GUIDANCE NOTES
GD11-2015

China Classification Society

GUIDELINES FOR SAFETY AND RELIABILITY ASSESSMENT FOR SHIPBOARD SOFTWARE

2015

Contents

1 Scope and description

1.1 This Guide is a guide for the safety and reliability assessment for the software of shipboard computer systems (hereinafter referred to as the computer systems, including the programmable electronic systems), which specifies the technical requirements for the safety and reliability in the development, testing, authentication, production and maintenance for the software of shipboard computer systems, and also specifies a number of requirements for the hardware of the corresponding software; these requirements shall be treated together with the technical requirements for the products. This section is applied to the computer system installed on the classified ships, and the system provides control, alarm, monitoring or safety functions including the programmable electronic systems, which meet the classification requirements.

1.2 This Guide mainly focuses on the lifecycle of software development with some processes in the integral safety lifecycle adopted. The V model is adopted as the development model for the software in this Guide. And the evolution of relevant models is not contained in this Guide.

1.3 Given the direct application of small simple computer systems and the application of some functions in complex systems, this Guide defines the small low complexity computer system and gives a simplified assessment method in Appendix2.

1.4 When this Guide is applied, the documentation referred in this Guide can be prepared according to the internal document management system of the manufacturer while the content of the documentation shall conform to the relevant content referred to in this Guide.

1.5 Additional markings

1.5.1 The following additional markings can be granted based on different system classifications (see Article 4.1 for the classification) for the computer systems with system lifecycle established, whose system and hardware are certificated according to the Rules for classification of sea-going steel ships and other requirements and meet all the technical requirements for software specified in this Guide:

(1) SLC1 for Category I system;

(2) SLC 2 for Category II system;

(1) SLC 3 for Category III system.

1.6Threeappendixes are attached to this Guide, which are:

1.6.1Appendix1, the verification form of testing and inspection for on-site ship surveyor to assess the safety and reliability of the software in the shipboard computer system.

1.6.2Appendix2,theassessment method for small low complexity computer system.

1.6.3 Appendix3, the technical proposal for the design and realization stages of the computer system.

2 Normative references

2.1 The following references are necessary to the application of this Guide. For the dated reference documents, only the referred versions are applicable while for those not dated, the latest versions shall be applicable.

Reference documents Table 2.1

1.		*Rules for the Classification of Sea-going Steel Ships* (2012) issued by China Classification Society and its modification notifications
2.		Article 55, Chapter II-1, International convention for the safety of life at sea
3.	GD01-2006	Guidelines for type approval test of electric and electronic products issued by China Classification Society
4.	IACS UR E22	Application and utilization of programmable electronic system on ships
5.	IEC 61508-1：2010	Functional safety of electrical/electronic/programmable electronic safety-related systems – Part 1: General requirements
6.	IEC 61508-2：2010	Functional safety of electrical/electronic/programmable electronic safety-related systems – Part 2: Requirements for electrical/electronic/programmable electronic safety-related systems
7.	IEC 61508-3：2010	Functional safety of electrical/electronic/programmable electronic safety-related systems – Part3: Software requirements
8.	IEC 61508-4：2010	Functional safety of electrical/electronic/programmable electronic safety-related systems – Part 4: Definitions and abbreviations
9.	IEC 61508-5：2010	Functional safety of electrical/electronic/programmable electronic safety-related systems – Part 5:Examples of methods for the determination of safety integrity levels
10.	IEC 61508-6：2010	Functional safety of electrical/electronic/programmable electronic safety-related systems – Part 6: Guidelines on the application of Part 2 and 3
11.	IEC 61508-7：2010	Functional safety of electrical/electronic/programmable electronic safety-related systems – Part7: Overview of techniques and measures

Reference documents **Table 2.1**

12.	IEC 60092-504：2001	Electrical installations in ships – Part 504:Special features – Control and instrumentation
13.	IEC 60812-2006	Analysis techniques for system reliability – Failure mode and effect analysis (FEMA)
14.	IEC 61025-2006	Fault tree analysis
15.	IEEE 730-2014	Software quality assurance plan
16.	ISO 9000-3	Guidelines for the application of ISO 9001 quality assurance standards in the development, supply, installation and maintenance of computer software
17.	ISO 17894-2005	Ships and marine technologies – Computer applications - General principles for the development and use of programmable electronic systems in marine applications

3 Terms and abbreviations

3.1 Terms

3.1.1 software

Intellectual creation comprising the programs, procedures, data, rules and any associated documentation pertaining to the operation of a data processing system.

3.1.2 computer system)

Consisting of hardware, software and I/O units based on computer technology.

Note: The term contains the microelectronic device based on one or more than one CPUs and relevant memories.

3.1.3 system

A group of interacting elements based on the design, which may include the interacting hardware, software and personnel, etc.

3.1.4 Subsystem

A model element with the semantics of package (other model elements may be included) and class (behaviors may be included) . The behavior of the subsystem is supplied by the classes contained in it or other subsystems. One or multiple interfaces are realized in the subsystem, which define the behavior that can be executed by the subsystem.

3.1.5 Module

A function set of programs, discrete components and packaged programs or a group of merged discrete components.

3.1.6 Software module

Construct that consists of specifications and/or data descriptions and can also interact with other such constructs.

3.1.7 Safety function

Function to be implemented by a computer system, other technical safety-related system or external risk reduction facilities to achieve or maintain a safe state for the EUC, in respect of specific hazardous event.

3.1.8 Equipment under control (EUC)

Equipment, machinery, apparatuses and/or plants used for manufacturing, processing, transportation or other activities

3.1.9 Small low complexity computer system

A computer system, in which the failure modes of each individual component are well-defined; and the behavior of the system under fault conditions can be completely determined.

Note: Behavior of the system under fault conditions may be determined by analytical and/or test methods.

3.1.10 Dynamic testing

Executing software and/or operating hardware in a controlled and systematic way, so as to demonstrate the presence of the required behavior and the absence of unwanted behavior.

3.1.11 Quality plan

Documents of quality control measures, resources and activity sequence specified for specific software products and projects, which are a series of documents describing the relevant quality standards and stating how to meet the corresponding standards.

3.1.12 System lifecycle

Necessary activities involved in the implementation of the system, occurring during a period of time that starts at the concept phase of a project and finishes when all the computer systems and their related items are no longer available for use.

3.1.13 Software lifecycle

Activities occurring during a period of time that starts when software is conceived and ends when the software is permanently decommissioned.

Note: A software lifecycle typically includes requirement phase, development phase, test phase, integration phase, installation phase and modification phase etc.

3.1.14 Software Configuration Management(SCM)

A technology for modification identification, organization and control, applying to the whole software operation.

3.2 Abbreviations

3.2.1 ISO: International Organization for Standardization

3.2.2 IEC: International Electrotechnical Commission

3.2.3 IEEE: Institute of Electrical and Electronics Engineer

3.2.4 FMEA: Failure Mode and Effects Analysis

3.2.5 FMECA: Failure Mode, Effects and Criticality Analysis

3.2.6 FAT: Factory Acceptance Test

3.2.7 PE: Programmable Electronic

4 Categorization of computer systems

4.1 Based on the fault effects and the functions supplied by the system, the computer systems shall be classified into three categories as Category I, II and III as specified in Table 4.1. Effects shall be analyzed based on the harshest environment during the classification.

Categorization of computer systems **Table 4.1**

Category	Effects	System functionality
I	Those systems, failure of which will not lead to dangerous situations for human safety, safety of the vessel and/or threat to the environment.	- Monitoring and daily management functions
II	Those systems, failure of which could eventually lead to dangerous situations for human safety, safety of the vessel and/or threat to the environment.	- Monitoring and alarm functions - Control functions which are necessary to maintain the ship in its normal operational and habitable conditions
III	Those systems, failure of which could immediately lead to dangerous situations for human safety, safety of the vessel and/or threat to the environment.	- Control functions for maintaining the vessel's propulsion and steering - Safety functions

4.2 Some examples of the categorization of computer systems are described in Table 4.2 for reference of the manufacturer.

Examples of system categories **Table 4.2**

System category	Example
I	Maintenance support systems Daily information process
II	Monitoring and alarm equipment Tank capacity measuring equipment Control system for auxiliary machinery Main propulsion remote control systems Fire detection and extinguishing systems Bilge water systems Governors
III	Machinery protection systems/equipment Burner control system Electronic fuel injector for internal combustion engine Control systems for propulsion and steering Synchronizing units for power generator

5 Quality system requirements

5.1 Quality assurance system

5.1.1 It shall be demonstrated that the manufacturer has certain product quality assurance ability and quality management level and that a management system ensuring that the product meets the rules issued by China Classification Society and relevant conventions has been established by the manufacturer via the quality assurance system.

5.1.2 The manufacturer shall establish and implement ISO9001 or equivalent quality management system and hold validate certificates. As is required in ISO9001, the manufacturer shall control the process related to the management, resource supply, and realization, measurement, analysis and improvement of the software products.

5.1.3 The manufacturer of Category II and III shall also meet the applicable requirements in ISO9000-3.

5.2 Software quality plan

5.2.1 The manufacturer shall make a quality plan for the lifecycle of software development.

5.2.2 The software quality plan shall regulate the software activities in the whole lifecycle and define relevant procedures, responsibilities and systematic documents. IEEE 730 can be referred to when making the quality plan.

5.2.3 For software of Category II and III system, safety function requirements shall be included in the quality plan, whose specific assurance method shall be designed to verify and affirm whether the safety function requirements are met.

5.2.4 Configuration management shall be developed for the shipboard computer system during the lifecycle of software development. See Article 5.6 for details.

5.3 Quality control in production

5.3.1 The product quality shall be ensured by means of practicable quality assurance measures, plans and organization.

5.3.2 The manufacturer shall have documents of product quality control which shall describe the producing process flow accurately and also describe the quality control requirements for each process flow with words, diagrams and table; and it shall contain specific control objects, standards and methods, inspection methods and all the supporting documents ensuring the implementation of quality assurance measures in the production. The supporting documents for passing the "test and simulation" are required for products with functions related to safety.

5.4 Final test report

5.4.1 The manufacturer shall make final tests for the products and submit the report. The final test report shall be a report prepared according to the tests of the finished products and the test results.

5.5 Software traceability

5.5.1 Requirements for software traceability: the modification of programming content and data as well as the version change shall be marked and documented according to the procedures. It shall be ensured that the forming process of software quality can be traceable when required. The procedures for the modification of programming content and data as well as the version change shall be defined in the software configuration management, version statements and other quality assurance documents. And the modifications and changes shall be determined in the documents.

5.5.2 The mentioned documents shall be kept for at least one year from the end of the lifecycle of software development. The manufacturer shall have solid evidence to prove that the software has retired indeed.

5.6 Configuration management

5.6.1 Configuration management is used to ensure the consistency of deliverable items in several developments when some deliverable items are changed. In general, it consists of hardware and software configuration management.

5.6.2 Requirements:

(1) Administrative and technical controls shall be applied during the lifecycle of software development to manage software changes and ensure that all the provisions for software safety are always met.

(2) It shall be ensured that all the necessary operations are executed to state that all the required software demands are obtained.

(3) The unique and accurate identification for all the configuration items that are necessary to maintain the integrity of the computer system shall be maintained. The configuration items include the following at least: safety analysis and requirements, software rules and design documentation, software source code modules, testing plan and results applied to the software components and packages of the computer system, and all the tools and development environments for the creation, testing or implementation of the software of computer system.

(4) Change control procedure shall be applied to avoid unlicensed modifications and the modification requests shall be documented; the effect of modification shall be suggested in the analysis to approve or reject the request; the details and authorization for all the approved modification shall be documented; the composition of all the software baselines(including the reconstruction of early baseline).

(5) The information for configuration status, release status, judgment and passing of all the modifications and the details of modification shall be documented for approval.

(6) The software release shall be officially documented. The main software backup and all the relevant documents shall be kept within the development lifecycle of the released software for maintenance and modification.

6 System lifecycle

6.1 Division of system lifecycle

6.1.1 The system lifecycle is divided into 5 stages, which respectively are concept, requirement, implementation, verification and running. Each stage is divided according to its different scopes of purposes, with a relationship and requirements in following table and figure.

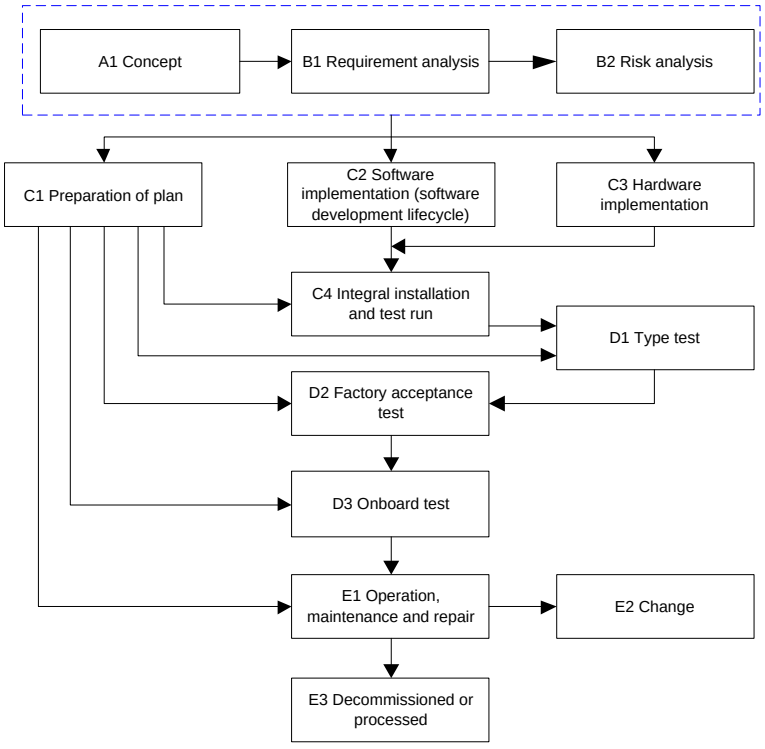


Fig. 6.1.1 System Lifecycle

Overview of System Lifecycle **Table 6.1.1**

System Lifecycle Stage		Purpose	Requirement	Input	Output
Box Number of Fig. 6.1.1	Title				
A Concept					
A1	Concept	Improve the understanding of controlled equipment and its environment (actual and legal) to meet the needs of the implementation of its lifecycle activities.	Ensure full understanding on controlled equipment and required control function and actual environment; determine the possible source of danger; obtain relevant information on determining danger; get the existing safety regulations; take into account the danger produced by interaction between adjacent controlled equipment; and keep the above required information and results in documentation.	All information necessary to meet all such requirements	Information obtained from concept to overall scope
B Requirement					
B1	Requirement analysis	All works required for describing the purpose, scope, definitions, and function of new system when creating or modifying the computer system. These include: determine the		Information obtained from concept to overall scope	Requirements and specifications of computer system

Overview of System Lifecycle **Table 6.1.1**

		computer system boundary; stipulate the scope of risk analysis			
B2	Risk analysis (only for functions related to safety)	Prove that the system enters a fail-safe condition in case of a single fault and the system in service would not loss or drop to the performance standard unacceptable of CCS in order to ensure the safety and reliability of computer system.	Adopt proper methods for analysis such as: - fault tree analysis; - risk analysis; - FMEA or FMECA analysis	Requirements and specifications of computer system	Requirements and specifications of functions related to safety (including information and records distributed according to safety requirements)
C Implementation					
C1	Preparation of plan.	Prove the computer system meets the requirements of installation, operation and maintenance by prescribed and technical procedures.	Develop a plan of installation, operation and maintenance of computer system, to ensure to keep the safety of required functions in operation and maintenance. Prepare the FAT test and onboard test programs that function tests related to safety are included.	Requirements and specifications of computer system; Requirements and specifications of functions related to safety.	Plan for software quality; installation, operation and maintenance plans for computer system; type test program; FAT test program; and onboard test program.
C2	Software implementation	Establish the computer system software conforming to requirements and specifications of computer system and of functions related to safety.	See Chapter 7 for software development lifecycle in details.	Requirements and specifications of computer system	Verification of each computer system that conforms to the requirements and specifications. See Chapter 7 for software development lifecycle in details.

Overview of System Lifecycle **Table 6.1.1**

C3	Hardware implementation	Establish the computer system hardware that conforms to requirements and specifications of computer system and of functions related to safety.		Requirements and specifications of computer system.	Verification of each computer system that conforms to the requirements and specifications.
C4	Integral installation and test run	Install the computer system; Give a test run for computer system;		Installation, operation and maintenance plans for computer system	Computer system installed; computer system after a full test run;
D Acceptance					
D1	Type test	Confirm the computer system conforms to requirements and specifications of computer system (including functions related to safety); Confirm the system confirms to GD01-2006.	Test based on GD01-2006.	Requirements and specifications of computer system; requirements and specification of functions related to safety; type test program.	Confirm the verification of computer system conform to requirements of functions related to safety. Type test report.
D2	FAT test	Make a computer system test in factory	FAT report shall include:①tools and equipment used;② FAT activity record;③ Differences and disposal of actual and expected results. Analyze and assess the differences between expected and actual results to ensure whether to continue the test or put forward a change request.	Requirements and specifications of computer system; FAT test program.	FAT test report.
D3	Onboard test	Perform the onboard test to	Verify the normal implementation of	Requirements and specifications of	Onboard test

Overview of System Lifecycle **Table 6.1.1**

		verify the system execution ability as scheduled after the interconnection between all systems, including full system test and integration test.	functions by the full system test under the condition of actual hardware parts and final application software. Verify the normal implementation of functions by the integration system in case of all systems in integration.	computer system; onboard test program.	report.
E Running					
E1	Operation, maintenance and repair	Operate, maintain and repair the computer system for keeping the functional safety as required.		Installation, operation and maintenance plans for computer system	Functions that continue to meet the requirements from the computer system; Documents of operation, repair and maintenance for computer system in time-ordered.
E2	Change	Ensure the computer system under control at and after the stage of change.	Verify the change returning the suitable stage of lifecycle. Keep records in documentation. Record modifications by manufacturer. Provide the subsequent significant modifications on software and hardware of systems II and III to CCS for approval. Notify the approved modification and change scheme in advance and analyze relevant	Requirements and specifications of computer system Plan for software quality Test program at corresponding stage	Realization of functional safety required by computer system at and after the stage of the change; Documents of operation, repair and maintenance for computer system in time-ordered.

			effects, meanwhile get an approval on modification from CCS. Give change verification on the software modified to prove that meets the computer system requirements. Note: major modifications refer to modifications impacting safe traffic and/or safety of ships.		
E3	Decommissioning or processing	Ensure the functional safety of computer system adapts to cases of being and after being decommissioned or processed.	Analyze effects and make a plan including the system shutdown and removal before the decommissioning or processing. Indicate disposal and process of sensitive information in the computer system operating instruction.	Request for decommissioning or processing based on functional safety management procedures.	

6.1.2 The documentation mentioned in this Guide can be prepared based on the manufacturer's internal document management system when using this Guide; however, its content shall comply with the related content mentioned in this Guide.

6.1.3 The content needed to be supplemented for system lifecycle in the above table will be separately listed below.

6.1.4 See Chapter 7 for software development lifecycle in details.

6.2 Software safety at maintenance stage

6.2.1 Purpose

It needs to meet the software safety requirements at maintenance stage.

6.2.2 Requirements

(1) The modification of the program and data, and changes of the version shall be recorded and submitted to our society for approval.

(2) To guarantee that the specifications of corresponding departments responsible for the software development Lifecycle (SDLC) are qualified to support their activities, the following is specially required:

①trainings on fault diagnosis and repairing and system testing are provided to the staff;

②trainings are provided to operating personnel;

③periodical retraining is provided to the staff.

(3) All trainings, experience and qualification of those personnel who are related to any activity concerning SDLC shall be documented.

(4) Analysis on risks (or potential risks) shall be conducted and specifications of suggestion on minimizing those risks shall be put forward.

(5) As for specifications applied to analysis on operation and maintenance of performance, it is especially required:

①that procedures used to identify system failures which will affect the function safety shall contain specifications used to test repeated failure for daily maintenance;

②that the requirement rate and the failure rate in operation and maintenance shall be estimated to make sure whether they are consistent with those presumed in system design.

(6) Procedures used for modification of safety-related systems shall be launched.

(7) Approval procedures and competent departments are required for the modification.

(8) Procedures shall maintain the accuracy of information on potential risks and safety-related systems.

(9) In the SDLC, configuration management of the computer particularly requires :

①specifications used to realize configuration control;

② specifications used for unique identification on all elements of a configuration management item (hardware and software);

③procedures used to prevent unauthorized entries from entering service.

(10) Training terms on suitable occasions and emergency service are required.

(11) The operators shall establish programs for receiving, recording, solving and tracking problems and modifying requests. The problems shall be dealt with as per troubleshooting programs. System operation plans shall be established and maintained, including identification of the configuration items, operation procedures and predicted maintenance. Transfer and retire of software shall be included in the plans.

(12) All new updates, releases or modifications of the system and (or) components version shall be tested by the operators. In addition, the components used for release shall meet specified standards. Integration testing shall be included if the interface of the released components has been modified.

(13) Configuration audit shall be performed regularly to verify the integrity of operation configuration.

6.2.3Input

(1) Operating instruction of computer systems

(2) Software quality plans

(3) Risk analysis and prevention procedures

(4) Training terms on suitable occasions and emergency services

6.2.4 Output

Records of modification of programs and data and change of version.

7 Softwaredevelopment Lifecycle

7.1 Quality plans of SDLC shall be made by manufacturers. Administrative and technical means shall be applied for control in software Lifecycle so as to manage the software changes and guarantee satisfaction in software safety-related aspects and to prove that manufacturers possess valid quality control programs satisfactory to all stages of SDLC.

7.2 Software quality plans shall include the following contents:

7.2.1 The SDLC shall meet the requirements specified in Article 5.2.

7.2.2 In the SDLC, configuration management of the shipboard computer system particularly requires:

(1) formal configuration control nodes to be executed in particular phases;

(2) procedures used for unique identification on all components of some items (hardware and software);

(3) procedures used to prevent unauthorized entries from entering service.

7.2.3 See the figure and table below for division of SDLC relations among divisions.

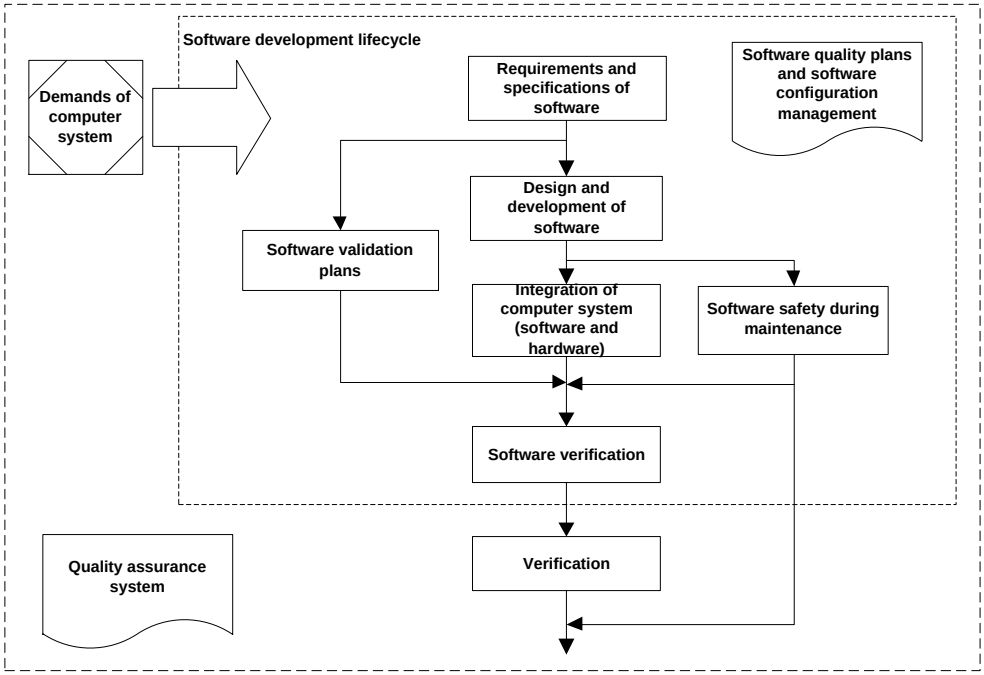


Fig.7.2.3-1 Software Development Lifecycle

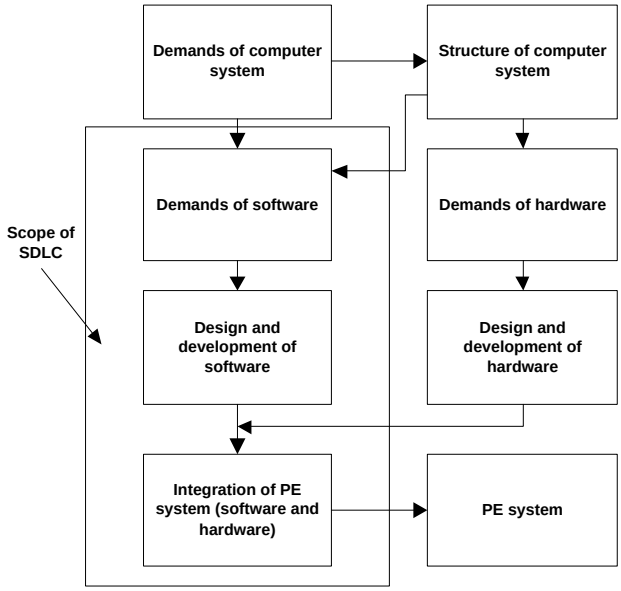


Fig. 7.2.3-2 Scope and External Relations of SDLC

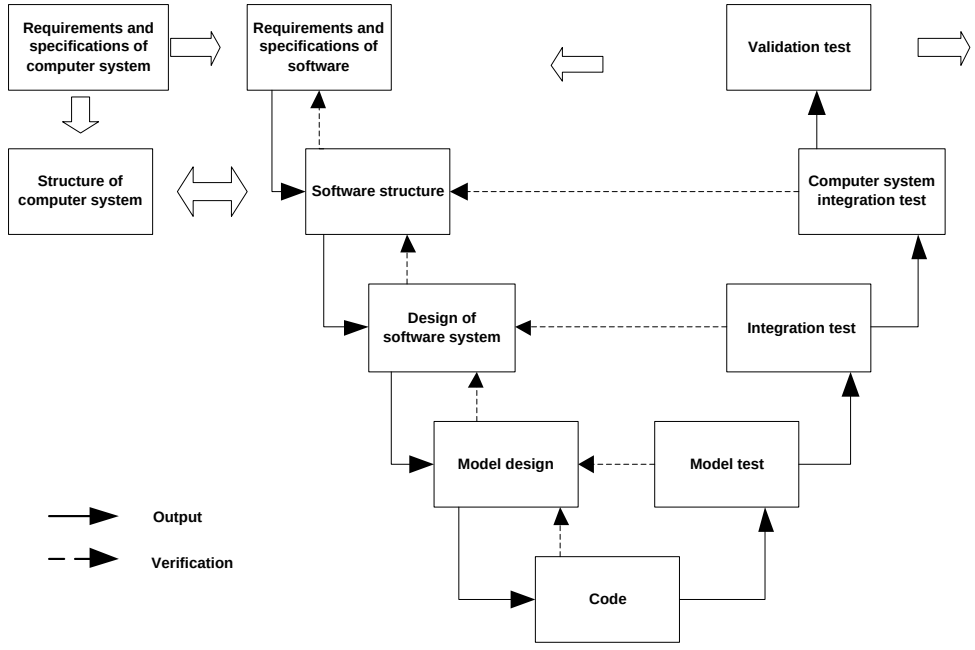


Fig. 7.2.3-3 SDLC Model (Model V)

Note: In addition to Model V, other SDLC models approved by our Society are accepted in the manual.

7.3Requirements and Specifications of Software

7.3.1 Objectives

(1) To specify requirements and specifications of software as per the requirements of system functions;

(2) To specify the requirements of software safety function for any computer system to achieve some safety functions;

(3) To specify the requirements of each computer system for software integration.

7.3.2 Requirements

(1) The information presented in Article 7.3.1 shall be reviewed by software developers for comprehensive provisions of requirements. The following factors shall be particularly taken into consideration:

①safety functions;

②system configuration or composition;

③hardware requirements;

④software requirements;

⑤capacity and response time performance;

⑥interface of equipment and operators

(2) The specification of software safety within required categories or levels shall be expressed and structured in such a way that they are:

①clear, precise, unambiguous, verifiable, testable, maintainable and feasible;

②traceable to provisions of safety requirements of computer system;

③ presented avoiding unclear terms and descriptions or incomprehensible terms and descriptions for the document users in any phase of the SDLC.

(3) If no special definitions concerning computer system safety is provided, all EUC-related operation modes shall be specified in the special requirements of software safety.

(4) Any safety-related or corresponding constraints between software and hardware shall be specified and documented in the requirements and specification of software.

(5) Within the scope of hardware design specifications of computer system, the following shall be taken into consideration for the requirements and specification of software:

①self-monitoring of software;

②monitoring of programmable electronic hardware, sensors and actuators;

③periodical tests on safety functions in the operation of system;

④tests on safety functions by EUC when operable.

(6) Clear distinction between non-safety and safety functions of computer system shall be presented in the requirements and specifications of software.

(7) Security attributes of products rather than that of engineering project shall be presented in the requirements and specifications of software.

7.3.3 Input

Requirements and specifications of computer system

7.3.4 Output

Requirements and specifications of software

7.4 Software validation plans

7.4.1 Objectives

Software validation plans are prepared as per the requirements and specifications of software.

7.4.2 Preparation Requirements

(1) Followings shall be taken into consideration in software validation plans:

①details during validation.

②details of personnel who execute the validation.

③identification of EUC operation-related models shall include:

preparation for application, including setting and adjustment;

— stable status of launching, teaching, automation, manual operation, semi-automation and operation.

— reset, power-off and maintenance;

— reasonable foreseeable abnormal conditions.

④identification of each EUC operation-related models shall be validated before trial operation.

⑤confirmed technology routes (e.g. analysis methods, statistical tests, etc.)

⑥measures and procedures used to confirm each function which meets the requirements of software functions.

⑦special references for the requirements and specifications of software.

⑧environment necessary to validation activities (e.g. adjustment tools and equipment are needed in testing).

⑨pass/failure criteria.

⑩records of evaluation validation, especially this Guidelines and procedures for failure evaluation.

(2) Technology strategies for software validation shall include:

①either or both of manual and automatic technology;

②either or both of dynamic and static technology;

③either or both of analysis and statistic technology.

(3) Pass/ failure criteria for software validation shall include:

① required input signal and its order and value;

② expected output signal and its order and value;

③other acceptable criteria, such as memory usage, timing and allowable deviation of value.

7.4.3Input

Requirements and specifications of software

7.4.4 Output

Software validation plans.

7.5 Design and development of Software

The design and development of software in the SDLC shall be described in this part.

7.5.1 Requirements for the software structure and coding language

(1) Objectives

① Software structure:

Software structure is established to meet requirements for software safety of different system levels.

The review and evaluation shall be performed for requirements for software of hardware in the computer system, including the impact on EUC safety due to the mutual effect of hardware and software.

② Requirements of coding language:

During the appropriate integration tools are selected in accordance with the required system levels, including language and compilers.

(2) Requirements of software structure

Software structure refers to the major components of software and subsystems, including how they realize the inner connection and how they achieve required attributes, especially the safety integrity. The major components include operation systems, database, large-equipment input/output subsystems, communication subsystems, applications, programming and diagnostic tools, etc.

The proposed design of software structure will be established by software suppliers and (or) developers. The design of software structure shall be presented in detail, which will include:

① Integration technology and measures selected and conformed to meet software requirements and specifications based on systems at different levels during the required software development Lifecycle. These technology and measures include software design strategies for fault allowable deviation (consistent with hardware) and fault prevention, redundancy and diversity (when applicable).

② According to the division of components/ subsystems, following information shall be provided for each part:

— whether they are new, existing or patented;

— whether they have been verified; and, if they have, their verification criteria;

— whether each component/ subsystem is safety-related;

③The mutual effect and evaluation of all software/hardware are confirmed and their importance is defined.

④ Structure which indicate clear definition or restrict clear definition are presented with symbols.

⑤ Design features used to maintain the safety integrity of all data are selected. Such data may include large-equipment input/output data, communication data, operation interface data, maintenance data and inner database data.

⑥ Proper integration test of software architecture are specified to ensure that the software architecture can meet the software safety requirements of specified system level.

(3) Requirements of supporting tools and programming language

① In the event that users adopting limited variable language during application programming with a low safety integrity, required tools and programming language may be confined as standard PLC language, editors and loaders. The responsibility of its compliance is mainly undertaken by the suppliers.

②It is required to limit the subsets of PLC language in systems with higher level, verify and validate the tools such as code analyzers, emulators, etc. Under these circumstances, the responsibility is jointly-undertaken by the suppliers and the users.

③Even in systems with lower level, total-variable languages are widely used for embedded application tools. The responsibility of its compliance is mainly undertaken by the developers. The PCL suppliers who use total-variable languages to provide users with low-variable languages for application programming are included.

④ According to the natural characteristics of software development, it is required to ensure that responsibility specified in following (a) ~ (d) requirements are undertaken by the suppliers and the users in separate or joint manner. The division of liabilities shall be documented during the preparation of the security technology.

(a) A set of appropriate integration tools shall be selected as per required system levels, including languages, compilers, configuration management tools and automatic testing tools during application. The availability of suitable developing tools for corresponding service in the whole Lifecycle of the computer system shall be taken into consideration (not those tools used during the initial phase of system developing).

Within the requirements of safety integrity levels, the selection of programming languages shall meet following requirements:

Translators/ compliers with validation certificates based on national or international standards are provided, or the evaluation for the suitability of its objectives is established;

The characteristics are completely and clearly defined or confined;

The selection shall be in accordance with characteristics of application;

Characteristics which contribute to detecting program errors are included;

Characteristics which match the design methods are supported.

(b) When it is failed to meet ①, reasons for another alternative language shall be recorded in the specification of software structure designs, which shall present the suitability of language objectives in detail and any additional measure about language drawbacks.

(c) The coding standards shall

be reviewed by the evaluating party to make sure whether they are suitable for application intentions;

be used for development of all safety software.

(d) Programming habits shall be specified in coding standards, which describe characteristics of non-safety languages (such as undefined language characteristics and unstructured designs) and provide procedures of source code documents. Source code documents shall at least include following information:

legal entities (such as companies, authors, etc.);

description;

input and output;

configuration management history.

(4) Input

① Requirements and specifications of software

② Designs of hardware structure in computer systems

(5) Output

① Specification for software structure design

② Development tools and coding standards

③ Selection of development tools

④ Specification for test of software structure integration

⑤ Specification for test of computer system integration

7.5.2 Detailed design and development

(1) Objectives

To design software to meet requirements of different system levels, which can be analyzed, verified and modified in a safe manner.

Designs for software structure and individual software modules are included in the detailed designs and development.

(2) Requirements

① Detailed design refers to software structure designs— the major components in the structure are divided in software modules, designs of single software modules and coding systems (such as basic software installed in each hardware unit and communication software and applications installed at network nodes).

②The detailed design and development of software are required to offer logic language for the formation of each piece of software and provide detailed design documents for the definition of the inner structure and interface of the interface of components, including the testing on each component.

③ The development of software shall possess modularization, testability and safe modification.

④As for each major component/ subsystem in the specification of software structure designs, the further optimization of designs shall be divided as per software modules. Each specified software modules shall be provided with applicable designs and tests for each one.

⑤ Specification for designs of software systems and single modules are required.

⑥The description documents shall include:

(a) description of basic software installed in each hardware unit;

(b) description of communication software installed at network nodes;

(c) description of applications (not the program list)

(d) tools used for system setting and equipment configuration;

(e) description of mutual restrictions and dependence between special functions, performance, modules and other parts.

Description of applications shall include:

(a) information on system modules (which must be in operation to maintain functions), including the dependency on other systems;

(b) details for each module which achieve enough levels to understand its functions;

(c) relations among software modules (which must be executed to maintain all functions)

(d) data flow and control flow among software modules;

(e) configuration of software which includes priority strategies;

(f) switch-over mechanism of the redundancy system (if any);

(g) self-monitoring of software (e.g. verification of watch-dog and data range of application driven);

(h) verification tests and external equipment diagnostic tests (e.g. sensor and terminal element);

(i) measures taken for unexpected process variables, such as exceeded sensor value, open circuit inspection, short circuit inspection.

(3) Input

① Specification for software structure design

⑤Supporting tools and coding standards

(4) Output

① Specification for software system designs

② Description of software system integration tests

③ Specification for software module designs

④ Specification for software module tests

7.5.3 Code implementation

(1) Objectives

To prepare single software module and design and implement the software with appropriate tool set (including language and compiler) in the whole lifecycle of software verification, validation, evaluation and modification.

(2) Requirements

Source code shall:

①be readable, understandable and testable;

②meet the requirements for software module design;

③meet the requirements for coding standards;

④meet relevant requirements in safety plans.

The code of each piece of software shall be reviewed to check whether the code compiling and its records meet the description of detailed design documents.

(3) Input

① Design specification for software module

②Support tools and coding standards

(4) Output

① List of source codes

② Code review record

7.5.4 Software module testing

(1) Objectives

Software module testing is a kind of verification activity. It is the combination of code review and software module test to provide evidence that a software module meets its relevant requirements, i.e. verified.

(2) Requirements

①Each software module shall be tested as per the regulations in software design.

The tests indicate that each software module performs its intended function and does not perform unintended functions.

② Records of software module tests shall be documented. The contents of documentation of records of software module tests of ships' service equipment of categories II and III include but not limited to software module test plan, software module test case, software module test record, test record analysis report, problem report ticket of software module test and summary report of test.

③ Specification for correction measures of test failures shall be stipulated.

④The manufacturer shall perform overall module test for the logic and requirements of software module using the testing method.

⑤For systems of categories II and III, the manufacturer shall verify each software module as per software module test specification, which is formulated in the design stage of software system.

⑥The manufacturer can perform module test with white-box testing and design test cases as per such methods as boundary value analysis, error guessing, equivalence class or input classification. The above methods can be selected as per the safety level requirements of software and characteristics of programmable equipment.

(3) Input

① Specification for software module testing

② List of source codes

③ Code review record

(4) Output

① Records of software module testing

② Verification and testing of software module

7.5.5 Software integration test

(1) Objectives

Software integration test is the integration from software component to software unit and the objective is to collect software component step by step, check the integration with preliminary and detailed design, provide evidence that all software modules, assemblies and subsystems interact correctly to perform their intended functions rather than unintended functions.

(2) Requirements

① Software integration tests shall be correctly specified in design and development stage.

② Software integration tests generally include: software subsystem test and software system test.

③The following contents shall be specified for software integration test:

(a) Classification of software managing integration;

(b) Test cases and test data

(c) Types of tests to be performed

(d) Test environment, tools, configurations and programs;

(e) Test completion criteria shall be determined; and

(f) Scale of correction action for test failure.

④ Software integration tests shall be performed as per the requirements for software integration tests. These tests shall indicate that all software modules and software assemblies/subsystems act correctly to perform their intended functions rather than unintended functions.

⑤ Records of software integration tests shall be documented and indicate whether the test results meet the objectives and test criteria. In case of a failure, the reason for the failure shall be recorded.

⑥ In software integration process, an impact analysis shall be carried out for any modification or change of the software to determine the impact on the software modules and re-verification and re-design activities needed.

(3) Additional requirements for software subsystem tests:

① It is suggested to perform subsystem tests with black-box testing and design black-box testing cases by such methods as dynamic test, equivalence class and input classification, including boundary value analysis. The above methods can be selected as per the safety level of software and characteristics of ship's service programmable equipment.

②For systems of categories II and III, subsystem tests shall be performed and test results shall be analyzed to verify whether software modules are correctly integrated. It shall be validated that the test results can be traced to the criteria established based on test traceability in test plan documentation.

(4) Additional requirements for software system tests:

①It shall be ensured that subsystems meet the requirements of categories II and III stated above.

② System test activities shall be carried out as specified in software-related system test plans of ship's service programmable equipment.

③The following properties shall be considered in system tests:

-Integrity for validation of relevant software design specification;

-Correctness for validation of relevant software design specification (successfully completed);

-Repeatability;

-Precisely defined validation configuration.

④For system tests of systems of categories II and III, protection function against modification shall be verified:

(a) Prevent users from modifying programs;

(b) Prevent users from modifying operation parameters of programs.

⑤For system tests of systems of categories II and III, the function that safe failure of system will occur under single fault condition shall be verified.

⑥ System tests shall be carried out by black-box testing, whose test cases shall be designed by such methods as equivalence class and process simulation. The above methods can be selected as per the safety level requirements and requirements for characteristics of programmable equipment.

⑦ In case of discrepancy between expected results and actual results, analysis and evaluation shall be made to determine whether to continue the test or issue a change request. If a change request is issued, then the progress shall return to early phase of development lifecycle. These decisions shall all serve as the validation results and documentation of system tests.

(5) Input

Test specification for software system integration (software subsystem/system tests)

(6) Output

① Records of software system integration tests

② Verification and testing software systems

7.6 Computer system integration (hardware and software)

7.6.1 Objectives

(1) To integrate software on target computer system hardware.

(2) To integrate software and hardware on the computer system to ensure its compatibility and meet the expected requirements.

7.6.2 Requirements for integration tests

(1) Integration tests shall be specified in the design and development stage to ensure the compatibilities of hardware and software in computer systems.

(2) The following shall be specified for integration tests of computer systems (hardware and software):

① System splitting by integration level;

② Test cases and test data;

③ Types of tests to be performed;

④ Test environment, including tools, support software and configuration description;

⑤ Criteria for determining test completion.

(3) During integration tests required for computer systems (hardware and software), activities performed by developers as per their own intentions shall be distinguished from activities performed from the standpoint of users.

(4) Integration tests required for computer systems (software and hardware) shall be distinguished in the following activities:

① Inclusion of software systems into target programmable electronic hardware;

② Computer system integration, i.e. adding sensors and actuators;

③ Full integration of EUC and computer systems.

(5) Software shall be integrated as per the integration tests required for programmable electronics (hardware and software) and safety-related programmable electronic hardware.

(6) During integration tests of safety-related programmable electronics (hardware and software), an impact analysis shall be carried out for any modification or change performed on integration system to determine the impacts on all software modules and re-verification activities needed.

(7) Test cases and their results shall be recorded for subsequent analysis.

(8) Integration tests of safety-related programmable electronics (hardware and software) shall be documented and it shall be indicated whether the test results meet the test objectives and test criteria. In case of a failure, the reason for the failure shall be recorded. Impact analyses shall be carried out for any modification or change of software to determine the impacts on all software assemblies/models and re-verification and re-design activities needed.

(9) For systems of category II, manufacturer shall reserve and submit supporting documents for integration tests, which shall include testing plan and testing report. For systems of category III, CCS shall witness the integration tests.

7.6.3 Requirements for fault simulation tests

(1) Manufactures shall formulate specification for system fault simulation tests as per system design specification. Fault simulation shall be carried out as real as possible to prove that it has appropriate system fault detection and system response. Any necessary fault analysis record shall be observed.

(2) Specification for fault simulation tests includes:

① Name of faulted assemblies or elements;

② Type of fault;

③ Insertion of fault mode;

④Required system response (output record).

(3) Test cases of fault simulation and their expectation results shall be documented. The document shall indicate the test results and whether the test results meet the test objectives and test criteria. In case of a failure result, the reason for the failure shall be documented.

7.6.4Input

(1) Specification for integration test of software architectures

(2) Specification for computer system tests (including specification for fault simulation tests)

(3) Computer system hardware

7.6.5 Output

(1) Records of integration tests of software architectures

(2) Records of integration tests of computer systems

(3) Verification and testing computer system

7.7 Change management

7.7.1 Objectives

To provide software-related information and specification to keep the safety of computer systems in operation and modification stage

7.7.2 Requirements

(1) Manufacturers shall record modifications. Subsequent major modifications of software of systems of category III shall be submitted to CCS for approval.

(2)Approved modification and change schemes shall be notified in advance and impact analysis shall be carried out, and the modification shall be approved by CCS.

(3) Change validation shall be carried out for modified software to the satisfaction of CCS.

Note: Major modifications refer to modifications impacting safe operation and/or safety of ships.

7.7.3Input

All input and output documents listed in 6.1~6.2

7.7.4 Output

Software operation and modification procedure

7.8 Software verification

7.8.1 Objective

To reach the required software system level, test and evaluate outputs of development lifecycle phase of given software to ensure the correctness and consistency of outputs and standards provided upon input of the phase.

7.8.2 Requirements

(1) Synchronization plan shall be prepared for software verification and development, and information shall be documented for each phase of software development lifecycle.

(2) The verification plan shall refer to the criteria, techniques and tools to be used in the verification activities, and the following shall be indicated:

① Evaluation of safety integrity requirements;

② Selection and documentation of verification strategies, activities and techniques;

③ Selection and use of verification tools (testing tools, professional testing software, input/output simulators etc.);

④ Evaluation of verification records;

⑤ Correction actions adopted.

(3) Software verification shall be carried out as per the plan.

(4) Documented evidence proving the phase verified have been completed in all aspects.

(5) After each verification, the verification documentation shall include:

① Identification of verified items;

② Identification of information of corresponding verification;

③ Nonconformance (e.g. software module, data architecture and algorithm that is not frequently used).

(6) All information required for the correct implementation of phase N+1 in stage N of software development lifecycle shall be available and verified. Outputs of phase N include:

① Specification, design specification or codes of phase N shall fully meet the following requirements:

- Functionality;
- Requirements for safety integrity, performance and preparation of other safety plans;
- Readable by the development team;
- Testability for further verification;

-Safe modification allowing further improvement.

② Designs specifying and describing phase N

Validation plans and/or tests specified in phase N fully meet the requirements.

③ Check incompatibility between:

-Tests specified in phase N and tests specified in phase N-1;

-Outputs of phase N.

(7) The following verification activities shall be carried out in phases of software development lifecycle:

① Verification of software requirements (see 7.8.2 (8));

② Verification of software architecture (see 7.8.2 (9));

③ Verification of software system design (see 7.8.2 (10));

④ Verification of software module design (see 7.8.2 (11));

⑤ Verification of codes (see 7.8.2 (12));

⑥ Data verification (see 7.8.2 (13));

⑦ Software module test (see 7.5.4);

⑧ Software integration test (see 7.5.5);

⑨ Computer system integration test (see 7.5.6);

(8) Verification of software requirements: once the software requirements are specified, the verification shall consider the following items before next phase, software design and development:

① Consider whether the specified software requirements have fully met the requirements on function, safety integrity, performance and other safety validation planning requirements specified by computer system.

② Consider whether the software safety validation plan has fully met the specified software safety requirements.

③ Check incompatibility between:

-Specified software requirements and specified computer system safety requirements;

-Specified software requirements and software validation plan.

(9) Verification of software architecture: after establishing software architecture design, the verification shall:

①Consider whether the software architecture design has fully met the specified software safety requirements.

②Consider whether the tests specified by software architecture integration fully meet the requirements of design specification for software architecture.

③ Consider whether the properties of each main component/subsystem fully meet the following requirements:

-Required flexibility of safety performance;

-Testability for further verification;

-Readable by development and verification team;

-Safe modification allowing further improvement.

④ Check the following incompatibilities among:

-Description of software architecture design and specified software safety requirements;

- Description of software architecture design and specified software architecture integration tests;

- Specified tests for software architecture integration and software safety validation plan.

(10) Verification of software system design: after specifying software system design, the verification shall:

① Consider whether the specified software system design has fully met the requirements of software architecture design.

② Consider whether the specified tests for software system integration have fully met the requirements for specified software system design.

③ Consider whether the properties of each main component specified in software system design can fully meet the following requirements:

- Required flexibility of safety performance;

- Testability for further verification;

- Readable by development and verification team;

- Safe modification allowing further improvement.

④ Check incompatibility among:

-Specified software system design and description of software architecture design;

-Design specification for software system and tests specified for software system integration;

-Tests specified for software system integration and tests specified for architecture integration.

(11) Verification of software module design: after specifying each software module design, the verification shall:

① Consider whether the specified software module design has fully met the requirements specified for software system design.

② Consider whether the tests specified for each software module fully meet the requirements specified for software module design.

③ Consider whether the properties of each software module fully meet the following requirements:

- Required flexibility of safety performance;

- Testability for further verification;

  -Readable by development and verification team;

- Safe modification allowing further improvement.

④ Check incompatibility among:

- Specified software module design and specified software system design;

- (For each software module) specified software module design and specified software module testing;

- Specified software module test and specified integration test of software system.

(12) Code verification: source code shall be verified by static method to ensure the conformance between the specified design of software module, required coding standards and safety planning requirements.

Note: In early phase of software safety lifecycle, the verification is static (e.g. check, review, formal proof etc.).Code verification includes such techniques as software inspection and walk-through. It combines the records of code verification and software module testing to ensure each software module meets relevant specifications. After then, the main verification method is forward testing.

(13) Data verification

①The following shall be verified for data architecture specified in the design:

- Integrity;

- Consistency;

- Protection against change or damage;

- Consistency of functional requirements of data driven system.

②The following shall be verified for application data:

- Compatibility with base system software (e.g. implementing sequence, operation time etc.);

- Correction of data value.

③All modified parameters shall be verified to prevent:

- invalid or undefined initial values;

- wrong, inconsistent or unreasonable values;

- Unauthorized change;

- Data damage.

④All large equipment interface and relevant software (i.e. sensors, actuators and offline interfaces) shall be verified to:

- be used for detection of failure of expected interface;

- be used for error tolerance for failure of expected interface.

⑤For all communication interfaces and relevant software, the sufficiency of the following events shall be verified:

- Failure detection;

- Error prevention;

- Data validation.

7.8.3 Input

Applicable verification plan (based on stages)

7.8.4 Output

Applicable verification report (based on stages)

8 Test and verification

8.1 Test and verification shall be performed to the computer system as required by Table 8.2, and specific requirements for software are proposed in this section.

8.2 Test and verification shall be performed to the assessment of small low complexity computer system as required by Appendix2 in this Guide.

Test and verification **Table 8.2**

S/N	Test and document	System category			Document to be provided
		I	II	III	
1.	**Quality system document**				
1.1	Plan for software quality		M	M	Plan for software quality
1.2	Inspecting parts provided by distribution partner (only for hardware)		M	M	Incoming inspection record
1.3	Quality control during production		M	M	Process inspection record
1.4	Report on completion test	M	M	S	Test report
1.5	Traceability of software	M	M	S	① Plan for software quality ② Record of modification of program and data and verification of revision.
2.	**Description of hardware and software**				
2.1	Software description		M	S	① Software requirements and specifications ② Specification for software architecture design ③ Specification for test of software architecture integration ④ Specification for test of computer system integration ⑤ Development tools and coding standard ⑥ Selection of development tools ⑦ Specification for software system design ⑧ Specification for test of

Test and verification **Table 8.2**

					software system integration ⑨ Design specification for software modules 10 Specification for software module test 11 Requirements and specifications of computer systems
2.2	Hardware description		M	S	Description of computer system hardware Requirements and specifications of computer system
2.3	Analysis only for fault of safety-related functions			S	Report for the assessment of software function safety
3.	**Supporting document for software test**				
3.1	Documents proving that the software test is performed based on quality plan		M	S	Code review report Record of software module tests Records of software system integration tests Records of software architecture integration tests Records of computer system tests
3.2	Analysis on whether programmable program related to safety-related functions is in existence, or whether it is observed.			S	Report for the assessment of software function safety
4.	**Hardware test**				
4.1	The test shall be performed based on GD01-2006.		W	W	Report on test of computer system types
5.	**Software test**				
5.1	Module test		M	S	Specification for software

Test and verification **Table 8.2**

					module test Record of software module tests
5.2	Subsystem test		M	S	Specification for software system test Record of software system tests
5.3	System test		M	S	Specification for software system test Record of software system tests
6.	**Performance test**				
6.1	Integration test		M	W	Records of software architecture integration tests Record of test of computer system integration
6.2	Fault simulation		W	W	Record of test of computer system faults
6.3	Factory acceptance test (FAT)	M	W	W	FAT report
7.	**Onboard test**				
7.1	Full system test	M	W	W	Report on test on board
7.2	Integration test		W	W	Report on test on board
7.3	Electromagnetic compatibility shall be proven by operating wireless functions		W	W*	Report on test on board
8.	**Modification**				
8.1	Test after modification	M	S/W	S/W	Software operation and modification procedure Record of analysis on software modification impact Test report

Note: M - Documents kept by manufacturer and submitted as required

S - Documents inspected by CCS

W - Witnessed by CCS

* - The Level of witness shall be determined based on the requirements below. If design or arrangement inconsistent with the planned requirements, it is required to submit project analysis performed as per related international (refer to Article 55 Chapter II-1 of SOLAS convention) or domestic standards to CCS for approval.

Appendix1 Verification of tests and inspections

S/N	Test and document	Reference item	System category			Document to be provided	Compliance to requirements
			I	II	III		
1.	**Quality system document**						
1.1	Plan for software quality	5.2		**M**	**M**	Plan for software quality	
a	Whether there is any confirmed standard and directive document for product definition?			×	×		
b	Whether related Parties (e.g. developer and project director) have reviewed products?			×	×		
c	Is acceptance standard of products prepared?				×		
d	Are the objectives and application scope defined?			×	×		
e	Are some of the software contents covered by software QAP defined?				×		
f	Are intended uses of software specified?				×		
g	If the part of software development Lifecycle covered by software QAP described?				×		
h	Are some available references included?			×	×		

i	Is the outline of project management structure included?				×		
j	Are documents for developing, verifying, validating, using and maintaining of management software refined?				×		
k	Are the documents listed and described?		×	×	×		
l	Are the documents to be assessed by software QAP listed.			×	×		
m	Are standards in use and requirements for practice and quality identified (e.g. Standards of IEC, ISO, IEEE, etc.)?			×	×		
n	Is how to monitor and guarantee the conformance of process and products described (e.g. review, report and trend)?				×		
o	Is the role that software management plan plays in software verification and validation specified and described?				×		
p	Are the method and programs for reporting, tracking and solving problems specified and described?				×		
q	Are the tools and technologies used for supporting software guarantee activities described the (e.g. inspection list, plan and report template, database for tracing)?						
r	Is it discussed that the control of suppliers meeting requirements of customers can be guaranteed by internal and external supervising (e.g. inspection, assessment/review, monthly status report)?				×		

s	Can the design and development of software be guaranteed to meet special design and development requirements, i.e. preventing and responding to potential failure conditions?			×	×		
1.2	Inspecting parts provided by distribution partner (only for hardware)	5		**M**	**M**	Incoming inspection record	
a	IsISO9000 quality system is established.			×	×		
b	Are inspection method, proportion, judgment method and control of non-conformed products for receiving inspection specified?			×	×		
1.3	Quality control during production	5.3		**M**	**M**	Process inspection record	
a	Are the production processes established?			×	×		
b	Are quality control requirements for each technology process described clearly with character and charts?			×	×		
c	Do quality control documents contain identified control object, control standard, control method and inspection method?			×	×		
d	Is there any measure or method for products not modified during production?			×	×		
e	Have proper measures been taken for protecting the completeness of software and preventing viruses?			×	×		
f	Are there documents of "test and simulation" for production of safety-related functions?			×	×		

1.4	Final test report	5.4	**M**	**M**	**S**	Test report	
a	Is a report based on finished products test and test results prepared?		×	×	×		
b	Does the final test report include the overall evaluation of tested software?		×	×	×		
c	Are the differences between test environment and operation environment and the assessment on the impact from the difference to test result provided?		×	×	×		
d	Does the summary of test result include "all results meet expectation", "problems occurred" (if applicable) and, "difference from requirements" (if applicable).		×	×	×		
1.5	Traceability of software	5.5 5.6	**M**	**M**	**S**	① Plan for software quality ② Record of modification of program and data and verification of revision.	
a	Are the changes to the programming content and data and the revision marked or documented based on programs.		×	×	×		
b	Are quality assurance documents of software configuration management and software revision instructions prepared.		×	×	×		
c	Are programming content, process for data modification and revision change specified and recorded in documents.		×	×	×		
2.	**Description of hardware and software**						

2.1	Software description			**M**	**S**	① Software requirements and specifications ② Specification for software architecture design ③ Specification for test of software architecture integration ④ Specification for test of computer system integration ⑤ Development tool and coding standard ⑥ Selection of development tools ⑦ Specification for software system design ⑧ Specification for test of software system integration ⑨ Design specification for software module 10 Specification for software module test 11 Requirements and specifications of computer system	
a	Are software requirements and specifications based on the requirements of system	7.3		×	×		

	functions specified?						
b	Are requirements of software safety function for computer system to achieve some safety functions specified?	7.3		×	×		
c	Are the requirements of each computer system for software integration specified?	7.3		×	×		
d	Does the description for software architecture design include the following: integration technology selected and conformed to meet software requirements and specifications based on systems at different levels during the required software development Lifecycle.	7.5.1		×	×		
e	Do the technologies and measures in the requirements and specifications of software include the following: software design strategies for fault allowable deviation (consistent with hardware) and fault prevention, redundancy and diversity (when applicable).	7.5.1		×	×		
f	Does the description for software architecture design include the following: confirming the interaction and evaluation of all software/hardware and refining their importance.	7.5.1		×	×		
g	Does the description for software architecture design includes include the following: specifying proper integration test of software architecture to ensure that the software architecture can meet the software safety requirements of specified system Level.	7.5.1		×	×		
h	Are standards and naming rule specified?	7.5.1		×	×		
i	Are design specifications for software system and individual modules specified?	7.5.2		×	×		
j	Do design specifications for software system and individual modules indicate mutual restrictions and dependence among function, performance, modules and other parts.	7.5.2		×	×		

k	Do design specifications for software system and individual modules indicate self-monitoring of software (e.g. verification of watch-dog and data range of application driven).	7.5.2		×	×		
l	Do design specifications for software system and individual modules require to perform verification test and external equipment diagnostic test (e.g. sensor and terminal element)	7.5.2		×	×		
m	Do design specifications for software system and individual modules take measures for bad process variables, e.g. exceeded sensor value, open circuit inspection and short circuit inspection.	7.5.2		×	×		
2.2	Hardware description			**M**	**S**	Description of computer system hardware Requirements and specifications of computer system	
a	Does hardware description include: system chart, equipment layout, input and output equipment and mutual connection relation.			×	×		
b	Does hardware description include: wiring diagram;			×	×		
c	Does hardware description include: detailed instruction of input and output equipment;			×	×		
d	Does hardware description include: detailed instruction of power supply.			×	×		
2.3	Analysis only for fault of safety-related functions				**S**	Report for the assessment of software function safety	

a	Is proper method, e.g. FTA, risk analysis, FMEA or FMECA analysis, adopted;	6.2			×		
b	If the following proven through fault analysis: for a single fault, system in safe status of fault, and system in operation would not loss or drop to the performance standard unacceptable of CCS.	6.2			×		
3.	**Supporting document for software test**						
3.1	Documents proving that the software test is performed based on quality plan			**M**	**S**	Code review report Record of software module tests Records of software system integration tests Records of software architecture integration tests Records of computer system tests	
a	Has the software developer reviewed the codes of software and checked whether the code compiling and its records meet the description of detailed design documents.	7.5.3		×	×		
b	Has the software developer obtained the record on whether the code compiling meets detailed design documents by checking code compiling.	7.5.3		×	×		
c	Are records of software module tests are documented.	7.5.4		×	×		

d	Do the contents of documentation of records of software module tests of ships' service equipment of categories II and III include but not limited to software module test plan, software module test case, software module test record, test record analysis report, problem report ticket of software module test and summary report of test.	7.5.4		×	×		
e	Do these tests indicate that all software modules and software assemblies/subsystems act correctly to perform their intended functions rather than unintended functions?	7.5.5		×	×		
f	Do the records of software integration tests is documented, and to indicate whether the test results meet the objectives and test criteria? In case of a failure, the reason for the failure shall be recorded.	7.5.5		×	×		
g	In software integration process, has an impact analysis been carried out for any modification or change of the software to determine the impact on the software modules and re-verification and re-design activities needed?	7.5.5		×	×		
h	Has the software test verified protection function against modification?	7.5.5		×	×		
i	Have system tests of systems of categories II and III verified the function that safe failure of system will occur under single fault condition.	7.5.5		×	×		
j	Does the computer system integration test regulate the following:① System splitting by integration level;② Test cases and test data;③ Types of tests to be performed;④ Test environment, including tools, support software and configuration description;⑤ Criteria for determining test completion.	7.6.2		×	×		
k	During integration tests required for computer systems, are activities performed by developers as per their own intentions distinguished from activities performed from the standpoint of users.	7.6.2		×	×		

l	During integration tests of safety-related programmable electronics (hardware and software), has an impact analysis been carried out for any modification or change performed on integration system to determine the impacts on all software modules and re-verification activities needed.	7.6.2		×	×		
m	Are integration tests of safety-related programmable electronics (hardware and software) documented and does the test record indicate compliance with the test objectives and test criteria. In case of a failure, the reason for the failure shall be recorded. Impact analyses shall be carried out for any modification or change of software to determine the impacts on all software assemblies/models and re-verification and re-design activities needed.	7.6.2		×	×		
n	For systems of category II, has the manufacturer reserved and submitted supporting documents for integration tests, which shall include testing plan and testing report.	7.6.2		×			
o	For systems of category III, has CCS witnessed the integration tests?	7.6.2			×		
p	Do specification for fault simulation tests includes the followings: ① Name of faulted assemblies or elements;② Type of fault;③ Insertion of fault mode;④ Required system response (output record).	7.6.3		×	×		
q	Are test cases of fault simulation and their expected results documented to indicate whether the test results meet the test objectives and test criteria? In case of failure record, are the failure causes documented?	7.6.3		×	×		
3.2	Analysis on whether programmable program related to safety-related functions is available and whether it is observed.				**S**	Report for the assessment of software function safety	
a	Has it been proven that for a single fault, system in safe status of fault, and system in operation would not loss or drop to the performance standard unacceptable of CCS through fault analysis.	6.2			×		

4.	**Hardware test**						
4.1	The test shall be performed based on GD01-2006.	Table 6.1.1 - D1		**W**	**W**	Report on test of computer system types	
a	The test shall be performed based on GD01-2006.			×	×		
5.	**Software test**						
5.1	Module test	7.8.2(11)		**M**	**S**	Specification for software module test Record of software module tests	
a	After specifying each software module design, it shall be verified that: whether the specified software module design has fully met the requirements specified for software system design.			×	×		
b	After specifying each software module design, it shall be verified that: whether the tests specified for each software module fully meet the requirements specified for software module design.			×	×		
c	Consider whether the properties of each software module fully meet the following requirements after the design of each specified software module: ① Required flexibility of safety performance;② Testability for further verification;③ Readable by development and verification team;④ Safe modification allowing further improvement.			×	×		
d	Check the incompatibility among the following three points after the design of each specified software module: ① Specified software module design and specified			×	×		

	software system design;② (For each software module) specified software module design and specified software module testing;③ Specified software module test and specified integration test of software system.						
5.2	Subsystem test			**M**	**S**	Specification for software system test Record of software system tests	
a	Do these tests indicate that all software modules and software assemblies/subsystems act correctly to perform their intended functions rather than unintended functions?	7.5.5		×	×		
b	Are the records of software integration tests documented to indicate whether the test results meet the objectives and test criteria? In case of a failure, the reason for the failure shall be recorded.	7.5.5		×	×		
c	In software integration process, has an impact analysis been carried out for any modification or change of the software to determine the impact on the software modules and re-verification and re-design activities needed?	7.5.5		×	×		
5.3	System test			**M**	**S**	Specification for software system test Record of software system tests	
a	Do these tests indicate that all software modules and software assemblies/subsystems act correctly to perform their intended functions rather than unintended functions?	7.5.5		×	×		
b	Are the records of software integration tests documented to indicate whether the test results meet the objectives and test criteria? In case of a failure, the reason for the failure	7.5.5		×	×		

	shall be recorded.						
c	In software integration process, has an impact analysis been carried out for any modification or change of the software to determine the impact on the software modules and re-verification and re-design activities needed?	7.5.5		×	×		
d	Does software test verified protection function against modification.	7.5.5		×	×		
e	Have system tests of systems of categories II and III verified the function that safe failure of system will occur under single fault condition.	7.5.5		×	×		
6.	**Performance test**						
6.1	Integration test			**M**	**W**	Records of software architecture integration tests Record of test of computer system integration	
a	After establishing software architecture design, it shall be verified that: whether the description of software architecture design has fully met the specified software safety requirements.	7.8.2(9)		×	×		
b	After establishing software architecture design, it shall be verified that: whether the tests specified by software architecture integration fully meet the requirements of design specification for software architecture.	7.8.2(9)		×	×		
c	Consider whether the properties of each main component/subsystem fully meet the following requirements after the establishment of software architecture design:① Required flexibility of safety performance;② Testability for further verification;③ Readable by development and verification team; ④ Safe modification allowing further	7.8.2(9)		×	×		

	improvement.						
d	Check the following incompatibilities after the establishment of software architecture design: ① Description of software architecture design and specified software safety requirements;② Description of software architecture design and specified software architecture integration tests;③ Specified tests for software architecture integration and software safety validation plan.	7.8.2(9)		×	×		
e	Does computer system integration test regulate the following:① System splitting by integration level;② Test cases and test data;③ Types of tests to be performed;④ Test environment, including tools, support software and configuration description;⑤ Criteria for determining test completion.	7.6.2		×	×		
f	During integration tests required for computer systems, are activities performed by developers as per their own intentions distinguished from activities performed from the standpoint of users.	7.6.2		×	×		
g	During integration tests of safety-related programmable electronics (hardware and software), has an impact analysis been carried out for any modification or change performed on integration system to determine the impacts on all software modules and re-verification activities needed.	7.6.2		×	×		
h	Are integration tests of safety-related programmable electronics (hardware and software) documents to indicate that the test results meet test objectives and test criteria. In case of a failure, the reason for the failure shall be recorded. Impact analyses shall be carried out for any modification or change of software to determine the impacts on all software assemblies/models and re-verification and re-design activities needed.	7.6.2		×	×		
i	For systems of category II, has manufacturer reserved and submitted supporting documents for integration tests, which shall include testing plan and testing report.	7.6.2		×			

j	For systems of category III, has CCS witnessed the integration tests.	7.6.2			×		
6.2	Fault simulation			**W**	**W**	Record of test of computer system fault	
a	Do specifications for fault simulation tests include the followings: ① Name of faulted assemblies or elements;② Type of fault;③ Insertion of fault mode;④ Required system response (output record)?	7.6.3		×	×		
b	Are test cases of fault simulation and their expected results documented? Whether the document indicated that the test results and whether the test results meet the test objectives and test criteria. In case of failure record, whether the failure causes are documented.	7.6.3		×	×		
6.3	Factory acceptance test (FAT)	Table 6.1.1-D2	**M**	**W**	**W**	FAT report	
a	Has FAT process been recorded based on time sequence for tracing the sequence of FAT activities.		×	×	×		
b	Are tools and equipment in use and their calibration data recorded?		×	×	×		
c	Are there differences between expected record and actual record. In case of discrepancy between expected results and actual results, analysis and evaluation shall be made to determine whether to continue the test or issue a change request. If a change request is issued, then the progress shall return to early phase of development lifecycle.		×	×	×		
7.	**Onboard test**						

7.1	Full system test	Table 6.1.1-D3	**M**	**W**	**W**	Report on test on board	
a	Verify whether functions can be achieved under the status of actual hardware parts and final application software.		×	×	×		
7.2	Integration test	Table 6.1.1-D3		**W**	**W**	Report on test on board	
a	Verify whether functions can be achieved under the status of all system integration.			×	×		
8.	**Modification**						
8.1	Test after modified		**M**	**S/W**	**S/W**	Software operation and modification procedure Record of analysis on software modification impact Test report	
a	Have manufacturers recorded modifications.	7.7	×	×	×		
b	Are approved modification and change schemes notified in advance and the impact analysis carried out by the manufacturer? Is the modification approved by CCS?	7.7	×	×	×		

c	Are subsequent major modifications of software of systems of category III submitted to CCS for approval by manufacturer?	7.7			×		
d	Has the software been returned to the proper stage of software development lifecycle for corresponding verification based on the record of impact analysis.	7.8	×	×	×		

Note: M - Documents kept by manufacturer and submitted as required, S - Documents inspected by CCS, W - Witnessed by CCS, X - This requirement is applicable.

Appendix2 Evaluation of small low complexity computer systems

1Objectives

1.1Single case evaluation method is adopted for small low complexity computer systems so as to reasonably and effectively simplify the evaluation method of software.

2 Requirements

2.1 Document

2.1.1 Software description shall be based on internal document management system of manufacturer and merge with the document provided by clause 2.1 in table 8.2, and the content includes:

(1) System function description, mutual restrictions and dependence between special functions, performance, modules and other parts;

(2) Software design description, especially the configuration of software which includes priority strategy;

(3) Software version description;

(4) Failure mode analysis;

(5) The switch-over mechanism of redundancy system (if any);

(6) System test, integration test and failure simulation testing method.

2.2 Test

(1) For newly designed software, failure mode analysis of the software shall be checked, and the software shall be tested using the confirmed testing method provided by manufacturer.

(2) For products produced before software reuse and modification, regression testing shall be carried out.

Note 1:Software reuse means using various knowledge of existing software to develop new software to cut the cost of the development and maintenance of software. Software reuse is an important technique to improve the productivity and quality of software. Software reuse mainly refers to code reuse, but not specifically program. Software reuse also includes all relevant aspects, such as domain knowledge, development experience, design decisions, system architecture, demands, design, documents and etc.

Note 2:Regression testing means retest of software after modification of the software to make sure that no new mistakes have been brought in or caused the other codes to generate mistakes.

3 Input

3.1 Requirements and specifications of computer system

4 Output

4.1 Software description

4.2 Hardware description

4.3 Test report

Appendix3Technical recommendations for the design and implementation stages of computer system

1 General requirements

1.1 The design of relevant system of computer system safety (including overall structure of hardware and software, sensors, actuators, embedded software, application software and etc., see the following figure) shall conform to all the following requirements from1.1.1 to 1.1.2.

1.1.1 Hardware safety integrity requirements include:

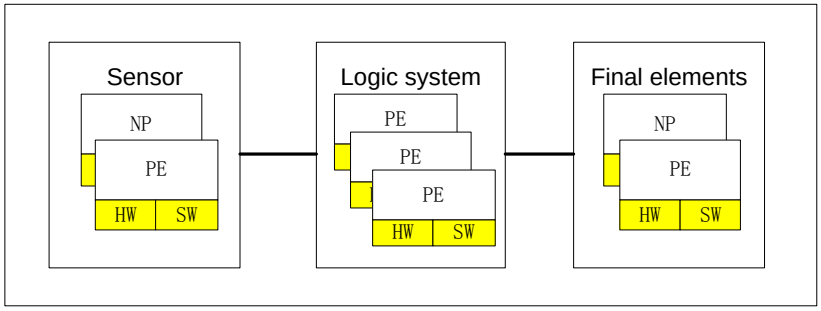
(1) structural constraints of hardware safety integrity; and

(2) requirements of probability of dangerous random hardware failure.

1.1.2 Requirements of system safety integrity include：

(1) requirements of avoidance of failure and system failure control; or

(2) evidence of equipment being verified through operation.



Programmable electronic structure		
PE Hardware structure	PE Software structure	
	PE Embedded software	PE Application software
Specific characteristics of PE hardware in general and at operation: Such as: --Diagnostic test; --Redundancy processor; --Dual I/O card	Such as: --Communication driver; --Failure processing; --Executable software	Such as: --Input/output function; --Derivative function (for example: sensor inspection when embedded software service is not provided)

PE： Programmable electronics, NP: Non-programmable devices, HW: Hardware, SW: Software.

Figure Appendix3-1.1.2Relationship between PE hardware and software structure

1.2 When safety function and non-safety function are executed at the same time, unless safety function and non-safety function are proved to be fully independent (that is to say, the failure of non-safety function will not cause the dangerous failure of safety function), the hardware and software of computer system shall be considered safety-related. Wherever practicable, safety function and non-safety function shall be divided.

1.3 The requirements of hardware and software shall be determined by the safety integrity level of safety function that has the highest safety integrity level, unless it can be proven that the implementation of safety function of different safety integrity level is totally independent.

1.4 When requiring independence among safety functions (see 1.2 and 1.3), the following items shall be documented during design.

1.4.1 Methods to obtain independence;

1.4.2Verification of the rationality of methods.

2. Techniques and measures of hardware safety integrity: failure control in operation

Appendix3Table 2-1~Table 2-6 provide recommendations of safety integrity techniques and measures

I/O unit and interface (external communication) Appendix3Table 2-1

Diagnostic techniques/measures	See IEC61508-7	Maximum diagnostic coverage that can be obtained after consideration	Notes
Detection failure with on-line monitoring	A1.1	Low (low demand mode) Medium(high demand mode or continuous mode)	Diagnostic coverage relying on failure detection
Test pattern	A6.1	High	
Code protection	A6.2	High	
Multi-channel parallel output	A6.3	High	Only effective if the dataflow changes during the diagnostic test interval
Monitored outputs	A6.4	High	Only effective if the dataflow changes during the diagnostic test interval

Data paths (internal communication) **Appendix3 Table 2-2**

Diagnostic techniques/measures	See IEC61508-7	Maximum diagnostic coverage that can be obtained after consideration	Notes
One-bit hardware redundancy	A7.1	Low	
Multi-bit hardware redundancy	A7.2	Medium	
Complete hardware redundancy	A7.3	High	
Inspection with test pattern	A7.4	High	Only effective on transient failures
Transmission redundancy	A7.5	High	
Information redundancy	A7.6	High	

Power supply **Appendix3 Table 2-3**

Diagnostic techniques/measures	See IEC61508-7	Maximum diagnostic coverage that can be obtained after consideration	Notes
Over-voltage protection with safety shut-off or switching to stand-by power unit	A8.1	Low	Techniques in the table shall be adopted, other techniques are also recommended
Voltage control with safety shut-off or switching to stand-by power unit (secondary)	A8.2	High	
Power cut with safety shut-off or switching to stand-by power unit	A8.3	High	Techniques in the table shall be adopted, other techniques are also recommended

Power supply **Appendix3 Table 2-3**

Idle current principle	A1.5	Low	Only effective when power is cut

Program sequence(watch-dog) **Appendix3Table 2-4**

Diagnostic techniques/measures	See IEC61508-7	Maximum diagnostic coverage that can be obtained after consideration	Notes
Watch-dog with separate time base but without time-window	A9.1	Low	
Watch-dog with separate time base and time-window	A9.2	Medium	
Logic monitoring of program sequence	A9.3	Medium	Relying on the quality of monitoring
Combination of temporal and logical monitoring of program sequences	A9.4	High	
Temporal monitoring with on-line check	A9.5	Medium	

Sensor **Appendix3Table2-5**

Diagnostic techniques/measures	See IEC61508-7	Maximum diagnostic coverage that can be obtained after consideration	Notes
Detection failure with on-line monitoring	A1.1	Low (low demand mode) Medium(high demand mode or continuous mode)	Diagnostic coverage relying on failure detection
Reactive current principle	A1.5	Low	Only effective on E/E/PE safety-related system that need no continuous control or maintain EUC safety status
Simulation signal monitoring	A2.7	Low	
Test pattern	A6.1	High	
Input comparison/voting	A6.5	High	Only effective if the dataflow changes during the diagnostic test interval
Reference to sensor	A12.1	High	Diagnostic coverage relying on failure detection
Switches that are reliably switched on.	A12.2	High	

Final elements (actuator) Appendix3Table 2-6

Diagnostic techniques/measures	See IEC61508-7	Maximum diagnostic coverage that can be obtained after consideration	Notes
Detection failure with on-line monitoring	A1.1	Low (low demand mode) Medium(high demand mode or continuous mode)	Diagnostic coverage relying on failure detection
Relay contact monitoring	A1.2	High	
Reactive current principle	A1.5	Low	Only effective on E/E/PE safety-related system that need no continuous control or maintain EUC safety status
Test pattern	A6.1	High	
Monitoring	A13.1	High	Diagnostic coverage relying on failure detection
Cross-monitoring of multiple actuators	A13.2	High	

3 Recommendations on techniques and measure for system integrity

3.1Appendix3Table 3.1-1, Appendix3Table 3.1-2 provide recommendations concerning techniques and measures for system safety integrity

3.1.1 Control failure caused by the design of hardware and software;

3.1.2 Control failure caused by environmental stress and influence;

3.1.3 Failure of control operation process.

Techniques and measures for controlling system failure caused by hardware design **Appendix3Table 3.1-1**

	Techniques and measures	See IEC61508-7	I	II	III
1	Program sequence monitoring	A.9	Highly recommended (HR) Low	HR Low	HR Medium
2	Detection failure with on-line monitoring	A1.1	R Low	R Low	R Medium
3	Test with redundancy hardware	A2.1	R Low	R Low	R Medium
4	Standard test of access port and boundary-scan structure	A2.1	R Low	R Low	R Medium
5	Code protection	A6.2	R Low	R Low	R Medium
6	Multi-hardware	B1.4	- Low	- Low	R Medium
Note: At least one technique of 2~6 shall be adopted.					

Techniques and measures for controlling system failures caused by environmental stress orinfluence **Appendix3Table 3.1-2**

	Techniques and measures	See IEC61508-7	I	II	III
1	Measures to prevent voltage breakdown, voltage fluctuation, over-voltage, low-voltage	A8	Must be adopted	Must be adopted	Must be adopted
2	Power wire and information wire shall be separated (note 1)	A11.1	Must be adopted	Must be adopted	Must be adopted
3	Improve anti-interference performance	A11.3	Must be adopted	Must be adopted	Must be adopted
4	Measures against physical environment (temperature, humidity, vibration and etc.)	A14	Must be adopted	Must be adopted	Must be adopted
5	Program sequence monitoring	A9	HR Low	HR Low	HR Medium
6	Measures against the rising of temperature	A10	HR Low	HR Low	HR Medium
7	Space division of multiple lines	A11.2	HR Low	HR Low	HR Medium
8	failure with on-line monitoring check (note 2)	A1.1	R Low	R Low	R Medium
9	Test with redundancy hardware	A2.1	R Low	R Low	R Medium

10	Code protection	A6.2	R Low	R Low	R Medium
11	Anti-synthetic signal transmission	A11.4	R Low	R Low	R Medium
12	Multi-hardware(Note 3)	B1.4	- Low	- Low	- Medium
13	Software structure	7.4.3 of GB/T20438.3	See table A.2 of GB/T20438.3		

Note: At least 1 technique of 8~13 shall be adopted.

Note 1: If optical media is adopted for information transmission, then there is no need to separate power wire and information wire. There is no need to power components of the system separately and separate the low power cables designed for the transmission of information to/ from these components.

Note 2: For a safety -related system operating in low demand mode (for example: emergency system shutdown), diagnostic coverage rate of failure detection by on-line monitoring is usually low or void.

Note 3:If confirmed or proved by experience of broad application: to meet target failure measure, hardware has fully got rid of design failure and strong enough to prevent common cause failures, then multi-hardware is not needed.