall also be applied to the work such as the partial modification of the software which is called patches, and to the addition and extension of software functions during operational periods.

   *This does not imply that their development processes shall be exactly the same as the development-to-retirement processes of the software that is being used for operations. For projects assuming reprogramming, the development-to-retirement processes regarding reprogramming shall be concretized and tailored in advance.

   In principle, this standard shall be applied and the compatibility shall be evaluated.

5.5.5 Migration
   This activity is a process for migrating software 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 shall be considered:
   (1) Requirements analysis and definition for migration
   (2) Migration tools
   (3) Conversion of software product and data
   (4) Migration rehearsal
   (5) Migration execution
   (6) Migration confirmation
   (7) Future support for old environments

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

5.5.6 Software retirement
   The following viewpoints shall be considered for the software retirement:
   (1) A retirement plan to remove active support by an institution engaged in operation and maintenance shall be developed. Users shall be included in 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 to the software or computer system with its date of availability

    (b)   Statement of why the software or computer 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 computer system shall be performed, for smooth transition to the new computer 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 computer 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 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 computer system life cycle, and the documents schedule for development shall be clarified.

(2)　Procedures for the development of documents (development, inspection, and approval), issue (issue, distribution, and storing), and revision (revision and disposal) shall be decided.

(3)　The form (content, format, and so on) appropriate for such documents shall be decided.

6.1.2　Development
Documents shall be developed in accordance with the decided procedure. The following shall be considered:

(1)　The appropriateness of sources of information for the documents shall be confirmed.

(2)　The documents shall be checked and approved regarding format, technical content, against their documentation standards.

6.1.3　Production
Documents production shall be implemented in accordance with the defined procedures. The following shall be considered:

(1)　Production and distribution of documents shall be at the appropriate version.

(2)　The documents shall be managed in accordance with requirements (security management, backup, and so on) including parties for distribution.

6.1.4 Revision

Documents shall be revised in accordance with the decided procedure. In addition, the revised documents shall be checked and approved regarding the form and technical content.

6.2 Configuration management process

Configuration management process is a process of applying the following administrative and technical procedures through the software life cycle:

(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

The configuration management plan shall be developed, and shall include the following:

(1) Configuration management activities
(2) Procedures and schedule for performing these activities
(3) The organization responsible for performing these activities
(4) The relationship with other organizations

6.2.2 Configuration identification

For configuration items and its versions, the following shall be defined:

(1) Baseline
(2) Version references
(3) Other identification details

6.2.3 Configuration change control

The following shall be identified and defined for the configuration changes:

(1) Identification and recording of change requests
(2) Analysis and evaluation of changes
(3) Approval or disapproval of change requests
(4) Implementation, verification, and release of modified configuration items
(5) Traceable review records against the reason for the modification, and authorization of the modification
(6) The control of accesses and the audit to controlled configuration items that handle safety or security critical functions

6.2.4 Record of configuration change status

Management records and status reports that show the status and history of controlled configuration items including baselines shall be prepared.

6.2.5 Evaluation of configuration change status

The functional completeness of the configuration items against their requirements and the physical completeness of the configuration items shall be ensured.

6.2.6　Release management and delivery

The release and delivery of software product shall be formally controlled in accordance with the procedure. The source code and documentation that contain safety or security critical functions shall be handled, stored, packaged, and delivered in accordance with the policies of organizations involved.

6.3　Quality assurance process

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

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 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

The quality assurance activity plan shall include the following:

(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 review
(7)　Procedures for the works, 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 Products quality assurance

(1) It shall be assured that the software products, or computer system, and its related documentation are developed without insufficiency with the plans.

(2) It shall be assured that the software products or computer 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

(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:

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 or computer system shall be identified, and education and training shall be conducted.

6.3.4.2 Purchase management and supplier management

(1) Purchase management

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

(2) Suppliers and purchase vendor selection

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

6.3.4.3 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. The items below are to be managed for applied projects and future reuse:

(1) Any objectives, reasons, and benefits in using existing software items

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

    (a) Applicability of existing software with regard to the developing software

    (b) Traceability relative to requirements applied to development software items

    (c) Risk obtained from the information such as the past performance of product, and so on, of software items to be used

    (d) Acceptance and assurance conditions

    (e) Associated documents, which are obtainable and usable

    (f) Introduction, preparation, training, and constraints

(g) Identification of 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, storing, and labeling of software or computer 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 consists of the following activities. This process may include the checking works such as test, review, analysis, and so on:

(1) Process Implementation

(2) Verification

6.4.1 Process implementation

(1) The necessity of verification and the degree of organizational independence for the verification organization shall be decided.

(2) If the project requires verification, a verification process including the following shall be established:

(a) An organization responsible for performing the verification shall be selected. This organization shall be assured of its independence and authority to perform the verification process.

(b) The target for verification shall be determined, and appropriate tasks defined in 6.4.2 below shall be selected according to degree of importance.

(c) Based upon the selected tasks, a verification plan shall be documented.

(d) The verification (test, review, analysis and so on) shall be performed according to the verification plan. Problems detected through verification shall be resolved in accordance with the problem resolution process (refer to 6.8).

6.4.2 Verification

This activity consists of the following tasks:

6.4.2.1 Process verification

The following shall be considered:

(1) Adequacy of the processes selected for the project

(2) Planning of the processes is adequate

(3) Applicable standards and environment for the project's processes in place

(4) Adequate levels of competent staff allocated to the processes

(5) Proper execution of the processes

6.4.2.2 Requirements verification

The following shall be considered:

(1) The requirements are consistent, feasible, and verifiable.

(2) Requirements for software items are 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 safety and security and so on, it is able to show the proper method that it satisfies the higher level requirements and the standards applied to items.

6.4.2.3 Design verification

The following shall be considered:

(1) The design shall meet requirements, and shall be traceable to requirements.

(2) Designed properly with respect to data interface, timing, computer resource (memory capacity, processing speed, and so on), logic design, processing sequence and processing contents (especially initialization, termination, exception handling and so on).

(3) The characteristics such as portability, modifiability and ease of problem resolution have been covered.

(4) Concerning to the design to be especially taken care such as safety and security and so on, it is able to show the proper method that it satisfies the requirements and the standards applied to items.

6.4.2.4 Source code verification

The following viewpoint shall be considered:

(1) Source code shall meet design, and shall be traceable to the design.

(2) Implemented properly with respect to data interface, timing, computer resources (memory capacity, processing speed, and so on.), logic design, processing sequence and processing contents (especially initialization, termination, exception handling and so on).

(3) The characteristics such as portability, modifiability and ease of problem resolution have been covered.

(4) Concerning to the source code to be especially taken care such as safety and security and so on, it is able to show the proper method that it satisfies the requirements and the standards applied to items.

(5) Source code shall conform to e.g. coding standards.

6.4.2.5 Integration verification

The following viewpoint shall be considered:

(1) The configuration of the software modules and data is in proper and correct versions. The software components and units of each software item have been completely and correctly integrated into the software item.

(2) The modules and data configuring software have been completely and correctly integrated into the system.

(3) The integration tasks have been performed in accordance with the plan.

6.4.2.6 Documentation verification

The following shall be considered:

(1) The documentation is adequate and consistent.

(2) The documentation has been organized in accordance with the plan.

(3) Configuration management of the documentation is performed in accordance with the proper procedures.

6.5 Validation process

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

This process consists of the following activities:
(1) Process implementation
(2) Validation

6.5.1 Process implementation

(1) The necessity of validation and the degree of organizational independence for the validation organization shall be determined.

(2) If the project requires validation, the validation process shall be established and include the following:

    (a) An organization responsible for performing the validation shall be selected. This organization shall be assured of its independence and authority to perform the validation process.

    (b) A validation plan shall be developed. The plan shall include the following:

        (i) Items subject to validation

        (ii) Validation tasks to be performed

        (iii) Resources, responsibilities, and schedule for validation

        (iv) Procedures for distributing validation reports

    (c) Validation shall be performed according to the validation plan. Problems detected by validation shall be resolved in accordance with the problem resolution process (refer to 6.8).

6.5.2 Validation

(1) Test requirements and test cases shall be selected for validation, and test specifications shall be prepared.

(2) Test cases shall be checked to ensure they reflect the method and use of the software or computer system.

(3) Tests in (1) and (2) above shall be performed to include the following view points, as necessary:

    (a) Testing with stress, boundary, and illegal data

    (b) Fault testing

    (c) Testing that representative users can successfully achieve their intended tasks

(4) Validation shall be performed so that the software or computer system satisfies its intended use.

(5) Tests for the software or computer system shall be performed as appropriate in the target environment. When a simulated instead of a real environment is used, the difference between them shall be evaluated.

    *As a validation method, methods other than testing (analysis, modeling, simulation, and so on) may be appropriate.

6.6 Joint review process

The joint review process consists of the following activities:

  (1) Process implementation

  (2) Project management reviews

  (3) Technical reviews

6.6.1 Process implementation

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

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

  (2) All resources required to perform the reviews shall be agreed on between all the parties. These resources include personnel, location, facilities, hardware, software, 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 appropriate action taken through the problem resolution process (refer to 6.8), as necessary.

  (5) The review results shall be documented and distributed.

  (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 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 schedule and software requirements specifications shall be considered.

6.6.3 Technical reviews

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

  This activity consists of the following tasks:

6.6.3.1　Review

  (1) The following shall be implemented for review preparation:

      (a) Reviewers shall be selected for review implementation. Reviewers shall include appropriate personnel, consisting not only of parties to the work but also personnel who have enough sufficient knowledge of the area under review, as well as project parties which include interface designers and the originator of the requirements.

      (b) The review subjects and purposes shall be clarified, and materials for a review shall be completed in advance.

      (c) The review purposes, subjects and reviewers shall be clarified, and documented.

  (2) Software items shall be evaluated in a step-by-step approach as developments progress.

  (3) Technical actions regarding software shall be evaluated.

  (4) Review reports shall be developed after the review is completed. Also, quantitative data, such as

the evaluation time, and the number of questions and comments at the review, shall be recorded, and a quality evaluation for the review shall be implemented.

6.6.3.2   Walk-through

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

(1)   As preparation for the walk-through, the reviewed party shall provide the relevant documents and codes, including those still in development.

(2)   Parties necessary for walk-through (such as persons in charge of higher level process, persons in charge of lower process, and so on) shall check the documents and codes.

(3)   Questions and problems found through walk-through shall be recorded and followed up until the resolution is completed, and mutually agreed.

6.7   Assessment process

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

This process consists of the following activities:

(1)   Process implementation

(2)   Assessment implementation

6.7.1 Process implementation

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

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

(2)   Personnel who are well informed of assessment methods and relevant standards shall be selected as assessors.

(3)   Assessment results, including improvement offers on items needing improvement, are reported in documents to sponsors.

6.7.2 Assessment implementation

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

6.8   Problem resolution process

The problem resolution process consists of the following activities:

(1)   Process implementation

(2)   Problem resolution

(3)   Prevention

(4)   Problem trend analysis

6.8.1 Process implementation

The following shall be determined in advance for preparation of problems (notifications) occurrence regarding software products:

 (1) Software products for management

 (2) Management period

 (3) Set of procedures for resolution (action)

If a problem has occurred, it shall be recorded and managed in accordance with the list above.

6.8.2 Problem resolution

 This activity consists of the following tasks.

6.8.2.1 Problem identification

 If a problem has occurred, it shall be identified in order to be managed. It is recommended that it shall be in prioritized for resolution. Relevant parties shall be informed.

6.8.2.2 Investigation and analysis

 The problem phenomenon, occurrence condition, and the cause shall be investigated, analyzed, and clarified. Problem resolution shall be requested to the other organizations, as necessary.

6.8.2.3 Consideration for problem resolution

 The following methods of problem resolution shall be considered:

 (1) Ways of avoiding modification to, or of decreasing the impact on, the software product shall be considered.

 (2) Determination of whether the software product requires a change shall be considered.

  In addition, it is desirable that multiple options for resolution be prepared, including solutions which do not modify the software. If software modifications are required, the options must be evaluated by taking into consideration the points of view, such as the cost, the schedule, the overall risk, and the extent of the impact.

6.8.2.4 Determination of problem resolution plan

 Plans shall be documented and agreed with the parties:

 (1) As a temporary measure, if there is a way of mitigating the problem, plans shall be documented and agreed with parties.

 (2) With respect to a permanent solution, the problem resolution methods shall be documented and agreed with parties.

6.8.2.5 Problem resolution implementation

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

6.8.2.6 Recording and monitoring

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

6.8.3 Prevention

 Preventive measures shall be implemented if the risks that need to be prevented can be assessed.

6.8.4 Problem trend analysis

 Problem trend analysis shall ideally be performed.

Appendix
Appendix I　Example of the overall configuration of the software development process (Water fall type)
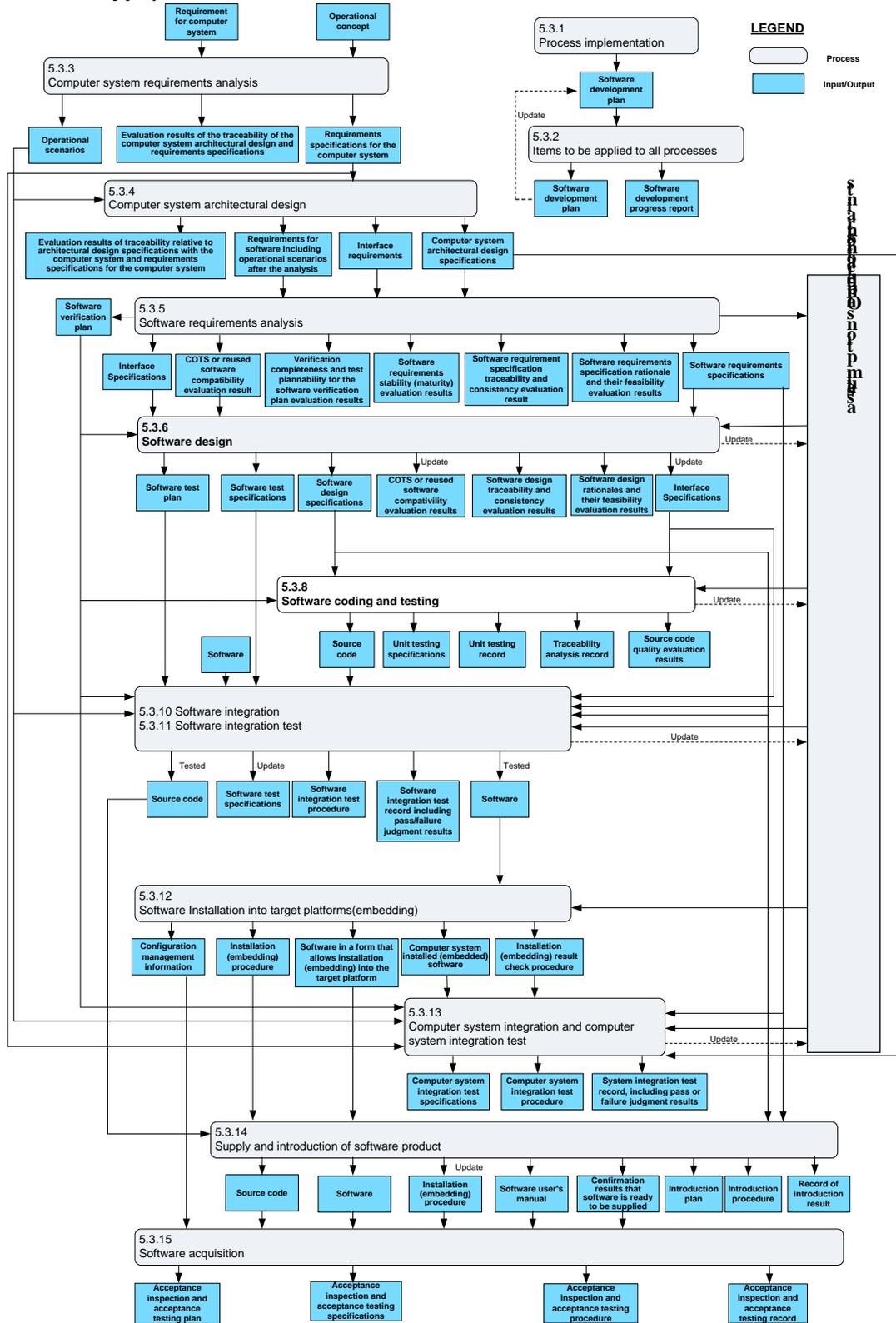

Figure I-1 Example of the overall configuration of the software development process

Appendix II Matrix of input, output and processes

Table II-1 Matrix of input, output and processes

Process names (columns 5.3.1–5.3.15):
- 5.3.1 Process implementation
- 5.3.2 Items to be applied to all processes
- 5.3.3 Computer system requirements analysis
- 5.3.4 Computer system architectural design
- 5.3.5 Software requirements analysis
- 5.3.6 Software design
- 5.3.7 Not used
- 5.3.8 Software coding and testing
- 5.3.9 Not used
- 5.3.10 Software integration
- 5.3.11 Software integration test
- 5.3.12 Software installation onto target platforms(embedding)
- 5.3.13 Computer system integration and computer system integration test
- 5.3.14 Supply and introduction of software product
- 5.3.15 Software acquisition

Input / Output	5.3.1	5.3.2	5.3.3	5.3.4	5.3.5	5.3.6	5.3.7	5.3.8	5.3.9	5.3.10	5.3.11	5.3.12	5.3.13	5.3.14	5.3.15
Software development plan	O	M													
Software development progress report		O													
Requirements for computer system			I												
Operational concept			I												
Operational scenarios			O	I							I		I		
Requirements specifications for computer system			O	I									I		
Evaluation results of the traceability of the computer system architectural design and requirements specifications			O												
Computer system architectural design specifications				O	I										
Requirements for the software including operational scenarios after the analysis				O	I										
Interface requirements				O	I										
Evaluation results of traceability relative to architectural design specifications with the computer system and requirements specifications for the computer system				O											
Software requirement specifications					O	I					I		I	I	
Interface specifications					O	M		I			I				
Operational assumptions and constraints					O	M		M			M	I	M		
COTS or reused software applicability evaluation result					O	M									
Software verification plan including validation plan					O	I		I			I		I		
Verification completeness and test plannability for the software verification plan evaluation results					O										
Software requirements stability (maturity) evaluation results					O										
Software requirements specification traceability and consistency evaluation results					O										
Software requirements specification rationale and feasibility evaluation results					O										
Software architectural or detailed design specifications						O		I			I			I	
Software design traceability/consistency evaluation results						O									
Software design rationales and their feasibility evaluation results						O									
Software test plan						O					I				
Software test specifications						O					M				
Unit testing specifications								O							
Unit testing record								O							
Recording unit testing								O							
Traceability analysis record								O							
Source code quality evaluation results								O							
Source code								O		M	M			M	I
Software										O	M	I			
Software in a form that allows installation (embedding) into the target platform												O		M	I
Configuration management information												O			I
Installation (embedding) procedure												O		M	I
Software integration test procedure											O				
Software integration test record including pass/failure judgment result											O				
Computer system installed (embedding) software												O	I		
Installation (embedding) result check procedure												O			
Computer system integration test specifications													O		
Computer system integration test procedure													O		
Computer system integration test record including pass or failure judgment results													O		
Software user's manual														O	I
Confirmation results that software is ready to be supplied														O	
Record of confirmation that software is ready to be supplied															I
Introduction plan														O	
Installation (embedding) procedure														O	
Record of introduction results														O	
Acceptance inspection and testing plan															O
Acceptance inspection and testing specifications															O
Acceptance inspection and testing procedure															O
Acceptance inspection and testing record															O

Legend: I->Input,   O->Output,   M->Input and Output(updated)   <Typical>

Appendix III "Verification" and "Validation"

The difference in concept between "Verification" and "Validation" is shown in the below example.
The purpose of "Verification" is the confirmation that the specified requirements, determined in the previous process, have been fulfilled.
The purpose of "Validation" is the confirmation that the requirements for a specific intended use or application have been fulfilled, and that it makes the accumulated difference from the requirements, occurred with the progress of development, smaller.

The explanation of verification and validation:

"Verification" refers to the provision of objective evidence, that specified requirements have been fulfilled.

"Validation" refers to the provision of objective evidence, that the requirements for a specific intended use or application have been fulfilled.
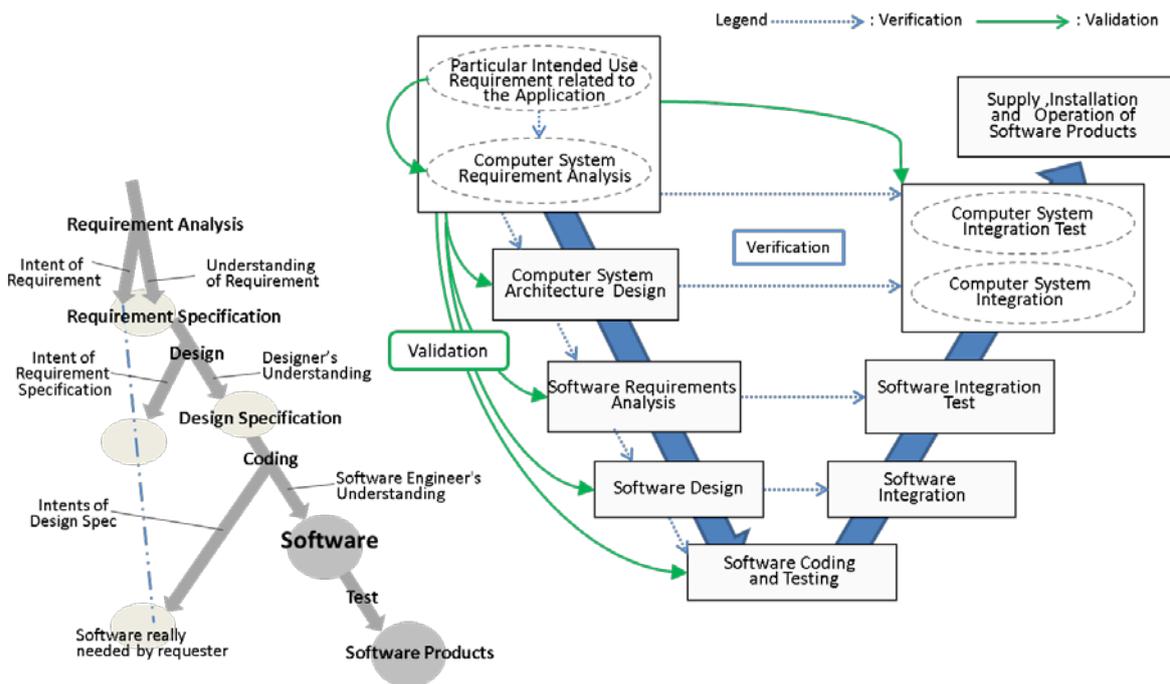

Figure III-1 Concept of verification and validation

Appendix IV Relationship between the problem resolution process and another process

If problems occur during an operation, the relationship between the problem resolution process and the operation process are shown in below.
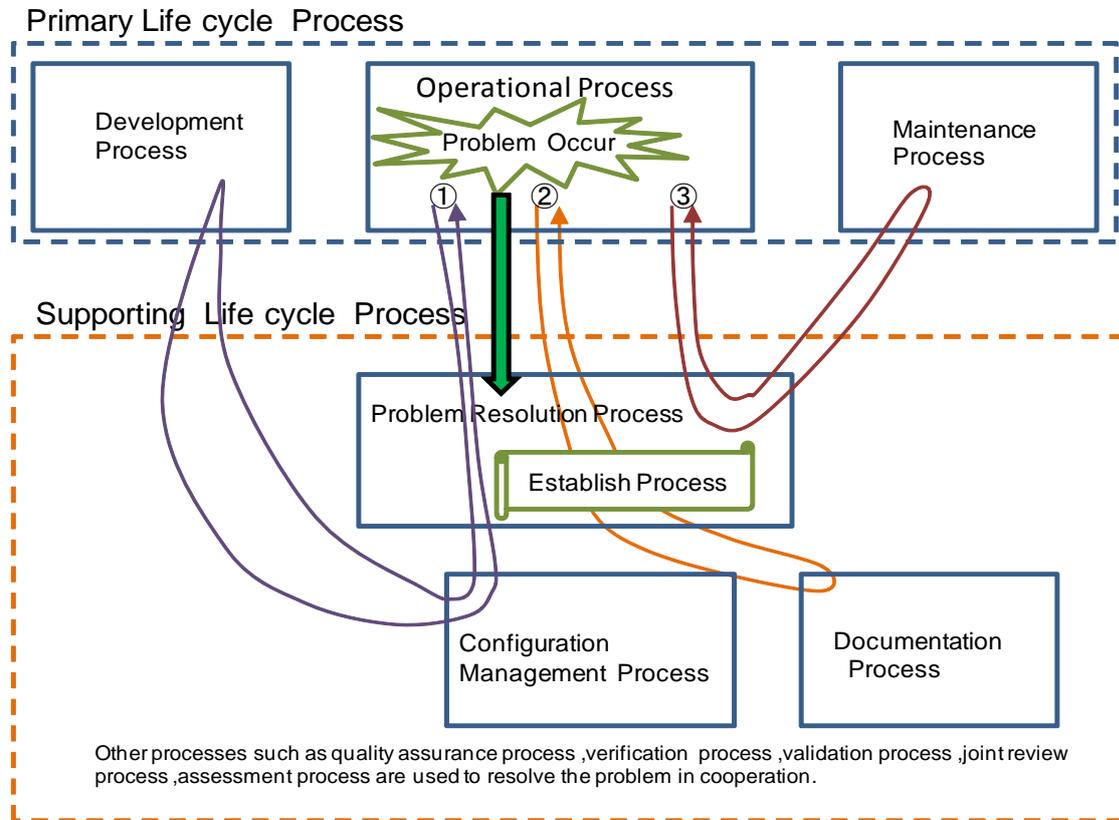

Figure IV-1 Relationship between the problem resolution process and another process

[(1)In case of reworked software]

Problems occur during operation. → Sent to the problem resolution process. As a result of the examination, a failure is identified and the countermeasure (modification of the software) is determined. → Take out software from the configuration management process. → Rework software in the development process. → Return software to the configuration management process. → Verify the results in the problem resolution process. → Return to the operation process.

[(2)In case of a document revision]

Problems occur during operation. → As a result of the examination, here was an identified document problems in the problem resolution process. → Fix in the documentation process. → Verify the results in the problem resolution process. → Return to the operation process.

[(3)In case of countermeasure by operation]

Problems occur during operation. → As a result of the examination with the problem resolution process, a failure is determined, and a countermeasure is determined without modifying the software in order to support the operation. → Implement countermeasures operation in the maintenance process. → Check the results in the problem resolution process. → Return to the operation process.