2022

Guidance for Computer-based System Conformity Assessment

GC-30-E

KR

APPLICATION OF “GUIDANCE FOR Computer-based System Conformity Assessment”

1. Unless otherwise noted, the requirements in the Guidance apply to software for which the application for software conformity assessment is dated on or after 1 July 2022.
2. The amendments to the Guidance for 2022 edition and their effective date are as follow;

Effective Date : 1 July 2022

Complete revision

CONTENTS

CHAPTER 1 GENERAL

Section 1 General

101. Application

1. This Guidance is applied to control, observation and safety software installed on ships and/or offshore plants, operating support software for ships and/or offshore plants, product design and technical/engineering software for marine equipment.

2. Items not included in this Guidance may comply with ISO, IEC or equivalent recognized standards by the appropriate consideration of the Society.

3. In application to 203. test result may be accepted in cases where test has been carried out or approved as follow.

   (1) At a laboratory accredited for all the required tests by an accreditation body being member of KOLAS in accordance with KS Q ISO/IEC 17025
   (2) At a laboratory accredited for all the required tests by an accreditation body being member of ILAC in accordance with ISO/IEC 17025
   (3) The assessment of Certificates issued by other Classification Society recognized by the Society

102. Definitions

The Definitions of terms are to follow the **Rules for Classification of Steel Ships**, unless otherwise specified in this Guidance.

1. **“Anomaly”** means a any condition that deviates from expectations based on requirement definition, product specifications, this Guidance, etc. or from someone‘s perceptions or experiences.

2. **“Application Software”** means a software that users use directly on computers where the operating system is installed.

3. **“Code Coverage”** means the percentage of codes tested in the entire software source code as a criterion for determining the level of dynamic testing, it is classified with 'Statement', 'Branch', 'MC/DC' according to testing level.

4. **“Coding Rule”** means a set of Guidance for a specific.

5. **“Conformity Assessment”** means a systematic examination of the extent to which a product, process or service fulfills specified requirements.

6. **“Dynamic Testing”** means a testing that requires the execution of the test item.

7. **“Embedded Software”** means a software developed to perform a predetermined specific function on a microprocessor installed in electronic products, information devices. etc.

8. **“False Alarm”** means a case that static analysis tool reports a fault when one does not exist.

9. **“Fault“** means incorrect step, process, or data definition in a computer program.

10. **“Function”** means a implementation of an algorithm in the software with which the end user or the software can perform part or all of a work task.

11. **“Product description”** means a document stating properties of software, with the main purpose of helping potential acquirers in the evaluation of the suitability for themselves of the software before purchasing it.

12. **"Source Code Metric"** means index that measures the quality of the source code.

13. **“Statement Coverage”** means a percentage of the set of all executable statements of a test item that are covered by a test set.

14. **“Computer-based System(Software)”** means a system of one or more computers(including programmable electronic device), associated software, peripherals and interfaces, and the computer network with its protocol.

15. **“Static Testing”** means a testing in which a test item is examined against a set of quality or other

criteria without code being executed.

16. **“Test Case”** means a set of inputs, execution conditions, and expected results developed for a particular objective, such as exercise a particular program path or to verify compliance with a specific requirement.
17. **“Test Plan”** means detailed description of test objectives to be achieved and the means and schedule for achieving them, organised to coordinate testing activities for some test item or set of test items
18. **“Weakness"** means as flaws, bugs, faults, or other errors, that create vulnerabilities that can be exploited by both internal and external forces.

103. Exclusion from the Guidance

The Society can not assume responsibility for use of unauthorized commercial products and other technical characteristics not specified in this Guidance.

Section 2 Assessment Process

201. Application

1. The applicant is, in principle, to be the manufacturer of the materials and equipment. however, the applicant, where deemed appropriate by the Society, need not always be the manufacturer of the materials and equipment.

2. The manufacturer wishing to obtain a conformity assessment is to submit a copy of the application of type approval of the Society, together with three copies of the required data for approval and two copies of the required data for reference, data previously submitted to the Society, according to the Technical Rules, may be exempted from submission.

3. Additional material not include this Guidance may be additionally required by the Society when deemed necessary by the Society.

4. 'Test Plan', 'Test Result‘, 'Anomaly Report‘ may be written separately in accordance with static and dynamic test, Each document can be submitted as an integrated or separate document.

5. Application document

(1) Test plan
   (A) Product purpose
   (B) Product boundaries and configuration
   (C) Product summaries
      (a) System name
      (b) Unique identifier(reference, version number, date of issue)
      (c) The history of changes or any other element that describes the process of revision of the document.
      (d) the identifier of the document referenced in the body of the document.
      (e) Information of writer
   (D) Test purpose and boundaries, method
   (E) Test environment
   (F) Hardware specification
   (G) Test tool;
   (H) Test item(static, dynamic, non-function) and acceptance criteria
(2) Static testing report
   (A) The identification of the static testing report
   (B) The date of the test execution
   (C) The name and the function of the person having carried out the test
   (D) The development constraints(compiler, OS, etc.), language
   (E) The list of acceptance criteria
   (F) The test tool
   (G) The execution result
   (H) The list of the found anomalies by coding rule
   (I) The anomaly description by coding rule
(3) Dynamic testing report
   (A) The identifier of the dynamic testing report
   (B) The date of the test execution
   (C) The name and the function of the person having carried out the test
   (D) The summary of conformity assessment results and, if any, test results
   (E) The test tool
   (F) The list of the found anomalies
   (G) for each anomaly, the reference to the corresponding anomaly report
(4) Anomaly reports
   (A) The identifier of the anomaly
   (B) The point in the test case the anomaly occurred
   (C) The severity(serious, interrupted, simple) and reproducibility of the anomaly
   (D) The anomaly description
(5) Function list
   (A) A table of hierarchical classification of the functions that make up the software.
(6) Mapping table between the product description or requirement definition and test case
   (A) All functions mentioned in the function list shall be classified according to the test case(1:1,

1:N, N:N)

6. Data for reference

(1) Outline of company

(A) Data on history, outline and layout of manufacturing plants

(B) The organization and management structure, including subsidiaries to be included in the approval/certification

(2) When plant audit is required in accordance with the requirements in **204.**, the following reference data may be submitted

(A) Data on major manufacturing facilities

(B) Data on manufacturing process

(C) Data of in-house standards or codes

(D) Data of quality control system

(E) Data on major inspection and test facilities

(F) Service records

(3) Document related to the recognition of test organization

(4) Document related to the recognition of test tools

7. Notwithstanding the requirements in the preceding **Sec 2**, where the applicant is already approved by the Society and the attachments are entirely equal in content to the documents previously submitted the submission of documents may be partly or wholly exempted except for the approval test program.

202. Document review

1. The Society examines the software conformity assessment test plan, drawings and data and where deemed appropriate, those are to be approved and returned to the manufacturers.

2. The document review is to evaluate the appropriateness of a document based on software conformity assessment requirement.

203. Conformity assessment test

1. After completion of the document reviews specified in **202.**, the type tests are to be carried out for the test products in the presence of the surveyor in accordance with the conformity assessment test program and test method as deemed appropriate by the Society.

2. Software which have been failed to pass the conformity assessment tests specified in 1. should not be retested without revision of drawings and/or specifications. If, following analysis of the experimental data from tests, it is found that the failure of type tests have been caused by the poor test conditions, etc., retest without revision may be permitted subject to the Society's approval.

3. Upon completion of the type test, the manufacturer is to submit to the Society the complete test report including test conditions, test results and required information.

204. Plant audit

This is to comply with the requirements in **Ch 3, 105.** of **Guidance for Approval of Manufacturing Process and Type Approval, etc.** where type approval of equipment is carried out simultaneously or already done, plant audit may be omitted.

205. Notification and announcement of approval

This is to comply with the requirements in **Ch 3, 106.** of **Guidance for Approval of manufacturing process and Type Approval, Etc.**

206. Changes in the approved contents

This is to comply with the requirements in **Ch 3, 107.** of **Guidance for Approval of Manufacturing process and Type Approval, Etc.**

207. Validity and renewal of approval certificate

1. The approval certificate will be valid within three years from the date of issue. In case where the approval certificate is renewed in accordance with the requirements specified in the preceding 206., the expiration date will not be changed.
2. This is to comply with the requirements in **Ch 3, 108.** of **Guidance for Approval of Manufacturing Process and Type Approval, Etc.** However, the renewed approval certificate will be valid within three years from the expiry date of old approval certificate.

208. Confirmation test and/or occasional plant audit

This is to comply with the requirements in **Ch 3, 109.** of **Guidance for Approval of Manufacturing Process and Type Approval, Etc.**

209. Suspension or withdrawal of approval

This is to comply with the requirements in **Ch 3, 110.** of **Guidance for Approval of Manufacturing Process and Type Approval, Etc.** ⚓

CHAPTER 2 COMPUTER BASED SYSTEM

Section 1 General

101. General

1. All functions mentioned in the user documentation(product description/requirement definition) shall be executable with the corresponding facilities, properties, and data, and within the given limitations, according to all the statements in the user documentation.

2. The function of the software shall be able to execute according to the description defined in the product description/requirement definition.

3. The software shall be free from contradictions within itself and with the product description/requirement definition.

4. The control of the software operation by the end user following product description/requirement definition and the software behaviour shall be consistent.

5. The software shall perform in accordance with the reliability features defined in the product description/requirement definition.

6. The function related to error handling shall be consistent with corresponding statements in the product description/requirement definition

7. The software shall not lose data when used within the limitations stated in the product description/requirement definition.

8. The software shall recognize violations of syntactic conditions for input and it shall not process this as permissible input.

9. The software shall perform in accordance with the effectiveness features stated in the product description/requirement definition.

10. The product description/requirement definition shall state whether maintenance is offered or not. If offered, the product description/requirement definition. shall describe the maintenance services in accordance with the release plan of the software.

11. If the user can carry out installation, the product description/requirement definition shall contain, as applicable, statements on Portability, taking into account adaptability, installability and replaceability, written such that verifiable evidence of compliance can be demonstrated, based on ISO/IEC 25010.

12. If the user can carry out the installation, the software shall be installed successfully by following the information in the the product description/requirement definition.

13. If the user can carry out installation, successful installation and correct operation of the software application shall be verified for all supported platforms and systems listed in the product description/requirement definition.

14. If the user can carry out installation, the software shall provide a mean for the user to uninstall all its installed components.

102. Test Plan

1. Information contained in the test plan shall be verifiable and correct.

2. All document shall be written based on **Ch 1, 201. 5** (1).

3. The test plan in accordance with the requirements of **Ch 1, 201. 5** (1) shall include:

(1) All the functions described in the product description/requirement definition, as well as the combinations of functions representative of the task to be achieved, shall be subject to test cases.
(2) Each function described in the product description/requirement definition shall be subject to at least one test case.
(3) All the installation procedures shall be subject to test cases.
(4) All the operational limits indicated in the product description and user documentation shall be

subject to test cases.
(5) The test plan shall indicate the criteria used to decide if the test results demonstrate the conformity of the software to the product description

103. Test report

1. The faults shall not be detected in software static testing. however, fault that are not correctable can be handled as an anomaly due to false alarms and software characteristics.

2. The detected false alarm shall be analyzed and made an anomaly report due to limitations of the static testing tool.

3. The alarms that are not correctable shall be made an anomaly report due to software characteristics, even if false alarms detected by software static testing tool.

4. For software dynamic testing, all function and non-function requirement described by product description shall be tested. if dynamic testing is not possible, anomaly report shall be made.

5. For software dynamic testing, the 100% coverage of product shall be achieved. however, where anomaly report shall be made in case of grey code or unable to measure.

6. The test report is to include the requirement specified in **Ch 1, 201. 5** (2), (3).

7. For the static testing result: at least following information is to be included;
(1) The computer systems used for testing (hardware, software, and their configuration)
(2) The test tool used for testing
(3) The software development environment, e.g., complier, development tool, OS, etc.
(4) The static testing process included test tool
(5) The criteria applied to the static testing
(6) Pass/fail criteria
(7) The overall summary of the potential product defects detected by static testing
(8) A number of potential defects classified by criteria type
(9) Where there is anomaly, the summary of the defects is written with identifier
(10) The weakness list applied to the testing
(11) The source code metric check list applied to the testing

8. For the dynamic testing result: at least following information is to be included;
(1) The testing completion date and product identifier shall be included.
(2) The overall summary of the result of all test case and coverage shall be included.
(3) The test report shall be proved that all of test cases in test plan were carried out.
(4) The configuration of hardware and software for testing shall be specified.
(5) All test case shall be included, as test case identifier, test case name, test case purpose and description, preconditions, test procedure, expected results, and actual results. however, where it is necessary to change product configuration because of product characteristic, the product configuration may be partly changed subject to the approval by the Society.
(6) The test tools used in the test shall be specified.
(7) The procedure for performing the tool for coverage measurement shall be specified.
(8) The results of coverage test shall contain, as overall coverage, file or module coverage, function coverage, etc.
(9) The errors found in the test shall be corrected prior to the application, and anomaly item shall be reported in the anomaly report.

104. Anomaly report

1. The anomaly report shall include an overall summary of the anomalies found.

2. The anomaly report separated by static and dynamic test shall be submitted to the Society.

3. The anomaly report shall include for each anomaly:
(1) the identifier and name of the anomaly
(2) the point in the test case the anomaly occurred
(3) the anomaly description

Section 2 Embedded software

201. Application

The requirements of this Section apply to tests and inspection for the conformity assessment of the software installed in embedded OS, real time OS or without OS software for use in the marine environment.

202. Data to be submitted

The following reference data are to be submitted to the Society in addition to those specified in **Ch 1, 201. 5.**

(1) Requirement definition
  (A) Software purpose
  (B) System configuration(hard disk size, memory, video card, LAN card etc.,)
  (C) Function requirement
  (D) Non-function requirement
  (E) Operational limitation related to function and non-function
  (F) Type of user interface(command, menu, help, etc.)

203. Conformity test

1. The static testing in accordance with the requirements of **Ch 1, 201. 5** (1) are to be as given in **Table 2.1.** however, where the language is not C, the rules/standards suitable for the language can be applied and the reference of the rules/standards should be specified.

Table 2.1 The criteria of static testing for embedded software

Name	Description
Defensive Programming	Prohibit to use the object which is not verified as a specific factor of specific function.
	Inspect the scope of constant value coming as a specific factor of specific function.
	When calling a function, check if the numbers of parameter are same.
	When calling a function, check if the numbers of parameter are same.
	Prohibit to assign constant which is out of the type size of variable.
	In Boolean type variables, prohibit to use values other than boolean type variables and the values of 0 and 1.
	Inspect a value assignment before using variables.
	Check if it verifies a divisor for avoiding division by zero.
	Prohibit explicit conversion for removing const or volatile.
	Check if a shift operator with value out of scope is used.
	Prohibit to use plain char type for the purpose other than using or saving character value.
	Prohibit to used signed and unsigned char types for the purpose other than using or saving numeric value.
	Prohibit to use a statement which the result is different depending on the assessment order(sequence point detection).
	Check if the parameter of function macro is enclosed with parenthesis(except that it is connected with # or ##)
	All macro identifiers in preprocessor directives shall be defined before use, except in #indef and #ifndef preprocessor directives and the defined() operator
	Prohibit to use the address of local variable to return statement
	Prohibit to assign an address of local variable to the variable having address which is beyond own scope.

Table 2.1 The criteria of static testing for embedded software (continued)

Name	Description
Defensive Programming	If a pointer type parameter of function prototype is not used to modify the object which the pointer directs to, the pointer shall be declared as const.
	Check all switch clauses having statement is ended by break statement.
	Check a switch statement has more than one case statement.
	The last clause of switch statement shall be default clause.
	Prohibit to use an expression which the operation of conditional expression has always the same result.
	Prohibit to use bitwise operators(&,|) in the conditional statement.
	If there is else if, check if there is else.
	Check if the bodies of switch, while, do-while, for and if statements are compound statement.
	Check if explicit return is existed in non-void return type function
Use of coding standard	Prohibit recursive call directly/indirectly.
	Prohibit to use exit function.
No dynamic variables or dynamic objects	Prohibit to assign dynamic memory.
Online checking during creation of dynamic variables or dynamic objects	Online inspection for installing dynamic variable or dynamic object.
Limited use of pointers	Prohibit to use a pointer which is not inspected by conditional expression.
Limited use of recursion	Prohibit recursive call directly/indirectly.
Structured programming	Prohibit to use goto statement.
	Check if initializer/loop-test/counting expressions of for statement are related to loop control.
Information hiding/encapsulation	Prohibit define a global variable to header file.
	Prohibit define a function to header file.
Modular approach	Check if a function has one exit point.
	Check if the parameters only related to function are declared.
	Prohibit to use longjmp function and setjmp macro.

2. The static testing in accordance with the requirements of **Ch 1, 201. 5** (1) are to be carried out the weakness check. the weakness check are to be done in accordance with the requirements in CWE(CWE-658 for C, CWE-659 for C++, CWE-660 for JAVA). however, where the language is not C or JAVA, the rules/standards suitable for the language can be applied and the reference of the rules/standards should be specified. and also, if there are no rules/standard for the weakness check, the weakness check is not performed.

3. The static testing in accordance with the requirements of **Ch 1, 201. 5** (1) are to be carried out the source code metric check as given in **Table 2.2.**

Table 2.2 The criteria of source code metric for embedded software

Kind of metric	Acceptance criteria	Note
Cyclomatic Complexity	Max. 20	
Number of Call Levels	Max. 6	Maximum nesting depth
Number of Function Parameters	Max. 8	
Number of Calling Functions	Max. 8	How many other functions are called for this function
Number of Called Functions	Max. 10	How many other functions does this function call?
Number of Executable Code Lines	Max. 200	

4. The dynamic testing in accordance with the requirements of **Ch 1, 201. 5** (1) are to be as given in **Table 2.3** and function requirement shall be carried out based on equivalence partitioning method, boundary value analysis method(refer to IEC 61508-3 Table b.3).

Table 2.3 The criteria of dynamic testing for embedded software

Criteria	Description	Acceptance criteria
Function requirement	How well does the system function meet the specified target as intended?	100%
Non-Function requirement	How well does the system response time meet the specified target?	100%
	How well does the turnaround time meet the specified targets?	100%
	How may users can access the system simultaneously at a certain time against the specified target?	100%

5. The dynamic testing in accordance with the requirements of **Ch 1, 201. 5** (1) are to be carried out the coverage check as given in **Table 2.4.**

Table 2.4 The criteria of dynamic testing for embedded software

Criteria	Description	Acceptance criteria
Code coverage	How much of the required test cases has been executed based on requirement definition during testing?	100%

204. Other requirements

1. Requirement definition

(1) The requirement shall be defined in a way that is clear, concise, clear, verifiable, testable, maintainable, and feasible.
(2) The requirement shall be free of terms and descriptions that are not understood by those who use the relevant documents
(3) In cases where a term used in a particular context could have multiple meanings, terms shall be included in a requirement definition as a glossary where its meaning is made more specific.
(4) The requirement shall include all requirements, whether relating to functionality, performance, design constraints, attributes, or external interfaces. In particular any external requirements imposed by a system specification should be acknowledged and treated.
(5) The requirement shall define responses of the software to all realizable input data in all realizable situations with the responses to both valid and invalid input values.
(6) In case of TBD(To be determined); following information shall be included.
  (A) A description of the conditions causing the TBD (e.g., why an answer is not known) so that the situation can be resolved;
  (B) A description of the customization timing and action to be taken.
(7) The requirement definition shall specify the logical characteristics of each interface between the software product and the hardware components of the system. this includes configuration characteristics (number of ports, instruction sets, etc.). it also covers such matters as what devices are to be supported, how they are to be supported, and protocols.
(8) For the external interfaces; at least following information is to be included;
  (A) Name of item
  (B) Description of purpose
  (C) Source of input or destination of output
  (D) Valid range, accuracy and/or tolerance
  (E) Units of measure
  (F) Timing
  (G) Relationships to other inputs/outputs
  (H) Screen formats/window formats
  (I) Data formats
  (J) Command formats
  (K) End messages
(9) For the external interfaces; at least following information is to be included;
  (A) Exact sequence of operations
  (B) Responses to abnormal situations(overflow, communication facilities, error handling and recovery)
(10) Effect of parameters
(11) Relationship of outputs to inputs
  (A) Input/output sequences
  (B) Formulas for input to output conversion
(12) For the performance requirements; at least following information is to be included;
  (A) The number of terminals to be supported
  (B) The number of simultaneous users to be supported
  (C) Amount and type of information to be handled
(13) For the database; at least following information is to be included;
  (A) Type of information used by various functions
  (B) Frequency of use
  (C) Accessing capabilities
  (D) Data entities and their relationships
  (E) Integrity constraints
  (F) Data retention requirements

Section 3 Application software

301. Application

The requirements of this Section apply to tests and inspection for the conformity assessment of standalone software installed on commercial OS for use in the marine environment.

302. Data to be submitted

The following reference data are to be submitted to the Society in addition to those specified in **Ch 1, 201.**

(1) Product description

(A) Product name, version, release date

(B) Name and address(postal or web) of the supplier and, if applicable, of the sellers, e-Commerce sellers or distributors

(C) The Intended work tasks and services that can be performed with the software

(D) License type

(E) Whether maintenance is offered or not. if offered, the product description shall describe the maintenance services offered

(F) Overview of end user callable functions of the product

(G) All known limitation that the user may encounter

(H) The ability of the software to continue operating (i.e. to be available) in the case of user interface errors, errors in the application's own logic, or errors due to availability of system or network resources

(I) Information on data saving and restoring procedures

(J) The type of user interface(command line, menu, window, function key)

(K) System configurations, resources needed for efficient working with the software, e.g, bandwidth, hard disk space, RAM, video card, wireless network card, CPU speed, etc.

(L) Information on maintenance for the user, e.g. logs, alert screens,

(M) The different configurations or supported configurations(hardware, software) for putting the software into use

(N) Information on the installation procedure

303. Conformity test

1. The static testing in accordance with the requirements of **Ch 1, 201. 5** (1) are to be as given in **Table 3.1.** however, where the language is not C, the rules/standards suitable for the language can be applied and the reference of the rules/standards should be specified.

Table 3.1 The criteria of static testing for application software

Name	Description
Defensive Programming	Prohibit to use the object which is not verified as a specific factor of specific function.
	Inspect the scope of constant value coming as a specific factor of specific function.
	When calling a function, check if the numbers of parameter are same.
	When calling a function, check if the numbers of parameter are same.
	Prohibit to assign constant which is out of the type size of variable.
	In Boolean type variables, prohibit to use values other than boolean type variables and the values of 0 and 1.
	Inspect a value assignment before using variables.
	Check if it verifies a divisor for avoiding division by zero.
	Prohibit explicit conversion for removing const or volatile.
	Check if a shift operator with value out of scope is used.
	Prohibit to use plain char type for the purpose other than using or saving character value.

Table 3.1 The criteria of static testing for application software (continued)

Name	Description
Defensive Programming	Prohibit to used signed and unsigned char types for the purpose other than using or saving numeric value.
	Prohibit to use a statement which the result is different depending on the assessment order(sequence point detection).
	Check if the parameter of function macro is enclosed with parenthesis(except that it is connected with # or ##)
	All macro identifiers in preprocessor directives shall be defined before use, except in #indef and #if ndef preprocessor directives and the defined operator
	Prohibit to use the address of local variable to return statement
	Prohibit to assign an address of local variable to the variable having address which is beyond own scope.
	If a pointer type parameter of function prototype is not used to modify the object which the pointer directs to, the pointer shall be declared as const.
	Check all switch clauses having statement is ended by break statement.
	Check a switch statement has more than one case statement.
	The last clause of switch statement shall be default clause.
	Prohibit to use an expression which the operation of conditional expression has always the same result.
	Prohibit to use bitwise operators(&,|) in the conditional statement.
	If there is else if, check if there is else.
	Check if the bodies of switch, while, do-while, for and if statements are compound statement.
	Check if explicit return is existed in non-void return type function
Use of coding standard	Prohibit recursive call directly/indirectly.
	Prohibit to use exit function.
No dynamic variables or dynamic objects	Prohibit to assign dynamic memory.
Online checking during creation of dynamic variables or dynamic objects	Online inspection for installing dynamic variable or dynamic object.
Limited use of pointers	Prohibit to use a pointer which is not inspected by conditional expression.
Limited use of recursion	Prohibit recursive call directly/indirectly.
Structured programming	Restrict function complexity(cyclomatic complexity number)
	Prohibit to use goto statement.
	Check if initializer/loop-test/counting expressions of for statement are related to loop control.
	Restrict the maximum nesting depth of function
Information hiding/encapsulation	Prohibit define a global variable to header file.
	Prohibit define a function to header file.
Modular approach	Restrict the line of code(LOC) for function.
	Check if a function has one exit point.
	Check if the parameters only related to function are declared.
	Prohibit to use longjmp function and setjmp macro.

4. The dynamic testing in accordance with the requirements of **Ch 1, 201. 5** (1) are to be as given in **Table 3.2.**

Table 3.2 The criteria of dynamic testing for application software

Criteria	Description	Acceptance criteria
Function requirement	How well does the software function meet the specified target as intended?	100%
Non-Function requirement	How well does the software response time, turnaround time, User access capacity, etc. meet the specified target?	100%

304. Other requirements

1. Product description

(1) The product description shall display a unique identification.
(2) The software shall be designated by its product identification(name, version, release date)
(3) The product description shall contain the name and address(postal or web) of the supplier and, if applicable, of the sellers, e-commerce sellers or distributors.
(4) The product description shall identify the intended work tasks and services that can be performed with the software.
(5) The product description shall identify the requirements documents when the supplier wants to claim conformity to documents defined by a law or by a regulatory body that affects the software
(6) The product description shall contain the license type
(7) If the product description documentation makes reference to known user callable interfaces to other software, these interfaces or software shall be identified.
(8) The product description documentation shall indicate where the software relies on specific software and/or hardware with appropriate references(name of software/hardware, version, specific operating system).
(9) The product description shall state whether maintenance is offered or not. if offered, the product description shall describe the maintenance services offered.
(10) The functionality in the product description shall include clear terms and criteria in order to avoid ambiguity. especially, the clear value shall be stated if there are performance efficiency, etc. non-function requirement. ⚓

Annex 1 Template for Test Plan

No. :

Date of issue :

Date of revision :

Test Plan

(Ver 1.1)

YY.MM.DD

System Name

History

Date	Version	Description	reference

Contents

Content of Table

Content of figure

1. General

1.1 System Purpose

1.2 System Scope

1.3 System configuration

Table 1 System information

Name	
Abbreviation	
Version	

1.3.1 System

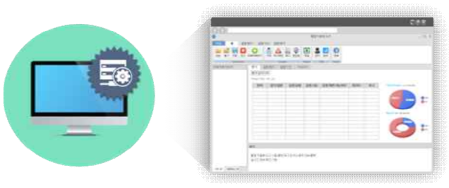

Figure 1 System configuration

2. Test

2.1 Name

2.2 Purpose

2.3 Scope

Section		Description
Static test		
Dynamic	Function	
	Non-function	

2.4 Method

Section	Description
Static test	
Function test	
Non-Function test	

3. Environment

3.1 Software configuration

		Name	Quantity	Purpose
A SW	OS			
	Compiler			
	Development tool			
B SW	OS			
	Compiler			
	Development tool			
...				

3.2 Hardware configuration

Name	Purpose	Quantity	Remark
PC	SW Operating environment	1	
...			

3.3 Test tool

Name	Purpose	Quantity	Remark
PC	SW Operating environment		
...			

3.4 Test drawing

Figure 2 Test environment

4. Test item

4.1 Test criteria

Section		Description	Criteria
Static test			
Dynamic	Function		
	Non-function		

4.2 Static test

4.2.1 Criteria of code rule

Code rule	Description	Remark
Rule 1		
...		

4.3 Dynamic test

4.3.1 Function test item

ID	Name	Description
FD-001		
...		

4.3.2 Non-function test item

ID	Name	Description
FD-001		
...		

Guidance for Computer-Based System Conformity Assessment

Published by
KR
36, Myeongji ocean city 9-ro, Gangseo-gu,
BUSAN, KOREA
TEL : +82 70 8799 7114
FAX : +82 70 8799 8999
Website : http://www.krs.co.kr