指导性文件
GUIDANCE NOTES
GD11-2015

中 国 船 级 社

船用软件安全及可靠性评估指南

GUIDELINES FOR SAFETY AND RELIABILITY ASSESSMENT FOR SHIPBOARD SOFTWARE

2015

目录

1 范围及说明.....1

2 规范性引用文件.....2

3 术语及缩略语.....2

4 计算机系统分类.....5

5 质量体系的要求.....6

6 系统生命周期.....8

7 软件开发生命周期.....14

8 试验和验证.....39

附录 1 测试和检验的验证表.....41
附录 2 小型低复杂度计算机系统的评估.....52
附录 3 计算机系统设计和实现阶段的技术建议.....54

1　范围及说明

1.1　本指南是对船用计算机系统（以下简称计算机系统，包括可编程电子系统）中软件的可靠性及安全性评估指南，对船用计算机系统中软件的开发、测试、认证、生产、维护提出了安全及可靠性的技术要求。本指南也对与软件相关的硬件制定了一些要求，这些要求需与产品的技术要求结合对待。本节适用安装于入级船舶上，提供符合入级要求的控制、报警、监测或安全功能的计算机系统，包括可编程电子系统。

1.2　本指南主要关注在软件开发生命周期，对整体安全生命周期中的一些环节也有所采用。本指南软件开发模型采用 V 模型，相关模型的演化并未包含在本指南中。

1.3　考虑到小型简单的计算机系统的直接应用和在复杂系统中对部分功能实现的应用，本指南定义了小型低复杂度计算机系统，并在附录 2 给出了简化的评估方法。

1.4　在应用本指南时，可根据制造厂内部文件管理系统编制本指南中提到的文档，但内容应符合指南中提及的相关内容。

1.5　附加标志

1.5.1　对于建立了系统生命周期的计算机系统，依照《钢质海船入级规范》等要求通过了系统和硬件的认证，并满足了本指南对软件的技术要求，根据其不同的系统等级（分类见 4.1 条），可授予下列附加标志：

（1）对于 I 类系统，SLC1；

（2）对于 II 类系统，SLC2；

（1）对于 III 类系统，SLC3。

1.6　本指南包括三个附录。其中：

1.6.1　附录 1 是现场验船师进行船用计算机系统中软件的安全及可靠性评估时使用的测试和检验的验证表。

1.6.2　附录 2 是小型低复杂度计算机系统的评估方法。

1.6.3　附录 3 是计算机系统设计和实现阶段的技术建议。

2　规范性引用文件

2.1　下列参考文件对于指南的应用是不可缺少的。凡是注日期的引用文件，仅引用版本适用。凡是不注日期的引用文件，其最新版本适用于本指南。

引用文件　　**表 2.1**

1.		中国船级社《钢质海船入级规范》（2012）及其修改通报
2.		SOLAS 公约第 II-1 章第 55 条
3.	GD01-2006	中国船级社 电气电子产品型式认可试验指南
4.	IACS UR E22	可编程电子系统船上使用和应用
5.	IEC 61508-1： 2010	电气/电子/可编程电子安全相关系统的功能安全 第 1 部分：一般要求
6.	IEC 61508-2： 2010	电气/电子/可编程电子安全相关系统的功能安全 第 2 部分：电气电子/可编程电子安全相关系统的要求
7.	IEC 61508-3： 2010	电气/电子/可编程电子安全相关系统的功能安全 第 3 部分：软件要求
8.	IEC 61508-4： 2010	电气/电子/可编程电子安全相关系统的功能安全 第 4 部分：定义和缩略语
9.	IEC 61508-5： 2010	电气/电子/可编程电子安全相关系统的功能安全 第 5 部分：确定安全完整性等级的方法示例
10.	IEC 61508-6： 2010	电气/电子/可编程电子安全相关系统的功能安全 第 6 部分：应用第 2 部分和第 3 部分的指南
11.	IEC 61508-7： 2010	电气/电子/可编程电子安全相关系统的功能安全 第 7 部分：技术和措施概述
12.	IEC 60092-504：2001	船舶电气设备 第 504 部分：专辑 控制和测量仪表
13.	IEC 60812-2006	系统可靠性分析技术-故障模式影响分析
14.	IEC 61025-2006	故障树分析
15.	IEEE 730-2014	软件质量保证计划
16.	ISO 9000-3	ISO9001 质量保证标准在计算机软件开发、供应、安装和维护中的应用指南
17.	ISO 17894-2005	船舶和海上技术 计算机应用 海上用可编程电子系统的开发和使用总则

3　术语及缩略语

3.1 术语

3.1.1　软件（software）

包括程序、规程、数据、规则以及相关的数据处理系统操作文档在内的智能创作。

3.1.2　计算机系统（computer system）

以计算机技术为基础，可以由硬件、软件及其输入和（或）输出单元构成的。

注：这个术语包括以一个或多个中央处理器（CPU）及相关的存储器等为基础的微电子装置。

3.1.3　系统（system）

根据设计相互作用的一组元素，可能包括相互作用的硬件、软件和人等。

3.1.4 子系统（Subsystem）

子系统是一种模型元素，它具有包（其中可包含其他模型元素）和类（其具有行为）的语义。子系统的行为由它所包含的类或其他子系统提供。子系统实现一个或多个接口，这些接口定义子系统可以执行的行为。

3.1.5　模块（Module）

程序、分立部件、封装程序的一个功能集、或一组归并在一起的分立部件。

3.1.6　软件模块（Software module）

由规程和（或）数据说明组成的构造，并能与其它这样的构造相互作用。

3.1.7　安全功能（Safety function）

针对特定的危险事件，为达到或保持 EUC 的安全状态，由计算机系统、其它技术安全相关系统或外部风险降低设施实现的功能

3.1.8　受控设备（Equipment under control（EUC））

用于制造、加工、运输或其它活动的设备、机器、器械和（或）成套装置。

3.1.9　小型低复杂度计算机系统(Small low complexity computer system)

一种计算机系统,其中：已很好确定了每个部件的失效模式；能完全确定在故障情况下的系统行为。

注：在故障状态下系统行为可用试验和（或）分析的方法确定。

3.1.10　动态测试（Dynamic testing）

用系统的和受控的方式执行软件和（或）操作硬件以证明所要求的行为的存在以及非要求行为的不存在。

3.1.11　质量计划（Quality plan）

针对特定的软件产品、软件项目所规定的质量措施、资源和活动顺序的文件。描述相关质量标准并且说明如何满足相应标准的一系列文件。

3.1.12　系统生命周期（System lifecycle）

系统实现过程中所必需的生命活动，这些活动发生在从一项工程的概念阶段开始，直至所有的计算机系统及其相关停止使用为止的一段时间内。

3.1.13　软件生命周期（Software lifecycle）

从软件开始构思到软件永久停用期间的活动。

注：一个典型的软件生命周期包括需求、开发、测试、集成、安装和修改等阶段。

3.1.14　软件配置管理（Software Configuration Management，SCM）

是一种标识、组织和控制修改的技术。软件配置管理应用于整个软件工程过程。

3.2　缩略语

3.2.1　ISO：International Organization for Standardization，国际标准化组织。

3.2.2 IEC：International Electrotechnical Commission，国际电工委员会。

3.2.3 IEEE：Institute of Electrical and Electronics Engineers，电气电子工程师学会。

3.2.4 FMEA：Failure Mode and Effects Analysis，故障模式影响分析。

3.2.5 FMECA：Failure Mode, Effects and Criticality Analysis，故障模式及影响分析和危害性分析。

3.2.6 FAT：Factory Acceptance Test，工厂验收试验。

3.2.7 PE：Programmable Electronic，可编程电子。

4 计算机系统分类

4.1 按照故障的影响并考虑系统提供的功能，应参照表 4.1 将计算机系统分为 3 类，分别是 I、II、III 类。分类过程中，应按最严酷环境进行影响分析。

计算机系统分类 **表 4.1**

类别	影响	系统功能
I	这些系统的故障不会对人员的安全、船舶的安全以及环境产生危害	——监视功能和日常管理功能
II	这些系统的故障最终会对人员的安全、船舶的安全以及环境产生危害	——监视和报警功能 ——对保持船舶处于正常运营和起居状况所必要的控制功能
III	这些系统的故障即刻会对人员的安全、船舶的安全以及环境产生危害	——保持船舶推进和操舵的控制功能 ——安全功能

4.2 在表 4.2 中，举例说明了一些计算机系统的分类供制造商参考。

系统分类举例 **表 4.2**

系统类别	举例
I	维修保养支持系统 日常信息处理
II	监视和报警装置 液柜容量测量设备 辅机控制系统

	主推进装置遥控系统 探火和灭火系统 舱底水系统 调速器
III	机械保护系统或设备 燃烧器控制系统 内燃机的电子喷油器 推进和操舵控制系统 发电机同步单元

5 质量体系的要求

5.1 质量保证体系

5.1.1 应通过质量保证体系证明制造厂具有一定的产品质量保证能力和质量管理水平，以及制造厂制定了能确保产品符合本社规范及相关公约的管理制度。

5.1.2 制造厂应建立并实施 ISO9001 或等效标准的质量管理体系并持有有效证书。按照 ISO9001 的要求，制造厂应对其管理活动、资源提供、软件产品实现和测量、分析和改进有关的过程进行控制。

5.1.3 II、III 类系统的制造厂还应满足 ISO9000-3 的适用要求。

5.2 软件质量计划

5.2.1 制造厂应制定针对软件开发生命周期的质量计划。

5.2.2 软件质量计划应规范该软件整个生命周期的活动，明确相关程序、职责和系统文件，包括配置管理。所制定的质量计划可参照 IEEE 730 的要求。

5.2.3 对于 II、III 类系统的软件，质量计划中应包含安全功能要求部分，应设计具体保证方法，以验证和确认安全功能要求是否得以满足。

5.2.4 在软件开发生命周期阶段中，应制定船用计算机系统的配置管理，详见 5.6 条。

5.3 生产中质量控制

5.3.1 通过切实可行的质量保证措施、计划和组织，确保产品的质量。

5.3.2　制造厂应具有针对产品的质量控制文件，该质量控制文件应准确描述产品的生产工艺流程，并用文字以及图表清晰描述各工艺流程的质量控制要求；还应包含明确的控制对象、控制标准、控制方法及检验方法和生产质量保证措施落实的证明文件。对于安全相关功能的产品，还要求提供通过“试验和模拟”的证明文件。

5.4　最终的试验报告

5.4.1　制造厂应对产品进行最终测试并提供报告。最终的试验报告是根据成品试验和试验结果记录生成的报告。

5.5　软件可追溯性

5.5.1　软件可追溯性要求：必须按程序对编程内容和数据的修改以及版本的变化进行标识并文档化。确保在需要时对软件产品质量形成过程实行可追溯。通过软件配置管理及软件版本说明等质量保证文件，明确编程内容、数据的修改以及版本的变化所必须遵循的流程，并确定在文件中记录这些修改或变化。

5.5.2　这些文件至少应保存至软件开发生命周期结束后一年。制造厂应有明确证据证明该软件确实退役。

5.6　配置管理

5.6.1　配置管理的目的是为了保证当某些可交付项有改变时，几种开发的可交付项的一致性。一般来讲，配置管理包括硬件配置管理和软件配置管理。

5.6.2　要求：

(1)　在软件开发生命周期阶段中应使用行政和技术控制，以管理软件变化和保证有关软件安全的规定要求始终能得到满足。

(2)　应确保所有必需的操作已被执行以说明获得了所要求的软件需求。

(3)　应保持精确的和维护计算机系统完整所必需的所有配置项的唯一识别。配置项至少包括：安全分析和要求；软件规范和设计文档；软件源代码模块；应用于计算机系统软件组件和软件包的测试计划和测试结果；所有用于创建、测试或执行计算机系统软件的工具和开发环境。

(4) 应采用变更控制规程来防止非授权的修改，修改请求应文档化；分析建议修改的影响以批准或拒绝请求；对所有准许修改的细节和授权应文档化；确保所有软件基线的构成（包括早期基线的重建）。

(5) 应对配置状态、发布状态、所有修改的判断和通过、修改的细节等信息文档化以便接受审核。

(6) 软件的发布应正式文档化。软件的主要备份和所有有关文档在已发布软件开发生命周期内应被保存以维护和修改。

6 系统生命周期

6.1 系统生命周期的划分

6.1.1 系统生命周期划分为五个阶段，分别为概念、需求、实现、验证、运行。每个阶段根据目的范围的不同又做了划分，其关系和要求见下表和图：

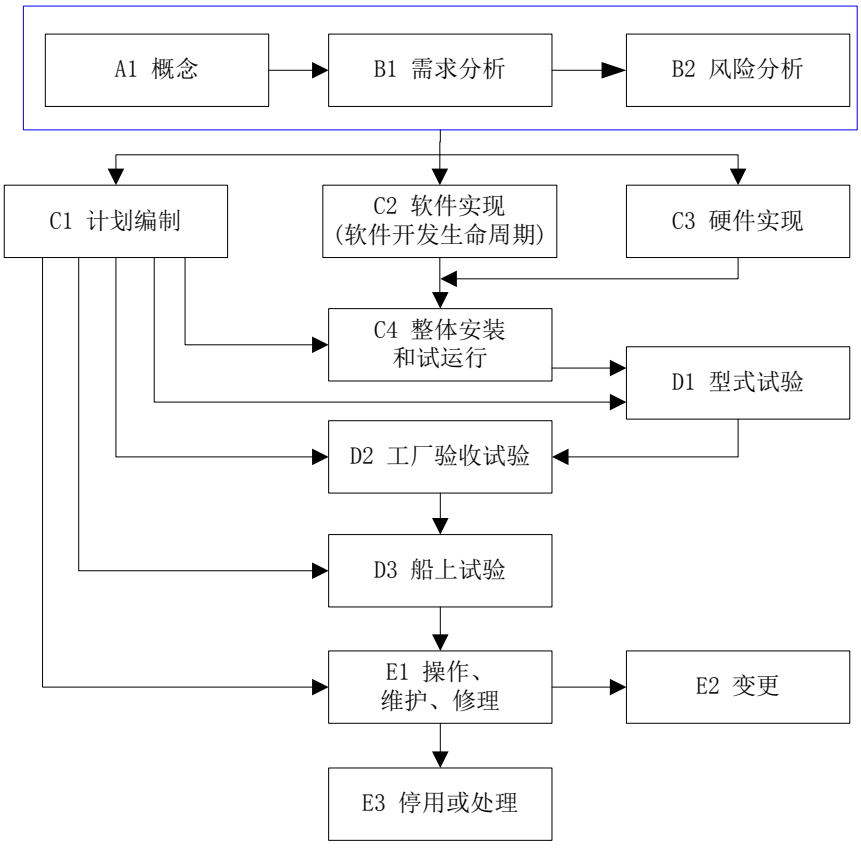


图 6.1.1 系统生命周期

系统生命周期概述 **表 6.1.1**

系统生命周期阶段		目的	要求	输入	输出
图 6.1.1 的方框号	标题				
A 概念					
A1	概念	提高对受控设备及其环境（实际的、法律的等）的理解水平，以满足执行其生命周期活动的需要。	对受控设备及其要求的控制功能和实际环境进行全面的了解；确定可能的危险源；获取确定危险的有关信息；获取当前的安全法规；考虑相邻近的受控设备之间相互作用所产生的危险；以上所要求的信息和结果应文档化。	满足该条要求所必需的所有有关信息	从概念至整体范围获取的信息
B 需求					
B1	需求分析	在新建或修改计算机系统时描写新系统的目的、范围、定义和功能时所要做的所有的工作。包括： 确定计算机系统的边界； 规定风险分析的范围。		从概念至整体范围获取的信息	计算机系统要求规范
B2	风险分析（仅针对安全相关功能）	为了保证计算机系统的安全可靠性，证明对于单一故障，系统进入故障安全状态，并且运行中的系统不会丢失或降低到不能满足船级社规定的可接受性能标准。	应采取适当的方法进行分析，如： ——故障树分析； ——风险分析； ——FMEA 或 FMECA 分析	计算机系统要求规范	安全相关功能要求规范（含安全要求分配的信息和记录）
C 实现					
C1	计划编制。	在规定规程上和技术上步骤，证明计算机系统满足安装、操作和维	拟定计算机系统的安装、操作和维护计划，以确保在操	计算机系统要求规范； 安全相关功	软件质量计划； 计算机系

		护要求。	作和维护过程中保持所要求的功能安全。 拟定 FAT 试验大纲和船上试验大纲，大纲中需包括安全相关功能试验。	能要求规范。	统安装、操作和维护计划； 型式试验大纲； FAT 试验大纲； 船上试验大纲。
C2	软件实现	建立符合计算机系统要求规范和安全相关功能要求规范的计算机系统软件。	详见第 7 章软件开发生命周期。	计算机系统要求规范	每个计算机系统满足计算机系统要求规范的证实。详见第 7 章软件开发生命周期。
C3	硬件实现	建立符合计算机系统要求规范和安全相关功能要求规范的计算机系统硬件。		计算机系统要求规范。	每个计算机系统满足计算机系统要求规范的证实。
C4	整体安装和试运行	安装计算机系统； 试运行计算机系统。		计算机系统安装、操作和维护计划	已安装就绪的计算机系统； 经充分试运行过的计算机系统。
D 验收					
D1	型式试验	确认计算机系统满足计算机系统要求规范（包含安全相关功能）； 确认系统满足 GD01-2006 的要求。	按 GD01-2006 进行试验。	计算机系统要求规范； 安全相关功能要求规范； 型式试验大纲。	确认计算机系统满足安全相关功能要求的证实。 型式试验报告。
D2	FAT 试验	对计算机系统在工厂进行测试	FAT 报告应记录①使用的工具和设备；②FAT 活动的	计算机系统要求规范； FAT 试验大	FAT 试验报告。

			记录；③实际结果和预期结果的差异及处理。 当预期结果和实际结果出现差异时，应经分析和评估确定是继续测试还是提出变更请求。	纲。	
D3	船上试验	通过船上试验，验证所有系统互连后，系统执行预定功能的能力。船上试验 ，包括完全系统试验和集成试验。	完全系统试验应验证在实际硬件部件及最终应用软件的条件下，功能可以正常实现。 集成试验应验证所有系统集成状态下的功能可以正常实现。	计算机系统要求规范； 船上试验大纲。	船上试验报告。
E 运行					
E1	操作、维护、修理	为保持要求的功能安全，操作、维护和修理计算机系统。		计算机系统安装、操作和维护计划	可持续满足计算机系统所需的功能； 按时间排序的计算机系统的操作修理和维护文档。
E2	变更	在变更阶段中及阶段后保证计算机系统受控。	变更应返回生命周期合适阶段，并加以验证。记录应文档化。 制造厂应对修改进行记录。对II、III系统的软件和硬件进行的后续重大修改应提交给船级社进行批准。 应提前告知对已批准系统的修改和变更方案，并	计算机系统要求规范 软件质量计划 相应阶段的试验大纲	在变更阶段中及阶段后，均可达到计算机系统要求的功能安全； 按时间排序的计算机系统的操作、修理和维护文档。

			进行影响分析，同时修改应获得船级社批准。 修改后的软件应进行变更验证以证明满足相关计算机系统要求。 注：重大修改指影响船舶安全行驶和/或安全的修改。		
E3	停用或处理	在受控设备的停用及处理活动中及活动后，保证计算机系统的功能安全适应这种情况。	在进行停用或处理活动之前应进行影响分析，并制定一个计划，包括系统的关闭、系统的拆除。在计算机系统使用说明书中提示对敏感信息的销毁和处理。	根据功能安全管理规程对停用或处理的请求	

6.1.2　在应用本指南时，可根据制造厂内部文件管理系统编制本指南中提到的文档，但内容应符合指南中提及的相关内容。

6.1.3　下面将对系统生命周期中上表中需补充的内容单独列出。

6.1.4　软件开发生命周期将在第 7 章详述。

6.2　维护阶段的软件安全

6.2.1　目的

在维护阶段达到软件安全的要求。

6.2.2　要求

(1)　程序和数据的修改，以及版本的改变，要被记录并提交我社批准。

(2)　保证软件开发生命周期活动的相应责任部门的规程能胜任其活动，尤其应满足以下要求：

①　对工作人员进行针对诊断和修复故障以及系统测试的培训；

② 操作人员的培训；

③ 对工作人员进行定期再培训。

(3) 与软件开发生命周期的任何活动有关的人员的培训、经验和资格都应文档化。

(4) 保证危险事故（或产生危险的潜在事故）分析，以及提出使其重复发生的概率降到最低的建议的规程。

(5) 对操作和维护性能进行分析的规程，尤其是：

① 识别危及功能安全的系统故障的规程，包括用于检测重复性故障的日常维护所使用的规程；

② 评估需求率和在操作和维护期间的失效率是否和系统设计期间的假设一致。

(6) 启动对安全相关系统进行修改的规程。

(7) 进行修改所需要的批准规程和主管部门。

(8) 保持潜在危险和安全相关系统信息准确的规程。

(9) 在软件开发生命周期中，计算机系统的配置管理，尤其要对以下各项进行规定：

① 实现配置控制的规程；

② 用于对一个配置管理项（硬件和软件）的全部要素进行唯一标识的规程；

③ 防止未授权项进入服务的规程。

(10) 在适当场合的培训条款和应急服务信息

(11) 操作者应建立接收，记录，解决，跟踪问题和修改请求的程序。问题应按问题处理程序处理。应建立和保持系统操作计划，包括识

别配置项，操作规程和预期的维护活动。计划应包括软件迁移和退休问题。

(12) 系统和/或组件版本的每一个新的升级、发布或修改，操作者应进行测试。并且，发布操作使用的组件应满足指定的标准。如果发布部件的接口已被修改，测试应包括集成测试。

(13) 应定期进行配置审核，来验证操作配置的完整性。

6.2.3 输入

(1) 计算机系统使用说明书

(2) 软件质量计划

(3) 风险分析及预防规程

(4) 在适当场合的培训条款和应急服务信息

6.2.4 输出

程序和数据的修改及版本变更的记录

7 软件开发生命周期

7.1 制造厂应制定针对软件开发生命周期的质量计划。应在软件的生命周期中使用行政和技术手段加以控制，以便于管理软件变化和保证有关软件安全方面的要求得到满足，证明制造厂存在有效的、能够满足软件开发生命周期的各阶段的质量控制程序。

7.2 软件质量计划应包含以下内容：

7.2.1 软件开发生命周期应满足第 5.2 条的要求。

7.2.2 在软件开发生命周期阶段中，船用计算机系统的配置管理，特别是在：

(1) 对于特定的阶段，即将执行正式的配置控制节点；

(2) 用来唯一识别某项（硬件和软件）所有构成部分的规程；

(3) 阻止非授权项进入服务的规程。

7.2.3 软件开发生命周期的划分及关系请见下图和表。

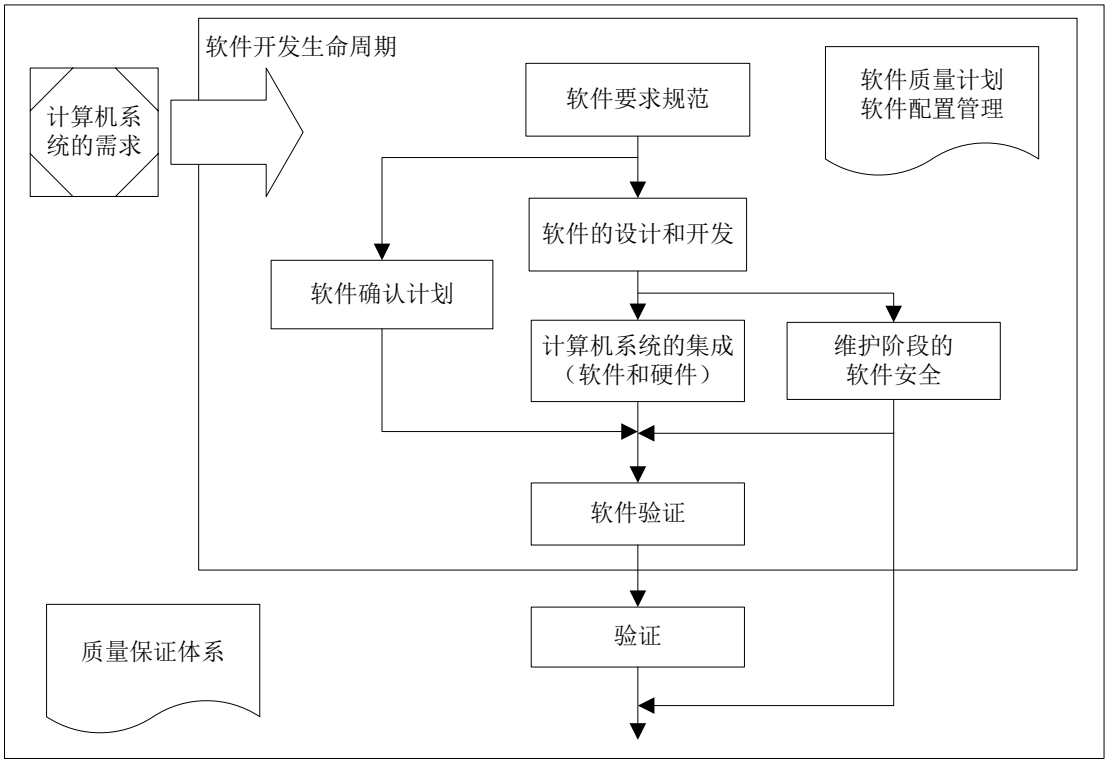


图 7.2.3-1 软件开发生命周期

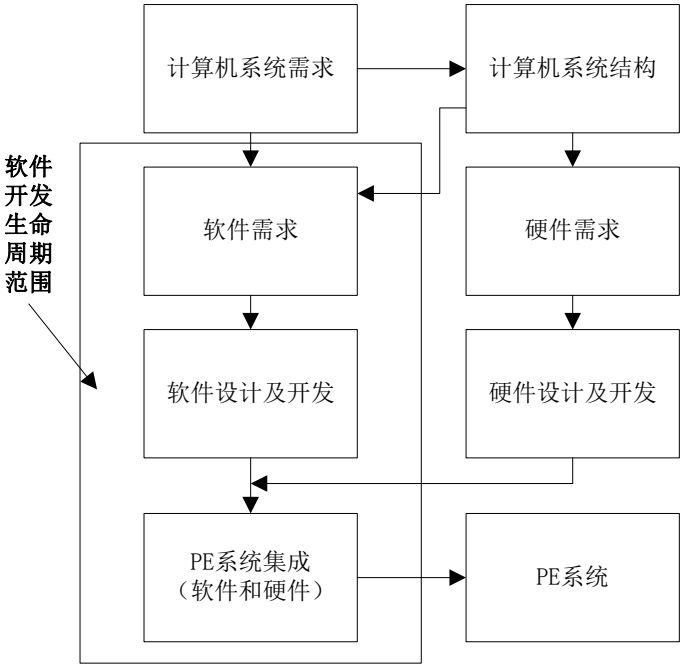


图 7.2.3-2　软件开发生命周期的范围及外部关系

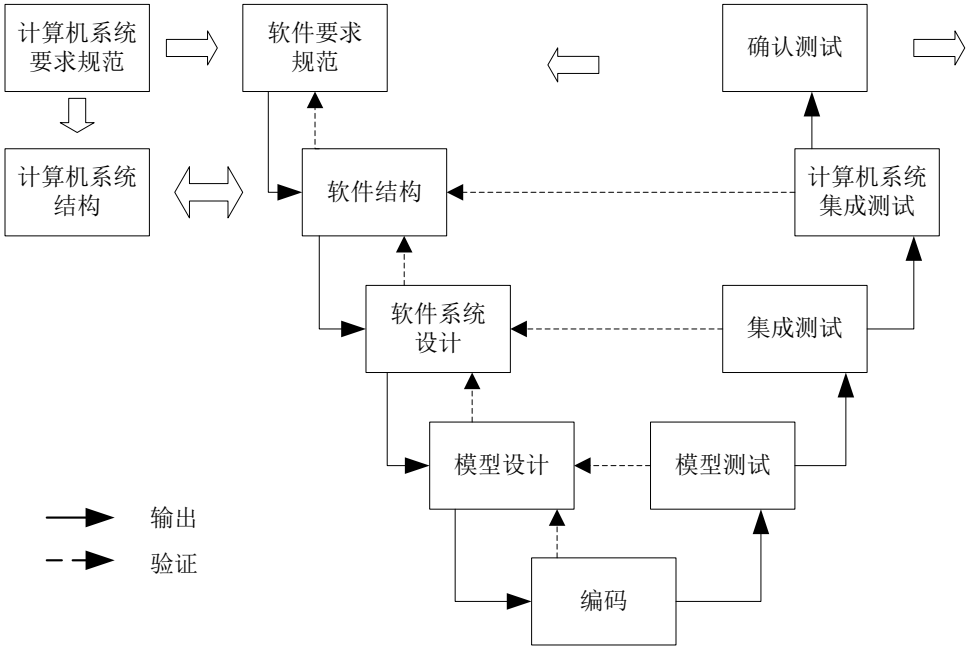


图 7.2.3-3　软件开发生命周期模型（V 模型）

注：除 V 模型外，本指南亦接受其他经本社同意的软件开发生命周期模型。

7.3　软件要求规范

7.3.1　目的

(1) 根据系统功能要求规定软件要求规范；

(2) 对每个需实现一定安全功能的计算机系统规定软件安全功能的要求；

(3) 规定每一个计算机系统对于软件集成的要求。

7.3.2 要求

(1) 软件的开发人员应复审 7.3.1 中的信息以确保对要求全面规定，应特别考虑以下环节：

① 安全功能；

② 系统配置或构成；

③ 硬件要求；

④ 软件要求；

⑤ 能力和响应时间性能；

⑥ 设备和操作人员界面。

(2) 在要求的系统类别等级范围内，软件安全的规定要求应得到表达和组织，以使其：

① 清楚、准确、不含糊、可验证、可测量、可维护、可行；

② 可回溯到计算机系统的安全要求的规定；

③ 不使用不明确的或在软件开发生命周期任一阶段使用这些文档的人所不能理解的术语和描述。

(3) 如果没有详细定义计算机系统的特殊安全要求，所有 EUC 的有关操作模式应在软件安全的特殊要求中详细说明。

(4) 软件要求规范应对软件和硬件间的任何与安全有关的或相应的约束进行规范并文档化。

(5) 在计算机系统硬件结构设计规范的范围内，软件要求规范应考虑如下内容：

① 软件自监视；

② 可编程电子硬件、传感器和执行器的监视；

③ 在系统运行时对安全功能进行的周期测试；

④ EUC 可操作时，能够对安全功能进行测试。

(6) 软件要求规范应将计算机系统的非安全功能和安全功能清晰区分。

(7) 软件要求规范将表示出产品要求的安全属性，但不是工程项目的安全属性。

7.3.3 输入

计算机系统要求规范

7.3.4 输出

软件要求规范

7.4 软件确认计划

7.4.1 目的

根据软件要求规范编制软件确认计划。

7.4.2 编制要求

(1) 软件确认计划应考虑：

① 确认时的细节问题。

② 执行确认的人员的细节问题。

③ EUC 操作的有关模式的识别应包括：

——使用的准备，包括设置和调整；

——启动、教学、自动化、手动、半自动化、操作的稳定状态；

——重置、关机、维护；

——合理的可预见异常条件。

④ 在开始试运行前，需要确认 EUC 操作的每一模式软件的识别。

⑤ 确认的技术路线（如分析方法、统计测试等） 。

⑥ 用于确定符合软件功能规定要求的每一功能的措施和规程。

⑦ 软件要求规范要求的特殊参考。

⑧ 确认活动所需要的环境（如测试将包括调校工具和设备）。

⑨ 通过/失败准则。

⑩ 评价确认记录，特别是评价失效的方针和规程。

(2) 确认软件的技术战略应包括下列信息：

① 手动或自动技术选一或选二；

② 动态或静态技术选一或选二；

③ 分析或统计技术选一或选二。

(3) 完成软件确认的通过/失败准则应包指：

① 要求的输入信号及其次序和值；

② 预期的输出信号及其次序和值；

③ 其他可接受的准则，如内存使用、定时、值的允许偏差。

7.4.3 输入

软件要求规范

7.4.4　输出

软件确认计划。

7.5　软件的设计与开发

这部分应描述软件开发生命周期中软件设计与开发活动。

7.5.1　软件结构和编码语言要求

(1)　目的

①　软件结构：

创建软件结构以满足不同系统等级对软件安全规定要求。

复审和评价计算机系统硬件对软件的要求，包括计算机系统中软件和硬件相互作用对 EUC 安全性的影响。

②　编码语言要求：

在用于辅助验证、确认、评价和修改软件的整个生命周期中，根据要求的系统等级选择合适的集成工具，包括语言和编译器。

(2)　软件结构的要求

软件结构是定义软件主要组件和子系统，包括它们如何实现内部连接，如何获得所要求的属性，特别是安全完整性。主要软件组件包括操作系统、数据库、大型设备输入/输出子系统、通信子系统、应用程序、编程和诊断工具等。

计划的软件结构设计将由软件供方和/或开发人员来建立，软件结构设计的描述应详细，描述将包括：

①　在所需的软件开发生命周期中，按不同等级的系统，选择和判断满足软件要求规范的集成技术和措施集。这些技术和措施包

括故障允许偏差（与硬件一致）和故障避免的软件设计策略，（适用时）冗余和多样性。

② 根据组件/子系统的划分，对每一部分应提供以下信息：

——它们是否是新的、已存在的或专利的；

——它们是否已被验证，如果是，它们的验证条件；

——每一个组件/子系统是否与安全有关；

③ 确定所有软件/硬件相互作用和评价及细化它们的重要性。

④ 使用符号表示法表示清楚定义的或限制清楚定义特性的结构。

⑤ 选择用于保持所有数据安全完整性的设计特征。这种数据可包含大型设备输入/输出数据、通信数据、操作界面数据、维护数据和内部数据库数据。

⑥ 规定适当的软件结构集成测试来保证软件结构满足规定系统等级上的软件安全要求。

(3) 支持工具和编程语言的要求

① 对于使用有限可变语言的用户应用程序编程，在一个低安全完整性等级下，要求的工具和编程语言可被限定为标准 PLC 语言、编辑器、加载器。其符合性的责任主要由供方承担。

② 在较高等级的系统上，需限制 PLC 语言的子集，验证和确认工具如代码分析器和仿真器等。在这些环境下责任由供方和用户共同承担。

③ 即便是在低等级的系统上，也成广泛使用完全可变性语言的嵌入应用工具，符合性的责任主要由软件开发人员来承担。这包括使用完全可变语言为用户应用程序编程提供低可变语言的 PLC 供方。

④ 根据软件开发的固有特性，确保以下(a)~(d)要求的符合性要求的责任由供方或用户单独承担，或由两者共同承担，责任的划分应在安全技术编制中文档化。

(a) 一套合适的集成工具，包括语言、编译器、配置管理工具、应用时的自动测试工具，应根据要求的系统等级选择。应考虑在计算机系统整个生命周期中提供相应服务的合适的开发工具（不是那些在系统开发的初始阶段期间使用的）的可用性。

在安全完整性等级要求的范围内，程序编程语言选择应：

具备有国家标准或国际标准认可的确认证书的翻译器/编译器，或对其目的的适宜性建立评估；

完全并清楚地定义或限制清楚定义特性；

与应用的特征匹配；

包括方便探测程序错误的特性；

支持与设计方法匹配的特性。

(b) 当不能满足①时，软件结构设计规范中应记录另一种可选择语言的理由，理由应足够详细说明语言目的的适宜性和任何说明语言缺点的附加措施。

(c) 编码标准应：

由评估方复审其与使用目的是否合适；

用于开发所有安全软件。

(d) 编码标准应规定好的编程习惯，描述非安全语言特性（如未定义的语言特性、非结构化设计等）和规定源代码文档规程。源代码文档中至少应包括下列信息：

法律实体（如公司、作者等）；

描述；

输入和输出；

配置管理历史。

(4) 输入

① 软件要求规范

② 计算机系统硬件结构设计

(5) 输出

① 软件结构设计规范

② 开发工具和编码标准

③ 开发工具的选择

④ 软件结构集成测试规范

⑤ 计算机系统集成测试规范

7.5.2 详细设计与开发

(1) 目的

设计软件，以满足不同的系统等级对软件的要求，这种软件可分析、验证并能被安全地修改。

详细设计与开发包括软件系统设计和单个软件模块设计。

(2) 要求

① 详细设计是指软件系统设计——结构中的主要组件划分在软件模块、单独的软件模块设计和编码系统（如装在每个硬件单元的基本软件、安装在网络节点中的通信软件、应用软件）中。

② 软件的详细设计与开发需对每一软件组成提供逻辑语言，并产生详细设计文件以定义内部结构和组成部分的界面，包括每一组成部分单元的测试部分。

③ 软件的开发应具有模块化、可测试性、安全修改能力。

④ 对于软件结构设计规范中的每一个主要组件/子系统，设计的进一步优化应根据软件模块的划分。每一软件模块的设计和测试应适用于规定的每一软件模块。

⑤ 需提供软件系统设计规范和单个模块设计规范。

⑥ 描述文档中应包括的内容如：

(a) 装在每个硬件单元的基本软件的描述；

(b) 安装在网络节点中的通信软件的描述；

(c) 应用软件（不是程序清单）的描述；

(d) 用于系统设置和设备配置的工具；

(e) 说明功能、性能、模块和其他部件之间的相互约束和依赖关系。

对应用软件的描述，应满足以下要求：

(a) 系统的模块（必须工作以维持功能）的信息，包括对其他系统的依赖关系；

(b) 每个模块的细节应达到足以了解其功能的水平；

(c) 软件模块（必须执行以保持各个功能）之间的关系；

(d) 软件模块之间的数据流和控制流；

(e) 软件的配置，其中包括优先级策略；

(f) 冗余系统的切换机制（若有）；

(g) 软件自我监控（例如，包括应用驱动的看门狗和数据范围验证）；

(h) 验证测试和外部设备诊断测试（例如，传感器和终端元件）；

(i) 对非预期的过程变量，如传感器值超出范围、检测开路、检测短路，应采取措施。

(3) 输入

① 软件结构设计规范

② 支持工具和编码标准

(4) 输出

① 软件系统设计规范

② 软件系统集成测试描述

③ 软件模块设计规范

④ 软件模块测试规范

7.5.3 代码实现

(1) 目的

编制单个软件模块，在验证、确认、评价和修改软件的整个生命周期中，利用合适的工具集包括语言和编译器来设计和实现软件。

(2) 要求

源代码应是：

① 可读、可理解和可测试的；

② 满足软件模块设计的规定要求；

③ 满足编码标准的规定要求；

④ 满足安全计划中规定的相关要求。

每一软件代码应复审，以检查代码编写和它的记录是符合详细设计文件的描述。

(3) 输入

① 软件模块设计规范

② 支持工具和编码标准

(4) 输出

① 源代码清单

② 代码复审记录

7.5.4 软件模块测试

(1) 目的

测试软件模块是一种验证活动，是代码复审和软件模块测试的结合，用以证明软件模块满足它的相关要求，即已验证。

(2) 要求

① 每一软件模块在软件设计中都应根据规定进行测试。

这些测试表明每一软件模块执行其预定功能且不执行其非预定功能。

② 软件模块测试的记录应文档化。II 类、III 类船用设备的软件模块测试记录的文档记录的内容应包括但不限于软件模块测试计划、软件模块测试用例、软件模块测试记录、测试记录分析报告、软件模块测试问题报告单和测试总结报告。

③ 应规定测试失效的纠正措施规程。

④ 制造厂应采用测试方法对软件模块的逻辑及需求进行全面的模块测试。

⑤ 制造厂应对 II 类、III 类系统的每一软件模块均根据软件模块测试规范的要求进行验证，该测试规范是在软件系统设计阶段制定的。

⑥ 制造厂可采用白盒测试的方法执行模块测试，根据边界值分析、错误推测、等价类或输入划分等方法设计测试用例。可根据软件的安全等级要求及船用可编程设备的特性要求选择上述方法。

(3) 输入

① 软件模块测试规范

② 源代码清单

③ 代码复审记录

(4) 输出

① 软件模块测试记录

② 验证和测试软件模块

7.5.5 软件集成测试

(1) 目的

软件集成测试是软件组件到软件单元的集成，目的是逐步收集软件组件，检查与初步和详细设计的整合。证明所有软件模块、组件和子系统相互正确作用来实现其预定的功能，不实现非预定的功能。

(2) 要求

① 软件集成测试应在设计和开发阶段正确规定。

② 软件集成测试一般包括：软件子系统测试和软件系统测试。

③ 软件集成测试应规定以下内容：

(a) 管理集成集中的软件划分；

(b) 测试用例和测试数据

(c) 执行测试的种类

(d) 测试环境、工具、配置和程序；

(e) 测试完成的准则应判断；并且

(f) 测试失效校正动作的规模。

④ 软件集成根据规定的软件集成测试要求进行测试。这些测试应表明所有软件模块和软件组件/子系统内部正确作用以执行其预定的功能而不执行非预定的功能。

⑤ 软件集成测试的记录应文档化，说明测试结果是否满足目的和测试准则。如果出现失效，记录失效原因。

⑥ 在软件集成过程中，应对软件的任何修改或改变进行影响分析以确定对所有软件模块的影响和所需要的再验正和再设计活动。

(3) 软件子系统测试附加要求：

① 建议采用黑盒测试执行子系统测试，采用动态测试、等价类和输入划分测试，包括边界值分析等方法设计黑盒测试用例。可根据软件的等级要求及船用可编程设备的特性选择上述方法。

② 对 II 类、III 类系统，应执行子系统测试，并分析测试结果以验证软件模块是否正确地集成。确认测试结果可追溯到测试计划文档中由测试可追溯性建立的测试准则。

(4) 软件系统测试附加要求：

① 应保证子系统满足 II、III 类上的要求。

② 应根据船用可编程设备的软件方面系统测试计划的规定进行系统测试活动。

③ 在系统测试中应考虑下列属性：

——有关软件设计规范的确认的完整性；

——有关软件设计规范（成功完成）的确认的正确性；

——可重复性；

——精确定义的确认配置。

④ II、III 类系统的系统测试，应验证防修改保护功能：

(a) 防止用户修改程序；

(b) 防止用户修改程序的运行参数。

⑤ II、III 类系统的系统测试，应验证单一故障条件下系统会安全失效的功能。

⑥ 应采用黑盒测试执行系统测试，采用等价类或过程仿真方法设计黑盒测试用例。可根据安全等级要求和船用可编程设备要求选择上述方法。

⑦ 当预期结果和实际结果出现差异时，应经分析和评估确定是继续测试还是提出变更请求，若提出变更请求，则应返回开发生命周期较早阶段。这些决定都应作为系统试验的确认结果文档化。

(5) 输入

软件系统集成测试规范（软件子系统/系统测试）

(6) 输出

① 软件系统集成测试记录

② 验证和测试软件系统

7.6 计算机系统集成（硬件和软件）

7.6.1 目的

(1) 在目标计算机系统硬件上集成软件。

(2) 将软件和硬件结合到计算机系统上以保证其兼容性和满足预定的要求。

7.6.2 集成测试要求

(1) 应在设计和开发阶段中规定集成测试.以保证计算机系统中硬件和软件的兼容性。

(2) 计算机系统（硬件和软件）的集成测试应规定：

① 根据集成水平拆分系统；

② 测试用例和测试数据；

③ 执行测试的种类；

④ 测试环境包括工具、支持软件和配置描述；

⑤ 判定测试完成的准则。

(3) 在进行计算机系统（硬件和软件）规定的集成测试时，应区别开发人员按自己的意图所执行的活动和从用户立场出发所进行的活动。

(4) 对计算机系统（硬件和软件）规定的集成测试应在下列行为中进行区分：

① 将软件系统纳入目标可编程电子硬件；

② 计算机系统集成，即增加接口如传感器和执行器；

③ EUC 和计算机系统的全部集成。

(5) 软件根据可编程电子（硬件和软件）规定的集成测试和与安全有关的可编程电子硬件进行集成。

(6) 在安全有关可编程电子（硬件和软件）集成测试中，应对集成系统的任何修改或改变进行影响分析，以确定对所有软件模块的影响和所需要的再验证活动。

(7) 测试用例及其结果应记录用于随后的分析。

(8) 安全有关可编程电子（硬件和软件）的集成测试应文档化，说明测试结果是否满足测试目的和测试准则。如果出现失效应记录失效原因。软件的任何修改或改变应进行影响分析以确定对所有软件组件/模型的影响和所需要的再验证和再设计活动。

(9) 对于 II 类系统，制造厂应保留并按需求提交集成测试的证明文件，文件包括测试计划和测试报告。对于 III 类系统，船级社应见证集成测试。

7.6.3 故障模拟测试要求

(1) 制造厂应根据系统设计规范制订系统故障模拟测试规范。应尽可能真实地进行故障模拟，以证明有适当的系统故障检测和系统响应。对任何所需的故障分析记录应进行观察。

(2) 故障模拟测试规范包括以下内容：

① 故障组件或元器件名称；

② 故障类型；

③ 插入故障方式；

④ 要求的系统响应（输出记录）。

(3) 故障模拟的测试用例及其预期结果应文档化。说明故障模拟的测试结果以及是否满足测试目的和测试准则。如果出现失败结果，应将失败原因文档化。

7.6.4　输入

(1)　软件结构集成测试规范

(2)　计算机系统测试规范（含故障模拟测试规范）

(3)　计算机系统硬件

7.6.5　输出

(1)　软件结构集成测试记录

(2)　计算机系统集成测试记录

(3)　验证和测试计算机系统

7.7　变更管理

7.7.1　目的

提供软件有关的信息和规程以保持操作和修改阶段中计算机系统的安全

7.7.2　要求

(1)　制造厂应对修改进行记录。对 III 系统的软件进行的后续重大修改应提交给船级社进行批准。

(2)　应提前告知对已批准系统的修改和变更方案，并进行影响分析，同时修改应获得船级社批准。

(3)　修改后的软件应进行变更验证并使船级社满意。

注：重大修改指影响船舶安全行驶和/或安全的修改。

7.7.3　输入

6.1～6.2 中所有输入输出文件

7.7.4　输出

软件操作和修改规程

7.8　软件验证

7.8.1　目的

达到所需的软件系统等级，测试和评价给定软件开发生命周期阶段的输出，以保证当输入该阶段时提供的输出与标准的正确性和一致性。

7.8.2　要求

(1) 软件验证应与开发做好同步计划，对软件开发生命周期的每一个阶段，信息应文档化。

(2) 软件验证计划编制应参考确认活动中使用的准则，技术和工具，并应注明：

① 安全完整性要求的评价；

② 验证战略、活动和技术的选择和文档；

③ 验证工具的选择和使用（测试工具、专业测试软件、输入/输出仿真器等）；

④ 验证记录的评价；

⑤ 采用的校正动作。

(3) 软件验证应根据计划执行。

(4) 表明被验证的阶段已在所有方面圆满完成的文档化证据。

(5) 每次验证后，验证文档应包括：

① 被验证的项目识别；

② 对应验证完成的信息识别；

③ 非符合性（如：软件模块、数据结构和不常采用的算法）。

(6) 软件开发生命周期 N 阶段中所有 N+1 阶段正确执行所需的信息都应可获得并被验证，N 阶段的输出包括：

① N 阶段的规范、设计规范或代码应充分满足：

——功能性；

——安全完整性、性能和其他安全计划编制的要求；

——开发小组可读；

——进一步验证的可测试性；

——允许进一步改进的安全修改。

② 对规定和描述 N 阶段的设计

N 阶段规定的确认计划和/或测试是充分性的。

③ 检查下列两点之间的不兼容性：

——N 阶段规定的测试和 N-1 阶段规定的测试；

——N 阶段中的输出。

(7) 软件开发生命周期的各阶段应执行下列验证活动：

① 软件要求的验证（见 7.8.2(8)）；

② 软件结构的验证（见 7.8.2(9)）；

③ 软件系统设计的验证（见 7.8.2(10)）；

④ 软件模块设计的验证（见 7.8.2(11)）；

⑤ 代码的验证（见 7.8.2(12)）

⑥ 数据验证（见 7.8.2(13)）；

⑦ 软件模块测试（见 7.5.4）；

⑧ 软件集成测试（见 7.5.5）；

⑨ 计算机系统集成测试（见 7.5.6）；

(8) 软件要求验证：一旦规定软件要求，在下一阶段、软件设计和开发开始前，验证应：

① 考虑规定的软件要求是否已充分满足计算机系统规定的对功能、安全完整性、性能和其他安全确认计划编制的要求。

② 考虑软件安全确认计划是否已充分满足规定的软件安全要求。

③ 检查下列两点之间的不兼容性：

——规定的软件要求和规定的计算机系统安全要求；

——规定的软件要求和软件确认计划。

(9) 软件结构验证：在建立软件结构设计后，验证应：

① 考虑软件结构设计的描述是否已充分满足规定的软件安全要求。

② 考虑软件结构集成规定的测试对软件结构设计规范是否充分。

③ 考虑每一主要组件/子系统的属性是否充分满足：

——要求的安全性能的柔性；

——进一步验证的可测试性；

——开发和验证小组可读；

——允许进一步改进的安全修改。

④ 检查下列不兼容性：

——软件结构设计的描述和规定的软件安全要求；

——软件结构设计的描述和规定的软件结构集成测试；

——软件结构集成的规定测试和软件安全确认计划。

(10) 软件系统设计验证：规定软件系统设计后，验证应：

① 考虑规定的软件系统设计是否已充分满足软件结构设计。

② 考虑软件系统集成规定的测试是否已充分满足规定的软件系统设计。

③ 考虑规定的软件系统设计的每一主要组件的属性是否已足够满足：

——要求的安全性能的柔性；

——进一步验证的可测试性；

——开发和验证小组可读；

——允许进一步改进的安全修改。

④ 检查下列二三点之间的不兼容性：

——规定的软件系统设计和软件结构设计的描述；

——软件系统设计规范和软件系统集成规定的测试；

——软件系统集成规定的测试和结构集成规定的测试。

(11) 软件模块设计验证：在规定每一软件模块设计后，验证应：

① 考虑规定的软件模块设计是否已充分满足规定的软件系统设计。

② 考虑每一软件模块的规定测试对规定的软件模块设计是否充分。

③ 考虑每一软件模块的属性是否充分满足：

——要求的安全性能的柔性；

——进一步验证的可测试性；

——开发和验证小组可读；

——允许进一步改进的安全修改。

④ 检查下列三点之间的不兼容性：

——规定的软件模块设计和规定的软件系统设计；

——（对每一个软件模块）规定的软件模块设计和规定的软件模块测试；

——规定的软件模块测试和规定的软件系统集成测试。

(12) 代码验证：源代码需通过静态方法验证，以确保软件模块的规定设计、要求的编码标准和安全计划编制要求之间的符合性。

注：软件安全生命周期的早期阶段，验证是静态的(如检查、复查、形式证明等)。代码验证包括软件检查和走查等技术。它是代码验证的记录和软件模块测试的结合，以保证每一软件模块满足其相关规施。此后，向前测试成为验证的主要方法。

(13) 数据验证

① 设计中规定的数据结构应验证：

——完整性；

——自身一致性；

——对改变或破坏的防范；

——数据驱动系统功能要求的一致性。

② 应用数据应验证：

——与数据结构的一致性；

——完整性；

——与基础系统软件的兼容性（如执行的次序、运行时间等）；

——数据值的校正。

③ 所有修改参数应验证以防止：

——无效或未定义初始值；

——错误、不连续或不合理值；

——非批准改变；

——数据损坏。

④ 所有大型设备接口和有关软件（即传感器、执行器和离线界面）应验证，以：

——用于预期界面失效的探测；

——用于预期界面失效的容错。

⑤ 所有通信接口和有关软件应对下列事件的充分水平进行验证：

——失效探测；

——错误防范；

——数据确认。

7.8.3 输入

适当的验证计划（根据阶段）

7.8.4 输出

适当的验证报告（根据阶段）

8 试验和验证

8.1 计算机系统应根据表 8.2 的要求进行试验和验证，本节仅针对软件提出了具体的要求。

8.2 小型低复杂度计算机系统的评估应按照本指南附录 2 的要求进行试验和验证。

试验和验证 **表 8.2**

序号	试验和证明	系统类别			需提供的文件
		I	II	III	
1.	**质量体系文件**				
1.1	软件质量计划		M	M	软件质量计划
1.2	检查由分销商供应的部件（仅对硬件）		M	M	进货检验记录
1.3	生产中的质量控制		M	M	过程检验记录
1.4	完工试验报告	M	M	S	试验报告
1.5	软件的可追溯性	M	M	S	①软件质量计划 ②程序和数据的修改及版本变更的记录
2.	**硬件和软件说明**				
2.1	软件说明		M	S	①软件要求规范 ②软件结构设计规范 ③软件结构集成测试规范 ④计算机系统集成测试规范 ⑤开发工具和编码标准 ⑥开发工具的选择 ⑦软件系统设计规范 ⑧软件系统集成测试规范 ⑨软件模块设计规范 ⑩软件模块测试规范 ⑪计算机系统要求规范
2.2	硬件说明		M	S	计算机系统硬件说明 计算机系统要求规范
2.3	仅针对安全相关功能的故障分析			S	软件功能安全评估报告
3.	**软件试验证明文件**				
3.1	证明软件测试依据质量计划进行的文件		M	S	代码复审报告 软件模块测试记录 软件系统集成测试记录 软件结构集成测试记录 计算机系统测试记录
3.2	分析安全相关功能的可编程程序是否存			S	软件功能安全评估报告

	在并得以遵守				
4.	**硬件试验**				
4.1	按 GD01-2006 进行试验		W	W	计算机系统型式试验报告
5.	**软件测试**				
5.1	模块测试		M	S	软件模块测试规范 软件模块测试记录
5.2	子系统测试		M	S	软件系统测试规范 软件系统测试记录
5.3	系统测试		M	S	软件系统测试规范 软件系统测试记录
6.	**性能测试**				
6.1	集成测试		M	W	软件结构集成测试记录 计算机系统集成测试记录
6.2	故障模拟		W	W	计算机系统故障测试记录
6.3	工厂验收试验（FAT）	M	W	W	FAT 报告
7.	**船上试验**				
7.1	完全系统试验	M	W	W	船上试验报告
7.2	集成试验		W	W	船上试验报告
7.3	通过对无线功能的操作证明电磁兼容能力		W	W*	船上试验报告
8.	**修改**				
8.1	修改后的试验	M	S/W	S/W	软件操作和修改规程 软件修改影响分析记录 试验报告

注： M - 制造厂保留的并按需要提交的证明文件

S - 经船级社检查的证明文件

W - 经船级社见证

*- 见证的等级按照下述的要求评估后决定，若采取与预定要求不一致的设计或布置，应向 CCS 提交按照相关国际（参见 SOLAS 公约第 II-1 章第 55 条。）或国内标准进行的工程分析，并获得认可。

附录 1　测试和检验的验证表

序号	试验和证明	参考条目	系统类别			需提供的文件	是否符合要求
			I	II	III		
1.	**质量体系文件**						
1.1	软件质量计划	5.2		**M**	**M**	软件质量计划	
a	是否有明确的标准和指导性文件来定义产品？			×	×		
b	所有的相关方（如：开发人员、项目负责人等）是否对产品进行了审查？			×	×		
c	产品的验收标准是否已经建立？				×		
d	产品是否已经明确的定义了目标和应用范围？			×	×		
e	是否明确了哪些软件内容已经被软件质量保证计划覆盖？				×		
f	是否规定了软件的预定用途？				×		
g	是否描述了软件开发生命周期哪部分已经被软件质量保证计划覆盖？				×		
h	是否包含了可用的参考资料？			×	×		
i	是否包括项目管理结构的概要？				×		
j	是否细化了用于管理软件的开发，验证，确认，使用和维护的文件？				×		
k	是否对文件进行列表和描述？		×	×	×		
l	是否已经列出需要被软件质量计划评估的文件？			×	×		
m	使用的标准、实践和质量要求是否被识别（如 IEC,ISO,IEEE 等标准）？			×	×		
n	是否描述如何监测和保证过程及产品的符合性（如：追溯、报告和趋势）？				×		
o	是否明确和描述软件管理计划在软件验证及确认中的角色？				×		
p	是否描述了对问题进行报告、跟踪和解决的方法和程序？				×		
q	是否描述哪些工具和技术被用于支持软件保证活动（如：检验清单，计划和报告模板，用于追溯的数据库）？						

r	是否讨论了通过内外部监督确保供应商的控制能满足客户的要求（如：检查，评估/审核，月度状态报告）？				×		
s	软件的设计和开发是否能确保其满足特殊的设计和开发要求即对潜在的失效条件进行预防和响应？			×	×		
1.2	检查由分销商供应的部件（仅对硬件）	5		**M**	**M**	进货检验记录	
a	是否建立了 ISO9000 质量体系			×	×		
b	对进货检验是否规定了检查方式、比例、判定方法、不合格品的控制			×	×		
1.3	生产中的质量控制	5.3		**M**	**M**	过程检验记录	
a	是否制定了生产工艺流程？			×	×		
b	各工艺流程的质量控制要求是否用文字以及图表清晰描述？			×	×		
c	质量控制文件是否包含明确的控制对象、控制标准、控制方法及检验方法？			×	×		
d	是否有产品在生产过程中没有被更改的措施及方法？			×	×		
e	是否对软件采取适当的措施保护软件的完整性及防止病毒？			×	×		
f	对于安全相关功能的产品，是否有“试验和模拟”的证明文件？			×	×		
1.4	最终的试验报告	5.4	**M**	**M**	**S**	试验报告	
a	根据成品试验和试验结果记录是否生成的报告		×	×	×		
b	最终试验报告是否包含对被测试软件的总体评价？		×	×	×		
c	是否给出测试环境与操作环境的差异及这种差异对测试结果的影响进行的评估？		×	×	×		
d	测试结果总结是否已包含“所有结果都符合预期”、“遇到的问题”（如适用）和，“与要求的偏差”（如适用）。		×	×	×		
1.5	软件的可追溯性	5.5 5.6	**M**	**M**	**S**	①软件质量计划 ②程序和数据的修改及版本变更的记录	

a	是否按程序对编程内容和数据的修改以及版本的变化进行标识并文档化。		×	×	×		
b	是否制定了软件配置管理及软件版本说明等质量保证文件。		×	×	×		
c	是否明确编程内容、数据的修改以及版本的变化所必须遵循的流程，并确定在文件中记录这些修改或变化。		×	×	×		
2.	**硬件和软件说明**						
2.1	软件说明			M	S	①软件要求规范 ②软件结构设计规范 ③软件结构集成测试规范 ④计算机系统集成测试规范 ⑤开发工具和编码标准 ⑥开发工具的选择 ⑦软件系统设计规范 ⑧软件系统集成测试规范 ⑨软件模块设计规范 ⑩软件模块测试规范	

						⑪计算机系统要求规范	
a	是否根据系统功能要求规定了软件要求规范？	7.3		×	×		
b	是否对每个需实现一定安全功能的计算机系统规定软件安全功能的要求	7.3		×	×		
c	是否规定每一个计算机系统对于软件集成的要求	7.3		×	×		
d	软件结构设计的描述是否包括：在所需的软件开发生命周期中，按不同等级的系统，选择和判断满足软件要求规范的集成技术。	7.5.1		×	×		
e	软件要求规范的技术和措施是否包括：故障允许偏差（与硬件一致）和故障避免的软件设计策略，（适用时）冗余和多样性。	7.5.1		×	×		
f	软件结构设计的描述是否包括：确定所有软件/硬件相互作用和评价及细化它们的重要性。	7.5.1		×	×		
g	软件结构设计的描述是否包括：规定适当的软件结构集成测试来保证软件结构满足规定系统等级上的软件安全要求。	7.5.1		×	×		
h	标准和命名规则是否明确？	7.5.1		×	×		
i	是否提供软件系统设计规范和单个模块设计规范	7.5.2		×	×		
j	软件系统设计和单个模块设计规范是否说明了功能、性能、模块和其他部件之间的相互约束和依赖关系	7.5.2		×	×		
k	软件系统设计和单个模块设计规范是否说明了软件自我监控（例如，包括应用驱动的看门狗和数据范围验证）	7.5.2		×	×		
l	软件系统设计和单个模块设计规范是否要求进行验证测试和外部设备诊断测试（例如，传感器和终端元件）	7.5.2		×	×		
m	软件系统设计和单个模块设计规范是否对坏的过程变量，如传感器值超出范围、检测开路、检测短路，采取了措施	7.5.2		×	×		
2.2	硬件说明			M	S	计算机系统硬件说明 计算机系统要	

						求规范	
a	硬件说明是否包括：系统框图，设备布置、输入输出设备以及相互连接关系；			×	×		
b	硬件说明是否包括：接线图；			×	×		
c	硬件说明是否包括：输入输出设备详细说明；			×	×		
d	硬件说明是否包括：电源详细说明。			×	×		
2.3	仅针对安全相关功能的故障分析				S	软件功能安全评估报告	
a	是否采用了适当的方法，如故障树分析、风险分析、FMEA 或 FMECA 分析；	6.2			×		
b	是否通过故障分析证明对于单一故障，系统进入故障安全状态，并且运行中的系统不会丢失或降低到不能满足船级社规定的可接受性能标准	6.2			×		
3.	**软件试验证明文件**						
3.1	证明软件测试依据质量计划进行的文件			M	S	代码复审报告 软件模块测试记录 软件系统集成测试记录 软件结构集成测试记录 计算机系统测试记录	
a	软件开发方是否复审了软件代码，检查代码编写和它的结果是符合详细设计文件的描述。	7.5.3		×	×		
b	软件开发方是否通过检查代码编写，得到代码编写是符合详细设计文件的记录。	7.5.3		×	×		
c	软件模块测试的记录是否已文档化。	7.5.4		×	×		
d	对 II 类、III 类船用设备的软件模块测试记录的文档记录的内容是否包括软件模块测试计划、软件模块测试用例、软件模块测试记录、测试记录分析报告、软件模块测试问题报告	7.5.4		×	×		

	单和测试总结报告。						
e	软件系统集成测试是否表明所有软件模块和软件组件/子系统内部正确作用以执行其预定的功能而不执行非预定的功能	7.5.5		×	×		
f	软件集成测试的记录是否文档化，并说明测试记录是否满足目的和测试准则。如果出现失效，记录失效原因	7.5.5		×	×		
g	在软件集成过程中，是否对软件的任何修改或改变进行影响分析以确定对所有软件模块的影响和所需要的再验正和再设计活动。	7.5.5		×	×		
h	软件系统测试是否验证了防修改保护功能	7.5.5		×	×		
i	II、III 类系统的系统测试，是否验证了单一故障条件下系统会安全失效的功能	7.5.5		×	×		
j	计算机系统集成测试是否规定了：①根据集成水平拆分系统；②测试用例和测试数据；③执行测试的种类；④测试环境包括工具、支持软件和配置描述；⑤判定测试完成的准则。	7.6.2		×	×		
k	在进行计算机系统集成测试时，是否对开发人员按自己的意图所执行的活动和从用户立场出发所进行的活动加以区别	7.6.2		×	×		
l	在安全有关可编程电子（硬件和软件）集成测试中，是否对集成系统的任何修改或改变进行影响分析，以确定对所有软件模块的影响和所需要的再验证活动。	7.6.2		×	×		
m	对安全有关可编程电子（硬件和软件）的集成测试是否已文档化，说明测试记录是否满足测试目的和测试准则。如果出现失效应记录失效原因。软件的任何修改或改变应进行影响分析以确定对所有软件组件/模型的影响和所需要的再验证和再设计活动。	7.6.2		×	×		
n	对于 II 类系统，制造厂是否保留并按需求提交集成测试的证明文件，文件包括测试计划和测试报告。	7.6.2		×			
o	对于 III 类系统，船级社是否见证了集成测试。	7.6.2			×		
p	故障模拟测试规范是否包括以下内容：①故障组件或元器件名称；②故障类型；③插入故障方式；④要求的系统响应（输出记录）。	7.6.3		×	×		
q	故障模拟的测试用例及其预期记录是否已文档化。说明故障模拟的测试记录以及是否满足测试目的和测试准则。如果出现失败记录，失败原因是否已文档化。	7.6.3		×	×		

3.2	分析安全相关功能的可编程程序是否存在并得以遵守				S	软件功能安全评估报告	
a	是否通过故障分析证明对于单一故障，系统进入故障安全状态，并且运行中的系统不会丢失或降低到不能满足船级社规定的可接受性能标准	6.2			×		
4.	**硬件试验**						
4.1	按 GD01-2006 进行试验	表 6.1.1-D1		W	W	计算机系统型式试验报告	
a	按 GD01-2006 进行试验			×	×		
5.	**软件测试**						
5.1	模块测试	7.8.2(11)		M	S	软件模块测试规范 软件模块测试记录	
a	在规定每一软件模块设计后，验证：考虑规定的软件模块设计是否已充分满足规定的软件系统设计。			×	×		
b	在规定每一软件模块设计后，验证：考虑每一软件模块的规定测试对规定的软件模块设计是否充分。			×	×		
c	在规定每一软件模块设计后，考虑每一软件模块的属性是否充分满足：①要求的安全性能的柔性；②进一步验证的可测试性；③开发和验证小组可读；④允许**进**一步改进的安全修改。			×	×		
d	在规定每一软件模块设计后，检查下列三点之间的不兼容性：①规定的软件模块设计和规定的软件系统设计；②（对每一个软件模块）规定的软件模块设计和规定的软件模块测试；③规定的软件模块测试和规定的软件系统集成测试。			×	×		
5.2	子系统测试			M	S	软件系统测试规范	

						软件系统测试记录	
a	软件系统集成测试是否表明所有软件模块和软件组件/子系统内部正确作用以执行其预定的功能而不执行非预定的功能	7.5.5		×	×		
b	软件集成测试的记录是否文档化，并说明测试记录是否满足目的和测试准则。如果出现失效，记录失效原因	7.5.5		×	×		
c	在软件集成过程中，是否对软件的任何修改或改变进行影响分析以确定对所有软件模块的影响和所需要的再验正和再设计活动。	7.5.5		×	×		
5.3	系统测试			M	S	软件系统测试规范 软件系统测试记录	
a	软件系统集成测试是否表明所有软件模块和软件组件/子系统内部正确作用以执行其预定的功能而不执行非预定的功能	7.5.5		×	×		
b	软件集成测试的记录是否文档化，并说明测试记录是否满足目的和测试准则。如果出现失效，记录失效原因	7.5.5		×	×		
c	在软件集成过程中，是否对软件的任何修改或改变进行影响分析以确定对所有软件模块的影响和所需要的再验正和再设计活动。	7.5.5		×	×		
d	软件系统测试是否验证了防修改保护功能	7.5.5		×	×		
e	II、III 类系统的系统测试，是否验证了单一故障条件下系统会安全失效的功能	7.5.5		×	×		
6.	**性能测试**						
6.1	集成测试			M	W	软件结构集成测试记录 计算机系统集成测试记录	

a	在建立软件结构设计后，验证：考虑软件结构设计的描述是否已充分满足规定的软件安全要求。	7.8.2(9)		×	×		
b	在建立软件结构设计后，验证：考虑软件结构集成规定的测试对软件结构设计规范是否充分。	7.8.2(9)		×	×		
c	在建立软件结构设计后，考虑每一主要组件/子系统的属性是否充分满足：①要求的安全性能的柔性；②进一步验证的可测试性；③开发和验证小组可读；④允许进一步改进的安全修改。	7.8.2(9)		×	×		
d	在建立软件结构设计后，检查下列不兼容性：①软件结构设计的描述和规定的软件安全要求；②软件结构设计的描述和规定的软件结构集成测试；③软件结构集成的规定测试和软件安全确认计划。	7.8.2(9)		×	×		
e	计算机系统集成测试是否规定了：①根据集成水平拆分系统；②测试用例和测试数据；③执行测试的种类；④测试环境包括工具、支持软件和配置描述；⑤判定测试完成的准则。	7.6.2		×	×		
f	在进行计算机系统集成测试时，是否对开发人员按自己的意图所执行的活动和从用户立场出发所进行的活动加以区别	7.6.2		×	×		
g	在安全有关可编程电子（硬件和软件）集成测试中，是否对集成系统的任何修改或改变进行影响分析，以确定对所有软件模块的影响和所需要的再验证活动。	7.6.2		×	×		
h	对安全有关可编程电子（硬件和软件）的集成测试是否已文档化，说明测试结果是否满足测试目的和测试准则。如果出现失效应记录失效原因。软件的任何修改或改变应进行影响分析以确定对所有软件组件/模型的影响和所需要的再验证和再设计活动。	7.6.2		×	×		
i	对于 II 类系统，制造厂是否保留并按需求提交集成测试的证明文件，文件包括测试计划和测试报告。	7.6.2		×			
j	对于 III 类系统，船级社是否见证了集成测试。	7.6.2			×		
6.2	故障模拟			W	W	计算机系统故障测试记录	
a	故障模拟测试规范是否包括以下内容：①故障组件或元器件名称；②故障类型；③插入故	7.6.3		×	×		

	障方式；④要求的系统响应（输出记录）。						
b	故障模拟的测试用例及其预期记录是否已文档化。说明故障模拟的测试记录以及是否满足测试目的和测试准则。如果出现失败记录，失败原因是否已文档化。	7.6.3		×	×		
6.3	工厂验收试验（FAT）	表 6.1.1-D2	**M**	**W**	**W**	FAT 报告	
a	FAT 过程是否按时间顺序记录，以便追溯 FAT 活动的顺序		×	×	×		
b	是否记录了使用的工具和设备及其校准数据		×	×	×		
c	是否有预期记录和实际记录的差异。当预期记录和实际记录出现差异时，应经分析和评估确定是继续测试还是提出变更请求，若提出变更请求，是否返回开发生命周期较早阶段。		×	×	×		
7.	**船上试验**						
7.1	完全系统试验	表 6.1.1-D3	**M**	**W**	**W**	船上试验报告	
a	验证在实际硬件部件及最终应用软件的条件下，功能是否可以正常实现。		×	×	×		
7.2	集成试验	表 6.1.1-D3		**W**	**W**	船上试验报告	
a	验证所有系统集成状态下的功能是否可以正常实现。			×	×		
8.	**修改**						
8.1	修改后的试验		**M**	S/W	S/W	软件操作和修改规程 软件修改影响分析记录 试验报告	
a	制造厂应对修改是否进行记录。	7.7	×	×	×		
b	制造厂是否提前告知对已批准系统的修改和变更方案，并进行影响分析，同时修改应获得船级社批准。	7.7	×	×	×		

c	制造厂对 III 系统的软件进行的后续重大修改是否提交给船级社并获批准。	7.7			×		
d	根据影响分析的记录，是否已返回软件开发生命周期的合适阶段并进行相应的验证。	7.8	×	×	×		

注：M – 制造厂保留的并按需求提交的证明文件，S – 经船级社检查的证明文件，W – 应经船级社见证，× – 本条要求适用。

附录 2　小型低复杂度计算机系统的评估

1　目的

1.1　对小型低复杂度计算机系统通过单案评估的方法，对软件评估方法进行合理有效的简化。

2　要求

2.1　文档

2.1.1　软件说明可根据制造厂内部文件管理系统，将表 8.2 中 2.1 条需提供的文件合并，但内容应包括：

(1)　系统功能描述，特别功能、性能、模块和其他部件之间的相互约束和依赖关系；

(2)　软件设计说明，特别是软件的配置，其中包括优先级策略；

(3)　软件版本说明；

(4)　失效模式分析；

(5)　冗余系统的切换机制（若有）；

(6)　系统测试、集成测试和故障模拟试验方法。

2.2　测试

(1)　对新设计的产品，应核查其失效模式分析，并按经确认的制造厂提供的试验方法进行测试。

(2)　对软件复用或修改使用之前软件的产品，还应注意回归测试。

注 1：软件复用是将已有软件的各种有关知识用于建立新的软件，以缩减软件开发和维护的花费。软件复用是提高软件生产力和质量的一种重要技术。软件复用主要是代码级复用，被复用的不专指程序，也包括领域知识、开发经验、设计决定、体系结构、需求、设计、代码和文档等一切有关方面。

注 2：回归测试是指修改了旧代码后，重新进行测试以确认修改没有引入新的错误或导致其他代码产生错误。

3　输入

3.1　计算机系统要求规范

4 输出

4.1 软件说明

4.2 硬件说明

4.3 测试报告

附录 3　计算机系统设计和实现阶段的技术建议

1　一般要求

1.1　计算机系统安全相关系统的设计（包括硬、软件的整体结构、传感器、执行器、可编程电子、嵌入式软件和应用软件等，见下图），应当符合以下 1.1.1～1.1.2 的全部要求：

1.1.1　硬件安全完整性要求包括：

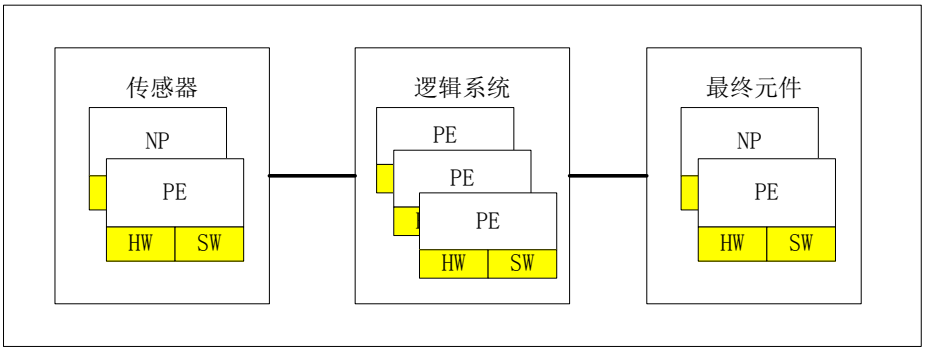
(1)　硬件安全完整性的结构约束；和

(2)　危险随机硬件失效概率的要求。

1.1.2　系统安全完整性要求包括：

(1)　避免失效的要求和系统故障控制的要求；或

(2)　设备"经使用证实"的证据。



可编程电子结构		
PE 硬件结构	PE 软件结构	
	PE 嵌入式软件	PE 应用软件
PE 硬件中通用的和应用时的具体特性： 例如包括： ——诊断测试； ——冗余处理器； ——双 I/O 卡	例如包括： ——通信驱动器； ——故障处理； ——可执行软件	例如包括： ——输入/输出功能； ——派生功能（例如，未提供嵌入式软件服务时的传感器检验）

PE：可编程电子，NP：非可编程装置，HW：硬件，SW：软件。

图附录 3-1.1.2　PE 硬件和软件结构的关系

1.2　在计算机系统既执行安全功能又执行非安全功能的地方，除非能够表明实现安全功能和非安全功能是充分独立的（也就是说，非安全功能的失效不会引起安全功能的危险失效），否则的软硬件都应被视为与安全相关的。只要可行，安全功能应与非安全功能分开。

1.3　软硬件的要求由拥有最高安全完整性等级的安全功能的安全完整性等级来决定，除非能够表明不同安全完整性等级的安全功能的实现是充分独立的。

1.4　在要求安全功能之间相互独立（见 1.2 和 1.3）时，在设计时以下几条应文档化：

1.4.1　达到独立的方法；

1.4.2　方法的合理性证明。

2　硬件安全完整性的技术和措施：操作中的失效控制

附录 3 表 2-1～附录 3 表 2-6 给出了有关硬件安全完整性技术和措施的建议

I/O 单元和接口（外部通信）　　附录 3 表 2-1

诊断技术/措施	见 IEC61508-7	经考虑能达到的最大诊断覆盖率	注
利用在线监视检测失效	A1.1	低（低要求模式） 中（高要求或连续模式）	依赖于失效检测的诊断覆盖率
测试模式	A6.1	高	
代码保护	A6.2	高	
多通道平行输出	A6.3	高	仅当诊断测试问陆内数据流改变时才有效
监视输出	A6.4	高	仅当诊断测试间隔内数据流改变时才有效

数据路径（内部通信）　　附录 3 表 2-2

诊断技术/措施	见 IEC61508-7	经考虑能达到的最大诊断覆盖率	注
1 位硬件冗余	A7.1	低	
多位硬件冗余	A7.2	中	
完全硬件冗余	A7.3	高	
使用测试模式进行检查	A7.4	高	仅对瞬时故障有效
传输冗余	A7.5	高	
信息冗余	A7.6	高	

电源　　附录 3 表 2-3

诊断技术/措施	见 IEC61508-7	经考虑能达到的最大诊断覆盖率	注
使用安全断电或切换到备用电源单元的过压保护	A8.1	低	应使用本表中的技术，也推荐使用其他技术
使用安全断电或切换到备用电源单元的电压控制（次级）	A8.2	高	
带安全断电或切换	A8.3	高	应使用本表中的技

到备用电源单元的断电			术，也推荐使用其他技术
无功电流原理	A1.5	低	仅对断电有用

程序顺序（看门狗） **附录 3 表 2-4**

诊断技术/措施	见 IEC61508-7	经考虑能达到的最大诊断覆盖率	注
具有分离时基但无时间窗的看门狗	A9.1	低	
具有分离时基和时间窗的看门狗	A9.2	中	
程序顺序的逻辑监视	A9.3	中	依赖于监视质量
程序顺序的时序和逻辑监视的组合	A9.4	高	
具有在线检验的时序监视	A9.5	中	

传感器 **附录 3 表 2-5**

诊断技术/措施	见 IEC61508-7	经考虑能达到的最大诊断覆盖率	注
利用在线监视检测失效	A1.1	低（低要求模式） 中（高要求或连续模式）	依赖于失效检测的诊断覆盖率
无功电流原理	A1.5	低	仅对无需连续控制未达到或保持 EUC 安全状态的 E/E/PE 安全先关系统才有效
模拟信号监视	A2.7	低	
测试模式	A6.1	高	
输入比较/表决	A6.5	高	仅当诊断测试间隔内数据流改变时才有效
参考传感器	A12.1	高	依赖于失效检测的诊断覆盖率
可靠开启的开关	A12.2	高	

最终元件（执行器） **附录 3 表 2-6**

诊断技术/措施	见 IEC61508-7	经考虑能达到的最大诊断覆盖率	注
利用在线监视检测失效	A1.1	低（低要求模式） 中（高要求或连续模式）	依赖于失效检测的诊断覆盖率
继电器触点监视	A1.2	高	
无功电流原理	A1.5	低	仅对无需连续控制未达到或保持 EUC 安全状态的 E/E/PE 安全先关系统才有效
测试模式	A6.1	高	
监视	A13.1	高	依赖于失效检测的诊

			断覆盖率
多个执行器的交叉监视	A13.2	高	

3 系统完整性的技术和措施的建议

3.1 附录 3 表 3.1-1、附录 3 表 3.1-2 给出了有关系统安全完整性技术和措施的建议

3.1.1 控制由硬件和软件设计引起的失效；

3.1.2 控制由环境应力或影响引起的失效；

3.1.3 控制操作过程的失效。

用于控制由硬件设计引起的系统失效的技术和措施 附录 3 表 3.1-1

	技术/措施	见 IEC61508-7	I	II	III
1	程序顺序监视	A.9	极力推荐 低	极力推荐 低	极力推荐 中
2	利用在线监视检测失效	A1.1	推荐 低	推荐 低	推荐 中
3	利用冗余硬件进行测试	A2.1	推荐 低	推荐 低	推荐 中
4	访问端口和边界扫描结构的标准测试	A2.1	推荐 低	推荐 低	推荐 中
5	代码保护	A6.2	推荐 低	推荐 低	推荐 中
6	多种硬件	B1.4	- 低	- 低	推荐 中
注：要求至少应用一种 2~6 中的技术。					

用于控制由环境应力或影响引起的系统失效的技术和措施 附录 3 表 3.1-2

	技术/措施	见 IEC61508-7	I	II	III
1	防止电压击穿、电压波动、过压、低压的措施	A8	必须采用	必须采用	必须采用
2	分隔开电力线和信息线（注 1）	A11.1	必须采用	必须采用	必须采用
3	提高抗干扰性	A11.3	必须采用	必须采用	必须采用
4	抗物理环境（如温度、湿度、振动等）的措施	A14	必须采用	必须采用	必须采用
5	程序顺序监视	A9	极力推荐 低	极力推荐 低	极力推荐 中
6	抗温升措施	A10	极力推荐 低	极力推荐 低	极力推荐 中
7	多线路的空间分隔	A11.2	极力推荐 低	极力推荐 低	极力推荐 中
8	利用在线监视检测失效（注 2）	A1.1	推荐 低	推荐 低	推荐 中
9	利用冗余硬件进行测试	A2.1	推荐	推荐	推荐

			低	低	中
10	代码保护	A6.2	推荐 低	推荐 低	推荐 中
11	抗合成信号传输	A11.4	推荐 低	推荐 低	推荐 中
12	多种硬件（注 3）	B1.4	- 低	- 低	- 中
13	软件结构	GB/T20438.3 的 7.4.3	见 GB/T20438.3 的表 A.2		

注：要求至少应用一种 8～13 中的技术。

注 1：若信息传输采用光介质，则无需分离电力线和信息线。并且也不需要分隔开为系统的部件通电，以及载送来自或传送到这些部件的信息而设计的低功率电缆。

注 2：对于在低要求工作模式下工作的安全相关系统（例如紧急关闭系统），通过在线监视由失效检测所达到的诊断覆盖率通常为低或无。

注 3：若通过确认和广泛工作经验证明:为满足目标失效量，硬件充分摆脱了设计故障并足以防止共同原因失效，则不需要多种硬件。