GUIDANCE NOTES
GD008-2025

China Classification Society

Guide for Safety and Reliability Assessment for Shipboard Software

2025

Contents

1 Scope and description

1.1 This Guide provides safety and reliability assessment guides for software in the shipboard computer-based systems (hereinafter referred to as computer-based systems, including programmable electronic systems), proposing safety and reliability requirements for the design, development, testing, certification and maintenance of software in the computer-based system. This Guide also establishes certain requirements for hardware related to software, which need to be used in combination with the technical requirements of the product.

1.2 This Guide is intended for computer-based systems of intelligent vessels, computer-based system to improve the intelligence of vessels, and computer-based systems based on programmable controllers. The following computer-based systems shall be tested, verified, and approved in accordance with the requirements of Chapter 10 of this Guide:

(1) Computer-based systems installed on classified vessels that provide control, alarm, monitoring, safety, or internal communication functions meeting classification requirements;

(2) Computer-based systems seeking to obtain the class notations of SLC1, SLC2, or SLC3.

This Guide does not apply to computer-based systems subject to statutory requirements, such as loading instruments, stability computers, and radio communication systems and navigation systems specified in Chapter IV and V of the SOLAS Convention.

1.3 This Guide focuses on the software development lifecycle and refers to some aspects of the overall safety lifecycle. For the purpose of this Guide, V model is adopted for the software development, and this Guide does not cover the evolution of other relevant models.

1.4 Considering the direct application of small and simple computer-based systems and their application in implementing partial functions in complex systems, this Guide defines small low complexity computer systems, and provides a simplified assessment method in Appendix 2.

1.5 For the application of this Guide, documents mentioned in this Guide can be prepared according to the internal document management systems of the stakeholders, but the content shall conform to the relevant requirements mentioned in this Guide.

1.6 This Guide includes four appendices, as described below:

1.6.1 Appendix 1 is a test and verification table used by attending surveyors to assess the safety and reliability of software in a computer-based system.

1.6.2 Appendix 2 is the assessment method for small low complexity computer system.

1.6.3 Appendix 3 is the technical proposal for the design and implementation of computer-based system.

1.6.4 Appendix 4 is the software testing requirements during the development phase.

2 References

2.1 The following references are indispensable for the application of this Guide. For dated references, only the referenced version applies. For undated references, the latest edition of the normative document referred to applies.

Table 2.1 References

1.	R001-2024	CCS Rules for Classification of Sea-going Steel Ships
2.	GD019-2024	CCS Guidelines for Type Approval Test of Electric and Electronic Products
3.	GB/T33783-2017	Testing Guideline for Programmable Logic Device Software
4.	IACS UR E22	Computer-based Systems
5.	IEC 61508	Functional Safety of Electrical/Electronic/Programmable Electronic Safety-related Systems
6.	IEC 61511	Functional safety - Safety Instrumented Systems for the Process Industry Sector
7.	IEC 60092-504	Electrical Installations in Ships - Part 504: Special Features - Control and Instrumentation
8.	IEC 60812	Failure Modes and Effects Analysis (FMEA and FMECA)
9.	IEC 61025	Fault Tree Analysis (FTA)
10.	IEEE 730	Software Quality Assurance Processes
11.	ISO 9001	Quality Management Systems – Requirements
12.	ISO/IEC 90003	Software Engineering - Guidelines for the Application of ISO 9001:2015 to Computer Software
13.	ISO/IEC 12207	Systems and Software Engineering - Software Life Cycle Processes
14.	ISO/IEC 15288	Systems and Software Engineering - System Life Cycle Processes
15.	ISO 17894	Ships and Marine Technology - Computer Applications - General Principles for the Development and Use of Programmable Electronic Systems in Marine Applications

16.	ISO/IEC 25000	Systems and Software Engineering - Systems and software Quality Requirements and Evaluation (SQuaRE) - Guide to SQuaRE
17.	ISO/IEC 25041	Systems and Software Engineering - Systems and Software Quality Requirements and Evaluation (SQuaRE) - Evaluation Guide for Developers, Acquirers and Independent Evaluators
18.	ISO 10007	Quality Management - Guidelines for Configuration Management
19.	ISO 24060	Ships and Marine Technology - Ship Software Logging System for Operational Technology

3 Terms and abbreviations

3.1 Terms

3.1.1 Software

Computer programs, regulations, rules, and any files, documents and data related to the operation of the computer-based system.

3.1.2 Software safety

The ability of shipboard software to protect itself against potential threats and attacks, which mainly refers to protection of the confidentiality, integrity and availability of software to prevent unauthorized access, data disclosure, malicious tampering or service interruption.

3.1.3 Software reliability

The ability of shipboard software to complete specified functions under specified conditions and within specified time interval.

3.1.4 Computer-based system

A programmable electronic device, or a group of interoperable programmable electronic devices, organized for one or more specific purposes, such as the collection, processing, maintenance, use, sharing, dissemination, or disposal of information. Shipboard computer-based systems usually include IT systems and OT systems. A shipboard computer-based system may be a combination of subsystems connected via a network, which is connected directly or via a public communication means (e.g., the Internet) to onshore computer-based systems, computer-based systems of other vessels and/or other facilities.

3.1.5 System

A combination of component, equipment and logic with clearly-defined purpose, function and performance. A computer-based system is a system. For the purpose of this Guide, a system refers specifically to a computer-based system, which is delivered by the system supplier.

3.1.6 Subsystem

An identifiable part of a system that implements a specific function or set of functions.

3.1.7 System of systems

A system consisting of several computer-based systems. For the purpose of this Guide, the system of systems is part of the vessel and includes all monitoring, control and safety systems delivered by the shipyard.

3.1.8 Module

Units that can be disassembled, assembled and replaced, which are the basic units that make up a system and are easy to handle.

3.1.9 Software module

A program component in software that is independent, has a specific function, and can be independently compiled and executed, which consists of program code and related data structures, and can interact with other modules through specific interfaces.

3.1.10 Software component

A separate piece of code that provides specific and tightly coupled functionality.

3.1.11 Software master files

The computer files that form the initial source of the software, which may be a readable source code file for custom software, and binary files of different forms for COTS software.

3.1.12 Software structure

An overview of the way through which different software components interact, often referred to as a software architecture or software hierarchy.

3.1.13 Software registry

The software name, version number, development completion date, release status, history of change and other information for registration of the shipboard software.

3.1.14 Safety function

Functions achieved by computer-based systems, other technology safety related systems, or external risk reduction facilities for specific hazardous events to achieve or maintain a safe state.

3.1.15 Equipment under control (EUC)

Equipment, machinery, apparatus and/or complete sets used for manufacturing, processing, transportation or other activities. A computer-based system may act as a EUC or a part of a EUC.

3.1.16 Small low complexity computer system

A computer-based system in which the failure modes of each component have been well-defined, enabling the complete determination of system behavior in fault conditions

Note: The behavior of the system under failure conditions can be determined by test and/or analysis.

3.1.17 Dynamic testing

A process in which the software execution and/or hardware operation are performed in a systematic and controlled manner to demonstrate the existence of the required behavior and the non-existence of the non-required behavior.

3.1.18 Quality plan

A document that specifies the responsible person and time to apply the quality system procedures and related resources for a particular project, product or contract.

3.1.19 Regulation

Implementation of working procedures in accordance with certain standards, requirements, and rules, under the premise of ensuring equipment and personal safety.

3.1.20 Procedure file (or procedure for short)

A QMS documentation of the next level under the quality manual, specifying the general process for a specific task. The procedure mentioned here is not a computer program.

3.1.21 System lifecycle

The whole process of computer-based system from project initiation, development, operation and maintenance to extinction.

3.1.22 Software lifecycle

The life process of software from conception to decommissioning.

Note: A typical software lifecycle includes requirements, development, testing, integration, installation, change, etc.

3.1.23 Software Configuration Management (SCM)

A technique for identifying, organizing, and controlling changes, which is applied throughout the software lifecycle.

3.1.24 Programmable device

A physical unit with software installed.

3.1.25 Vessel

Ships or offshore installations with computer-based system installed.

3.1.26 Stakeholders

A person or organization that can influence, be affected, or feel affected by decisions or activities.

3.1.27 Owner

The organization or individual who orders the vessel during the vessel building phase. The organization that owns or manages a vessel in service during the vessel operation phase. For the purpose of this Guide, the owner is a role with clear responsibilities.

3.1.28 System integrator

A single organization or individual allocated to coordinate the interaction between system and subsystem suppliers throughout the lifecycle of a computer-based system, which is responsible for integrating the system and subsystems into a validated, ship-wide system of systems and providing appropriate operation and maintenance services for the computer-based system. For the purpose of this Guide, the system integrator is a role with clear responsibilities. During the design and delivery phases, the shipyard is the default system integrator. During the operation phase, the Owner is the default system integrator.

3.1.29 Supplier

A general term referring to a contracted or subcontracted provider of services, system components, or software, which may be an organization or an individual.

3.1.30 System supplier

A contractor or subcontractor who provides system components or software under the coordination of the system integrator, which may be an organization or an

individual. For the purpose of this Guide, the system supplier is a role with clear responsibilities.

3.1.31 Service supplier

Individuals or companies employed by non-IACS members, at the request of equipment manufacturer, shipyard, owner or other customers, engaging in related inspection work and providing services such as measurement, testing or maintenance of safety systems and equipment for ships (including offshore mobile platforms). Classification society surveyors may make decisions based on the results they provide when carrying out classification or statutory certification services.

3.1.32 Black-box description

A description of system functions, behaviors, and performances observed from outside the system.

3.1.33 Black-box test methods

A test method which verifies the functionality, performance, and robustness of a system, subsystem or component only by manipulating inputs and observing outputs, which does not require any knowledge about the internal work of the system, but only requires a focus on the observable behavior of the system/ component under test to achieve the required level of verification.

3.1.34 Failure mode description

Documentation describing the effects of system failures (other than failures of system supporting devices), which shall include a list of failures to be evaluated, the system response to each failure, and evaluation of the consequences of each failure.

3.1.35 Parameterization

A process to configure and adjust the functions of the system and software by changing parameters, which usually does not require computer programming and is done by the system supplier or service provider, rather than by the operator or end user.

3.1.36 Robustness

An indicator that can be used to reflect the ability of a system to maintain the stable operation of its functions in the face of changes in internal structure or external environment.

3.1.37 Simulation test

A method for monitoring, controlling or safety system testing in which the EUC is partially or completely replaced by simulation tools, or the communication network and lines are partially replaced by simulation tools.

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 Engineers.

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 SAT: System Acceptance Test.

3.2.8 SOST: System of Systems Test.

3.2.9 PE: Programmable Electronic.

3.2.10 COTS: Commercial Off-The-Shelf.

3.2.11 IT: Information Technology.

3.2.12 OT: Operational Technology.

3.2.13 PMS: Planned Maintenance System.

3.2.14 SSLS: Ship Software Logging System.

4 Categorization of computer-based system

4.1 Based on the impact of system function failure, the computer-based systems are classified as shown in Table 4.1.

Table 4.1 System categories

Category	Failure effects	Typical system functionality
I	The failure of those systems will not endanger the safety of personnel, the safety of the vessel and/or the environment.	- Monitoring, informational and administrative functions.
II	The failure of those systems will ultimately endanger the safety of personnel, the safety of the vessel and/or the environment.	- Alarm, monitoring and control functions necessary to maintain the vessel in normal operational and habitable conditions.
III	The failure of those systems can result in an immediate hazard or disaster to the safety of personnel,	- Control functions to maintain the propulsion and steering of the vessel;

	the safety of the vessel and/or the environment.	- Vessel safety function.

4.2 Category I systems generally do not require verification by China Classification Society (hereinafter abbreviated as CCS) because failures in these systems do not lead to dangerous situations. However, the information related to Category I systems shall be provided to determine the correctness of the system category or to ensure that Category I systems do not affect the operation of Category II and Category III systems. The category of a system shall always be assessed in conjunction with a specific vessel, so the categorization of a system varies from one vessel to the next, and the system categorization in Table 4.2 is for reference only and the examples are not exhaustive.

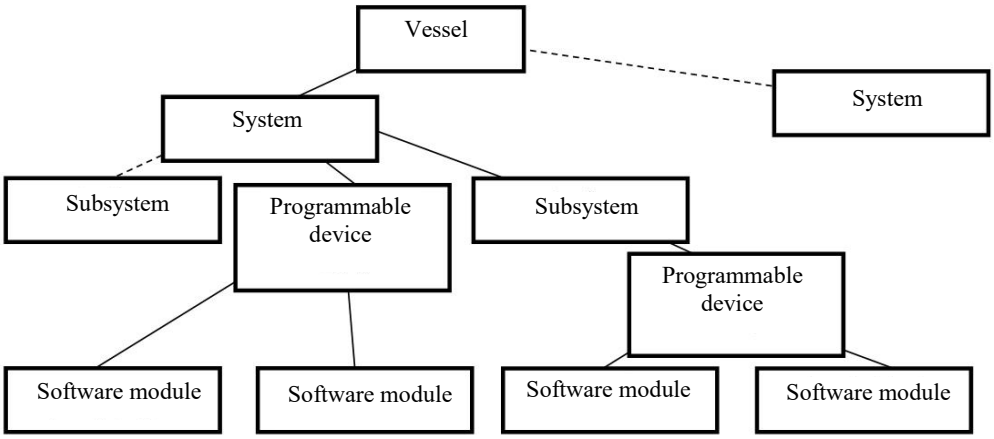
Table 4.2 System category examples

System category	Examples
I	(1) Fuel monitoring system; (2) Maintenance support system; (3) Diagnostic and troubleshooting systems; (4) CCTV; (5) Cabin security and entertainment system; (6) Fish detection system.
II	(1) Control, monitoring and safety systems for cargo containment system; (2) Bilge water detection and related control of bilge pumps; (3) Fuel oil treatment system; (4) Inert gas system (5) Remote control system for ballast water valve; (6) Stable and floating control systems, such as anti-pitching fin control system; (7) Alarm, monitoring and safety systems for propulsion and auxiliary machinery.
III	(1) Propulsion control system, i.e., generating and controlling mechanical thrust to move the vessel (excluding equipment used only under maneuvering conditions, such as bow thruster); (2) Steering gear control system; (3) Electric power system (including power management system); (4) Vessel safety systems, including fire detection and extinguishing, water ingress detection and drainage,

	internal communication system related to evacuation, and vessel system related to the operation of life-saving appliances; (5) Dynamic positioning system marked with DP2 and DP3; (6) Drilling system.

4.3 Object

4.3.1 A typical hierarchy and relationship of a computer-based system is shown in the figure below.



Note: Dashed lines in the figure indicate undeveloped branches.

Figure 4.3.1 Hierarchy of Computer-based System

5 Quality system requirements

5.1 Quality assurance system

5.1.1 In the design and development stage of the software and hardware of computer-based system, as well as the integration stage of subsystems, systems and system of systems, a global top-down management method spanning the whole lifecycle shall be adopted. The method shall be developed according to standards accepted by CCS and shall be validated by CCS.

5.1.2 The system supplier and system integrator shall demonstrate a certain level of product quality assurance capability and quality management through a quality assurance system. The system supplier and the system integrator shall establish management system to ensure that the computer-based system comply with CCS specifications and relevant conventions.

5.1.3 The system supplier and system integrator shall establish and implement a quality management system certified against ISO9001 or equivalent standard, and hold

valid certificates issued by a nationally recognized institution, or be confirmed by CCS through specific assessments. The system supplier and system integrator shall follow recognized quality standards, such as those specified in ISO9001 and IEC/ISO90003, when developing computer-based systems. The quality management system shall include at least the following:

Table 5.1.3 Quality Management System

Area		Role	
No.	**Topic**	**System supplier**	**System integrator**
1	Responsibilities and competencies of employees.	x	x
2	The complete lifecycle of delivered software and of associated hardware.	x	x
3	Specific procedures for unique identification of a computer-based system, it's components and versions.	x	
4	Creation and update of the vessel's system architecture.		x
5	The organization setting for purchasing software and related hardware from suppliers.	x	x
6	The organization setting for software code writing and verification.	x	
7	The organization setting for system validation before integration in the vessel.	x	
8	Specific procedures for conducting and approving of the system during the FAT and SAT.	x	x
9	Creation and update of system documentation.	x	
10	Specific procedures for software modification and installation on board the vessel, including interactions with the shipyard and the owner.	x	x
11	Specific procedures for the verification of software code.	x	
12	Procedures for integrating a system with other systems and testing of the system of systems for the vessel.	x	x
13	Procedures for managing changes to software and configurations before FAT.	x	
14	Procedures for managing and documenting changes to software and configurations after FAT.	x	x
15	Tracking checkpoints of the organization for compliance with the quality management system. A checkpoint may be a required deliverable, a test, a technical review meeting, or an expert review meeting.	x	x

5.2 Quality plan

5.2.1 The system supplier and the system integrator shall develop a quality plan for the lifecycle of the system, including the software quality plan.

5.2.2 The software quality plan shall regulate the activities of the software throughout its lifecycle, specifying the relevant procedures, responsibilities and documents, including configuration management, whose development may be referenced to the requirements of IEEE 730.

5.2.3 For the software of a Category II or Category III system, the quality plan shall include the safety function requirements, and specific assurance methods shall be designed to verify and validate whether the safety function requirements are met.

5.2.4 The configuration management of the computer-based system shall be established, as detailed in 5.7.

5.3 In-production quality control

5.3.1 Sound quality assurance measures, plans and organizations shall be established to ensure the quality of the computer-based system.

5.3.2 The system supplier and system integrator shall have product quality control documents for the computer-based system, which shall accurately describe the development process or production process of the computer-based system, and clearly describe the quality control requirements of each process in words and charts. It shall also include clear control objects, control standards, control methods, inspection methods, and documents proving the implementation of production quality control measures. For products with safety-related functions, documentary evidence of passing experiments or tests is also required.

5.4 Final test report

The system integrator shall conduct final tests to the computer-based system and provide test reports (such as SAT report and SOST report).

5.5 Software traceability

5.5.1 The modification of programming content and data as well as the version change must be identified and documented in accordance with the requirements of the quality management system, so as to ensure that the forming process of software quality can be traceable when required. Through quality control documents such as software configuration management and software version description, the procedures must be followed for the modification of programming content and data as well as for the version change (especially the process of informing the owner of software change and onboard installation) , and these modifications or changes shall be recorded.

5.5.2 These documents shall be kept for at least one year after the decommissioning of the software. The system integrator shall keep evidences that are clear enough to prove the software decommissioning.

5.6 Security policy

5.6.1 The software shall not be modified unless authorized. Regardless of a physical system or remotely controlled system, physical and logical security measures shall be taken to prevent unauthorized or inadvertent modification.

5.6.2 All firmware, software codes, executable programs and physical media intended for installation on board shall be scanned for vulnerabilities, viruses, malware, etc. prior to installation. The results of the security scan shall be recorded in the test report, software registry, or similar documents.

5.7 Configuration management

5.7.1 The configuration management is intended to ensure the consistency of developed deliverables when some deliverables change. Configuration management generally includes hardware configuration management and software configuration management.

5.7.2 Requirements:

(1) During the software development lifecycle, administrative and technical means shall be used to manage software changes and ensure that specified software safety requirements, such as security policy, are always met.

(2) All necessary operations shall have been performed to meet the software requirements.

(3) Accurate and unique identification of all configuration items necessary to ensure the integrity of the computer-based system shall be maintained. Configuration items shall at least include: safety analysis and requirements; software requirement documents and design documents; software source code module; test plan and test results applied to software components and packages of the computer-based system; and all tools and development environments used to create, test, or implement the software of the computer-based system.

(4) Physical and logical security measures shall be taken to prevent unauthorized or unintentional changes in accordance with the change management procedure. Change requests shall be documented. The impact of the change shall be analyzed for the approval or rejection of the request. The details and authorization of all permitted changes shall be documented. The composition of all software baselines (including reconstruction of earlier baselines) shall be ensured.

(5) Information such as configuration status, release status, judgment and approval of all changes, and details of changes shall be documented for review.

(6) Software release should be documented. The main backup of the software and all related documents shall be stored throughout the lifecycle of the released software for the maintenance and change of the software.

6 Technical requirements

6.1 System identification

The methods and applications for identifying the name, version, identifier and manufacturer of the computer-based system shall be provided. It is recommended that the computer-based system automatically report its software status to the Ship Software Logging System (SSLS) in accordance with international standard ISO24060.

6.2 Data links

6.2.1 The data link failure of Category II and Category III computer-based systems shall be specified in the risk assessment and analysis /FMEA, including:

(1) A single failure of the data link of Category III systems shall not result in malfunction of the vessel. Any effect of such failures shall comply with the fail -to-safe principle.

(2) For Category II and Category III systems, the loss of any function of the remote control system shall be capable of being compensated locally/manually.

(3) The data link shall be designed with means to prevent or cope with excessive communication rates.

(4) The data link shall be configured with built-in test capability to detect the failure or performance problems of the link and the data communication failure of the nodes connected to the link.

(5) An alarm shall be initiated when a failure is detected.

6.2.2 Wireless data link shall not be used for Category III systems unless specifically considered by CCS after engineering analysis in accordance with acceptable international or national standards. However, for systems of other categories, wireless data link may be used, provided the following requirements are met:

(1) The recognised international wireless communication system protocols shall be used and shall meet the following requirements:

a. Message integrity: The received message will not be destroyed or changed compared with the sent message through failure prevention, detection, diagnosis and correction.

b. Configuration and device authentication: Shall only permit connections of devices that are included in the system design.

c. Message encryption: Confidential and/or critical data shall be protected.

d. Security management: Network assets shall be protected from illegal access.

(2) The internal wireless system within the vessel shall meet the requirements of the International Telecommunication Union and the Flag State's authorities for radio frequency and power levels.

(3) The system operation shall be implemented with the provisions of port and local regulations on radio frequency transmission considered, and the use of wireless data communication links is prohibited due to frequency and power restrictions.

(4) The wireless data communication equipment shall be tested during harbour and sea trials to demonstrate that, under the expected operating conditions, the radio frequency transmission will not cause failure of itself and any other equipment due to electromagnetic interference.

6.3 CCS verification

The system design documents shall indicate the degree of compliance of the computer-based system with the technical requirements. CCS will verify the implementation of technical requirements as part of system description, FAT and SAT.

7 Documentation requirements

At least the following documents related to the computer-based system shall be submitted to CCS as required.

7.1 Basic documents to be submitted by the system supplier:

No.	Document name	System category		
		Category I	Category II	Category III
1	Quality plan	Ⓘ (when necessary)	Ⓘ	Ⓘ
2	System description	Ⓘ (when necessary)	Ⓐ	Ⓐ
3	Environmental compliance test report	Ⓘ (when necessary)	Ⓘ	Ⓘ
4	Software test report	Ⓘ (when necessary)	Ⓘ (when necessary)	Ⓘ (when necessary)
5	System test report	Ⓘ (when necessary)	Ⓘ (when necessary)	Ⓘ (when necessary)
6	FAT program	-	Ⓐ	Ⓐ
7	FAT report	-	Ⓘ	Ⓘ

No.	Document name	System category		
		Category I	Category II	Category III
8	Other FAT documents (e.g., User Manual, etc.)	-	Ⓘ (when necessary)	Ⓘ (when necessary)
9	Change management procedure	Ⓘ (when necessary)	Ⓘ	Ⓘ

Note: The symbols used in the table and their meanings are as follows: Ⓐ Submit to CCS for approval; Ⓘ Submit to CCS for reference.

7.2 Basic documents to be submitted by the system integrator:

No.	Document name	System category		
		Category I	Category II	Category III
1	Quality plan	Ⓘ (when necessary)	Ⓘ	Ⓘ
2	List of system categorizations	Ⓐ	Ⓐ	Ⓐ
3	Risk assessment report	Ⓐ (when necessary)	Ⓐ (when necessary)	Ⓐ (when necessary)
4	System architecture description	Ⓘ	Ⓘ	Ⓘ
5	SAT program	-	Ⓐ	Ⓐ
6	SAT report	-	Ⓘ	Ⓘ
7	SOST program	-	Ⓐ	Ⓐ
8	SOST report	-	Ⓘ	Ⓘ
9	Software change management procedure	Ⓘ (when necessary)	Ⓘ	Ⓘ

Note: The symbols used in the table and their meanings are as follows: Ⓐ Submit to CCS for approval; Ⓘ Submit to CCS for reference.

8 System lifecycle

8.1 Division of system lifecycle

8.1.1 The lifecycle of the system is divided into five stages: concept, requirement, implementation, verification and operation. Each stage is further divided according to the purpose and scope, and the relationship and requirements are shown in the following table and figure.

Note: As per IEC61508, the computer-based system is generally considered as an EUC in the definition of system lifecycle.

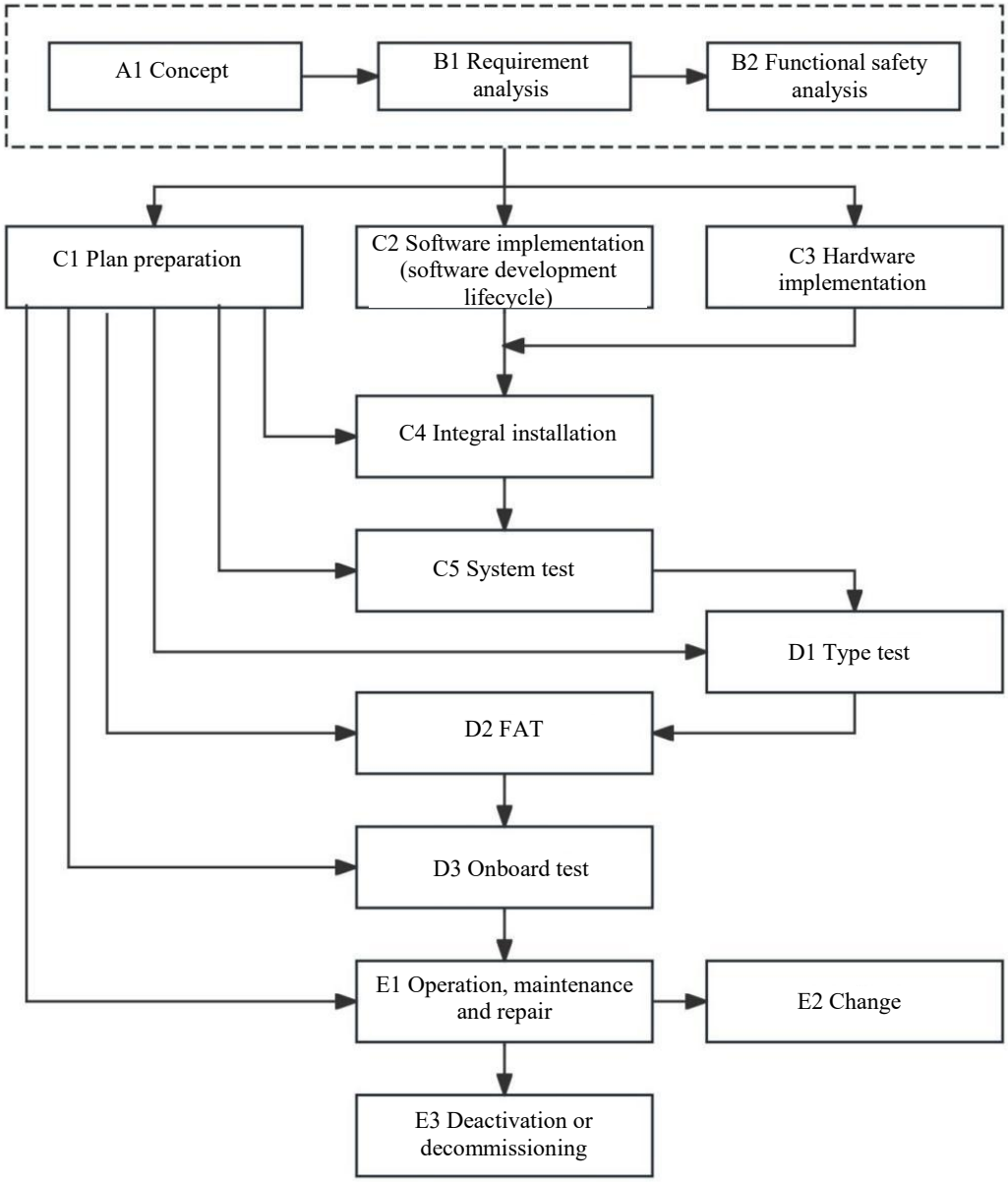


Figure 8.1.1 System Lifecycle

Table 8.1.1 Overview of System Lifecycle

System lifecycle stage		Purpose	Requirements	Input	Output
Figure 8.1.1 Box No.	Title				
A Concept					

A1	Concept	Improve the understanding of computer-based system and it's environments (actual, legal environments etc.) to better implement the lifecycle activities.	Get a comprehensive understanding of the computer-based system and its required control functions and the actual environment; identify possible hazard sources; obtain relevant information on the identified hazards; obtain current safety regulations; consider the hazards arising from the interaction between adjacent EUCs; document the information and results required above.	All relevant information necessary to satisfy the requirements of this clause.	Information from concept to overall scope.
B Requirement					
B1	Requirement analysis	When developing or modifying a computer-based system, describe the purpose, scope, definition and function of the system, including: (1) Determine the boundary of the computer-based system; (2) Define the scope of the safety analysis.		Information from concept to overall scope.	System requirement specifications.

B2	Functional safety analysis	In order to ensure the safety and reliability of the computer-based system, it shall be proved that for a single failure, the system shall enter the fail-to-safe status, and the functions of the system in operation shall not be lost or degrade to the extent that the acceptable performance standards specified by the CCS cannot be met.		System requirement specifications.	Safety-related functional requirement documentation (containing information and records of the allocation of safety requirements).
C Implementation					
C1	Preparation of plan	Clearly define the working steps involved in the specified regulations and technologies, and prove that requirements for the installation, operation and maintenance of the computer-based system are met.	Develop a plan for the installation, operation and maintenance of the computer-based system to ensure that the required functional safety is maintained during operation and maintenance. Develop the FAT program and onboard test programs (including SAT program and SOST program), which shall include safety-related function tests.	System requirement specifications; Safety-related functional requirement documentation.	Quality plan; System installation, operation and maintenance plan; Type test program; FAT program; SAT program; SOST program.

C2	Software implementation	Develop computer-based system software that meets the system requirement specifications and safety-related functional requirement documentation.	See Chapter 9 "Software development lifecycle" for details.	System requirement specifications.	Evidences proving that each computer-based system software meets the computer system requirement specifications, including software test reports. See Chapter 9 "Software development lifecycle" for details.
C3	Hardware implementation	Develop computer-based system hardware that meets the system requirement specifications and safety-related functional requirement documentation.	See 8.2.1.3, 8.2.1.4, 6.2.	System requirement specifications.	Evidences proving that each computer-based system hardware meets the system requirement specifications.
C4	Integral installation	Install the software and hardware of the computer-based system to form a complete computer-based system.		System requirement specifications; System installation, operation and maintenance plan.	An assembled computer-based system ready for use.

C5	System test	The system test is mainly intended to allow the system supplier to internally verify that the entire computer-based system complies with specifications, approval documents, and applicable regulations.	See 8.2.1.6.	System requirement specifications; List and version number of software installed in the system; Software function description; Software operation and maintenance manual; List of interfaces between the system and other shipboard systems; List of data transmission standards.	System test report
D Verification					
D1	Type test	Verify that the computer-based system meets the system requirement specifications (including safety-related functions); Verify that the system meets the requirements of GD019-2024.	For Category II and Category III systems, the environmental tests shall be carried out as per GD019-2024; For Category I systems, the environmental tests, if necessary, can be carried out according to GD019-2024.	System requirement specifications; Safety-related functional requirement documentation; Type test program.	Evidence proving that the computer-based system meets the requirements of safety-related functions; Type test reports, such as environmental compliance test report.

D2	FAT	Perform acceptance test of the computer-based system at the factory.	The FAT report shall record: ① tools and equipments used; ② activities involved in FAT; ③ differences between actual results and expected results,as well as their handling. When there is a difference between the expected result and the actual result, an analysis shall be made to determine to continue the test or propose a change request.	System requirement specifications; FAT program.	FAT report; Other FAT documents (e.g., User Manual, etc.)
D3	Onboard test	Verify the ability of the system to perform its intended function after all systems are integrated through onboard test. Onboard test includes SAT and SOST.	The SAT shall verify that the functions of a system can be realized normally under the actual hardware environment with the final application software. The SOST shall verify that all functions can be realized normally when all systems are integrated.	System requirement specifications; List of system categorizations; System architecture description; SAT program; SOST program.	SAT report; SOST report.
E Operation					

E1	Operation, maintenance and repair	Operate, maintain and repair the computer-based system to maintain the required functional safety.	See 8.3.	System installation, operation and maintenance plan.	Continued maintenance of functions of the computer-based system; Time-ordered documentation for the operation, maintenance and repair of the computer-based system.
E2	Change	Ensure that the computer-based system is still under control during and after the change phase.	Stakeholders shall be informed of changes and change plans for approved systems in advance and conduct impact analysis. The change shall be implemented by returning to the appropriate lifecycle stage. The changed software shall be verified to demonstrate continued compliance with the relevant requirements of the computer-based system. Changes shall be documented by stakeholders. For Category II and Category III systems, subsequent major changes to the software and	System requirement specifications; Quality plan; Description of change; Change impact analysis; Change management procedure; Test programs for the corresponding stages.	Continued compliance of the computer-based system with the requirements for functional safety during and after the change; Time-ordered documentation for the operation, maintenance and repair of the computer-based system; Test report or summary.

			hardware shall be applied to CCS for approval. Note: Major changes refer to the modifications that affect the safe operation and/or safety of the vessel.		
E3	Deactivation or decommissioning	During and after the deactivation or decommissioning activities of the computer-based system, it shall be ensured that the functional safety of the computer-based system is adapted to this situation.	Prior to deactivation or decommissioning activities, an impact analysis shall be performed and a plan developed, including the shutdown and dismantling of the system. The operation manual of the computer-based system shall give prompts about the destruction and disposal of sensitive information.	Requirements in functional safety management regulation for deactivation or decommissioning of the computer-based system.	Safe deactivation or decommissioning of the computer-based system.

8.1.2 The following lists the supplements to the system lifecycle table by role.

8.1.3 The software development lifecycle is detailed in Chapter 9.

8.2 Development requirements for computer-based system

The development requirements of computer-based system cover the system lifecycle stages of concept, requirement, implementation and verification. The development requirements for role are described in detail below for different computer-based system.

8.2.1 Requirements for system supplier

8.2.1.1 For the design, manufacture, delivery and maintenance of the specific computer-based system to be delivered, the system supplier shall formulate a quality plan and implement a quality management system. A verification shall be performed to demonstrate that the system supplier has implemented all the relevant requirements in Table 5.1.3. For Category II and III systems, the quality plan shall be submitted to CCS for reference during the FAT.

8.2.1.2 A method for unique identification of the computer-based system, different software components, and different versions of the same software components shall be adopted. This approach shall be applied throughout the lifecycle of system and software, and is part of the quality management system.

8.2.1.3 The system's specification and design shall be determined in the system description. The system description, in addition to being served as a specification for design and implementation, is intended to ensure that the entire system is delivered in compliance with the applicable specifications and clauses. The system description shall contain the following:

(1) Purpose and main functions, including safety-related functions;

(2) Defined system category;

(3) Key performance characteristics;

(4) Compliance with technical requirements and CCS rules;

(5) User interface/mimics;

(6) Communication and interface: identify and describe interfaces with other vessel systems;

(7) Hardware arrangement:

a. Network architecture/topology, including all network components such as switches, routers, gateways, firewalls, etc.;

b. The internal structure of all interfaces and hardware nodes of the system (e.g., operator stations, displays, computers, programmable devices, sensors, actuators, I/O modules, etc.);

c. I/O allocation (mapping of field devices to channels, communication links, hardware units, logical functions);

d. List of technical specifications of hardware and related external equipments;

e. Power supply arrangement;

f. Failure mode description.

The above information is collectively referred to as the system description, which can be divided into different documents and/or models.

For Category II and III systems, the system description shall be submitted to CCS for approval; For Category I systems, the system description shall be submitted to CCS for reference when necessary.

8.2.1.4 Environmental test shall be carried out for the hardware of the system and subsystems according to CCS Guidelines for Type Approval Test of Electric and Electronic Products. For Category II and III systems, the environmental compliance test report or the type approval certificate shall be submitted to CCS for future reference; For Category I systems, the environmental compliance test report or type approval certificate shall be submitted to CCS for reference when necessary.

8.2.1.5 The software created, changed or configured for the delivery of the project shall be developed in accordance with the standards selected in the quality plan and its quality assurance activities shall be evaluated. Quality assurance activities can be carried out at multiple levels of the software structure and shall include customized software and configured component (such as software libraries) as appropriate.

If black-box test methods are used, the verification of the software shall include at least the following:

(1) Correctness, completeness and consistency of any parameterization and configuration of all software components;

(2) Intended functionality;

(3) Intended robustness.

In software quality assurance activities, test methods such as "software unit test" or "developer test" are often used, and verification methods such as "code review", "static code analysis", etc. are also used.

For the softwares in Category I, Category II and Category III systems, the scope, purpose and results of all reviews, analyses, tests and other verification activities performed shall be documented in the test report. The test report shall be submitted to CCS for future reference when necessary.

8.2.1.6 Before FAT, the internal system test shall be carried out as far as practicable. The system test is mainly intended to allow the system supplier to verify that the entire system delivered complies with specifications, approval documents and applicable regulations. Further, the system is built and ready for FAT.

The system integration test shall be performed between the system, subsystems, and software modules in order to check the correct execution of software functions, as well as the normal interaction and function execution between the software and the

hardware it controls. The failures shall be simulated as realistically as possible to verify that the system has the appropriate failure detection and failure response capabilities.

Some tests can be performed by using simulation tools and replica hardware. The test environment shall be documented, including a description of any simulators, emulators, test stubs, test management tools, or other tools that affect the test environment and their limitations. The test cases and test results shall be recorded in the test programs and test reports respectively.

For the purpose of system test, the following materials shall be prepared:

(1) System description, including system requirement specifications;

(2) List and version numbers of software installed in the system;

(3) Software function description;

(4) Software operation and maintenance manual;

(5) List of interfaces between the system and other shipboard systems;

(6) List of data transmission standards.

The system test shall verify at least the following aspects of the system:

(1) Functionality;

(2) Failures and their effects (including diagnostic functions, detection, alarm response);

(3) Performance;

(4) Integration between software and hardware;

(5) Human-machine interfaces;

(6) Interfaces with other systems.

For Category I, II and III systems, the system test report shall be submitted to CCS for reference when necessary.

8.2.1.7 The FAT shall be arranged before the system is installed onto the vessel. The FAT is mainly intended to demonstrate to CCS that the system has been completed and complies with the applicable classification rules so that the issuance of a CCS certificate is allowed.

The FAT program shall include the representative test items selected from the system test (see 8.2.1.6), including normal system functional test and failure response test. All software and physical media intended for installation on board shall be scanned

for vulnerabilities, viruses, malware, etc. prior to installation. For a Category II or Category III system, a cyber test shall also be performed to verify compliance with cyber resilience requirements. The cyber test may be carried out as part of the onboard test, provided an agreement for this is reached among stakeholders.

Generally, the FAT shall be performed with the project specific software operating on the actual hardware components to be installed on board and with the necessary means for functional simulation and failure response. Other test solutions, such as the use of replica hardware or simulation tools (emulators), shall be subject to CCS approval before implementation.

Functional test and failure response test programs shall be submitted to CCS, which may require a FMEA to support the fault response test program. For Category II and III systems, CCS will also validate the identification application as part of the FAT.

For each test case, it is required to indicate whether the test is passed or not, and the test result shall be recorded in the test report. The test report shall also contain a list of software (including software versions) that have been installed in the system at the time of test.

For complex systems, there may be a large difference in the test range between the internal system test before the FAT and the FAT; For some systems, however, the test range may be the same.

For Category II and III systems, the FAT program shall be subject to CCS approval in advance; The FAT shall be conducted under the witness of CCS, and shall include functional test and failure response test; The FAT report shall be submitted to CCS for future reference, and other FAT documents (such as User Manual, system test report, etc.) shall be submitted to CCS for future reference when necessary.

8.2.1.8 The initial installation and subsequent update of the computer-based system software on the vessel shall be carried out in accordance with the change management procedure agreed between the system supplier and the system integrator. The change management procedure shall comply with the relevant requirements in 8.4; Cyber security measures shall comply with the relevant requirements of Guidelines for Ship Cyber Security adopted by CCS.

For Category II and III systems, the change management procedure and related records shall be submitted to CCS for future reference (see 8.4.12).

8.2.2 Requirements for system integrator

8.2.2.1 The shipyard is regarded as the system integrator during the vessel development and delivery phase, unless another organization or individual is specifically designated.

8.2.2.2 For the installation, integration, completion and maintenance of shipboard computer-based system, the system integrator shall develop a quality plan and

implement the quality management system. A verification shall be performed to demonstrate that the system integrator has implemented all the relevant requirements in Table 5.1.3. For Category II and III systems, the quality plan shall be submitted to CCS for reference during SAT/SOST.

8.2.2.3 For a computer-based system delivered to a specific vessel, the category of the system shall be determined based on the effects of the failure of the system. The system category, once determined, shall be notified to the system supplier concerned. For the categorization of Category I, Category II and Category III systems, a list of system categorizations shall be prepared and submitted to CCS for approval.

8.2.2.4 Upon the request of CCS, the system integrator shall conduct a risk analysis of the specific system of the vessel and prepare a risk assessment report to determine the applicable category of the system. The method of risk assessment can be determined according to IEC/ISO 31010 Risk Management - Risk Assessment Techniques.

The risk assessment report of Category I, II and III systems shall be submitted to CCS for approval when necessary. If the system category is amended based on risk assessment, a consent from CCS and the system supplier may be required. When the risk of the computer-based system is obvious, the submission of the risk assessment report is allowed to be exempted, but the system integrator shall submit supporting documents to explain the reason for the exemption.

8.2.2.5 The system of systems of the vessel shall be specified and a system architecture description shall be prepared. The system architecture description, which assigns functions to different systems and defines the main interfaces between systems, lays a foundation for the determination of system category and the development of different integrated systems. At the same time, it also serves as the basis for the test of the integrated systems on the vessel level (see 8.2.2.7).

The system architecture description shall contain at least the following:

(1) Overview of the total system architecture (the system of systems);

(2) The purpose and main functionality of each system;

(3) Communications and interfaces between different systems.

For Category I, II and III systems, the system architecture description shall be submitted to CCS for future reference.

8.2.2.6 The SAT shall be carried out on the vessel. The SAT is mainly intended to verify the functional operation of the computer-based system after its installation and integration with the related shipboard mechanical/electrical/process systems, including possible interfaces with other control and monitoring systems. For Category II and III systems, CCS will also validate the identification application as part of the SAT.

For each test case, it is required to indicate whether the test is passed or not, and the test result shall be recorded in the test report. The test report shall also contain a list of software (including software versions) that have been installed in the system at the time of test.

For Category II and III systems, the SAT program shall be subject to CCS approval in advance; The SAT shall be performed under the witness by CCS; The SAT report shall be submitted to CCS for reference.

8.2.2.7 After different systems are installed and integrated in the final environment on board, a system of systems test (i.e., SOST) shall be carried out. The SOST is mainly intended to verify the functionality of different systems after a complete installation, including all interfaces and interdependencies, for compliance with requirements and regulations. The test shall verify at least the following aspects of the system of systems:

(1) The overall functionality of the system of systems as a whole;

(2) The failure response between systems, which shall comply with the fail-to-safe principle;

(3) Performance;

(4) Human-machine interfaces;

(5) Interfaces and interconnections between different systems.

For complex systems, the test range of the SAT and SOST may vary greatly, while for some systems, the test range of these two tests may overlap or be the same. When the test range is similar, the two tests can be combined into one.

For Category II and III systems, the SOST program shall be subject to CCS approval in advance. The SOST shall be performed under the witness by CCS. The SOST report shall be submitted to CCS for reference.

8.2.2.8 The system integrator shall follow the change management procedure described in 8.4 to make changes to the system. For Category II and III systems, the change management procedure and related records shall be submitted to CCS for future reference (see 8.4.12).

8.3 Maintenance requirements for computer-based system

8.3.1 Requirements for stakeholders

8.3.1.1 The owner is regarded as the system integrator during the vessel operation phase, unless another organization or individual is specifically designated. The owner shall promptly inform CCS of the system integrator designated by it. Any changes to

the system shall be jointly implemented by the system integrator and the system supplier.

8.3.1.2 The system integrator shall ensure that necessary change management procedures for software and hardware are stored on the vessel and that any software modifications/upgrades are carried out in accordance with these procedures. See 8.4 for specific requirements of change management. The system integrator shall record the changes of the computer-based system during the operation phase. The record shall contain relevant software version information and other relevant information as described in 8.4.10.

8.3.1.3 The system supplier shall follow the maintenance procedures for the computer-based system, including the change management procedure described in 8.4. Before making changes to the shipboard computer-based system, the system supplier shall ensure that the planned system changes have passed the relevant internal tests.

8.3.2 System safety during the maintenance phase

8.3.2.1 The maintenance department shall be competent for the activities concerned and, in particular, meet the following requirements:

(1) Necessary training shall be organized for maintenance personnel engaging in failure diagnosis, repair and system test;

(2) Operators shall be trained as required;

(3) Periodic retraining shall be organized of maintenance personnel.

8.3.2.2 The training, experience and qualifications of personnel involved in maintenance activities shall be documented.

8.3.2.3 For the operator, the procedures for receiving, recording, resolving, and tracking problem shall be established, as well as applying for changes. An operational plan for the computer-based system shall be established and maintained, including identification of configuration items, operation regulations and planned maintenance activities. The plan shall also include issues related to software migration and decommissioning.

8.3.2.4 Regulations shall be established for the analysis of operation and maintenance performance, especially the following:

(1) Regulations for identifying system failures that endanger functional safety, including regulations used for routine maintenance to detect repetitive faults;

(2) Regulations to evaluate whether the demand rate and the failure rate during operation and maintenance are consistent with the assumptions made during system design.

8.3.2.5 A systematic analysis of hazardous accidents (or potential accidents that create hazards) shall be conducted, and regulations shall be prepared to minimize the probability of their recurrence.

8.3.2.6 Regulations to maintain accurate information on potential hazards and safety related system shall be prepared.

8.3.2.7 Emergency service information and training provisions shall be identified where appropriate.

8.3.2.8 Regulations for making modifications to safety related systems shall be prepared.

8.3.2.9 The approval regulations and competent authorities required for modification shall be clearly defined.

8.3.2.10 The operator shall carry out a test for each new modification, upgrade or release of the system and/or component. The component released for operation and use shall meet the specified standards. If the interface of the released component has been modified, the integration testing is required.

8.3.2.11 Major modifications to software and data, as well as version changes, must be recorded and submitted to CCS approval.

8.3.2.12 The configuration management of computer-based system during the maintenance phase includes the following:

(1) Development of configuration control regulations;

(2) Unique identification of all elements of a configuration management item (software and hardware);

(3) Prevention of unauthorized items from application in service.

8.3.2.13 Configuration audits shall be performed regularly to verify the integrity of the operational configuration.

8.4 Change management requirements for computer-based system

8.4.1 Change management of computer-based system throughout its lifecycle shall be implemented based on the negotiation between stakeholders. In general, change management includes at least the following three stages:

(1) Development and internal validation stage prior to FAT, in which system supplier and sub-suppliers are involved;

(2) Stage from FAT to delivery to the owner, in which system supplier, system integrator, CCS and the owner are involved;

(3) Operation stage, in which system supplier, service provider, the owner and CCS are involved.

8.4.2 If changes are required to computer-based system after approval by stakeholders (usually the system integrator and CCS during FAT), the established change management procedures shall be followed.

8.4.3 Stakeholders should develop the documented change management procedures covering both the software and hardware of the computer-based system. After FAT, the system supplier shall manage all changes to the system according to the procedures, including the purchase of new versions of software, new hardware, modified control logic, and changed configuration parameters, etc. The change management procedure shall at least describe the activities listed in 8.4.4 to 8.4.11.

8.4.4 The system supplier shall ensure that each system and software version is uniquely identified, as detailed in 8.2.1.2.

8.4.5 A mechanism shall be established for processing the files that make up the software master files. The permissions of the personnel involved shall be clearly specified, as well as the tools and mechanisms to ensure the integrity of the master files.

8.4.6 The methods to back up and restore the software of the shipboard computer-based system shall be clearly defined.

8.4.7 An impact analysis shall be performed for any change to the computer-based system prior to its implementation. The results of the impact analysis determine the extent to which the change activities are performed. The purposes of this impact analysis include:

(1) Determining the criticality of the change;

(2) Determining the impact on existing documents;

(3) Determining the needed verification and test activities;

(4) Determining whether other stakeholders need to be informed of the change;

(5) Determining whether an approval from other stakeholders (e.g., CCS and/or the owner) is required.

8.4.8 When the maintenance includes the installation of a new version of software in the system, it shall be possible to perform a rollback of the software to the previous installed version to restore the system to a known stable state. The rollback shall be recorded and analyzed to identify and eliminate the root cause of the change failure.

8.4.9 System changes shall be tested and verified as far as possible before being installed onboard. After installation, the testing and verification shall be carried out again onboard according to the documented verification program, including:

(1) Verifying that the new functions and/or improvements are working as intended;

(2) Verifying through regression test that the changes don't have any negative impact on the functionality or capabilities that should not have been affected.

Note: Regression test refers to a re-testing carried out after the old code is modified to confirm that the modification has not introduced new errors or caused errors in other code.

8.4.10 Changes to the system and software shall be recorded to ensure visibility and traceability of changes. The change records shall contain at least the following:

(1) The purpose of change;

(2) The description of changes and modifications;

(3) The main conclusions of the change impact analysis (see 8.4.7);

(4) The identity and version of any new system or software (see 8.4.4);

(5) Test reports or test summaries (see 8.4.9).

8.4.11 Software changes may be recorded in the planned maintenance system (PMS), the software registry, or similar files. If necessary, the software changes shall be subject to a security scan.

8.4.12 CCS verification of change management

8.4.12.1 During the operation phase of the vessel, CCS will usually verify the change management as a part of the annual survey of the vessel. At the time of survey, the change management procedures and relevant change records shall be provided to CCS. If the changes require prior approval from CCS, the procedures and documents related to the changes may be verified during the application for approval.

8.4.12.2 During the shipbuilding phase, CCS's verification of change management is divided into two parts:

(1) Verification of change management procedures as part of the quality management system (see 5.1.3);

(2) Verification during and after the FAT, in combination with the actual application of the change management procedures in the specific project.

9 Software development lifecycle

9.1 The system supplier and the system integrator shall develop a quality plan for the software development lifecycle. Administrative and technical means shall be used for control in the lifecycle of the software so as to manage software changes and ensure that software safety requirements are met, and to demonstrate that the system supplier

and system integrator have effective quality control programs that can meet requirements at all stages of the software development lifecycle.

9.2 The quality plan for software shall include the following:

9.2.1 The software development lifecycle shall meet the requirements specified in 5.2.

9.2.2 Throughout the software development lifecycle, configuration management shall be implemented for the computer-based system, especially the following shall be included:

(1) Necessary configuration control nodes for a specific phase;

(2) Unique identification of all components of software and hardware of the computer-based system;

(3) Prevention of unauthorized items from application in service.

9.2.3 The division and relationship of software development lifecycle are shown in the figure and table below.

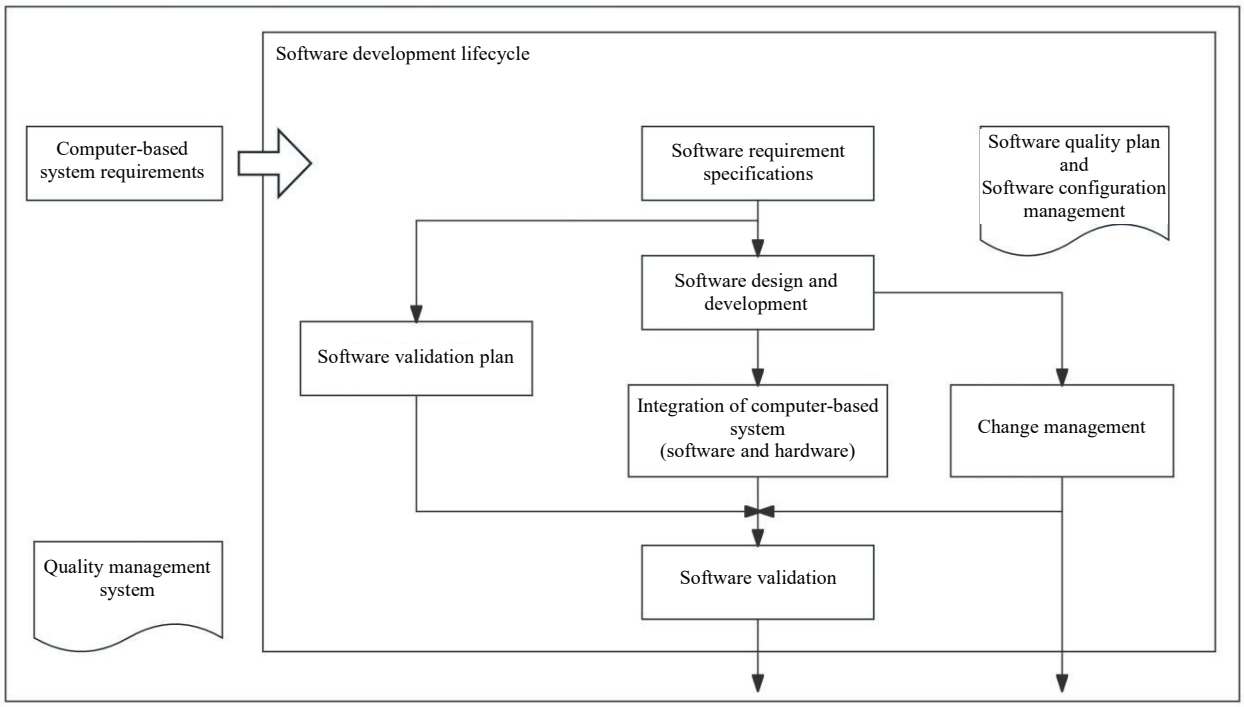


Figure 9.2.3-1 Software Development Lifecycle

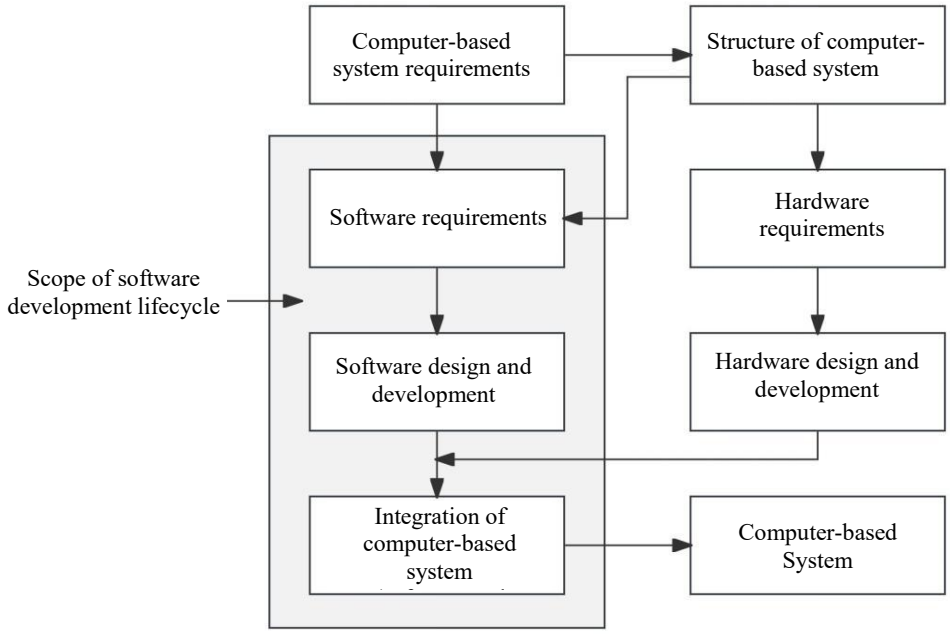


Figure 9.2.3-2 Scope and External Relationship of Software Development Lifecycle

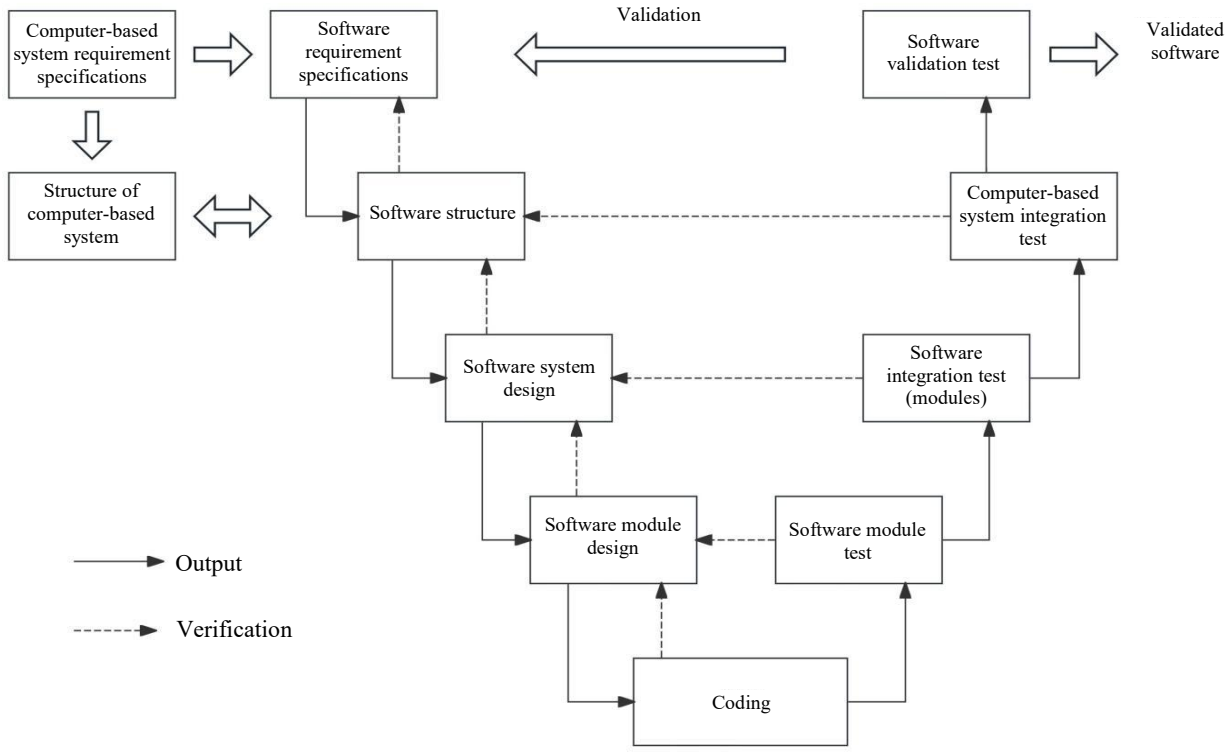


Figure 9.2.3-3 Model of Software Development Lifecycle (V Model)

Note: In addition to the V model, other software development lifecycle models agreed by CCS are accepted for this Guide.

9.3 Software requirement specifications

9.3.1 Purpose

(1) Specify the software requirement specifications according to the system functional requirements;

(2) Specify the software safety function requirements for each computer-based system requiring a certain software safety;

(3) Specify the requirements of the computer-based system for software testing.

9.3.2 Requirements

(1) The software developer shall review the information in 9.3.1 to ensure that the software requirements are fully specified, and shall give particular consideration to the following:

① Safety functions;

② System configuration or composition;

③ Hardware requirements;

④ Software requirements;

⑤ Capacity and response time;

⑥ Device and operator interface.

(2) Within the required system category level, the specified requirements of software safety shall be such expressed and organized that:

① they are clear, accurate, unambiguous, verifiable, measurable, maintainable and feasible;

② they can be traced back to the safety requirements of computer-based system;

③ ambiguous terms and descriptions that are not understood by those who use these documents at any stage of the software development lifecycle are not used.

(3) If the special safety requirements of the computer-based system are not defined in detail, the computer-based system and its operation mode shall be specified in the special requirements for software safety.

(4) The software requirement specifications shall standardize and document any safety-related constraints between software and hardware.

(5) Based on the hardware structure design requirements of computer-based system, the software requirement specifications shall consider the following:

① Software self-monitoring;

② Monitoring of programmable electronic hardware, sensors and actuators;

③ Periodic tests of safety functions during system operation;

④ Possibility of safety function tests during system operation.

(6) The software requirement specifications shall clearly distinguish between non-safety functions and safety functions of the computer-based system.

(7) The software requirement specifications shall express the safety attributes required by the computer-based system, but not the safety attributes of the project.

9.3.3 Input

System requirement specifications

9.3.4 Output

Software requirement specifications

9.4 Software validation plan

9.4.1 Purpose

A software validation plan shall be prepared according to the software requirement specifications.

9.4.2 Preparation requirements

(1) The following shall be considered for the software validation plan:

① Details at the time of validation;

② Details of the personnel performing the validation;

③ Determination of the relevant operating mode of the system, which shall include:

- Preparation before use, including setting and adjustment;

- Start-up, teaching, and automatic, manual, semi-automatic and stable operation;

- Reset, shutdown and maintenance;

- Reasonably foreseeable abnormalities and misoperations.

④ Determination of software requirements for each system operation mode before commissioning;

⑤ Technical strategy for validation activities (e.g., analytical methods, statistical testing, etc.);

⑥ Measures and regulations to ensure that each software function meets the specified requirements;

⑦ Special requirements determined according to the software requirement specifications;

⑧ Environment required for validation activities (e.g., calibration tools and equipment required for testing);

⑨ Pass/fail criteria;

⑩ Policies and regulations for evaluating validation results, especially in the event of failure.

(2) The technical strategy for software validation shall include the following:

① Manual and automatic technologies (one or both);

② Dynamic and static technologies (one or both);

③ Analytical or statistical technologies (one or both).

(3) The pass/fail criteria for software validation shall include:

① Required input signals, their sequence and values;

② Expected output signals, their sequences and values;

③ Other acceptable criteria, e.g., memory usage, timing, permissible variation of values.

9.4.3 Input

Software requirement specifications

9.4.4 Output

Software validation plan.

9.5 Software design and development

This section describes software design and development activities in the software development lifecycle.

9.5.1 Requirements for software structure and tool set

(1) Purpose

① Software structure:

Create a software structure to meet the specified requirements of different system category for software safety.

The computer-based system's hardware requirements for software shall be reviewed and evaluated, including the impact of software and hardware interaction on the safety of the computer-based system.

② Tool set:

Throughout the lifecycle of the software, the appropriate tool set (including programming languages, compilers, etc.) shall be selected according to the required system category to assist in verifying, validating, evaluating, and modifying the software.

(2) Requirements for software structure

The software structure defines the main components and subsystems of the software, including how they implement internal connections and how they obtain the required attributes, especially the safety integrity. The main software components include operating system, database, large device input/ output subsystem, communication subsystem, application program, programming and diagnostic tools, etc.

The software structure shall be designed by the software supplier and/or the developer. The description of the software structure design shall be detailed, including:

① During software development lifecycle, a set of necessary technologies and measures shall be selected and demonstrated to meet the safety requirements of the system. These technologies and measures include failure tolerance (consistent with hardware) and software design strategies for fault avoidance, including (where applicable) redundancy and diversity.

② According to the division of components / subsystems, each part shall provide the information about the following:

- whether they are new, currently available, or patented;

- whether they have been verified. If verified, please provide the criteria under which the verification is performed;

- whether each component / subsystem is related to safety.

③ All software/ hardware interactions shall be identified, and evaluated to detail their importance.

④ The software structure shall be represented using symbolic notation.

⑤ Design features shall be such selected as to maintain the safety integrity of all data, which may include large device input/ output data, communication data, operation interface data, maintenance data and internal database data.

⑥ According to the software structure, appropriate integration test shall be specified to ensure that the software structure meets the specified software safety requirements.

(3) Requirements for tool set

① For application programming using limited variability languages, the required tools and programming languages can be limited to a standard set of programming languages, editors, and loaders at a low safety integrity level. The responsibility for conformity shall be borne by the supplier.

② In higher-level systems, it is necessary to limit the subset of programming language, and verify and validate tools such as code analyzers and emulators. Responsibility in this context rests with the supplier and the user.

③ Even in low-level systems, full variability languages can be widely applied to develop embedded applications. Responsibility for compliance rests primarily with the software developer.

④ According to the inherent characteristics of software development, the responsibility for compliance with requirements in (a)-(e) below shall be borne by the supplier or the user alone or jointly, and the division of responsibility shall be incorporated in the safety technical document.

(a) A suitable set of tools, including programming language, compiler, configuration management tool, automatic test tool, etc., shall be selected according to the safety requirements. Consideration shall be given to the availability of appropriate development tools (not necessarily those used in the early stages of system development) that provide appropriate services throughout the lifecycle of computer-based system.

(b) Within the safety integrity level requirements, the selected tools or design expressions (including programming language) shall have the following features:

- Translators/compilers that comply with national or international standards are available, or their fitness for the purpose are assessed;
- Only defined language features are used;
- The application characteristics are matched;
- Features that facilitate the detection of program errors are included;

- The characteristics match the design approach.

(c) When (b) cannot be fully met, the software structure design specifications shall give a detailed reason for an alternative programming language to demonstrate its suitability, and give additional measures to address the identified shortcomings of the programming language.

(d) The coding standards shall be:

- reviewed by the assessors for their fitness with the purpose;
- used to develop all software related to functional safety.

(e) The coding standards shall specify good programming practices, prohibit unsafe language features (such as undefined language features, unstructured design, etc.), and establish regulations for creating source code documentation. The following information shall be included in the source code documentation:

- Legal entity (e.g., Company, author, etc.);
- Description;
- Input and output;
- Configuration management history.

(4) Input

① Software requirement specifications

② Hardware structure design of computer-based system

(5) Output

① Software structure design specifications

② Tools and coding standards supported

③ Selection of development tools

④ Software integration test specifications

⑤ Integration test specifications of computer-based system

9.5.2 Detailed design

(1) Purpose

Design software to meet the requirements of different safety level. Ensure that the software is analytical, verifiable and capable of safe modification.

The detailed design includes software system design and software module design.

(2) Requirements

① Detailed design first refers to the software system design, including dividing the main components of the software structure into software modules, separate software module design, code implementation method, etc.

② The detailed design of the software needs to provide a logical design for each software component and generate a detailed design document to define the internal structure and the interfaces of the components, including the relevant test content.

③ The software shall be designed to be modular, testable and capable of safe modification.

④ For each major component / subsystem in the software structure design, further detailing of the design shall be based on the division of the software module. The design and verification of each software module shall be specified.

⑤ Design specifications of software system and software module shall be provided.

⑥ The design specifications shall include the following:

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

(b) Description of communication software installed in network nodes;

(c) Description of the application software (not the program list);

(d) Tools for system setup and equipment configuration;

(e) Description of the mutual constraints and dependencies between functions, performance, modules and other components.

The description of the software shall meet the following requirements:

(a) Dependent system modules (which must work to maintain functionality), and dependencies on other systems;

(b) Description of each module that is detailed enough to understand its function;

(c) The relationship among software modules which must work to maintain the relevant functionality;

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

(e) Configuration of software, including priority policies;

(f) Switching mechanism for redundant systems (if any);

(g) Software self-monitoring (e.g., application-driven watchdog, data range verification, etc.);

(h) Verification tests and external device diagnostic tests (e.g., sensors and terminal elements);

(i) The measures to be taken for unexpected process variables (e.g., out-of-range sensor value, open circuit, short circuit).

(3) Input

① Software structure design specifications

② Tools and coding standards supported

(4) Output

① Design specifications of software system

② Specifications for software module design

③ Specifications for software system integration test

④ Software module test specifications

9.5.3 Code implementation

(1) Purpose

To implement the software with appropriate tool set, including programming language, compiler, etc.

(2) Requirements

The source code shall:

① Be readable, understandable and testable;

② Meet the specified requirements of software module design;

③ Meet the requirements of coding standards;

④ Meet the relevant requirements specified in the Safety Plan.

Each software code module shall be reviewed to check whether the code writing and its records comply with the description of the detailed design documents.

(3) Input

① Design specifications of software system

② Specifications for software module design

③ Tools and coding standards supported

(4) Output

① List of source code

② Code review report

9.5.4 Software module test

(1) Purpose

Software module test is a verification activity, which is a combination of code review and test work, to prove that the software module meets its relevant requirements.

(2) Requirements

① Each software module shall be tested according to the test specifications determined in the software design phase. These tests shall prove that each software module performs the intended functions and does not perform the non-intended functions.

② The software module tests shall be documented. The software module test documents of Category II and Category III systems shall include but not be limited to software module design specifications, software module test plan, software module test cases, software module test records, test record analysis report, software module test problem report and test summary report.

③ The regulations for correction measures of test failures shall be established.

④ Appropriate test methods shall be used to comprehensively test the logic and requirements of the software module.

⑤ White-box testing can be used to perform module testing, and test cases can be designed according to boundary value analysis, error speculation, equivalence class or input partitioning. The above methods can be selected according to the safety level requirements of the software and the characteristic requirements of the marine programmable device.

(3) Input

① Software module test specifications

② List of source code

③ Code review report

(4) Output

① Software module test records

② Verified and tested software modules

9.5.5 Software integration test

(1) Purpose

Software integration test is an activity to verify whether the software can be correctly integrated. The test shall prove that all software modules, components and subsystems can correctly interact with each other to achieve their intended functions, and do not achieve non-intended functions.

(2) General requirements

① The software integration test shall be specified during the design and development phase.

② The software integration test generally includes: software subsystem test and software system test.

③ The software integration test shall specify the following:

(a) Division of the software into integration sets that are manageable;

(b) Test cases and test data;

(c) Types of tests performed;

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

(e) Criteria for judging the completion of the test;

(f) Regulations for correction measures of test failures.

④ The software integration tests shall be carried out according to the specified software integration test requirements. These tests shall prove that all software modules, components and subsystems can interact with each other correctly to perform their intended functions without performing non-intended functions.

⑤ The software integration tests shall be documented and whether the test results meet the test objectives and test criteria shall be indicated. If a failure occurs, the cause of the failure shall be recorded.

⑥ During software integration, impact analysis shall be performed for any modification or change of the software to determine all affected software modules and required revalidation and redesign activities.

(3) Additional requirements for software subsystem test:

① It is recommended to use black-box test methods to perform subsystem tests. Test cases can be designed using methods such as dynamic testing, equivalence class partitioning and boundary value analysis. The above methods can be selected according to the safety level requirements of the software and the characteristic requirements of the marine programmable device.

② For Category II and Category III systems, subsystem tests shall be performed and the test results shall be analyzed to verify that the software modules are properly integrated. The confirmation basis of the test results can be traced back to the test criteria established by the test traceability in the test plan document.

(4) Additional requirements for software system test:

① For Category II and Category III systems, it shall be ensured that they meet the subsystem test requirements.

② The software of the marine programmable device shall be tested according to the provisions of the system test plan.

③ The software system test shall verify the anti-modification protection function:

(a) Preventing users from modifying the software;

(b) Preventing users from modifying the operating parameters of the software.

④ For the software system tests of Category II and Category III systems, it shall be verified that the software system meets the fail-to-safe principle under the condition of single failure.

⑤ Black-box test methods shall be used to perform software system tests, and such methods as equivalence class and process simulation shall be used to design test cases. The above methods can be selected according to the safety level requirements and the requirements of marine programmable device.

⑥ When there is a difference between the expected result and the actual result, an analysis shall be made to determine to continue the test or propose a change request. If a change request is made, it shall be returned to an earlier stage of the software development lifecycle. These decisions shall be used as the confirmation results of the software system tests and shall be documented.

(5) Input

Software integration test specifications (software subsystem/system test)

(6) Output

① Software integration test records

② Verified and tested software system

9.6 Integration of computer-based system (software and hardware)

9.6.1 Purpose

(1) The software is integrated on the hardware of the target computer-based system.

(2) The integration test of the computer-based system can ensure the compatibility of software and hardware and meet the intended requirements.

9.6.2 General requirements

(1) The integration test shall be specified in the design and development phase to ensure the compatibility of software and hardware in the computer-based system.

(2) The computer-based system (software and hardware) integration test, which can be abbreviated as computer-based system integration test, is the main system test method in this Guide and shall specify:

① Division of the system into integration sets;

② Test cases and test data;

③ Types of tests performed;

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

⑤ Criteria for judging test completion.

(3) When an integration test specified for the computer-based system (software and hardware) is conducted, a distinction should be made between activities performed by developers on their own intent and those performed from the user's standpoint.

(4) For integration tests of computer-based systems (both software and hardware), the following activities should be distinguished.

① Incorporation of software into the computer-based system of the target hardware;

② Integration of computer-based system, i.e., connecting sensors, actuators and other equipments by adding interfaces;

③ Complete integration of the computer-based system with the EUC (other systems).

(5) The integration tests of software and hardware shall be carried out in accordance with the specifications of computer-based system (software and hardware) integration test.

(6) Impact analysis shall be performed for any modification or change to the computer-based system (software and hardware) to identify all affected software components / modules and the required revalidation and redesign activities.

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

(8) The integration tests of the computer-based system (software and hardware) shall be documented to indicate whether the test results meet the test objectives and test criteria. If a failure occurs, the cause of the failure shall be recorded.

(9) For Category II and Category III systems, the supporting documents of the integration tests shall be retained and submitted as required, including test plan and test report.

9.6.3 Requirements for fault simulation test

(1) Fault simulation test, also known as fault response test. According to the design specifications of the computer-based system, formulate the fault simulation test specifications of the system. The fault simulation shall be carried out as realistically as possible to demonstrate that the system has an appropriate fault response capability.

(2) The specifications of fault simulation test include the following:

① Name of faulty component or element;

② Fault type;

③ Fault injection method;

④ Fault response required (output record).

(3) The test cases and expected results of the fault simulation test shall be documented. The test results of the fault simulation test and whether the test objectives and test criteria are met shall be stated. If the test fails, it should be analyzed and documented.

9.6.4 Input

(1) Computer-based system integration test specifications (including fault simulation test)

(2) Computer-based system with integrated software and hardware

9.6.5 Output

(1) Computer-based system integration test records

(2) Verified and tested computer-based system

9.7 Software validation

9.7.1 Purpose

The purpose is to ensure that the integrated computer-based system (software and hardware) meets the software requirement specifications.

9.7.2 Requirements

(1) Software validation cannot normally be separated from its associated hardware and system environment.

(2) During software validation, the following attributes shall be considered:

① The completeness of the validation for the software requirement specifications;

② The correctness of the validation for the software requirement specifications;

③ Repeatability;

④ Precisely defined validation configuration.

(3) Software validation activities shall be carried out according to the software validation plan.

(4) The following software validation results shall be documented:

① Validation records in chronological order to trace the sequence of activities;

② The version of the software validation plan used;

③ Software requirements to be validated (by test or analysis) according to the software validation plan;

④ Tools, equipments, and calibration data used;

⑤ Results of validation activities;

⑥ Differences between actual and expected results.

(5) When there is a difference between the actual result and the expected result, necessary analysis shall be carried out to decide whether to continue the validation, or to make a change request and return to an earlier stage of the software development lifecycle.

(6) The software validation shall meet the following requirements:

① Software validation test shall be the primary method of software validation. Analysis, animation and modeling can be used as a supplement to validation activities;

② If necessary, simulation or emulation test methods can be used for software validation;

③ The system developer shall be provided with the results of the software validation and its associated documentation.

(7) The software validation results shall meet the following requirements:

① The software validation test shall demonstrate that all software specified requirements are met and that the software does not perform non-intended functions;

② Test cases and test results shall be documented for subsequent analysis and independent evaluation;

③ Documented software validation test results shall indicate that (a) the software has passed the validation, or (b) the reason for failure to pass the validation.

9.7.3 Input

Software validation plan

9.7.4 Output

(1) Software validation results;

(2) Validated software.

9.8 Change management

9.8.1 Purpose

The purpose is to guide the correction, enhancement and adjustment of validated or approved software in accordance with the change management procedure to ensure the safety and controllability of the computer-based system after software change.

9.8.2 Requirements

(1) The software change management procedure shall be refined and improved based on the change management requirements for computer-based system (refer to section 8.4).

(2) The stakeholders shall be informed in advance of the software change plan for the approved system, and the impact analysis shall be carried out, and the CCS shall be reported. Major software changes for Category II and III systems shall be submitted to CCS for approval.

(3) Software changes shall be verified and recorded. For Category II and III systems, CCS shall witness the verification process of major software changes.

(4) Verification activities for software changes include regression test. When the software code change rate exceeds 30%, a complete and comprehensive test shall be carried out for the changed software and system.

9.8.3 Input

(1) Software change management procedure

(2) Software change request

9.8.4 Output

(1) Impact analysis results of software change

(2) Software change records

(3) Corresponding test reports

9.9 Software verification

9.9.1 Purpose

The purpose is to test and evaluate the output of the software development lifecycle at a given stage to ensure the correctness and consistency of the output of the stage with respect to the corresponding input.

9.9.2 Requirements

(1) For each stage of the software development lifecycle, the software verification shall be planned in synchronization with the development process, and the software verification shall be documented.

(2) The preparation of the software verification plan shall involve the criteria, techniques and tools used in the verification activities and shall include the following:

① Evaluation of safety integrity requirements;

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

③ Selection and use of verification tools (test tools, special test software, input/output simulator, etc.);

④ Evaluation of verification results;

⑤ Corrective actions taken.

(3) The software verification shall be performed according to the plan.

(4) The evidence of software verification shall be documented to prove that the verification of the relevant stage has been successfully completed in all respects.

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

① Identification of the item to be verified;

② Identification of the information on which verification is based;

③ Non-conformities (e.g., software modules, data structures, and algorithms that do not apply).

(6) All information required for the correct execution of the N+1 phase of the software development lifecycle shall be available and verified. The output of the N phase shall include:

① Phase N documentation, design or code shall adequately satisfy:

- Functionality;

- Requirements for the preparation of safety integrity, performance and other safety plans;

-Readability to the development team;

- Testability for further verification;

- Safety modifications that allow for further improvement.

② The validation plan and/or tests specified in Phase N should be sufficient for the design requirements and design statements of Phase N.

③ Inconsistencies between the following items shall be identified:

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

- Each output in Phase N.

(7) The following verification activities shall be performed at each stage of the software development lifecycle:

① Verification of software requirements;

② Verification of software structure;

③ Verification of software system design;

④ Verification of software module design;

⑤ Code verification;

⑥ Data verification;

⑦ Verification of time performance;

⑧ Software module test;

⑨ Software integration test;

⑩ Computer-based system integration test;

⑪ Software validation;

⑫ Additional requirements for data links for Category II and III systems.

(8) Verification of software requirements: after specifying the software requirements and before the subsequent software design and development phases, the verification shall:

① Consider whether the specified software requirements adequately meet the functionality, safety integrity, performance, and other aspects specified by computer-based system.

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

③ Inconsistencies between the following items shall be identified:

- Software requirements and requirements for computer-based system;

- Software requirements and software validation plan.

(9) Verification of software structure: after completing the software structure design, the verification shall:

① Consider whether the software structure design meets the software requirement specifications adequately;

② Consider whether the integration test specified by the software structure design is sufficient;

③ Consider whether the properties of each major component / subsystem adequately meet:

- Feasibility of required safety performance;

- Testability for further verification;

- Readability to the development team;

- Safety modifications that allow for further improvement.

④ Check for inconsistencies between:

-Software structure design and software requirement specifications;

-Software structure design and software structure integration test;

- Software structure integration test and software validation plan.

(10) Verification of software system design: After the software system design is completed, the verification shall:

① Consider whether the software system design meets the software structure design adequately;

② Consider whether the tests specified for software system integration fully meet the software system design;

③ Consider whether the properties of each major component of the software system design adequately meet:

- Feasibility of required safety performance;

- Testability for further verification;

- Readability to the development team;

- Safety modifications that allow for further improvement.

④ Check for inconsistencies between:

- Software system design and software structure design;

- Software system design and software system integration test;

-Software system integration test and software structure integration test.

(11) Verification of software module design: After completing software module design, the verification should:

① Consider whether the software module design fully meets the software system design requirements;

② Consider whether the specified tests of each software module fully meet the software module design requirements;

③ Consider whether the properties of each software module adequately meet:

- Feasibility of required safety performance;

- Testability for further verification;

- Readability to the development team;

- Safety modifications that allow for further improvement.

④ Check for inconsistencies between:

- Software module design and software system design;

- (For each software module) software module design and software module test;

- Software module test and software system integration test.

(12) Code verification: the source code shall be verified by static methods to ensure compliance with the software module design, required coding standards and software validation plan.

Note: In the early stages of the software development lifecycle, verification is static (e.g., review, re-examination, formal proof, etc.). Code verification includes techniques such as review and code walkthrough. Code verification is combined with software module test to ensure that each software module meets the requirements of relevant documents. Thereafter, testing became the primary method of verification.

(13) Data verification

① Data structure verification includes:

- integrity;

- self-consistency;

- protection against change or damage;

- consistency with the functional requirements of the data-driven system.

② Application data verification includes:

- consistency with the data structure;

- integrity for application requirements;

- compatibility with relevant system software (e.g., sequence of execution, runtime, etc.);

- correctness of data values.

③ All operating parameters shall be verified against application requirements to prevent:

- invalid or undefined initial values;

- incorrect, discontinuous or unreasonable values;

- non-approval changes;

- data corruption.

④ All device interfaces and associated software (i.e., sensors, actuators and offline interfaces) shall be verified as follows:

-Detection of expected interface failure;

-Fault tolerance for expected interface failure.

⑤ For all communication interfaces and associated software, the following items shall be properly verified:

- Failure detection;

- Error prevention;

- Data validation.

(14) Verification of time performance: Verification of the predictability of behavior in the time domain.

Note: Time behavior may include performance, resources, response time, worst-case execution time, overload, deadlock-free, runtime system, etc.

(15) Other verification: No additional requirements are placed on software module test, software integration test, and computer-based system integration test, because these tests are verification activities in themselves. Similarly, no additional requirements are imposed on software validation, because software validation is a verification activity to prove compliance with software requirements.

9.9.3 Input

Appropriate verification plan (depending on the phase)

9.9.4 Output

Appropriate verification report (depending on the phase)

10 Test, verification and approval

10.1 The computer-based system shall be tested and verified according to the requirements in the table below. This section only puts forward specific requirements for software. The evaluation of the small low complexity computer system shall be tested and verified in accordance with the requirements of Appendix 2 of this Guide.

Table 10.1 Test and Verification

S/N	Requirements	System supplier	System integrator	Owner	Category I System	Category II System	Category III System	Documents to be provided
1	**Quality management**							① Quality plan ② Procedures and documents related to security policy
1.1	**ISO9001 or equivalent standard quality management system shall be implemented.**	x	x		Ⓘ (when necessary)	Ⓘ	Ⓘ	
1.2	**Quality plan**	x	x		Ⓘ (when necessary)	Ⓘ	Ⓘ	
1.3	**Software traceability**	x	x	x	Ⓘ (when necessary)	Ⓘ	Ⓘ	
1.4	**Security policy**	x	x	x	Ⓘ (when necessary)	Ⓘ	Ⓘ	
2	**Technical requirements for computer-based system**							① Specific procedure for unique identification of a computer-based system, it's components and versions.
2.1	**System identification requirements**	x	x			Ⓘ	Ⓘ	

2.2	**Data link Requirements for Category II and III Systems**	x	x			Ⓘ	Ⓘ	
2.3	**Supplementary requirements when using wireless data link**	x	x			Ⓘ	Ⓘ	
3	**System description (software description and associated hardware description)**	x			Ⓘ (when necessary)	Ⓐ	Ⓐ	① System description (which can be decomposed into relevant documents) ② Computer-based system requirement specifications ③ Computer-based system hardware description ④ Software requirement specifications ⑤ Software structure design specifications ⑥ Software structure integration test specifications ⑦ Software module design specifications ⑧ Software module test specifications ⑨ Software system design specifications
3.1	**Software description**							
3.2	**Hardware description**							
3.3	**Technical requirements**							

								⑩ Software system test specifications ⑪ Software integration test specifications ⑫ Computer-based system integration test specifications ⑬ Supporting tools and coding standards ⑭ Selection of development tools
4	**Software code verification**	x				① (when necessary)	① (when necessary)	① Code review report
5	**Software module test**	x				① (when necessary)	① (when necessary)	① Software module test specifications ② Software module test records
6	**Software integration test**	x			① (when necessary)	① (when necessary)	① (when necessary)	① Software integration test specifications ② Software integration test records ③ Software system test specifications ④ Software system test records ⑤ Software structure integration test specifications ⑥ Software structure integration test records ⑦ Software test report
6.1	**General requirements for software integration test**							
6.2	**Additional requirements for software subsystem test**							
6.3	**Additional requirements for software system test**							

7	**Computer-based system (software and hardware) integration test, which is part of the system test**	x			Ⓘ (when necessary)	Ⓘ (when necessary)	Ⓘ (when necessary)	① Computer-based system integration test specifications ② Computer-based system integration test records ③ Computer-based system fault simulation test specifications ④ Computer-based system fault simulation test records ⑤ System test report
7.1	**General requirements for computer-based system integration test**							
7.2	**Requirements for fault simulation test of computer-based systems**							
8	**Factory acceptance test (FAT), including software validation test**							① System description ② FAT program ③ FAT report ④ User manual ⑤ System test report, etc.
8.1	**Preparing FAT program (test outline)**	x				Ⓐ	Ⓐ	
8.2	**Performing FAT**	x				Ⓦ	Ⓦ	
8.3	**Preparing FAT report**	x				Ⓘ	Ⓘ	
8.4	**Other FAT documents**	x				Ⓘ (when necessary)	Ⓘ (when necessary)	
9	**Environmental compliance requirements for hardware**	x			Ⓘ (when necessary)	Ⓘ	Ⓘ	① Environmental compliance test report or type approval certificate.
10	**Preparations before onboard tests**							① System description ② List of system categorizations

10.1	**List of system categorizations**		x		Ⓐ	Ⓐ	Ⓐ	③ Risk assessment report ④ System architecture description ⑤ FAT report ⑥ User manual
10.2	**Risk assessment report**		x		Ⓐ (when necessary)	Ⓐ (when necessary)	Ⓐ (when necessary)	
10.3	**System architecture description**		x		Ⓘ	Ⓘ	Ⓘ	
11	**Onboard test**							① SAT program ② SAT report ③ SOST program ④ SOST report
11.1	**Preparing onboard test programs (test outlines)**		x			Ⓐ	Ⓐ	
11.2	**Performing onboard tests**		x			Ⓦ	Ⓦ	
11.3	**Preparing onboard test reports**		x			Ⓘ	Ⓘ	
12	**Verification of changes in computer-based system**							① Change management procedure ② Change request or change description ③ Change impact analysis results ④ Change records ⑤ Corresponding test reports
12.1	**General verification requirements**	x	x	x	Ⓘ (when necessary)	Ⓘ	Ⓘ	
12.2	**Additional verification requirements for major changes**	x	x			Ⓐ Ⓦ	Ⓐ Ⓦ	

Note 1: The symbols used in the table and their meanings are as follows:

Ⓐ Submit to CCS for approval; Ⓘ Submit to CCS for reference; Ⓦ CCS witness required

Note 2: If the "role" and "system category" of the specific item in the table are not specified (blank), then the options of the title item are adopted.

Note 3: The "documents to be provided" in the table can be merged or decomposed, or other names can be used, as long as the contents required by this Guide are available.

Note 4: The level of witness shall be determined after evaluation according to the above requirements. If the design or layout is inconsistent with the predetermined requirements, the engineering analysis carried out in accordance with relevant international (see Article 55 of Chapter II-1 of the SOLAS Convention) or domestic standards shall be submitted to CCS for approval.

10.2 The following activities shall be witnessed or surveyed by CCS, and the relevant responsible roles shall facilitate the activities.

S/N	Activity	Responsible role	System category		
			Category I system	Category II system	Category III system
1	FAT	System supplier	-	Ⓦ	Ⓦ
2	SAT	System integrator	-	Ⓦ	Ⓦ
3	SOST	System integrator	-	Ⓦ	Ⓦ
4	Major change verification	System integrator	-	Ⓦ	Ⓦ

Note: The symbols used in the table and their meanings are as follows: Ⓐ Submit to CCS for approval; Ⓘ Submit to CCS for reference; Ⓦ CCS witness required.

10.3 The system supplier or system integrator may apply to CCS for inspection or evaluation of cyber security in accordance with the requirements of CCS Guidelines for Ship Cyber Security. If professional reliability verification is required, an application for the certificate of conformity for reliability verification can be submitted to CCS in accordance with the requirements of CCS Guidelines for Reliability Verification of Ship Equipment and Systems.

10.4 The system supplier or system integrator may obtain the approval for the programmable device integrated in a Category II or Category III system in accordance with CCS Guidelines for Type Approval Test of Electric and Electronic Products. After verification of the required tests by CCS, the approval of programmable device can be completed by single-piece inspection or as part of the type approval. The approval documents shall describe the compatibility of the programmable device in ship applications and the necessity of onboard test.

10.5 Type approval of computer-based system

The computer-based system that is routinely manufactured and includes standardized software functions may be type approved according to CCS requirements. Type approval consists of two main verification activities:

(1) Assessment of the type approval documents, including the documents listed in 7.1 and other documents required to apply for CCS type approval;

(2) The standardized functions shall be inspected and tested, and the software testing can be carried out by a professional organization (see Annex 4).

If a computer-based system, of which the system lifecycle has been established, has passed the system and hardware certification in accordance with the requirements of CCS Rules for Classification of Sea-going Steel Ships, and meets the software requirements of this Guide, the following class notations can be granted according to their different system categories (see 4.1 for categorization):

(1) SLC1 for Category I system;

(2) SLC2 for Category II system;

(3) SLC3 for Category III system.

Since the functions, parameter configurations and installation elements of a real ship demand vessel-specific verification, even if the computer-based system has completed the type approval, the product certificate shall still be required.

10.6 Product certificate of computer-based system

Category II or Category III computer-based system necessary to realize ship functions shall come with product certificate. The purpose of the product certificate is to confirm that the design and manufacture of the system have been completed and meet the applicable classification requirements. To obtain a product certificate, two main verification activities are required:

(1) The documents of the computer-based system, including the documents listed in 7.1, shall be evaluated;

(2) The on-board computer-based system shall be inspected and tested, of which the software testing may be carried out by a professional organization (see Appendix 4).

CCS can accept Alternative Certification Scheme (ACS) and issue the product certificate provided the requirements are met.

Appendix 1 Table of Test and Verification

Name of the applicant:_____ Work control No.:_____

Product name:_____ Product model:_____

Software name:_____ Software version number:_____

S/N	Requirements	Reference item	System supplier	System integrator	Owner	Category I System	Category II System	Category III System	Documents to be provided	Whether it meets the requirement ?
1	**Quality management**								① Quality plan ② Procedures and documents related to security policy	
1.1	**The ISO9001 or equivalent quality management system shall be implemented.**	5.1.3								
a	The responsibilities and competencies of employees shall be defined.		x	x			①	①		
b	The complete lifecycle of delivered software and of associated hardware shall be implemented.		x	x			①	①		
c	Specific procedures for unique identification of a computer-based system, it's components, and versions shall be established.		x			① (when necessary)	①	①		
d	The vessel's system architecture			x			①	①		

S/N	Requirements	Reference item	System supplier	System integrator	Owner	Category I System	Category II System	Category III System	Documents to be provided	Whether it meets the requirement ?
	shall be created and updated.									
e	An organization shall be set up to purchase software and related hardware from suppliers.		x	x			ⓘ	ⓘ		
f	An organization for writing and verifying the software code shall be set up.		x				ⓘ	ⓘ		
g	An organization for system validation shall be set up before integration in the vessel.		x				ⓘ	ⓘ		
h	Specific procedures for conducting and approving the system at the time of FAT and SAT shall be formulated.		x	x			ⓘ	ⓘ		
i	System documentation shall be created and updated.		x			ⓘ (when necessary)	ⓘ	ⓘ		
j	Specific procedures for modification and installation of onboard software, including interactions with the shipyard and the owner, shall be developed.		x	x		ⓘ (when necessary)	ⓘ	ⓘ		
k	Specific verification procedures for software code shall be developed.		x				ⓘ	ⓘ		

S/N	Requirements	Reference item	System supplier	System integrator	Owner	Category I System	Category II System	Category III System	Documents to be provided	Whether it meets the requirement ?
l	Procedures for integrating a system with other systems and testing of the system of systems for the vessel shall be developed.		x	x			Ⓘ	Ⓘ		
m	Procedures for managing changes to software and configurations before FAT shall be developed.		x			Ⓘ (when necessary)	Ⓘ	Ⓘ		
n	Procedures for managing and documenting changes to software and configurations after FAT shall be developed.		x	x		Ⓘ (when necessary)	Ⓘ	Ⓘ		
o	The tracking checkpoints for the organization's compliance with the quality management system shall be determined. A checkpoint may be a required deliverable, a test, a technical review meeting, or an expert review meeting.		x	x			Ⓘ	Ⓘ		
1.2	**Quality plan**	5.2	x	x		Ⓘ (when necessary)	Ⓘ	Ⓘ		
a	There are clear standards and guidance documents to define the computer-based system.									

S/N	Requirements	Reference item	System supplier	System integrator	Owner	Category I System	Category II System	Category III System	Documents to be provided	Whether it meets the requirement ?
b	All stakeholders involved (e.g., developers, project leaders, etc.) have reviewed the computer-based system.									
c	Acceptance criteria for the computer-based system have been established.									
d	The computer-based system has clearly defined objectives and scope of application.									
e	The software content that is covered by the Software Quality Assurance Plan is identified.									
f	The intended use of the software is specified.									
g	The part of the software development lifecycle that has been covered by the quality plan is described.									
h	Available references are included.									
i	An outline of the project management structure is included.									
j	The files for the development, verification, validation, use and									

S/N	Requirements	Reference item	System supplier	System integrator	Owner	Category I System	Category II System	Category III System	Documents to be provided	Whether it meets the requirement ?
	maintenance of the system and software are detailed.									
k	The files are listed and described.									
l	The files to be evaluated by the quality plan are listed.									
m	The standards, practices and quality requirements used are identified (e.g., IEC, ISO, IEEE, etc.).									
n	The measures to monitor and ensure compliance of computer-based system and processes (such as traceability, reporting, and trends) are described.									
o	The role of software management plans in software verification and validation is defined and described.									
p	The methods and procedures for reporting, tracking, and resolving issues are described.									
q	The tools and technologies that are used to support software quality assurance activities (e.g., checklists, plans and report									

S/N	Requirements	Reference item	System supplier	System integrator	Owner	Category I System	Category II System	Category III System	Documents to be provided	Whether it meets the requirement ?
	templates, databases for traceability) are described.									
r	The idea to ensure that the supplier's controls meet customer requirements by internal and external supervision (e.g., inspections, assessments/audits, monthly status reports) is discussed.									
s	The design and development of software can ensure that it meets special design and development requirements. In another word, it can prevent and respond to potential failure conditions.									
t	The process for informing the owner of software modification and installation on board has been clearly defined.									
1.3	**Software traceability**	5.5	x	x	x	Ⓘ (when necessary)	Ⓘ	Ⓘ		

S/N	Requirements	Reference item	System supplier	System integrator	Owner	Category I System	Category II System	Category III System	Documents to be provided	Whether it meets the requirement ?
a	According to the quality management procedure, the modification of programming content and data and the change of version shall be identified and documented.									
b	The configuration management of the software and hardware of computer-based system is implemented.									
c	The process that must be followed for the modification of programming content and data and the change of version is clarified, and it is confirmed that these modifications or changes have been recorded in documents.									
d	The process for informing the owner of software modification and installation on board has been clearly defined.									
e	If the Owner designates the system integrator as the party responsible for the software change, the Owner shall inform CCS.									

S/N	Requirements	Reference item	System supplier	System integrator	Owner	Category I System	Category II System	Category III System	Documents to be provided	Whether it meets the requirement ?
f	The software change impact analysis results and test reports shall be submitted to CCS for reference.									
g	The system integrator can record the changes by updating the software registry or equivalent file.									
1.4	**Security policy**	5.6	x	x	x	Ⓘ (when necessary)	Ⓘ	Ⓘ		
a	The Owner, system integrator and system supplier shall adopt security policy in the quality system and procedures.									
b	The software shall not be modified without authorization. Regardless of a physical system or remotely controlled system, physical and logical security measures shall be taken to prevent unauthorized or inadvertent modification.									

S/N	Requirements	Reference item	System supplier	System integrator	Owner	Category I System	Category II System	Category III System	Documents to be provided	Whether it meets the requirement ?
c	All firmware, software code, executable program, and physical media intended for installation on board shall be scanned for vulnerabilities, viruses, malware, etc. prior to installation. Scan results are saved in software registry or equivalent files.									
2	**Technical requirements for computer-based system**								① Specific procedure for unique identification of a computer-based system, it' s components and versions	
2.1	**System identification requirements**	6.1	x	x			ⓘ	ⓘ		
a	Methods and applications for identifying the name, version, identifier and manufacturer of the system, including software and hardware, shall be provided.									
2.2	**Data link Requirements for Category II and III Systems**	6.2.1	x	x			ⓘ	ⓘ		
a	A single failure can be handled automatically to restore normal operation of the system.									

S/N	Requirements	Reference item	System supplier	System integrator	Owner	Category I System	Category II System	Category III System	Documents to be provided	Whether it meets the requirement ?
b	The loss of any function of the remote control system shall be compensated by local/manual means.									
c	The communication rate of the data link can be prevented from being overloaded under any operating condition.									
d	The data link has a self-test function to detect its own link failure and the communication failure of the nodes connected to the link.									
e	An alarm shall be given when a failure occurs.									
2.3	**Supplementary requirements when using wireless data link**	6.2.2	x	x			①	①		
a	Wireless data link shall not be used for Category III systems unless specifically considered by CCS.									
b	Recognised international wireless communication system protocols shall be used.									

S/N	Requirements	Reference item	System supplier	System integrator	Owner	Category I System	Category II System	Category III System	Documents to be provided	Whether it meets the requirement ?
c	Message integrity: The received message will not be destroyed or changed compared with the sent message through failure prevention, detection, diagnosis and correction.									
d	Configuration and device authentication: Connections to devices included in the system design shall only be permitted.									
e	Message encryption: Confidential and/or critical data shall be protected.									
f	Security management: For protecting network assets and preventing illegal access to network assets.									
g	The internal wireless system within the vessel shall meet the requirements of the International Telecommunication Union and the Flag State's authorities for radio frequency and power levels.									

S/N	Requirements	Reference item	System supplier	System integrator	Owner	Category I System	Category II System	Category III System	Documents to be provided	Whether it meets the requirement ?
h	The system operation shall be implemented with the provisions of port and local regulations on radio frequency transmission considered, and the use of wireless data communication links is prohibited due to frequency and power restrictions.									
i	The wireless data communication equipment shall be tested during harbour and sea trials to demonstrate that, under the expected operating conditions, the radio frequency transmission will not cause failure of itself and any other equipment due to electromagnetic interference.									
3	**System description (software description and associated hardware description)**	8.2.1.3	x			Ⓘ (when necessary)	Ⓐ	Ⓐ	① System description (which can be decomposed into related documents) ② Requirement specifications	
3.1	**Software description**									
a	The software requirement specifications is specified according to the system functional requirements.									

S/N	Requirements	Reference item	System supplier	System integrator	Owner	Category I System	Category II System	Category III System	Documents to be provided	Whether it meets the requirement ?
b	The requirements for software safety functions for each computer -based system that needs to implement certain safety functions shall be specified.								for computer-based system ③ Description of computer-based system hardware ④ Software requirement specifications ⑤ Specifications for software structure design ⑥ Specifications for software structure integration test ⑦ Specifications for software module design ⑧ Software	
c	The system category of the computer-based system is defined.									
d	The software integration requirements for each computer-based system, including communication and interface aspects, are specified.									
e	The description of software structure design includes: In the required software development lifecycle, according to different levels of systems, the integration technologies that meet the software requirement specifications shall be selected and determined.									
f	The technologies and measures for documentation of software requirements include: failure tolerance (consistent with hardware), software design strategies for fault avoidance, as									

S/N	Requirements	Reference item	System supplier	System integrator	Owner	Category I System	Category II System	Category III System	Documents to be provided	Whether it meets the requirement ?
	well as (where applicable) redundancy and diversity.								module test specifications ⑨ Design specifications of Software System ⑩ Software system test specifications ⑪ Software integration test specifications ⑫ Computer-based system integration test specifications ⑬ Supporting tools and coding standards ⑭ Selection of development tools	
g	The description of the software structure design includes the identification and evaluation of all software/ hardware interactions and the refinement of their importance.									
h	The description of the software structure design includes specifying appropriate software structure integration test to ensure that the software structure meets the requirements of software safety at the specified system level.									
i	Standards and naming conventions have been clarified.									
j	The documents of software system design and module design specifications are provided.									
k	The software system design and module design documents describe the mutual constraints and dependencies between functions, performance, modules and other components.									

S/N	Requirements	Reference item	System supplier	System integrator	Owner	Category I System	Category II System	Category III System	Documents to be provided	Whether it meets the requirement ?
l	The software system design and module design documents describe software self-monitoring (e.g., application-driven watchdog and data range verification).									
m	The software system design and module design documents shall require verification tests and external device diagnostic tests (e.g., sensors and terminal elements).									
n	The software system design and module design documents include measures against bad process variables, such as sensor values out of range, open circuit and short circuit.									
3.2	**Hardware description**									
a	Hardware description shall include network architecture / topology, and all network components, such as switches, routers, gateways, firewalls, etc.									

S/N	Requirements	Reference item	System supplier	System integrator	Owner	Category I System	Category II System	Category III System	Documents to be provided	Whether it meets the requirement ?
b	Hardware description shall include the internal structure of all interfaces and hardware nodes of the system (such as operation stations, displays, computers, programmable devices, sensors, actuators, I/O modules, etc.).									
c	Hardware description shall include I/O allocation (mapping field devices to channels, communication links, hardware units, logical functions).									
d	Hardware description shall include a detailed list of technical specifications for hardware and external related equipments.									
e	Hardware description shall include power supply arrangement.									
f	Hardware description shall include a description of the failure mode.									
3.3	**Technical requirements**									
a	For Category II and III systems, the implementation of the system technical requirements is verified as part of the system description.									

S/N	Requirements	Reference item	System supplier	System integrator	Owner	Category I System	Category II System	Category III System	Documents to be provided	Whether it meets the requirement ?
4	**Software code verification**	9.5.3	x				① (when necessary)	① (when necessary)	① Code review report	
a	Check whether the software code writing and its results comply with the description of the detailed design documents.									
b	Check whether the software code meets the requirements of the software module design, the required coding standards and the software validation plan.									
5	**Software module test**	9.5.4	x				① (when necessary)	① (when necessary)	① Software module test specifications ② Software module test records	
a	Software module test activities have been documented.									
b	The software module test documents of Category II and Category III systems include software module test specifications, software module test plan, software module test cases, software module test									

S/N	Requirements	Reference item	System supplier	System integrator	Owner	Category I System	Category II System	Category III System	Documents to be provided	Whether it meets the requirement ?
	records, test record analysis report, software module test problem report and test summary report.									
c	After each software module design is specified, the following shall be verified: Whether the specified software module design fully meets the specified software system design.									
d	After each software module design is specified, the following shall be verified: Whether the specified test of each software module fully meets the specified software module design.									
e	After each software module design is specified, the following shall be verified: Whether the attributes of each software module fully meet requirements in terms of: ① Feasibility of the required safety performance; ② Testability for further verification; ③ Readability to the development team; ④ Safety modifications that allow for further improvement.									

S/N	Requirements	Reference item	System supplier	System integrator	Owner	Category I System	Category II System	Category III System	Documents to be provided	Whether it meets the requirement ?
f	After each software module design is specified, the inconsistency between the following contents shall be checked: ① Software module design and software system design; ② (For each software module) software module design and software module test; ③Software module test and software system integration test.									
6	**Software integration test**	9.5.5	x			Ⓘ (when necessary)	Ⓘ (when necessary)	Ⓘ (when necessary)	① Software integration test specifications ② Software integration test records ③ Software system test specifications ④ Software system test records ⑤ Specifications	
6.1	**General requirements for software integration test**									
a	After the software structure design is finished, the following shall be verified: Whether the software structure design fully meets the software requirement specifications.									
b	After the software structure design is finished, the following shall be verified: Whether the integration test specified by the software structure design is sufficient.									

S/N	Requirements	Reference item	System supplier	System integrator	Owner	Category I System	Category II System	Category III System	Documents to be provided	Whether it meets the requirement ?
c	After the software structure design is finished, the following shall be verified: Whether the attributes of each major component / subsystem fully meet requirements in terms of: ① Feasibility of the required safety performance; ② Testability for further verification; ③ Readability to the development team; ④ Safety modifications that allow for further improvement.								of software structure integration test ⑥ Software structure integration test records ⑦ Software test report	
d	After the software structure design is finished, the inconsistency of the following contents shall be checked: ① software structure design and software requirement specifications; ② Software structure design and software structure integration test; ③ Software structure integration test and software validation plan.									
e	The software integration test shall include the following contents: ① Division of the software into manageable integration sets; ② Test cases and test data; ③ Types									

S/N	Requirements	Reference item	System supplier	System integrator	Owner	Category I System	Category II System	Category III System	Documents to be provided	Whether it meets the requirement ?
	of tests performed; ④ Test environment, tools, configurations and programs; ⑤ Criteria for judging the completion of the test; ⑥ Correction measures of test failures.									
f	Whether the software integration test shows that all software modules, components and subsystems can interact with each other correctly to perform their intended functions and do not perform non-intended functions.									
g	The software integration test shall be documented and state whether the test results meet the test objectives and test criteria. If a failure occurs, the cause of the failure shall be recorded.									
h	During software integration, impact analysis shall be performed for any modification or change of the software to determine all affected software modules and required revalidation and redesign activities.									

S/N	Requirements	Reference item	System supplier	System integrator	Owner	Category I System	Category II System	Category III System	Documents to be provided	Whether it meets the requirement ?
6.2	**Additional requirements for software subsystem test**									
a	For a Category II or Category III system, perform software subsystem tests and analyze the test results to verify that the software modules are properly integrated.									
6.3	**Additional requirements for software system test**									
a	Verify the anti-modification protection function of the software system through the software system test.									
b	For a Category II or Category III system, perform software system test to verify that the software system meets the fail-to-safe principle under the condition of single failure.									
7	**Computer-based system (software and hardware) integration test, which belongs to system test**	9.6	x			① (when necessary)	① (when necessary)	① (when necessary)	① Specifications for computer-	

S/N	Requirements	Reference item	System supplier	System integrator	Owner	Category I System	Category II System	Category III System	Documents to be provided	Whether it meets the requirement ?
7.1	**General requirements for computer-based system integration test**								based system integration test ② Computer-based system integration test records ③ Specifications for computer-based system fault simulation test ④ Computer-based system fault simulation test records ⑤ System test report	
a	The following documents shall be prepared for the computer-based system integration test: ① System description, including software requirement specifications; ② List and version numbers of software installed in the system; ③ Software function description; ④ Software maintenance and operation manual; ⑤ List of interfaces between the system and other shipboard systems; ⑥ List of data transmission standards.									
b	The computer-based system integration test shall specify: ① Division of the system into various integration sets; ② Test case and test data; ③ Types of tests performed; ④ Test environment, including tools, support software and configuration description; ⑤ Criteria for judging test completion.									

S/N	Requirements	Reference item	System supplier	System integrator	Owner	Category I System	Category II System	Category III System	Documents to be provided	Whether it meets the requirement ?
c	The computer-based system integration test shall verify the following aspects of the system: ① Functionality; ② Failures and their effects; ③ Performance; ④ Integration between software and hardware; ⑤ Human-machine interfaces; ⑥ Interfaces with other systems.									
d	When the computer-based system integration test is conducted, a distinction should be made between activities performed by developers based on their own intentions and activities performed from the standpoint of users.									
e	In the computer-based system integration test, an impact analysis is performed on any modification or change to the computer-based system to identify all affected software components / modules, as well as the required revalidation and redesign activities.									
f	The computer-based system integration test should be									

S/N	Requirements	Reference item	System supplier	System integrator	Owner	Category I System	Category II System	Category III System	Documents to be provided	Whether it meets the requirement ?
	documented to determine whether the test results meet the test objectives and test criteria. If a failure occurs, the cause of the failure should be recorded.									
g	For Category II and Category III systems, documentary evidence of the computer-based system integration test shall be retained and submitted to CCS as required, including test plan and test report.									
7.2	**Requirements for fault simulation test of computer-based system**									
a	It is proved by failure analysis that for a single failure, the system can enter the fail-to-safe status, and the running system will not be degraded to the extent that cannot meet the acceptable performance standards specified by CCS.									
b	The fault simulation shall be carried out as realistically as possible. The specifications of fault simulation test include the following contents: ① Name of									

S/N	Requirements	Reference item	System supplier	System integrator	Owner	Category I System	Category II System	Category III System	Documents to be provided	Whether it meets the requirement ?
	faulty component or element; ② Fault type; ③ Fault injection mode; ④ Fault response required (output record).									
c	The test cases and expected results of the fault simulation test shall be documented. The results of the fault simulation test as well as whether the test objectives and test criteria are met shall be stated. If a failure occurs, the cause of the failure shall be recorded and analyzed.									
8	**Factory acceptance test (FAT), including software validation test**	8.2.1.7 9.7							① System description ② FAT program ③ FAT report ④ User manual ⑤ System test report, etc.	
8.1	**Preparing FAT program (test outline)**		x				Ⓐ	Ⓐ		
a	The FAT program shall include the selection of representative test items from the system test, including normal functional test and fault simulation test (i.e., fault response test). The principles of selecting test items are: ①									

S/N	Requirements	Reference item	System supplier	System integrator	Owner	Category I System	Category II System	Category III System	Documents to be provided	Whether it meets the requirement ?
	Completeness of FAT content according to software requirement specifications; ② Correctness of FAT content according to software requirement specifications; ③ Repeatability; ④ Precisely defined test configuration.									
8.2	**Performing FAT**		x				Ⓦ	Ⓦ		
a	Perform FAT on actual hardware and have the necessary tools or means for simulating functions and failure responses. Other test solutions, such as the use of replica hardware or simulation tools (emulators), shall be subject to CCS approval before implementation.									
b	For a Category II or Category III system, a cyber test shall be performed to verify compliance with cyber resilience requirements. The cyber test may be carried out as part of the onboard test with the consent of CCS.									

S/N	Requirements	Reference item	System supplier	System integrator	Owner	Category I System	Category II System	Category III System	Documents to be provided	Whether it meets the requirement ?
c	For a Category II or Category III system, the implementation of the system technical requirements is verified as part of the FAT.									
d	For a Category II or Category III system, there should be credible third-party test reports.	Appendix 4: 2.3								
8.3	**Preparing FAT report**		x				①	①		
a	Test results such as FAT report shall be documented and include the following contents: ① The FAT process shall be recorded in chronological order to trace the sequence of FAT activities; ② Version of FAT program; ③ Software requirements verified by FAT; ④ Tools and equipments used and their calibration data; ⑤ List of software (including software versions) installed in the system at FAT. ⑥ Differences between actual and expected results.									

S/N	Requirements	Reference item	System supplier	System integrator	Owner	Category I System	Category II System	Category III System	Documents to be provided	Whether it meets the requirement ?
b	When there is a difference between the actual result and the expected result, analysis and evaluation shall be carried out to determine whether to continue the test, or to make a change request and return to an earlier stage of the software development lifecycle.									
c	The FAT report shall demonstrate that all software specified requirements have been met and that the software does not perform unintended functions. The FAT report shall also indicate that: (a) the software has passed the test; Or (b) the reason for the failure.									
8.4	**Other FAT documents**		x				Ⓘ (when necessary)	Ⓘ (when necessary)		
a	Other FAT documents, such as user manual, system test report, etc., shall be submitted to CCS for reference if necessary.									
9	**Environmental compliance requirements for hardware**	8.2.1.4	x			Ⓘ (when necessary)	Ⓘ	Ⓘ	① Environmental	

S/N	Requirements	Reference item	System supplier	System integrator	Owner	Category I System	Category II System	Category III System	Documents to be provided	Whether it meets the requirement ?
a	Environmental compliance test report or type approval certificate that meets the requirements of CCS Guidelines for Type Approval Test of Electric and Electronic Products.								compliance test report or type approval certificate.	
10	**Preparations before onboard tests**								① System description ② List of system categorizations ③ Risk assessment report ④ System architecture description ⑤ FAT report ⑥ User manual	
10.1	**List of system categorizations**	8.2.2.3		x		Ⓐ	Ⓐ	Ⓐ		
a	The category which the system belongs to shall be determined based on the impact of the system failure, and a list of system categorizations shall be developed.									
10.2	**Risk assessment report**	8.2.2.4		x		Ⓐ (when necessary)	Ⓐ (when necessary)	Ⓐ (when necessary)		
a	Upon the request of CCS, the system integrator shall conduct a risk analysis of the specific system of the vessel and prepare a risk assessment report to determine the applicable categroy of the system.									

S/N	Requirements	Reference item	System supplier	System integrator	Owner	Category I System	Category II System	Category III System	Documents to be provided	Whether it meets the requirement ?
b	If the system categroy is amended based on risk assessment and analysis, a consent from CCS and the system supplier may be required.									
c	When the risk of the computer-based system is obvious, the submission of the risk assessment report is allowed to be exempted, but the supporting documents shall be submitted to explain the reason for the exemption. Clauses d and e are not applicable at this time.									
d	Appropriate analysis methods are adopted, such as fault tree analysis, risk analysis, FMEA or FMECA analysis; (This clause does not apply when clause c comes into force)									
e	It is proved by failure analysis that for a single failure, the system can enter the fail-to-safe status, and the running system will not be degraded to the extent that cannot meet the acceptable performance standards specified by CCS. (This									

S/N	Requirements	Reference item	System supplier	System integrator	Owner	Category I System	Category II System	Category III System	Documents to be provided	Whether it meets the requirement ?
	clause does not apply when clause c comes into force)									
f	The data link failure of Category II and Category III computer-based system shall be specified in the risk assessment and analysis. (When applicable)									
10.3	**System architecture description**	8.2.2.5		x		①	①	①		
a	The system of systems of the vessel shall be specified and a system architecture description shall be prepared.									
b	The system architecture description shall at least include the following contents: ① Overview of the total system architecture (the system of systems); ② Purpose and main functionality of each system; ③ Communications and interfaces between different systems.									
11	**Onboard test**	8.2.2.6 8.2.2.7							① SAT program	

S/N	Requirements	Reference item	System supplier	System integrator	Owner	Category I System	Category II System	Category III System	Documents to be provided	Whether it meets the requirement ?
11.1	**Preparing onboard test programs (test outlines)**			x			Ⓐ	Ⓐ	② SAT report ③ SOST program ④ SOST report	
a	The onboard test programs include verification of design functions.									
b	The onboard test programs include verification of safety responses triggered by internal failures or external system equipment failures.									
c	The onboard test programs include verification of secure interconnection with other systems onboard.									
d	The onboard test programs include SAT program and SOST program. When the test range is similar, the two tests can be combined into one.									
11.2	**Performing onboard tests**			x			Ⓦ	Ⓦ		
a	After the computer-based system is installed and integrated with the relevant mechanical / electrical / process systems on board,									

S/N	Requirements	Reference item	System supplier	System integrator	Owner	Category I System	Category II System	Category III System	Documents to be provided	Whether it meets the requirement ?
	including possible interfaces with other control and monitoring systems, the SAT shall be carried out to verify the functional operation of the computer-based system.									
b	After the different computer-based systems have been installed and integrated in the final environment on board, an integration test of the whole ship (i.e., SOST) is carried out to verify the functionality of the different systems after complete installation, including whether all interfaces and inter-dependencies are in accordance with requirements and regulations. The SOST shall verify at least the following aspects: ① The overall functionality of the system of systems; ② The failure response between systems, which shall comply with the fail-to-safe principle; ③ Performance; ④ Human-machine interfaces; ⑤ Interfaces and interconnections between different systems.									

S/N	Requirements	Reference item	System supplier	System integrator	Owner	Category I System	Category II System	Category III System	Documents to be provided	Whether it meets the requirement ?
c	The wireless data communication equipment shall be tested during harbour and sea trials to demonstrate that, under the expected operating conditions, the radio frequency transmission will not cause failure of itself and any other equipment due to electromagnetic interference.									
d	For Category II and III systems, the implementation of the technical requirements is verified as part of the SAT.									
e	Depending on CCS requirements, the cyber test of a Category II or Category III system may be performed to verify whether it meets cyber resilience requirements.									
11.3	**Preparing onboard test reports**			x			①	①		
a	The onboard test reports shall be generated, including SAT report and SOST report, based on the test results.									

S/N	Requirements	Reference item	System supplier	System integrator	Owner	Category I System	Category II System	Category III System	Documents to be provided	Whether it meets the requirement ?
b	The onboard test reports shall contain an overall evaluation of the tested software.									
c	The difference between the testing environment and the actual operating environment, as well as the impact of this difference on the test results, shall be provided.									
d	The test summary shall include "All results met expectations", "Problems encountered" (if applicable) and "Deviations from requirements" (if applicable).									
12	**Verification of changes in computer-based system**	8.4 9.8							① Change management procedure ② Change request or change	
12.1	**General verification requirements**		x	x	x	① (when necessary)	①	①		

S/N	Requirements	Reference item	System supplier	System integrator	Owner	Category I System	Category II System	Category III System	Documents to be provided	Whether it meets the requirement ?
a	The system supplier and system integrator shall conduct an impact analysis on the change plan for an approved system and report to CCS if necessary. The impact analysis includes: ① Determining the criticality of the change; ② Determining the impact on existing documents; ③ Determining the needed verification and test activities; ④ Determining whether other stakeholders need to be informed of the change; ⑤ Determining whether an approval from other stakeholders (e.g., CCS and/or the owner) is required.								description ③ Change impact analysis results ④ Change records ⑤ Corresponding test reports	
b	Return to the appropriate stage of the software development lifecycle based on the results of the change impact analysis.									
c	The methods to backup and restore the software and data of the shipboard computer-based system shall be clearly defined. During system maintenance, it shall be possible to roll back the software									

S/N	Requirements	Reference item	System supplier	System integrator	Owner	Category I System	Category II System	Category III System	Documents to be provided	Whether it meets the requirement ?
	to the previous version to restore the system to a known stable state. The rollback shall be recorded and analyzed to identify and eliminate the root cause of the change failure.									
d	The Change shall be verified as far as possible prior to installation on board. After installation, it shall be verified on board according to the documented verification program, including: ① Verifying whether the new function and/or improvement achieves the expected effect; ② Verifying through regression test that the change does not have any negative impact on a function or capability that should not have been affected.									
e	Changes to the system and software shall be documented to ensure visibility and traceability of changes. The change records shall contain at least the following: ① Change purpose; ② Description of change; ③ Main conclusion of change impact analysis; ④									

S/N	Requirements	Reference item	System supplier	System integrator	Owner	Category I System	Category II System	Category III System	Documents to be provided	Whether it meets the requirement ?
	Identity and version of any new system or software; ⑤ Test report or test summary. Software changes may be recorded in the software registry, or similar files.									
f	CCS will perform the verification in the shipbuilding phase (during and after the FAT), in combination with the actual application of the change management procedures in the specific project.									
g	During the operation phase of the vessel, CCS will usually verify the changes as a part of the annual survey of the vessel. At the time of survey, the owner shall provide the change management procedures and relevant change records to CCS.									
12.2	**Additional verification requirements for major changes**									
a	Subsequent major changes to the software for a Category II or Category III system by the system supplier and system integrator		x	x			Ⓐ	Ⓐ		

S/N	Requirements	Reference item	System supplier	System integrator	Owner	Category I System	Category II System	Category III System	Documents to be provided	Whether it meets the requirement ?
	shall be submitted to CCS for approval.									
b	For a Category II or Category III system, CCS shall witness the verification process of major changes to the software.			x			Ⓦ	Ⓦ		

Note 1: The symbols used in the table and their meanings are as follows:

Ⓐ Submit to CCS for approval; Ⓘ Submit to CCS for reference; Ⓦ CCS witness required

Note 2: If the "role" and "system category" of the specific item in the table are not specified (blank), then the options of the title item are adopted.

Note 3: The specific items in the table can be tailored according to the actual situation.

Note 4: The "documents to be provided" in the table can be merged or decomposed, or other names can be used, as long as the contents required by this Guide are available.

Note 5: The meaning of the mark under "whether it meets the requirement" in the table: X, Pass; O Fail; -Not applicable. For the items that "fail" the test or are "not applicable", they can be left blank and not marked.

Acceptance conclusion of test and verification: _____

Approved by:_____ Approved on:_____

Appendix 2 Evaluation of Small Low Complexity Computer System

1 Purpose

1.1 Through the single-case evaluation of a small low complexity computer system, the evaluation method for its software is reasonably and effectively simplified.

2 Requirements

2.1 Documentation

2.1.1 The software description can combine the documents to be provided in Table 10.1 according to the internal document management system of the system supplier and the system integrator, but the contents shall include:

(1) System function description, including function description of software modules and hardware description of relevant programmable devices, list of interfaces between the system and other systems of the vessel, list of data transmission standards, especially the mutual constraints and dependencies between functions, performance, modules and other components;

(2) Software design description, including software function description, software maintenance and operation manual, especially software configuration including priority strategy;

(3) List and version number of software installed in the system;

(4) Failure mode analysis;

(5) Switching mechanism for redundant systems (if any);

(6) System test, integration test and fault simulation test methods.

2.2 Test

(1) For newly designed products, their failure mode analysis shall be checked and tested according to the confirmed testing methods provided by the system supplier and system integrator.

(2) The evidence of functional test and integration test of programmable device shall be provided at the level of software module, subsystem and system.

(3) For software reuse or modification, attention shall also be paid to regression test.

Note: Software reuse is to use all kinds of relevant knowledge of existing software to build new software to reduce the cost of software development and maintenance. Software reuse is an important technique to improve software productivity and quality. Software reuse is mainly code-level reuse, which does not specifically refer to programs, but also includes domain knowledge, development experience, design decisions, architecture, requirements, design, code and documents.

3 Input

3.1 Computer-based system requirement specifications

4 Output

4.1 Software description

4.2 Hardware description

4.3 Test report

Appendix 3 Technical Proposal for Design and Implementation Phases of Computer-based System

1 General requirements

1.1 The design of computer safety related system (including the overall structure of software and hardware, sensors, actuators, programmable electronics, embedded software and application software, etc., as shown in the figure below) shall meet all the requirements of 1.1.1~1.1.2 below:

1.1.1 Hardware safety integrity requirements include:

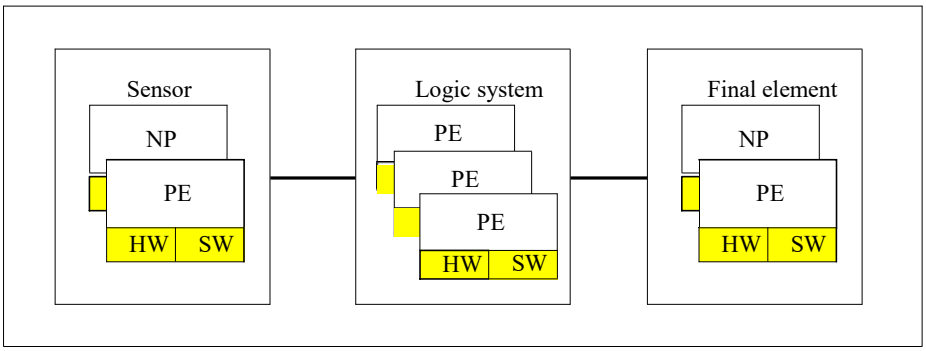
(1) Structural constraints of hardware safety integrity; and

(2) Requirements for the probability of dangerous random hardware failure.

1.1.2 System safety integrity requirements include:

(1) Requirements for failure avoidance and system failure control; or

(2) Evidence that the equipment is verified through use.



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, e.g.: - Diagnostic test; - Redundant processor; - Dual I/O card	Examples include: - Communication driver; - Failure handling; - Executable software	Examples include: - Input/ output function; - Derivative functions (e.g., sensor inspection when embedded software service is not provided)

PE: programmable electronics, NP: non-programmable devices, HW: hardware, SW: software.

Figure 3-1.1.2 Relationship between PE Hardware and Software Structure

1.2 Each part of the computer-based system that performs both safety function and non-safety function shall be considered safety-related unless it can be proved that the safety function and non-safety function are sufficiently independent (that is, the failure of non-safety function will not cause dangerous failure of safety function). The safety function shall be separated from the non-safety function whenever practicable.

1.3 The safety integrity level of the computer-based system is determined by the safety function with the highest safety integrity level, unless it can be proved that the safety functions of different safety integrity levels are fully independent.

1.4 If the safety functions are required to be independent of each other (see 1.2 and 1.3), the following shall be documented in the design:

1.4.1 Methods of achieving independence;

1.4.2 Verification of the rationality of methods.

2 Techniques and measures for hardware safety integrity: failure control during operation

Table 2-1 to Table 2-6 in Appendix 3 give recommendations on hardware safety integrity techniques and measures.

Appendix 3 Table 2-1 I/O Unit and Interface (External Communication)

Diagnostic techniques/measures	See IEC61508-7	Maximum diagnostic coverage considered achievable	Note
Failure detection by online monitoring	A1.1	Low (low demand mode) Medium (high demand or continuous mode)	Depends on diagnostic coverage of failure detection
Test pattern	A6.1	High	
Code protection	A6.2	High	
Multi-channel parallel output	A6.3	High	Valid only if the data flow changes during the diagnostic test interval
Monitored outputs	A6.4	High	Valid only if the data flow changes during the diagnostic test interval

Appendix 3 Table 2-2 Data Path (Intercom)

Diagnostic techniques/measures	See IEC61508-7	Maximum diagnostic coverage	Note

		considered achievable	
One-bit hardware redundancy	A7.1	Low	
Multi-bit hardware redundancy	A7.2	Medium	
Complete hardware redundancy	A7.3	High	
Inspection using test patterns	A7.4	High	
Transmission redundancy	A7.5	High	Valid for instantaneous failure only
Information redundancy	A7.6	High	

Appendix 3 Table 2-3 Power Supply

Diagnostic techniques/measures	See IEC61508-7	Maximum diagnostic coverage considered achievable	Note
Overvoltage protection with safety shut-off or switch-over to standby power unit	A8.1	Low	The techniques in this table shall be used, and other techniques are also recommended.
Voltage control with safety shut-off or switch-over to standby power unit (secondary)	A8.2	High	
Power-down with safety shut-off or switch-over to standby power unit	A8.3	High	The techniques in this table shall be used, and other techniques are also recommended.
Idle current principle	A1.5	Low	For power failure only

Appendix 3 Table 2-4 Program Sequence (Watchdog)

Diagnostic techniques/measures	See IEC61508-7	Maximum diagnostic coverage considered achievable	Note
Watchdog with separate time base but without time window	A9.1	Low	
Watchdog with separate time base and time window	A9.2	Medium	
Logical monitoring of program sequence	A9.3	Medium	Rely on monitoring quality

Combination of temporal and logical monitoring of program sequences	A9.4	High	
Temporal monitoring with online check	A9.5	Medium	

Appendix 3 Table 2-5 Sensor

Diagnostic techniques/measures	See IEC61508-7	Maximum diagnostic coverage considered achievable	Note
Failure detection by online monitoring	A1.1	Low (low demand mode) Medium (high demand or continuous mode)	Depends on diagnostic coverage of failure detection
Idle current principle	A1.5	Low	Only valid for E/E/PE safety related systems that do not require continuous control and do not achieve or maintain EUC safety status.
Analogue signal monitoring	A2.7	Low	
Test pattern	A6.1	High	
Input comparison/voting	A6.5	High	Valid only if the data flow changes during the diagnostic test interval
Reference sensor	A12.1	High	Depends on diagnostic coverage of failure detection
Positive-activated switch	A12.2	High	

Appendix 3 Table 2-6 Final Element (Actuator)

Diagnostic techniques/measures	See IEC61508-7	Maximum diagnostic coverage considered achievable	Note
Failure detection by online monitoring	A1.1	Low (low demand mode) Medium (high demand or continuous mode)	Depends on diagnostic coverage of failure detection
Relay contact monitoring	A1.2	High	

Idle current principle	A1.5	Low	Only valid for E/E/PE safety related systems that do not require continuous control and do not achieve or maintain EUC safety status.
Test pattern	A6.1	High	
Monitoring	A13.1	High	Depends on diagnostic coverage of failure detection
Cross monitoring of multiple actuators	A13.2	High	

3 Recommendations on technology and measures for system integrity

3.1 Table 3.1-1 and Table 3.1-2 in Appendix 3 give recommendations on technology and measures for system safety integrity.

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

3.1.2 Control of failures caused by environmental stress or influence;

3.1.3 Control of operation process failures.

Appendix 3 Table 3.1-1 Techniques and Measures for Controlling System Failures Caused by Hardware Design

	Techniques/measures	See IEC61508 -7	I	II	III
1	Program sequence monitoring	A.9	Strongly recommended (Note 1) Low (Note 2)	Strongly recommended Low	Strongly recommended Medium
2	Failure detection by online monitoring	A1.1	Recommended Low	Recommended Low	Recommended Medium
3	Test with redundant hardware	A2.1	Recommended Low	Recommended Low	Recommended Medium
4	Standard test of access ports and boundary scan structure	A2.1	Recommended Low	Recommended Low	Recommended Medium
5	Code protection	A6.2	Recommended Low	Recommended Low	Recommended Medium

6	Diverse hardware	B1.4	- Low	– Low	Recommended Medium
Note: It is required to apply at least one technology in 2~6. Note 1: Importance of the techniques/measures, including required, strongly recommended, recommended, - (neither recommended nor opposed). Note 2: Minimal effectiveness level that the techniques/measures adopted shall achieve, including low, medium and high.					

Appendix 3 Table 3.1-2 Techniques and Measures for Controlling System Failures Caused by Environmental Stress or Influence

	Techniques/measures	See IEC61508 -7	I	II	III
1	Measures to prevent voltage breakdown, voltage fluctuation, over-voltage and low-voltage	A8	Strongly recommended	Required	Required
2	Separation of electrical energy lines from information lines (Note 1)	A11.1	Strongly recommended	Required	Required
3	Increase of interference immunity	A11.3	Strongly recommended	Required	Required
4	Measures against the physical environment (such as temperature, humidity, vibration, etc.)	A14	Strongly recommended	Required	Required
5	Program sequence monitoring	A9	Strongly recommended Low	Strongly recommended Low	Strongly recommended Medium
6	Measures against temperature increase	A10	Strongly recommended Low	Strongly recommended Low	Strongly recommended Medium
7	Spatial separation of multiple lines	A11.2	Strongly recommended Low	Strongly recommended Low	Strongly recommended Medium
8	Failure detection by online monitoring (Note 2)	A1.1	Recommended Low	Recommended Low	Recommended Medium
9	Test with redundant hardware	A2.1	Recommended Low	Recommended Low	Recommended Medium
10	Code protection	A6.2	Recommended Low	Recommended Low	Recommended Medium
11	Antivalent signal transmission	A11.4	Recommended Low	Recommended Low	Recommended Medium
12	Diverse hardware (Note 3)	B1.4	– Low	- Low	– Medium
13	Software architecture	GB/T20438.3 Of 7.4.3	See Table A.2 of GB/T20438.3		

Note: It is required to apply at least one technology in 8~13.

Note 1: If optical media are used for information transmission, there is no need to separate electrical energy lines from information lines. For low-power energy lines designed to supply power to system components and transmit information to these components, it is also not necessary to separate electrical energy lines from information lines.

Note 2: For safety related systems (e.g., emergency shutdown system) working in low demand mode, the diagnostic coverage achieved from failure detection by online monitoring is usually low or non-existent.

Note 3: If it is proved through validation and extensive operational experience that, to fulfil the target failure measures, the hardware has fully avoided design faults and is sufficient to prevent common cause failures, then there is no need for diverse hardware.

Appendix 4 Software Testing Requirements during Development Phase

1 Software testing problem level

Software testing problems are divided into: fatal problems, serious problems, general problems, minor problems, and improvement suggestions. The developed software shall be free of fatal problems and serious problems.

(1) Fatal problem: A software problem that inevitably leads to the failure of a major software task in full completion, or a software problem that causes the system to crash during important work.

(2) Serious problem: A software problem that may result in the failure of a software task in full completion or partially affect the completion of a software task.

(3) General problem: A software problem that does not affect the completion of a software task, but affects the important quality characteristics of the software such as functionality, performance, reliability and safety; Or a software problem that it is difficult to determine the external impact on the software, but the program code itself has a severe flaw.

(4) Minor problem: A software problem that affects the usability, efficiency, maintainability, portability and other general characteristics of the software, although it does not affect the important quality characteristics of the software such as functionality, performance, reliability and safety; Or a software problem that it is difficult to determine the external impact on the software, but the program code itself has a non-severe flaw.

(5) Improvement suggestion: A software problem that has no impact on the use of the software but can be further improved in terms of the standardization, clarity and understandability of the software, or a constructive proposal.

2 Software testing requirements for stakeholders

2.1 In the software development lifecycle, the software test of the computer-based system can be divided into: developer test, third-party test, and acceptance test according to the implementation party.

2.2 Developer test

The general requirements for the developer test are as follows:

(1) The developer test is implemented by the software manufacturer. The software manufacturer is usually the system supplier or sub-supplier.

(2) The developer test generally includes software module test, software integration test, computer-based system integration test and software validation test.

(3) The developer shall classify and deal with the problems found in the test.

(4) After software change, the developer needs to carry out regression test at the corresponding level.

(5) The documents generated during the test shall be included in the configuration management.

(6) For different test levels, the documents generated by the developer test include at least: test plans, test records and test reports (including software problems, or independent software problem report).

2.3 Third-party test

The general requirements for the third-party test are as follows:

(1) The third-party test shall be implemented by the software testing laboratory of CCS, or a professional software testing agency recognized by CCS.

(2) The software of Category II and Category III computer-based systems shall be subject to third-party test.

(3) The documents generated during the test shall be included in the configuration management.

(4) Access conditions for third-party test:

① The system under test includes software (source code, executable program, project file, configuration file), project charter, software requirement specifications, design specifications, and necessary communication protocol, model formula, user's manual /instructions for use, etc.

② The system under test and related documents shall come from a controlled library or product library.

(5) The software problems found shall be classified according to the problem attribute and problem level.

(6) The process of handling problems found in the third-party test shall be as follows:

① The third party shall formally submit the software problems found in the test to the software manufacturer in the form of a software problem report.

② The software manufacturer shall confirm the software problems, fill in the handling opinions, and return them to the third-party test agency.

③ After the software is modified by the software manufacturer, the system under test and the analysis results of the change impact shall be submitted. The third party shall perform regression test on the modified software.

(7) For different levels of test, the documents generated by the third-party test include at least: test plans, test records and test reports (including software problems, or independent software problem report).

2.4 Acceptance test

Acceptance test is a software validation activity, and the general requirements are as follows:

(1) The acceptance test is implemented by the software customer. The software customer is usually the system integrator or the owner.

(2) The system under test shall be from a controlled library or product library.

(3) Acceptance test generally requires testing of the software's functions, performance and interfaces, and if necessary, testing of safety.

(4) The software problems found shall be classified and handled in the acceptance test.

(5) For software problems found in the acceptance test, the software problem report form shall be filled in detail and the software manufacturer shall be notified in time.

(6) The documents generated during the acceptance test shall at least include: test plans, test records and test reports (which may include software problems or may be a separate software problem report).

3 Software testing tools and testing environment

3.1 Testing tools

(1) Appropriate testing tools shall be selected according to the requirements and characteristics of test project, including purchased commercial testing tools and self-developed testing tools.

(2) Technical means shall be taken to ensure that the functionality and performance of self-developed testing tools meet the requirements.

(3) Software testing tools with indicators or range requirements shall be checked for their scope of application before being put into use.

(4) The software testing tools shall be managed, and there shall be methods for version control, upgrade and technical support of the software testing tools.

3.2 Testing environment

(1) The testing environments shall include the operating environment of the tested software and the environment for test cases (simulation).

(2) If a test project has specific requirements for the operating environment, a specific testing environment shall be developed (including partial self-development or secondary development); The specific environment of the software developer or user can also be used for testing.

(3) The differences between the software testing environment and the real operating environment shall be analyzed. Generally, the differences in data, external interfaces, system load, etc. should be considered.

(4) During software validation, the tested software shall be operated in a real system working environment or a compatible system operating environment. If a simulation testing environment is selected, it shall be demonstrated and approved by CCS.

4 Software testing types

Select the appropriate software testing type below according to the test level and requirements of the software. This Guide does not list all software testing types.

4.1 Documentation review

Review the tested software documents according to the document checklist, which generally includes the following:

(1) Review the completeness of documents;

(2) Review the completeness of document identification and signing;

(3) Review the completeness, accuracy, consistency and traceability of document content;

(4) Review the standardization of the document format.

4.2 Code review

Review the tested software according to the code checklist, which generally includes the following:

(1) Review the completeness and consistency of engineering documents;

(2) Review the consistency between code and design;

(3) Review the implementation of coding standards;

(4) Review the correctness of the code logic expression;

(5) Review the reasonableness of the code structure;

(6) Review the code readability;

(7) Review the compliance of binding documents.

4.3 Code walkthrough

Find the defects of the tested software according to the code logic, generally including the following:

(1) A code walkthrough shall be conducted on at least one complete functional module or complete topic;

(2) Manually check the code logic and record the code walkthrough results;

(3) If necessary, the structure diagram, state transition diagram, and sequence diagram can be drawn.

4.4 Static analysis

Static analysis is a method of analyzing the mechanical and procedural characteristics of the code. The general requirements are as follows:

(1) For static analysis, the name and version of the analysis tool shall be specified;

(2) The main contents of static analysis include:

① Control flow analysis;

② Data flow analysis;

③ Interface analysis;

④ Expression analysis;

⑤ Code static quality metrics;

⑥ Coding rule check.

4.5 Functional testing

Functional testing is to test the functional requirements stipulated in software requirement specifications or design specifications item by item, aiming to verify if the functions meet the requirements.

The general requirements for the functional testing are as follows:

(1) Determine the software input through equivalence class analysis;

(2) The input equivalence classes shall include normal equivalence class and abnormal equivalence class;

(3) The functional testing can be carried out in combination with other testing types, such as: boundary testing, strength testing, safety testing, etc;

(4) In the integration test or software validation test of computer-based system, the correctness and rationality of the control process shall be verified.

4.6 Performance testing

Performance testing is to test the performance requirements stipulated in software requirement specifications or design specifications item by item, aiming to verify if the performance meets the requirements.

The general requirements for the performance testing are as follows:

(1) Test the accuracy performance and time performance:

① Test the accuracy value of actual data processing for functions with data accuracy requirements;

② Test the accuracy value of actual data processing for functions with time accuracy requirements;

(2) Test the amount of data that needs to be processed to complete the function;

(3) Test the space occupied by the software operation;

(4) Test the integration performance of software and hardware;

(5) Test the processing ability of computer-based system for concurrent transactions and concurrent user access;

(6) Specific quantitative values shall be obtained from the test results, and at least 3 groups of measured values shall be obtained;

(7) Give the statistical results of the maximum, minimum and average values of the performance testing;

(8) The performance testing can be carried out in combination with other testing types, such as: margin testing, strength testing, etc.

4.7 Interface testing

Test each interface specified in the software requirement specifications and other documents, generally including the following:

(1) Test all external interfaces and check the correctness of interface implementation;

(2) Each feature of the interface is covered by at least one normal test case and one accepted abnormal test case;

(3) Test the impact of different interface data, communication rates, error types, etc. on software functions and performance.

4.8 Strength testing

In the process from normal software operation to failure, strength testing is used to verify the critical point at which the software can work under extended conditions, generally including the following:

(1) Provide the maximum amount of information to be processed;

(2) Provide saturation experimental indicator of data processing capability;

(3) Test the software response in the error state;

(4) Carry out continuous and uninterrupted testing within a specified duration.

4.9 Margin testing

The margin testing is used to determine whether the software meets the margin requirements specified in the requirement specifications. If there are no specific requirements, a margin of more than 20% is generally required.

The margin testing shall test the time margin, space margin and transmission margin. The general requirements are as follows:

(1) For functions with time constraint requirements, the margin of the actual execution time relative to the time constraint requirements shall be tested;

(2) For functions with space constraint requirements, the margin of the actual occupied space relative to the space constraint requirements shall be tested;

(3) For external communication interfaces, the actual transmission time and data transmission amount shall be tested to determine the margin relative to the hardware configuration capability.

4.10 Boundary testing

Boundary testing is the testing of the operating state of the software at the boundary or endpoint. The general requirements are as follows:

(1) Test the boundaries or endpoints of the input or output domain;

(2) Test the boundaries or endpoints of state transition;

(3) Usually, the integer field shall be tested. However, the real number field with extremely large number that cannot be exhausted shall also undergo boundary testing;

(4) The boundary testing can be combined with other testing types, such as functional testing and performance testing. Test the boundaries or endpoints of the functionality / performance.

4.11 Human-machine interface (HMI) testing

The general requirements for HMI testing are as follows:

(1) Test the consistency and compliance of the operation and display interfaces with the requirements specified in the software requirement specifications.

(2) Test the robustness of the HMI with unconventional operation, misoperation and rapid operation;

(3) Test the detection ability and prompt of wrong commands or illegal data;

(4) Test the detection and prompt of wrong operation process;

(5) Verify the consistency of text and reality item by item according to the user manual or operation manual;

(6) The HMI testing can be carried out in combination with other testing types, such as functional testing, performance testing, boundary testing, etc.

4.12 Recovery testing

Test all circumstances leading to the recovery or rest of the software with recovery or reset functions, so as to verify its recovery or reset functions. The recovery testing is to verify that the system can continue to work normally after overcoming the hardware failure without causing any damage to the system.

The general requirements for the recovery testing are as follows:

(1) Test the function of software to detect errors;

(2) Test the ability to restore normal operation through fault-tolerant measures after failure;

(3) Test the ability to resume working through self-resetting or standby machine switching after failure;

(4) Test the ability of the software to resume continuous operation according to the recorded data when the system is re-operated after failure;

(5) Recovery testing can be carried out in combination with other testing types, such as: safety testing, functional testing, performance testing, etc.

4.13 Safety testing

Safety testing is the testing to check whether the existing safety and confidentiality measures in the software are effective. The testing shall be carried out under conditions consistent with actual use as far as possible.

The general requirements for the safety testing are as follows:

(1) Carry out a software safety analysis and clearly identify every dangerous state and the possible causes of danger in the software requirements, and fully inspect the software's response in these dangerous states during testing;

(2) Test the software failure modes identified in the software safety requirements;

(3) Test the implementation of software reliability and safety design criteria;

(4) Test the structures, algorithms, fault tolerance, redundancy, interrupt handling and other schemes used to improve safety in software design;

(5) Test possible exception events, including:

① Possible hardware exception events;

② Possible software exception events;

③ Possible operation exception events;

④ Possible input exception events.

(6) The testing shall be carried out under conditions consistent with actual use as far as possible;

(7) In addition to the testing under normal conditions, the software shall be tested under abnormal conditions to indicate that an unsafe state will not be caused by possible single or multiple input errors;

(8) Hardware and software input failure mode testing shall be included;

(9) The testing of boundary, out-of-boundary and boundary junction shall be included;

(10) Inputs for "0", crossing "0", and approaching "0" in both directions shall be included;

(11) Minimum and maximum input data rates in the worst-case configuration should be included to determine the inherent capabilities of the system and its response to these environments;

(12) The operator interface testing shall include operator errors in safety critical operations to verify the system's response to these errors;

(13) Test the correctness and continuity of duplex switching and multi-machine replacement;

(14) Test the security protection capability of important data;

(15) Safety testing can be carried out in combination with other testing types, such as interface testing, strength testing, recovery testing, etc.

4.14 Logic testing

The internal logic structure and related information of the software, with the designed or selected test cases, shall be used to test the logic path and check the software status to determine whether the actual status is consistent with the expected status. Generally, it includes the following contents:

(1) Statement coverage;

(2) Branch coverage;

(3) Conditional coverage,

(4) Expression coverage;

(5) Bit flip coverage;

(6) State machine coverage.

4.15 Timing sequence testing

Under typical working conditions, maximum working conditions and minimum working conditions, the software delay, setup time, hold time and other indicators are tested, generally including the following:

(1) Test whether the setup and hold time meet the requirements;

(2) Test whether the phase, delay, level width, etc. of the timing control signal meet the requirements;

(3) Test whether the frequency and duty cycle of the pulse signal meet the requirements.

4.16 Power consumption analysis

Power consumption analysis of the tested software during operation generally includes the following contents:

(1) Perform power consumption analysis under the constraints of rated operating frequency, operating voltage, ambient temperature, input signal frequency, output load capacitance and drive current, and internal signal flip rate;

(2) Perform power consumption analysis under rated operating time conditions.