

该出版物已被第 SSG-39 号取代。

IAEA

国际原子能机构

安全标准

丛书

核动力厂基于计算机的
安全重要系统的软件

安全导则

No. NS-G-1.1



IAEA

国际原子能机构

该出版物已被第 SSG-39 号取代。

国际原子能机构安全相关出版物

国际原子能机构（原子能机构）安全标准

根据原子能机构《规约》第三条的规定，原子能机构授权制定或采取旨在保护健康及尽量减少对生命与财产的危险的的安全标准，并规定适用这些标准。

原子能机构借以制定标准的出版物以国际原子能机构安全标准丛书的形式印发。该丛书涵盖核安全、辐射安全、运输安全和废物安全以及一般安全（即涉及上述所有安全领域）。该丛书出版物的分类是安全基本法则、安全要求和安全导则。

安全标准按照其涵盖范围编码：核安全（NS）、辐射安全（RS）、运输安全（TS）、废物安全（WS）和一般安全（GS）。

有关原子能机构安全标准计划的信息可访问以下原子能机构因特网网址：

<http://www-ns.iaea.org/standards/>

该网址提供已出版安全标准和安全标准草案的英文文本。也提供以阿拉伯文、中文、法文、俄文和西班牙文印发的安全标准文本、原子能机构安全术语表以及正在制订中的安全标准状况报告。欲求详细信息，请与原子能机构联系（P.O. Box 100, A-1400 Vienna, Austria）。

敬请原子能机构安全标准的所有用户将其使用方面的经验（例如作为国家监管、安全评审和培训班课程的基础）通知原子能机构，以确保原子能机构安全标准继续满足用户需求。资料可以通过原子能机构因特网网址提供或按上述地址邮寄或通过电子邮件发至 Official.Mail@iaea.org。

其他安全相关出版物

原子能机构规定适用这些标准，并按照原子能机构《规约》第三条和第八条 C 款之规定，提供和促进有关和平核活动的信息交流并为此目的充任各成员国的居间人。

核活动的安全和防护报告以其他出版物丛书的形式特别是以**安全报告丛书**的形式印发。安全报告提供能够用以支持安全标准的实例和详细方法。原子能机构其他安全相关出版物丛书是**安全标准丛书适用规定**、**放射学评定报告丛书**和**国际核安全咨询组丛书**。原子能机构还印发放射性事故报告和其他特别出版物。

安全相关出版物还以**技术报告丛书**、**国际原子能机构技术文件丛书**、**培训班丛书**、**国际原子能机构服务丛书**的形式以及作为**实用辐射安全手册**和**实用辐射技术手册**印发。保安相关出版物则以**国际原子能机构核保安丛书**的形式印发。

该出版物已被第 SSG-39 号取代。

核动力厂基于计算机的 安全重要系统的软件

安 全 标 准 调 查

国际原子能机构欢迎您回复。请访问网址：

<http://www-ns.iaea.org/standards/feedback.htm>

该出版物已被第 SSG-39 号取代。

下述国家是国际原子能机构的成员国：

阿富汗	希腊	尼日利亚
阿尔巴尼亚	危地马拉	挪威
阿尔及利亚	海地	巴基斯坦
安哥拉	教廷	巴拿马
阿根廷	洪都拉斯	巴拉圭
亚美尼亚	匈牙利	秘鲁
澳大利亚	冰岛	菲律宾
奥地利	印度	波兰
阿塞拜疆	印度尼西亚	葡萄牙
孟加拉国	伊朗伊斯兰共和国	卡塔尔
白俄罗斯	伊拉克	摩尔多瓦共和国
比利时	爱尔兰	罗马尼亚
贝宁	以色列	俄罗斯联邦
玻利维亚	意大利	沙特阿拉伯
波斯尼亚和黑塞哥维那	牙买加	塞内加尔
博茨瓦纳	日本	塞尔维亚和黑山
巴西	约旦	塞舌尔
保加利亚	哈萨克斯坦	塞拉利昂
布基纳法索	肯尼亚	新加坡
喀麦隆	大韩民国	斯洛伐克
加拿大	科威特	斯洛文尼亚
中非共和国	吉尔吉斯斯坦	南非
智利	拉脱维亚	西班牙
中国	黎巴嫩	斯里兰卡
哥伦比亚	利比里亚	苏丹
哥斯达黎加	阿拉伯利比亚民众国	瑞典
科特迪瓦	列支敦士登	瑞士
克罗地亚	立陶宛	阿拉伯叙利亚共和国
古巴	卢森堡	塔吉克斯坦
塞浦路斯	马达加斯加	泰国
捷克共和国	马来西亚	前南斯拉夫马其顿共和国
刚果民主共和国	马里	突尼斯
丹麦	马耳他	土耳其
多米尼加共和国	马绍尔群岛	乌干达
厄瓜多尔	毛里塔尼亚	乌克兰
埃及	毛里求斯	阿拉伯联合酋长国
萨尔瓦多	墨西哥	大不列颠及北爱尔兰联合王国
厄立特里亚	摩纳哥	坦桑尼亚联合共和国
爱沙尼亚	蒙古	美利坚合众国
埃塞俄比亚	摩洛哥	乌拉圭
芬兰	缅甸	乌兹别克斯坦
法国	纳米比亚	委内瑞拉
加蓬	荷兰	越南
格鲁吉亚	新西兰	也门
德国	尼加拉瓜	赞比亚
加纳	尼日尔	津巴布韦

原子能机构《规约》于 1956 年 10 月 23 日在纽约联合国总部召开的国际原子能机构规约会议上通过，于 1957 年 7 月 29 日生效。原子能机构总部设在维也纳。原子能机构的主要目标是“加速和扩大原子能对全世界和平、健康及繁荣的贡献”。

该出版物已被第 SSG-39 号取代。

国际原子能机构安全标准丛书第 NS-G-1.1 号

核动力厂基于计算机的 安全重要系统的软件

安全导则

国际原子能机构
维也纳，2005 年

该出版物已被第 SSG-39 号取代。

版 权 说 明

国际原子能机构的所有科学和技术出版物均受1952年（伯尔尼）通过并于1972年（巴黎）修订的《万国版权公约》之条款的保护。自那时以来，世界知识产权组织（日内瓦）已经扩大了这一版权，以包括电子形式和虚拟形式的知识产权。必须获得许可而且通常需要签订版税协议方能使用原子能机构印刷形式和电子形式出版物中所载全部或部分内容。欢迎有关非商业性翻印和翻译的建议并将在个案基础上予以考虑。询问事宜应通过电子邮件地址 sales.publications@iaea.org 发至原子能机构出版科或按以下地址邮寄：

Sales and Promotion Unit, Publishing Section
International Atomic Energy Agency
Wagramer Strasse 5
P.O. Box 100
A-1400 Vienna
Austria
传真：+43 1 2600 29302
电话：+43 1 2600 22417
网址： <http://www.iaea.org/books>

© 国际原子能机构 • 2005 年
国际原子能机构印制
2005 年 8 月 • 奥地利

核动力厂基于计算机的安全重要系统的软件

国际原子能机构，奥地利，2005 年 8 月

STI/PUB/1095

ISBN 92-0-513705-7

ISSN 1020-5853

序

总 干 事

穆罕默德·埃尔巴拉迪

国际原子能机构《规约》授权原子能机构制定旨在保护健康及尽量减少对生命与财产的危险的的安全标准。原子能机构必须使这些标准适用于其本身的工作，而且各国通过其对核安全和辐射安全的监管规定能够适用这些标准。原子能机构对这样的一整套安全标准定期进行审查并协助实施这些安全标准已经成为全球安全体制的一个关键要素。

在 20 世纪 90 年代中期，原子能机构开始对其安全标准计划进行大检查，包括修改监督委员会的结构和确定旨在更新整套标准的系统方案。已经形成的新标准具有高水准并且反映成员国的最佳实践。在安全标准委员会的协助下，原子能机构正在努力促进全球对其安全标准的认可和使用。

诚然，只有对这些安全标准在实践中加以适当应用，它们才会是有效的。原子能机构的安全服务——其范围包括工程安全、运行安全、辐射安全、运输安全和废物安全，直至监管事项和组织中的安全文化——协助成员国适用安全标准和评价其有效性。这些安全服务能够有助于共享真知灼见，因此，我继续促请所有成员国都能利用这些服务。

监管核安全和辐射安全是一项国家责任。目前，许多成员国已经决定采用原子能机构的安全标准，以便在其国家条例中使用。对于各种国际安全公约缔约国而言，原子能机构的安全标准提供了确保有效履行这些公约所规定之义务的一致和可靠的手段。世界各地的设计者、制造者和营运者也适用这些标准，以加强电力生产、医学、工业、农业、研究和教育领域的核安全和辐射安全。

原子能机构认真看待世界各地用户和监管者正在面临的挑战，这就是确保世界范围内的核材料和辐射源在使用中的高水平安全。必须以安全的方式管理核材料和辐射源的持续利用以造福于全人类，原子能机构安全标准的目的正是要促进实现这一目标。

该出版物已被第 SSG-39 号取代。

编 者 按

如果列入附录，该附录可被视为标准的一个不可分割的组成部分并具有与主文本相同的地位。如果列入附件、脚注和文献目录，它们可被用来为用户提供可能是有用的补充信息或实例。

英文文本系权威性文本。

援引其他组织的标准不应被解释为国际原子能机构认可这些标准。

目 录

1. 引言	1
背景 (1.1—1.4)	1
目的 (1.5)	1
范围 (1.6—1.10).....	2
结构 (1.11—1.14).....	2
2. 基于计算机的系统的技术因素	3
基于计算机的系统的特征 (2.1—2.3).....	3
开发过程 (2.4—2.8).....	3
安全性和可靠性问题 (2.9—2.11).....	6
组织和法律问题 (2.12—2.16).....	6
3. 基于计算机的系统安全管理要求的应用 (3.1).....	7
安全管理要求 (3.2—3.20).....	7
设计和开发活动 (3.21—3.27).....	11
管理和质量保证 (3.28—3.33).....	12
文件 (3.34—3.44).....	14
4. 项目规划 (4.1—4.2).....	16
开发计划 (4.3—4.10).....	16
质量保证 (4.11).....	18
验证和确认计划 (4.12—4.18).....	18
配置管理计划 (4.19—4.24).....	19
安装和调试计划 (4.25—4.26).....	20
5. 计算机系统的要求 (5.1—5.3).....	21
建议 (5.4—5.23).....	22
文件 (5.24—5.40).....	24
6. 计算机系统设计 (6.1—6.2).....	27
建议 (6.3—6.26).....	27
文件 (6.27—6.42).....	31
7. 软件要求 (7.1—7.4).....	34

建议 (7.5—7.17).....	35
文件 (7.18—7.21).....	37
8. 软件设计 (8.1—8.3).....	38
建议 (8.4—8.12).....	38
文件 (8.13—8.23).....	40
9. 软件实现 (9.1—9.2).....	41
建议 (9.3—9.27).....	42
文件 (9.28—9.32).....	46
10. 验证和分析 (10.1).....	46
建议 (10.2—10.33).....	47
文件 (10.34—10.41).....	51
11. 计算机系统的集成 (11.1—11.2).....	52
建议 (11.3—11.13).....	52
文件 (11.14—11.15).....	54
12. 计算机系统的确认 (12.1—12.2).....	54
建议 (12.3—12.15).....	55
文件 (12.16)	56
13. 安装和调试 (13.1—13.4).....	56
建议 (13.5—13.10).....	57
文件 (13.11)	58
14. 运行 (14.1—14.2).....	59
建议 (14.3—14.9).....	59
文件 (14.10—14.12).....	60
15. 交付后修改 (15.1).....	60
建议 (15.2—15.8).....	61
文件 (15.9—15.12).....	61
参考文献	63

该出版物已被第 SSG-39 号取代。

附录：现成软件的使用和确认	65
术语表	69
参与起草和审订的人员	71
认可安全标准的机构	73

该出版物已被第 SSG-39 号取代。

1. 引言

背景

1.1. 随着基于计算机的系统在新的和已服役多年的核动力厂中使用的迅速增加，它们对核动力厂安全的重要性也越来越大。它们除了用于诸如反应堆保护或安全设施驱动等安全重要的应用外，还用于诸如过程控制和监测系统的某些功能等安全相关的应用。因此，基于计算机的安全重要系统的可信性十分重要，必须加以确保。

1.2 采用现代技术，原则上有可能为安全重要系统开发基于计算机的测量控制系统，使这些系统具有改善安全性和可靠性水平的潜力，并拥有足够的可信性。但是，只有遵循系统的、全部文件化的以及可审查的工程过程，它们的可信性才能得到预测和证明。尽管处理基于计算机的安全重要系统的质量保证有许多国家标准和国际标准已经或正在起草，但目前还没有国际商定的用于证明此类系统安全的准则可供使用。人们认识到，除本报告推荐的方法以外，可能还存在着能够提供必要的安全证明的其他方法。

1.3. 核动力厂安全系统设计的基本要求见IAEA“安全标准丛书”“设计要求”[1]。参考文献[2]针对保护系统和相关设备的设计对这些要求进行了扩展和解释；参考文献[3]则针对安全相关的测量控制系统的设计对这些要求进行了扩展和解释。为了反映包括数字技术在内的技术现状，目前正在对参考文献[2、3]进行修订。

1.4. IAEA已印发了一份“技术报告”[4]，目的是帮助成员国确保核动力厂中的基于计算机的安全重要系统是安全的和完全得到许可的。该报告提供了有关现行的软件工程规定和相关标准（例如参考文献[5]）的信息，是本“安全导则”的技术基础。

目的

1.5. 本安全导则的目的是指导收集证据和编写文件资料，这些证据和文件资料用于证明核动力厂基于计算机的安全重要系统的软件在系统生存周期的所有阶段是安全的。

范 围

1.6. 本导则可适用于参考文献[2、3]中定义的安全重要系统。由于目前无法仅仅依靠设计过程预测或通过设计过程建立基于计算机的系统的可靠性，因此如何将这指导性意见适当放松后适用于安全有关系统软件的问题很难做出规定和系统的商定。只要可能，对于只适用于安全系统而不适用于安全有关系统的建议，都将明确指出。

1.7. 本安全导则主要涉及在基于计算机的安全重要系统中使用的软件。有关基于计算机的系统其他方面（诸如与基于计算机的系统本身及其硬件的设计有关的内容）的指导性意见，仅限于由软件的开发、验证和确认所引起的问题。

1.8. 本安全导则的重点是起草充分证明基于计算机的安全重要系统的安全性和可靠性所需的文件。

1.9. 本安全导则适用于所有类型的软件：已有软件或固件（例如操作系统）、准备专门为该项目开发的软件、或准备针对先前开发的已有硬件或软件模块的设备家族开发的软件。有关将已有的或商业性现成软件用于安全功能的问题（有关这方面发展的信息很少）在附录中论述，该附录复制自参考文献[6]中的一章（亦可参见参考文献[7]的第6.3条）。有关测量控制系统升级过程的具体要求的信息见参考文献[8]。

1.10. 本安全导则拟供参与基于计算机的系统的生产、评定和许可证审批的工作人员使用，包括核动力厂系统的设计人员、软件设计人员和程序员、确认人员、验证人员、证明人和监管人员，也可供核动力厂操纵员使用。并考虑了这些人员间的各种接口。

结 构

1.11. 本安全导则第2章提供基于计算机系统技术因素的建议，并论述了此类系统的利弊、安全性和可靠性问题，以及开发该项目的组织条件。

1.12. 第3章提供用于基于计算机的安全重要系统的安全管理要求的建议。

1.13. 第4章提供系统开发项目规划阶段的建议，并介绍了相关文件资料的结构和内容，其中包括开发计划、质量保证大纲、验证和确认计划以及配置管理计划。

1.14. 第5—15章分别介绍开发过程的各个阶段。这些章节均以一小段介绍该阶段的引言作为开始。然后，在标题为《建议》的名下，有一组与该阶段有关的

推荐意见。在标题为《文件资料》的名下，有一张有待该阶段产生的文件的清单，并提供有关这些文件的内容的指导性意见。此外，还提供一些有关该阶段产品的特点和表述方面的一般性推荐意见。在所有这些部分中，都没有打算详尽地叙述开发工作肯定会需要的全部材料，而只是简要地叙述对安全证明来说最重要的推荐意见、材料及材料的特点。

2. 基于计算机的系统的技术因素

基于计算机的系统特征

2.1. 就安全性和可靠性评定而言，基于计算机的系统有2个基本特性。它们都是可编程的，它们的硬件以离散的数字逻辑为基础。与其他系统一样，硬件故障可能是由于设计或制造方面的错误引起的，但这些故障通常起因于磨损、质量变差或环境过程，具有随机性。软件——计算机的可编程部分——不会发生磨损，但可能会因为运行环境的变化而受到影响。软件错误可能是由于有关要求的规定不对或含混不清（这将导致逻辑设计或实现方面的错误）引起的，或是由于实现阶段或维护阶段纳入的错误引起的。

2.2. 基于计算机的系统的可编程特性加上离散逻辑，意味着它们相对于非数字的和非可编程的系统有许多优点。它们有利于实现复杂的功能；尤其是，它们能改善对核动力厂变量的监测、改善操纵员的界面、提高测试、标定、自检和故障诊断的方便性。它们还能提高准确度和稳定性。使用多路传输的“总线”结构，可以减少布线的需求。修改软件时很少要求中断设备，这有利于维护。

2.3. 与这些优点并存的是许多缺点。软件实现往往比纯硬连线系统的实现更加复杂，因而更容易发生设计错误。此外，软件实现是真实世界的离散逻辑模型。这将产生两种后果。软件对“小”错误较敏感（即宽容性较差）。对软件进行测试的难度较大，因为与传统的模拟系统相比，将内插和外推用于基于计算机的系统的难度大得多，而且最终并不是全都有效的。

开发过程

2.4. 安全重要系统的开发应当是一个一步一步受到控制的过程。在这种处理办法中，开发过程由许多有序的不同阶段组成。每个阶段都使用前面各阶段开

发出的信息，并为后面各阶段提供输入信息。安全重要系统的开发本质上是一个迭代过程。随着设计的推进，前面各阶段中发生的错误和遗漏就显露出来了，因此必须以前面的结果为基础进行迭代。这种处理方法的本质特点是每个开发阶段的产品要对照前一阶段的要求进行验证。在开发工作的某些阶段，要实施确认过程，以证实产品符合所有的功能要求和非功能要求，并没有无意识行为。这样一种一步一步受到控制过程的重要好处见参考文献[4]第3.2.2条。

2.5. 开发过程的典型阶段和可能采用的过程简图见图1。各个方框示出了将要完成的开发活动，箭头表示的是预定的次序和主要的信息流。图1括号内的数字表示本安全导则中介绍有关开发活动和产品的章节号。未注出数字的活动不属于本安全导则讨论的范围，这里示出这些活动只是为了保持完整。图2说明了验证和确认与要求、设计和实现的关系。特定开发活动的选择及其在这些图中和本安全导则中的次序，并不意味着要求人们使用特定的开发方法；变动也许还带有这些必要的属性，以便能满足这些要求。

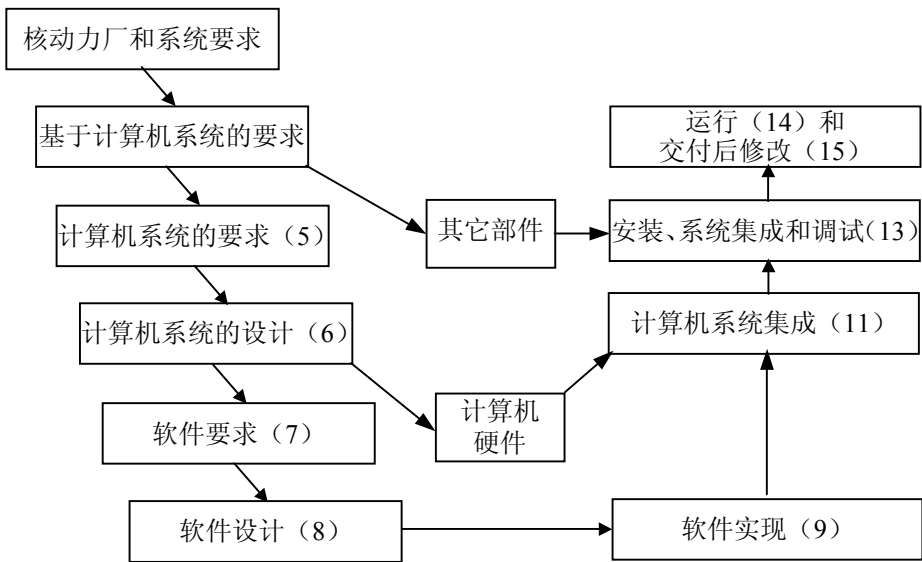


图1 基于计算机的安全重要系统的软件开发（数字表示本安全导则的章节号）。

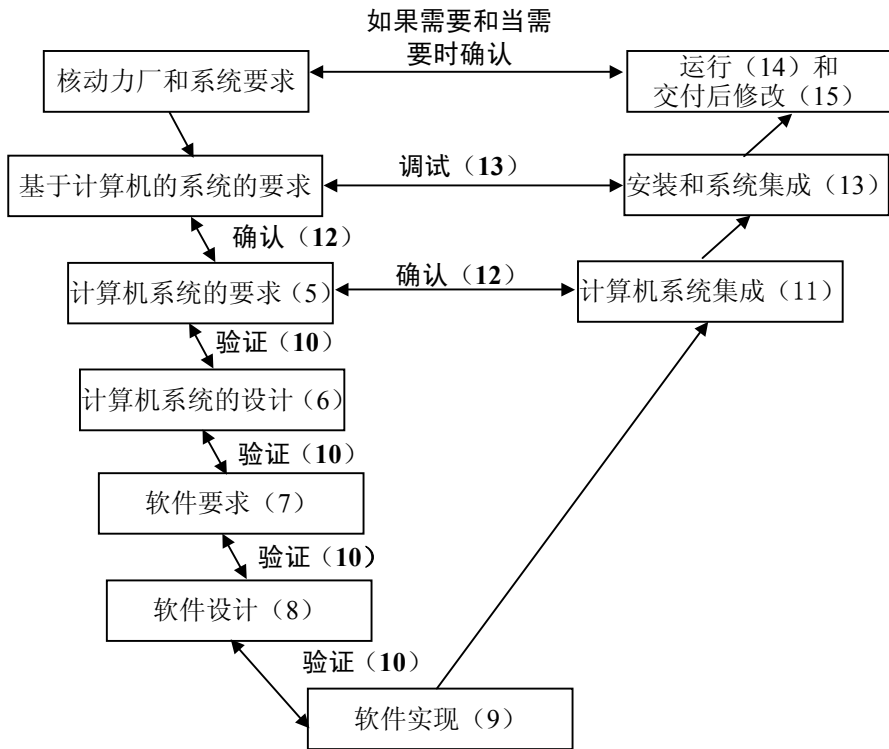


图2 验证、确认和调试（数字表示本安全导则的章节号）

2.6. 基于计算机的系统的开发应当从分析核动力厂和系统的要求开始，此事由包括安全工程师和软件工程师在内的工程专家进行。然后以这种分析的结果为基础，导出对基于计算机的系统的要求。在最终要求确定之前，这种分析通常需要反复进行。由于这个阶段的遗漏可以导致对安全措施的要求规定得不正确，并由此可以产生不安全的系统，因此应当对系统的推导过程提供明确的证明。

2.7. 第一个设计决定应当是在计算机系统（计算机系统的要求）和供测量核动力厂变量以及驱动控制设备（其他部件）用的常规电气与电子设备之间分配基于计算机的系统的要求。设计人员或许有理由选用模拟设备实现某些功能要求¹。

1 本安全导则不讨论其他设备的评价问题。它集中精力关注计算机系统，尤其是在计算机上运行的软件。

2.8. 计算机系统设计过程中，要在软件要求和计算机硬件之间分配计算机系统的要求。然后将软件设计成一整套交互作用的模块，这些模块将作为能在计算机硬件上运行的代码被实现（软件设计和软件实现）。接着，将软件与计算机硬件集成生成计算机系统（计算机系统集成）。最后，把计算机系统安装到核动力厂中（安装），以便下一步的调试和运行。调试阶段的步骤之一是将计算机系统与其他部件集成（这是系统集成阶段的一部分）。

安全性和可靠性问题

2.9. 鉴于上文所提到的不利因素，对基于软件的系统进行可靠性定量评价的难度，要大于对非可编程系统进行类似评价的难度，这可能在证明基于计算机的系统的预期安全性时产生一些特殊的困难。目前，软件具有高的可靠性这类断言是无法得到证明的。因此，应当谨慎处理要求单个基于计算机的系统的软件故障概率低于 10^{-4} 这一类设计（参照参考文献[9]第8.2.2条）。

2.10. 既然软件故障本质上是系统故障而不是随机故障，因此对于安装了使用同一软件版本的冗余子系统的基于计算机的系统来说，共模故障是个严重的问题。不容易采取防范措施。设计人员假设子系统具有独立性，并使用多样性和综合质量评定策略来防止共模故障。但是，当涉及到软件时，或许难以估计这些策略的成功程度及具体的收益。

2.11. 随着现代技术的发展，原则上有可能为安全重要系统开发出具有必需可信性的基于计算机的仪表和控制系统，并充分证明它们是安全的。但是，只有小心地进行开发并全部有文件记录，才能证明可信性。这个过程可以包括遵照特定的要求对使用先前的软件获得的运行经验进行评价（亦见附录）。本安全导则提供了有关获得充分的安全证明的建议。

组织和法律问题

2.12. 对于基于计算机的系统的开发项目来说，存在着各种各样的组织方面和法律方面的问题。为确保项目的成功，这些问题应该在该项目的早期阶段就仔细地考虑。其中包括这样一些因素，诸如是否具备审批基于计算机的安全重要系统的合适的法律和行政管理体系，以及参与系统开发过程的单位是否具有足够的能力和资源。如果没有在项目的概念设计阶段仔细地考虑这些因素，则项目的进度和费用可能会受到相当大的影响。其中的某些因素可能会使这个选择不切实际乃至费用高得惊人，因而影响到是否使用可编程数字技术的决策。

2.13. 软件可靠性的量化是一个尚未解决的问题。软件测试有其局限性，对基于计算机的系统来说，软件可靠性的量化可能是很难或者说不可能证明的。监管部门有关安全性证明和对软件可靠性要求的立场，应当在项目规划阶段就尽早阐明。处理安全性和可靠性问题所用的方案应得到规定、形成文件、提供给监管者以及必要时征得同意。其中可能包括特定的监管停工待检点。

2.14. 应当确保监管机构、许可证持有者的设计组、监管部门的技术支持人员和供应商有足够的能力和资源实施有关软件开发过程及评定其安全证明的建议（例如本安全导则中的建议）。许可证持有者还应确保计算机和/或软件的供应商肯定愿意公开办理许可证过程所需的一切知识产权资料。

2.15. 许可证持有者（即计算机系统的用户）应当建立适当的组织来处理运行与维护问题。偶而的情况是，许可证持有者可能需要组建自己的队伍进行软件交付使用后的改进。这支队伍的能力和水平应当与由最初的生产者提供的队伍和设备的水平相同。

2.16. 这些因素会使所需技术资源大幅增长，并因此对费用产生影响。财务负担或许要高出一个数量级，以致选择基于计算机的系统的费用高得惊人。

3. 基于计算机的系统安全管理要求的应用

3.1. 多年来为非基于计算机的系统开发的大部分安全管理要求，可适用于基于计算机的系统。但是，鉴于软件与硬件的种种差别，并不总是能将这些安全管理要求直截了当地适用于基于计算机的系统，在许多情况下，这种适用会产生许多新问题。第3章简短地回顾了相关的大部分安全管理要求，重点讨论与设计、设计和开发、管理和质量保证、以及文件资料相关的一些问题。这些安全管理要求构成了导出基于计算机的系统要求的基础，尤其是非功能要求。鉴于文件资料在软件设计和安全证明中的重要性，因此文件资料应当是高质量的。

安全管理要求

设计的简明性

3.2. 应当证明无论在系统的功能方面还是在其实现方面都已避免出现一切不必要的复杂性。鉴于使用可编程技术允许实现较复杂的功能，因此这种证明对

安全来说十分重要，并且不是简单易行的。提供有关设计结构要严谨、编程规则和编码规则都已得到遵守的证据，应当是这种证明的一部分。

3.3. 对安全系统而言，打算由计算机系统满足的功能要求，对于实现安全功能来说应当全都是必不可少的；应当将非安全所必需的功能与安全所必需的功能分离，并证明它们不会对安全功能产生影响。

3.4. 对于基于计算机的系统的各种应用而言，自顶向下的分解、多层次的抽象和模块结构，是应付不可避免的复杂性所带来的问题的重要思路。它们不仅允许系统开发人员解决若干较小的、较易处理的问题，还允许验证人员进行比较有效的审查。系统模块化的逻辑和接口的定义应当尽可能简单（例如通过应用“信息隐藏”（见参考文献[4]的第3.3.4条））。

3.5. 在设计系统模块时，应当在复杂的算法中选择较简单的算法。不应为实现并不需要的性能而牺牲简明性。应规定安全系统中使用的计算机硬件应具有足够的容量和性能，以防止软件过于复杂。

安全文化

3.6. 从事安全重要性较高系统的软件开发项目的人员，除了应当包括计算机的软件和硬件专家外，还应当包括应用专家。这些专门人才的结合，有助于确保将工业上成熟且已经广为人知的那些安全要求有效地传达给计算机专家。应当确保所有人员都了解他们的工作与满足安全要求之间的关系，并且应当鼓励他们就可能危及系统安全性的活动或决定提出质疑。这意味着软件专家也应当对应用有很好的了解，能够使用适宜的规范语言（例如图形语言）来描述安全功能。

安全分级方案

3.7. 规定仪表和控制系统功能[2、3、9]安全重要性的安全分级方案，可用于系统开发过程。它指导核动力厂的设计人员、操纵员和管理部門适当重视那些确保安全的系统和设备的技术条件、设计、质量合格鉴定、质量保证、制造、安装、维护和测试（参看第3.8条）。

降低风险与开发力度之间的平衡

3.8. 在系统及其相关软件的每个设计阶段，应谨慎评定各种相互矛盾的设计目标达成的折衷方案。应采用自顶至底的设计和开发过程，以便进行这种评定。当分级设计和质量合格鉴定要求（当适用于计算机系统功能时）可从安全分级方案导出时（第3.7条），为了在设计工作和质量合格鉴定要求之间取得平衡，可以使用这种分级方法。计算机系统应当满足其实现最高安全级别的准则。

3.9. 为保证必要的置信度水平而采取的适当措施应当与每个安全级别相适应。应当注意到可以使用定量技术来评定系统硬件部分的置信度水平，但是，对软件可能只能进行定性评定。

纵深防御

3.10. 应当在基于计算机的系统及其相关软件的开发中使用核动力厂[1、10]设计中使用的纵深防御原则。如果主要的安全措施是由基于软件的系统组成的，则应当提供纵深防御，例如通过使用不同的二级保护系统。

冗余性

3.11. 就向通常在模拟应用中使用的一样，基于带有表决设置的冗余测量通道的传统设计对计算机系统的硬件有益。但是，这种冗余不能防止系统因在硬件和软件设计中可能导致所有冗余通道受损的共因故障而失效。

单一故障准则

3.12. 直接把单一故障准则应用于硬件的随机故障[1、11]：单一故障应当不会导致安全功能的丧失。但是，在软件中很难满足这种准则，因为导致软件失效的故障肯定存在于该软件的所有拷贝中（参见第3.13条）。

多样性

3.13. 通过采用多样性来降低出现软件共因故障的可能性，能够提高基于计算机的系统的可靠性。应当在设计的不同层次考虑采用不同的功能和系统部件。还应当考虑方法、语言、工具和人员的多样性。但是，应当注意的是，尽管不同的软件可能改进对共模软件错误的防护，但它不能保证不会发生同时出错。

应当证明许可证持有者有关使用多样性决策、选择多样性类型的决策或不使用多样性的决策的合理性。

故障安全设计、监督和故障容错

3.14. 应当在软件中添加系统故障安全特征、监督和容错机制，但仅能达到如下程度：新增的复杂性可通过安全性的显著整体提高来解释其合理性。当软件用于检查承载其运行的硬件时，还应当证明软件作出正确响应的能力。使用程控时钟之类的外部设备使系统对故障检测的响应更加可靠。只要可以实现，就应当使用防御设计、适当语言、包括安全子设备和编码技术来确保任何情况下的安全响应。计算机系统要求阶段的目标之一是十分详细地说明对输入的所有综合结果作出的既定和安全响应。

保 安

3.15. 应当证明已在基于计算机的系统的整个生存周期中对它采取了保护措施，使其免遭实体攻击、故意和非故意的侵扰、虚假信息、病毒等[12、13]。在不能保证其安全性时，不能将安全系统与外部网络相连。

可维护性

3.16. 基于计算机的系统的设计，应当便于故障的检测、定位和诊断，以便能对系统进行有效的修理或更换。具有模块化结构的软件更便于修理，也更便于检查和分析，因此这种软件的设计也更便于理解，更容易在不引入新错误的情况下进行改进。软件可维护性还包括实现功能变更的方案。基于计算机的系统的设计应当确保变更被限制在软件的一小部分。

运行模式的充分展示

3.17. 安全重要系统软件的要求和设计应当明确规定每种运行模式中输入和输出间的所有关系。软件设计应当简明，足以允许考虑代表了所有运行模式的全部输入的综合。

人-机接口和预期的人为局限性

3.18. 人-机接口的设计可能对安全有重要影响。人-机接口的设计应当能为操作人员提供足够的结构性信息，而不是太多太泛滥的信息，以及足够的反应时间（例如，见参考文献[2]的30分钟准则）。在向操作人员分派任务时，计算机系统应当为操作人员留出从提供的信息中获得有关人的行动指令所需时间。作为防止因操作人员错误造成损伤的防御手段，应当检查所有操作人员的输入是否正确。应当谨慎研究操作人员越过这种正确性检查的可能性。

可证明的可信性

3.19. 系统不仅应当是可信的，它还应当可能向监管者证明它是可信的。本安全导则向许可证持有者建议，如何通过提高可追溯性的设计和质量合格鉴定方法以及制订足够的文件来实现可证明的可信性。

可测试性

3.20. 为了确定每个要求和设计特性是否已经得到正确实现，应当说明每个要求和设计特性的测试方法。功能要求和非功能要求都应当是可测试的。测试结果应当可追溯到相关要求。

设计和开发活动

3.21. 在确定必要的设计和开发活动时，应对下列主题给予特别关注（第3.22—3.27条）。

逐步控制过程

3.22. 应当对设计和开发过程进行逐步控制。该开发过程能够通过其每一步建设来证明其正确性。这还能便于实施验证过程，并确保在设计过程初期发现错误。

可审查性

3.23. 应当对开发过程的所有阶段准确编写便于审查的文件。用于向监管者证明开发过程适用的文件应当与在设计中实际使用的文件相同。

综合测试

3.24. 应当进行综合测试。测试是开发、验证和确认工作的重要组成部分。应当提供测试范围的证明，包括从测试案例追溯到原始文件。应当将测试结果、测试范围的证明及其他测试记录提交第三方监察。应当制订综合测试的计划，并且在项目的初期提交给监管者。

自动工具的使用

3.25. 使用的所有工具应当相互兼容。工具应当经过鉴定使其适于软件开发和安全证明过程中要求的功能。应当对获得这些工具的置信度所采用的技术作出详细说明，并形成文件。

可追溯性

3.26. 各种要求应当可追溯到设计；设计应当可追溯到编码；要求、设计和编码应当可追溯到测试。在作出变更时应当保持可追溯性。还应当具有反向可追溯性，以确保没有非预期的功能。

与标准的符合性

3.27. 应当确定在开发中使用的安全管理、安全要求和技术标准。应当对计算机系统涉及的技术条件和实现中使用的重要标准[5]进行符合性分析。

管理和质量保证

明确规定工作人员的职责和资格

3.28. 核动力厂管理应当证明工作人员的水平是足够的。管理者应当确定参与软件设计、生产和维护的工作人员职责和资格，应当保证只有合格且有经验

的人员参加了这些工作。应针对在软件开发和维护过程中的每项任务进行人员资格鉴定，包括质量保证大纲的实施[14]。

可接受的作法

3.29. 在软件开发过程中应当使用沿用已久的方法、语言和工具。不应使用尚处于研究阶段的方法、语言和工具。

质量保证大纲

3.30. 组织的质量保证大纲应当扩展到软件开发过程，还应涵盖软件交付后的配置管理和变更控制。与独立的辅助监察一起，至少还应当涵盖安全系统、独立验证、确认和测试。应当在软件质量保证大纲中规定质量保证大纲所规定的软件质量要求。

责任的分配

3.31. 应当在设计组织内部以及设计组织、其客户和参与系统开发的其他组织之间建立接口。应当建立有关设计接口的控制，并应当包括责任分配以及发放进、出接口组织的文件。

第三方评定

3.32. 应当对安全系统进行第三方评定。应当在描述质量保证大纲时详细规定评定的范围和程度。执行质量保证以及验证和确认任务的工作组应当独立于开发工作组。本安全导则的下文将涵盖这些问题（第3.33和4.17条）。

3.33. 独立于供应商和用户（许可证持有者）的第三方评定的目的是提供对系统及其软件适用性的看法。此类评定可以由监管者或监管者认可的实体进行。评定应当提供对许可证持有者和供应商的生产过程的置信度。为提供必要的置信度，应当仔细考虑第三方评定所必备的评定策略、能力和对项目的了解。此外，参与各方（监管者、许可证持有者、供应商）应当一致认可第三方评定，因此可在既定时限内有适当的资源可供使用。一些第三方评定应当包括过程的检查（例如通过质量保证监察和技术检查）。其他第三方评定应当包括产品检查（例如通过静态分析、动态分析、编码和/或数据检查以及测试范围分析）。

应当进行有关软件最后版本的最终产品评定（鉴于时间限制，应尽可能）。这可能包括对开发过程的中间产品进行第三方评定，例如软件规范。

文 件

3.34. 对软件产品可靠性的置信度，应当提供有关开发过程完好性的证据。文件在提供这种方法所需的“透明度”和“可追溯性”中将发挥重要作用。有关可信赖软件产品的设计和实现的文件应当明确而严谨。

3.35. 一套文件应当保证设计决策的可追溯性（14、Q3）。应当在开发过程的每个阶段都编制适当的文件。应当以迭代开发的方式来更新文件，包括调试和不断进行的维护过程。监管者获得的文件应当与设计人员使用的文件相同。应当在项目初期将文件交给设计人员。

3.36. 有关要求、设计和编码的文件应当明确而严谨，以便设计人员、程序员和独立的审查人员能够全面了解开发的每个阶段，并验证其完整性和正确性。

3.37. 好的文件对维护也是非常重要的。应当使用适当的文件格式，以便在未来与维护相关的变更中降低发生不一致和错误的可能性。与第3.38—3.44条所描述的一样，文件应当具有可读性、严谨性、可追溯性和完整性、一致性、以及可验证性和可修改性。

可读性

3.38. 文件应当能被具有不同背景和专门知识的人所理解。文件中使用的语言应当明确，当采用形式语言（包括图表形式）时，应当使用定义明确的语法和语义。

严谨性

3.39. 应当使用形式语言描述（使用定义明确的语法和语义）要求和设计说明，并用自然语言给出解释。每条说明或每条要求都应当只能有一种可能的解释（见参考文献[4]的第5.1.2条）。

可追溯性和完整性

3.40. 可追溯性的目的是证明计算机系统的要求和设计的实现是完整的，并有助于检测实现过程中的不安全因素。从上级文件追溯到软件文件核查了文件的完整性，从软件文件追溯到上级文件核查了可能不安全的未指明物项。应当惟一地标识每条要求。

3.41. 应当采用可清晰显示基于计算机的系统的要求和在计算机系统要求、计算机系统要求、软件要求、软件设计和软件实现中实现基于计算机的系统的每条要求的物项间联系的可追溯性矩阵。该矩阵应当证明基于计算机的系统的要求在实现、集成、安装和调试过程中的测试范围十分完整。

一致性

3.42. 文件不应当包含相互矛盾或不一致的内容。在文件中，每条信息都应当有一个惟一的可识别段，一条信息不应当重复出现，或被分散在2个或更多的段。每条要求、每个设计单元或程序模块都应当有惟一标识符（这也有助于实现可追溯性）。在文件编制过程中应当用相同的方法来使用注释、术语、注解和技术。

可验证性

3.43. 当文件可以理解、不含糊和可追溯时，可验证性就得到了改进。在软件要求和软件设计中对计算机系统要求使用相同的模型或语言也有助于可验证性的改进，但不是对所有问题都必须采用一种解决办法。在使用形式语言来详细说明要求或设计时，定理证明程序和模型检查程序可能也有助于可验证性的改进。

可修改性

3.44. 文件应当是可修改的，即对文件的结构和风格做必要改动，使其更容易编制、更加完整和一致，并且更便于识别。

4. 项目规划

4.1. 为了促进安全重要系统的许可证审批，应当谨慎规划开发过程，并提供证明该过程已得到遵照的明确证据。项目规划应当以一套针对基于计算机的系统的综合的专项安全计划或涵盖项目所有内容的一套计划的形式进行文件编制并存档。第4章描述一项单独计划的每个方面，但它同样适用于在单份文件中涵盖所有这些计划的情况。不应将系统的修改排除在外，应当规定就安全分析进行尽可能多的迭代。

4.2. 开发计划应当详细说明一组开发活动以及每项活动的基本特性。项目应当规划的其他内容质量保证、验证和确认、配置管理以及调试与安装。图1显示了本安全导则假定使用的基于计算机的系统的大致开发过程（有关选择特定开发模型的资料见第2章）。图2说明了按照要求、设计和实现进行验证和确认的关系。

开发计划

4.3. 开发计划应当列出和详细规定打算用于特定项目的开发过程。第4.4—4.10条简要地说明计算机系统开发计划应当涵盖的内容。

阶段

4.4. 应当将开发过程的所有阶段（图1）列出。举例子来说，计算机系统设计每个阶段都由规格说明、设计和实现组成。一个阶段的设计活动为下一阶段设置要求，对某个阶段的要求是前一阶段设计活动的一部分。例如，实现是一个选择现成编码（包括库存程序）以及编写所需的任何附加代码的过程。在开发工作的每个阶段中间或下一阶段开始前，都应进行验证（见图2和第4.12—4.18条有关验证和确认计划的内容）。

方法

4.5. 应当列出准备在开发工作中使用的方法。方法的选择应当与建立了标准和规程的质量保证大纲说明有关。

工 具

4.6. 应当在开发计划中列出准备使用的工具。应当以有利于正确应用所选定的方法、标准和程序为原则选择工具。对整个项目预先进行规划将有助于选择一套完整的工具。这些工具对于它们要在系统的开发、管理或验证工作中发挥的作用来说应当是完全合格的。应当通过工具鉴定过程、交叉验证或反求工程来确保其产出的正确性。

4.7. 工具是应该使用的，因为它们可以免除工作人员从事编程和验证之类易出错的手工作业。它们有助于达到可再现的质量水平，这种质量水平是依靠过去应用这些工具时所获得的证据部分地得到保证和证明的。

文 件

4.8. 应当确定每个阶段准备产生的文件，并规定文件的内容。应当指出哪里能找到所有的要求、质量属性和性能特点，整个项目将来要使用哪些验收标准。文件应当提供该项目完全是按开发计划行事的证据。

进度和里程碑

4.9. 应当建立有关文件的时间表，在项目审查时应确定时限。管理任务的产品包括：

- 评定资源的可用性；
- 估计每个阶段的持续时间，将迭代包括在内；
- 评定培训要求；
- 评定可利用的设施和工具的适宜性；
- 估计监管部门的审查和批准所需时间；
- 估计关键点的项目审查所需时间。

人 员

4.10. 应当制订计划，以确保参加开发活动的人员有能力使用相关标准、规程和方法；有能力使用设计、编程和分析工具与方法；在配置管理和变更控制的实践中也具备相应能力。应当保存有关他们能力的记录。如果有新成员加入工作组，他们应受到严格监督，直到他们已向其直接管理人员证明他们的能力。

质量保证

4.11. 应当由许可证持有者起草和实施质量保证大纲说明[14、15]，并且应在项目开始前提交给监管者审查（和可能的批准）。应当在项目开始时产生软件质量保证计划。该计划应当涵盖外部供应商，并至少包括以下内容：

- (1) 确定适用于质量保证大纲说明的硬件和软件物项以及在该项目中使用的指导标准、规程和工具；
- (2) 对于每份编制的文件，在质量保证大纲说明中指出应当由谁来审查和批准该文件，以便正式发放；
- (3) 包括确保质量保证监察员独立性等内容的项目组织结构说明；
- (4) 对参加项目的人员的能力和培训要求的说明；
- (5) 不符合（标准和程序）项的识别、报告和纠正机制；
- (6) 确定所有必要计划，诸如开发计划、验证计划、配置管理计划、以及调试与安装计划。
- (7) 根据系统的安全级别指出质量保证监察的次数和范围；
- (8) 证明工具合格的规程（参见第4.6条）；
- (9) 检验由外部供应商提供的元件质量的机制；如果该机制要依靠外部的认证程序，例如型式试验，那么还应当将这些认证规范的说明包括在内。

验证和确认计划

4.12. 系统正确性和安全性的证明要求实施各种各样的验证和确认活动[16]。在基于计算机的系统的生存周期中，验证是检查产品是否满足上一阶段的产出，确认是检查产品是否满足较高层次的要求和目标（图2）。

4.13. 应实施确认，以证明计算机系统已实现其总体安全要求和功能要求。确认可明显地分为两步：第一步是确认计算机系统的要求是否满足核动力厂要求和系统要求（参见第2.6条）。在确认报告中应明确对这些较高层次要求和安全分析进行确认的基础。第二步是对照计算机系统的要求确认计算机系统的实现过程。应当在验证和确认计划中确定技术和清楚的规程。

4.14. 应对下列开发阶段的产品进行验证（正如本安全导则规定的一样）：

- (1) 计算机系统的设计；
- (2) 软件要求；
- (3) 软件设计；
- (4) 软件实现。

4.15. 应当在验证和确认计划中规定用于软件验证的技术，应当明确这些技术的规程，并应证明这些技术对系统安全级别的适合性。预计验证和确认将是许多技术的综合，包括文件的静态检查和实现的动态执行。

4.16. 应当确定系统的整个生存周期中应当保存哪些验证和确认的结果记录。这些记录应当提供结果已记录的、异常已调查过并解决的所有已实施规定活动的证据。应当将这些记录提交给第三方，供监察和审查。这些记录应能清楚证明从所有验证和确认工作到有关原始文件的可追溯性。

4.17. 应当在计划中列出进行验证和确认的工作组。应当在工作组之间分配验证和确认任务。实施验证和确认的工作组应当独立于开发队伍。应当证明有关系统安全级别的独立验证和确认的次数和类型的合理性；例如，对于安全有关系统而言，可以不要求经费独立性。有关的独立性包括：

- (1) 技术独立性；这项工作应当由不同的人、使用不同的技术和工具完成；
- (2) 管理独立性；这项工作应当由不同的人来领导和推动；验证和确认工作组和开发工作组应当有不同的管理渠道；独立工作组之间的正式交流应当记录在案。
- (3) 财政独立性；在开发及验证与确认间的资金转帐应有限制并单独预算。

4.18. 验证与确认计划应当包括一种机制，用于记录在分析期间发现的一切不符合情况，并确保这些不符合能通过变更控制过程得到适当解决（第4.23—4.24条）。

配置管理计划

4.19. 配置管理是开发计划和质量保证大纲说明的延伸，是非常重要的一个问题，值得在此进行单独介绍（亦可参见参考文献[15]的第5.2条和有关配置管理的现行标准）。

版本控制

4.20. 应当将软件开发的所有物项，例如编译程序、开发工具、配置文件和操作系统，置于配置管理控制之下。所有可识别的物项，例如文件、软件部分或数据结构，都应当给予包括版本号在内的惟一标识。这些物项应当包括已开发的物项和正在被重复使用或重新运用的现有物项。应当有专门保存处于配置控制之下所有物项的信息库或存储区，以便能够在其现有的任何版本中发现和检

索任何已标识物项。应当有描述何时以及怎样将已开发的或已获得物项置于配置控制之下的规程。应当有凭此能在时间表的特定点确定一组代表后续工作“基线”的配置控制物项的机制。每个物项都应当有包含相关资料的记录，例如载有它何时完成，与上一个版本相比纳入了什么变化，它目前的审批状况以及负责建立、审查和批准的人员。应当有与质量保证大纲说明相关的核准程序。

4.21. 要求在规程及数据库之间建立适当链接，以进行软件和硬件的配置管理。硬件配置的变更可能影响软件的验证和确认活动。

4.22. 在任何时候，质量保证监察员或监管监察员都应当能够要求得到和能够直接获得物项组中的任何物项，这些物项是对系统及项目（现行基线）的最新版和最全面的描述。这组物项应当与作为正在进行的开发或分析活动基础的现行物项相同，除非进行中的工作尚未获得批准或放行。

变更控制

4.23. 一旦物项已置于配置控制之下，该物项就只能根据包括影响分析在内的定义明确的规程进行变更，该规程应当能够产生新版本的物项，而不是取代现有的已标识物项。变更控制规程应当保存有关以下内容的记录：变更的原因、怎样分析已确定的问题、哪些物项可能会受到影响、作出了哪些专门变更来纠正问题以及为解决问题产生了哪些版本和基线。变更控制规程还可能确定审批变更的责任，如果增加或改变了基本物项的审批机制。通常，变更应当涉及到重复最初用于建立物项的所有过程，包括从受到影响的最高层次物项开始的所有分析。应当采用回归分析，以确定保持验证和确认记录所需的测试。回归分析规程及其结果应当记录在案。

4.24. 在开发工作完成（并交付核动力厂）之后，可以采用不同的变更过程，因为负责维护的组织和人员可能不是最初的开发人员。这个问题将在15章进一步讨论。在适用维护变更过程之前，应当编制并保存有关变更规程的文件，并应当考虑是否需要重复开发期间实施的分析。应当编制并保存有关不重复任何上述分析的任何决策的文件并证明其合理性。

安装和调试计划

4.25. 在作为独立系统进行了建造和确认后，还应当将该系统与核动力厂的其他系统集成，并在实际核动力厂环境中进行测试。应当谨慎规划安装和调试过

程，以便协调从开发到使用的转移和从开发人员和验证人员到用户与维护人员的交接。

4.26. 安装和调试计划应当涵盖下列内容：

- (1) 将系统正确集成到核动力厂中的步骤序列以及为安全纳入新的或变更过的系统所需的核动力厂相应状态；
- (2) 与监管者必要的相互配合，包括在系统能够投入运行前应当注意的任何审批点或控制点；
- (3) 调试测试用例和顺序以及证实核动力厂环境中系统正确运行所需的核动力厂相应状态；
- (4) 其他部件与新的或现有的核动力厂系统之间的接口、以及检查每个接口正确有效运行所需的测试；
- (5) 任何试用期的持续时间；
- (6) 将产生的用于描述调试结果的记录和报告的说明；
- (7) 指导和通知用户和维护人员过程的启动；
- (8) 配置管理和变更控制过程从开发人员到维护人员的交接，包括在调试期间发现的任何异常的解决机制。

5. 计算机系统的要求

5.1. 对计算机系统的要求至少应规定计算机系统的功能特征和非功能特性，这些特性对满足核动力厂在较高设计层次上已规定的安全要求来说应是必要而充分的。计算机系统要求的技术条件是对计算机系统必要特性的说明。与操作人员、维护人员和外部系统的接口的精确定义是作为此阶段输出结果的组成部分。在此阶段，接口的定义仅限于这些接口的功能特征和非功能特征；它们的设计或实现可能尚未确定。

5.2. 计算机系统要求的确立是基于核动力厂高层次设计的成果（图1）。事故分析、瞬态分析、或核动力厂安全分析之类的安全分析（基于假设始发事件和要满足的安全准则）是该设计的基本组分。除了安全要求外，在这个设计阶段要增加一些与安全没有直接关系的补充要求，例如对可用性的要求。因此，安全系统要求的定义是不同学科专家共同努力的结果。设计部分的内容超出了本安全导则的范围，但它提供了这些系统要求应当涵盖的保护动作和性能准则的技术条件。反之，还应当确认系统要求是否满足这些技术条件，以确保完整性和一致性。

5.3. 这些要求的推导是开发过程的关键步骤，因为如果没有在这个阶段被发现，这个阶段的错误和缺陷将最终向确认过程发出挑战。此外，计算机系统要求中的缺陷是在包含相同软件的冗余子系统中引发共因故障的潜在因素。

建 议

概 述

5.4. 计算机系统要求应当与其实现无关。它们应当将计算机系统当作黑箱。应当采用所有有关各方都能理解、带有定义明确的语法和语义的规范格式来记录工作成果。对计算机系统和软件规范的许可证发放者、供应商和设计人员而言，这种格式应该是可以理解的。这种格式应当能够得到有助于确认要求的完整性和一致性的工具的支持。

5.5. 在项目的下一个阶段开始前，应当准确而清楚地书面描述这些要求。这种描述对于所有监管者和许可证持有者的下述有关专家应当是可以理解的：过程工程师、安全工程师、计算机系统工程和核动力厂工程师。

5.6. 计算机系统要求的说明应当易于使用，以便验证计算机系统和软件技术条件的恰当性，并规定与在设计完成时对系统进行确认有关的测试规范。

5.7. 这些要求的说明应当基于系统的精确模型，并且有与核动力厂环境的接口。参考文献[4]第5章给出了根据受到监测和控制的变量建立的此类模型。

确 认

5.8. 应当对系统要求进行确认，即应当在其与由核动力厂安全分析所产生的规范的联系中建立正确性、一致性和完整性。

5.9. 应当对分析计算机系统要求的一致性和完整性时所用的工具进行确认，以便证实其置信度达到如第5.10—5.23条中描述的设计过程的要求。

系统接口

5.10. 应当将系统接口设计得有利于操作人员参与保护活动，例如在反应堆紧急停堆后的人工接替、介入和手动复位。输入数据和用户命令应当由操作人员重新确认，并且由本系统进行确认后才可供处理。

5.11. 当操作人员接口涉及可视显示设备时，应当注意在设计时应给予足够的响应时间、引导和帮助设备[2、3]。每个显示区应当按给定的模式和前后关系仅用于一个目的。例如，应当避免使用一个显示区显示不同工程设备的同一参数。所有接口都应保持一致，应当使用同样的名称和标识符。应当进行任务分析，以证明手动安全措施和相关的任何显示信息的适宜性。

5.12. 系统接口还应当设计成有利于在役检查但又不会导致假保护动作。

5.13. 与核动力厂网络的接口之类的系统接口，应当设计成不妨碍由外部系统履行的其他保护功能。外部系统或支持系统的失灵和故障应当促使系统进入安全状态。

5.14. 应该精确定义系统的边界，即定义基于计算机的系统和核动力厂之间的接口。尤其应当详细说明系统与传感器和驱动器、操作人员、维护人员和其他的任何外部系统的接口。

5.15. 应当谨慎地详细说明输入、输出和数据显示的格式和警报的组织。

5.16. 应当将诸如来自软盘的自动加载数据对照原始数据进行机器验证。应当为人工加载的校准数据提供回读设备。应当禁止系统不加警告地自动生成缺省值。

5.17. 对于替换系统，实际环境和以前安装的系统对环境变量和参数的数值设置了限制。应当识别这些限制，并记录在案。尤其是，应当解决与核动力厂中现有系统的兼容性和防止非安全系统故障扩展的措施。

功能安全需求

5.18. 对于安全系统而言，为满足核动力厂的安全要求所必需的（也是充分的）系统功能，应给予规定。这些功能安全要求应表示为对系统维持核动力厂参数的附加限制，即保持参数在规定的限值内，以及表示为系统对核动力厂的影响。

5.19. 还应当对安全有关系统进行安全分析，以确定功能安全要求。

非功能需求

5.20. 应当详细说明下列非功能需求：

- 系统工况特性，即对工况有影响的环境、准确性、时间和性能限制（时间和性能限制应当在核动力厂安全分析确定的容许范围内）。

- 系统工况要求的有关可信性特性，诸如可靠性、有用性和保安。尤其应当考虑对基于计算机的系统给予可靠性限制。应当从已经为基于计算机的系统环境制订的安全政策推导保安要求，并应该考虑应当执行的保安规程（参见第14.5条）。
- 坚固性，即基于计算机的系统将怎样对核动力厂接口的潜在故障做出响应，例如传感器故障、或输入超出预期范围。
- 是否需要或在哪里需要实体分隔（例如安全功能和控制功能之间）。
- 在必须满足可靠性要求时，在系统设计中纳入多样性形式。能够详细说明各种检测、测量、表决和驱动方法的不同特点；对同样的假设始发事件做出响应的不同功能；以及独立的各种系统部件。应当对照每种功能的可靠性要求评价必要的多样性类型是否合适。
- 什么情况下可能需要更换的某些系统零部件。

分 析

5.21. 对安全系统而言，应当表明所有的功能安全要求和非功能安全要求都是核动力厂安全分析的结果，并都由其决定。

5.22. 作为系统安全证明的一部分，应对计算机系统要求进行失效分析。应当识别潜在的错误系统输出，评价它们对核动力厂和环境的影响，并证实纵深防御的适宜性。

5.23. 应当识别潜在的共因故障源。应当识别类似信号电路能够存在于一条以上通道或子系统的运行模式。就这方面而言，尤其应当注意系统的零部件或将用软件实现的功能。

文 件

内 容

5.24. 这个阶段编制的文件应当涵盖下列物项：

- 功能要求规范；
- 非功能要求规范；
- 操纵员、维护人员和外部系统的接口规范
- 事故后分析需要的档案记录和数据；
- 有关确认测试及其范围的规范；

- 有关计算机系统要求的确认报告。

功能要求规范

5.25. 应当以与实现手段无关的形式详细说明功能要求，例如根据应当由系统保持的功能关系。这些功能关系应当详细说明以下各项：

- 系统不能违反的由环境和其他系统施加的实体的、天然的或其他限制引起的规定环境变量之间的关系；
- 当基于计算机的系统在这种环境中运行时，应当由该系统建立和保持的监测变量（工艺参数、操作人员信号）与控制变量（给驱动器和指示器的输出）之间的附加关系。

5.26. 应当以能反映系统功能的方式安排这些关系，以便将必要的映射纳入到计算机系统结构的组织以及将在以后阶段定义的软件模块化结构的组织中。

5.27. 应当使用标准的数学工程符号来表示变量。应当谨慎定义这些变量，例如借助图表、坐标系、符号或比例尺。

5.28. 还应清楚确定系统可能必须采用的各种运行模式（状态组）及其类别，尤其在那些模式中的系统、核动力厂和操作人员之间采用不同接口的情况下。

5.29. 应当使用定义明确的格式来描述功能关系。为此目的可以使用包含输入和输出之间的功能关系的决策表，尤其是在验证和定义确认测试的过程中。

非功能要求规范

5.30. 对安全系统而言，应当采用可证明的方式从核动力厂安全分析中推导规范，规范应当以文件的形式记录时间限制、性能限制和可信性要求。

5.31. 除详细说明所需的可靠性和可用性外，可信性要求还应用文件记录并证明下列事项的合理性：

- 要求的零部件质量合格鉴定；
- 适用的安全要求，例如单一故障准则；
- 对实体的和功能隔离的要求（例如，利用一个以上的核动力厂参数和独立的参数处理过程来检测假设事件）。

操作人员、维护人员和外部系统的接口规范

5.32. 应当针对系统工况的所有可能的不同模式，清楚描述系统接口的功能规范和非功能规范。可以使用诸如用于描述通信协议的专门模式和规范方法。

事件后分析所需的档案记录和数据

5.33. 应当在文件中详细说明供事件分析使用的档案记录和数据保存的频率和必需的准确性，以及是否需要连续或仅根据规定的核动力厂工况生成此类数据。

确认测试及其范围的规范

5.34. 测试规范应当覆盖系统及其与核动力厂接口的全部功能，并应当表明规范来源于上述功能要求和非功能要求。

5.35. 对安全系统而言，测试规范还应当详细说明在系统层次统计有效并具有要求的可靠性的测试（参见第2.9条）。

5.36. 测试规范应当陈述预期的结果。

5.37. 对安全系统而言，应当由未参加系统要求规范制订的人员对测试规范文件进行独立验证和审批。这些测试规范文件还应当可供监管者审查。

计算机系统要求的确认报告

5.38. 确认报告应当包括下列内容：

- 来源于核动力厂安全分析的规范，并已根据该规范对系统要求进行确认；
- 实施此类确认的步骤；
- 此次确认的结论。

5.39. 确认报告尤其应当是可审查的，并应尽早提交给许可证发放机构。

文件格式

5.40. 有关文件的一般建议见第3.34—3.44条。

6. 计算机系统设计

6.1. 计算机系统设计的最初版本来源于有关计算系统要求的系统要求图（系统图、结构图或形式图），这些系统要求通过下列物项的适当组合来满足：

- 现有的或预先开发的软件或固件（例如操作系统）；
- 硬件或因应用而异的专用集成电路；
- 有待通过配置预制软件开发或生产的软件。

对计算机系统设计后续改进，将会由于第6.3—6.6条中讨论的活动而增加一些新内容。

6.2. 第7章介绍了如何改进计算机系统设计规范中与软件相关的内容，以制订软件要求。计算机系统设计规范中与硬件相关的内容，也应当以类似的方式进行改进，并选择适当的标准（例如参考文献[17]），但这种活动超出了本安全导则的范围。

建议

6.3. 选定的计算机系统体系结构应当提供计算机系统要求规定的系统接口，并应当实现计算机系统的非功能要求，例如性能和可靠性方面的要求。

6.4. 对设计、验证和确认以及维护目的，需要将“新增”要求纳入到在适当抽象层次的相关规范文件中。这些要求可能包括：

- (1) 由有关计算机系统设计或软件设计的低层次决策引起的要求；
- (2) 由有关计算机系统以外的系统零部件设计的低层次决策（例如选择需要软件补偿或硬件滤波的仪器）引起的、可能对计算机系统产生影响的要求。

6.5. 无论这些要求是否会在转换为系统要求规范文件或较低层次的规范文件的过程中得到反映，都应当有保持有关这些新增要求的配置控制的方法可供使用，以便能评定它们的确认范围是否必要。

6.6. 应在计算机系统设计文件中论述证明计算机系统是否足够安全用的方法。应当考虑在第6.7—6.26条中列出的设计问题。

安全与非安全的隔离

6.7. 安全系统应当致力于执行发挥其安全功能的任务。在必要以及证明非安全功能是安全系统组成部分的合理性时，应当分析整套系统是否应该归为安全系统，且安全功能不应当受到其他功能的影响。

6.8. 由于简明性有利于可靠性的实现，因此应当考虑是否把涉及安全功能的部件与其他系统隔离。这可以通过转移计算机系统的非安全功能和部件、采用分布式计算机系统或在集中式计算机系统内设置合适的“防火墙”来实现。

6.9. 隔离后，可以使用限制较少或资源较密集的方法来执行非安全功能和部件。此外，安全功能和部件的隔离还有益于满足单一故障准则，因为此时的非安全功能和部件故障不会影响到安全功能和部件。当然，应当注意确保隔离已经实现。

6.10. 安全功能和部件的隔离可以导致增加整个系统的复杂性。为实现隔离而在接口处增加的复杂性，应当不超过预期的复杂性减少。计算机系统体系结构越复杂，证明其安全性就越困难。

6.11. 万一发生安全功能和非安全功能共同访问某个硬件或软件部件（例如数据链）的情况，部件故障不应妨碍安全功能的执行。如果共同访问可能损害安全功能，则在计算机系统设计应该加以避免。

冗余性、通道化和表决逻辑

6.12. 通常应该为计算机系统提供冗余性，以满足可靠性和联机测试的要求。冗余性能使系统处理随机硬件故障，但它不能防止共因故障的发生。

6.13. 为了降低冗余设备发生交叉故障的概率，应当在电路和实体上对冗余子系统加以隔离。即使在部件失效或子系统因维修故障部件而停止使用的情况下，通过对每个子系统获得的结果进行表决，也可以获得总体上正确的结果。由于表决逻辑本身必需是所有子系统共同使用的，因此应当使用适当且足够的隔离电路来确保子系统冗余设备保持隔离。此外，冗余子系统应当以异步方式运行。

6.14. 对集成计算机系统进行调试应当考虑过载要求，因为调试可能影响硬件-软件接口。

多样性

6.15. 为了降低发生共因故障的可能性，应当将多样性纳入计算机系统体系结构中，以达到对整个计算机系统的安全性和可靠性要求所必需的程度为限。设备多样性意味着使用不同类型的设备来履行冗余功能。功能多样性意味着使用不同方法来完成某一特定功能。

6.16. 目前尚不完全清楚软件能够在多大程度上实现多样性并且没有共因故障。与使用相同的软件相比，在冗余系统中使用不同的软件（将在第9章详细介绍）能够降低共因故障发生的概率。但是，目前还没有一种办法能足够准确地预测多样性的效果。测试结果已经表明，在不同的版本中，相同的软件缺陷数量占软件缺陷总数的10%—100%[18]。在决定硬件和软件之间的功能分配方案时，应考虑这方面因素以及能够要求的软件可靠性限值。应当考虑的另一个因素是表决逻辑是一个复杂的问题，对于软件来说尤其如此。

故障检测能力和故障安全能力

6.17. 应当把计算机系统设计得能够检测到可以影响其发挥安全功能的内部和外部故障。如果发生了这类故障，计算机系统应当默认“安全状态”，即使是计算机系统及其输出驱动器发生供电中断。判断是不是处于安全状态并不总是简单易行的，很可能随着核动力厂情况而变（在有些情况下，计算机系统最安全的响应是没有响应，向核动力厂运行人员发出告警信号除外）。

6.18. 如果特定类型的某起故障没有严重到需要作出故障安全反应，则计算机系统仍然应当向核动力厂操作和维护人员发出警告，以便使他们能够在这种小故障与其他多个故障叠加成确实有安全意义的失效以前消除该故障。

故障容错

6.19. 计算机系统的硬件和软件体系结构应当能保证系统在数量有限的硬件和软件故障发生后仍能够按预定的安全方式运行。应当考虑诸如使用备用数据源和冗余体系结构等技术。应当在计算机系统设计中详细介绍这些技术和相关规程的局限性，并将其反映在核动力厂运行规程手册中。

系统过程监测和设备监测

6.20. 设计计算机系统体系结构时，除了应当考虑监测计算机系统本身部件工作状态外，还应当考虑监测核动力厂各个系统过程和核动力厂设备。例如，在有必要比较来自若干通道的相似信号以辨别出核动力厂过程故障信号的控制系统中，就应当确定在不危及通道隔离的情况下完成该项任务的方法。

运行时的可测试性

6.21. 应当将安全重要系统中使用的计算机系统设计得能够在核动力厂运行期间进行定期测试，以便验证该系统的连续可操作性和可靠性。这可能需要调整计算机系统的体系结构。

6.22. 应当根据安全分析的结果（分析时已经确定了必要的测试间隔时间）确定在役测试的频率。应当在计算机系统设计时确定在役测试的范围，并证明该测试范围对测试的目的而言是合适的。

准确性和响应时间

6.23. 组成计算机系统的硬件和软件的组合及其他部件，应当达到要求的准确性和响应时间，以满足总体的系统性能要求。准确性和时限分析应当考虑一切可能的时延或信号失真，这些时延或信号失真有可能是从检测核动力厂物理状态的传感器到最终的控制驱动机构这条链上的所有元件引入的。其中包括软件引入的和在计算机系统硬件—软件界面处引入的错误或时延。尤其应当考虑的是，利用计算机系统的硬件或软件调整机组的信号（诸如滤除噪声和消除继电器触点的不稳定性）可以导致响应时间的缩短。

6.24. 为了帮助识别和分析系统的响应问题，应当对计算机资源采用确定性通信协议和规划技术。分析应当包括考虑情况最坏时（例如同时使许多输入信号发生变化并要求许多输出驱动的事故工况）对系统的要求。计算机系统内的通信体系结构越复杂，提供令人信服的安全证明就越困难。可能有助于提供这种证明的技术是分析通信过程的那些技术（诸如通信系统的算法、Petri网或状态转换图[4]）。

非功能要求

6.25. 计算机系统的设计要求还应当涵盖计算机系统非功能方面的要求，诸如有能力应付极端的温度、辐射照射或静电（当可适用于特定的装置（即设备鉴定）时）、抗射频干扰性和抗电磁干扰性（例如通过电源、信号、通信或接地线传导的噪声或电涌产生的干扰）。应当注意，抗射频干扰性和抗电磁干扰性只能在已安装好的设备上才能得到证明。

6.26. 应当在计算机系统设计纳入未来更换该系统的某些零件的需要。

文件

计算机系统体系结构

6.27. 有关计算机系统设计文件，应当载有对选定的计算机系统体系结构的明确说明。此外，文件还应证明计算机系统设计在要求的系统可信性和实现所有功能要求的能力方面是合理的。使用结构化方法所获得的证据，能够部分地提供此类合理性证明。此外，选定的体系结构应当证明在概念的简明性和满足性能要求能力之间是平衡的。如果可能，使用模拟与分析、正式的方法迭代或制作样机，也能够部分地证明计算机系统设计的合理性。此类分析的结果应当在有关计算机系统的文件中介绍。

6.28. 除了对系统要求规范中的计算机系统的功能和性能要求进行细加工外，有关计算机系统设计文件应当涵盖由以下各节导出的补充要求（第6.29—6.40条）。

系统的在役测试

6.29. 有关将来如何进行在役系统测试（例如联机的或脱机的、自动的或手动的在役系统测试）的决定以及将来如何满足维护需求（例如通过校准、隔离和修理已失效部件，使用已安装的备用设备以及借手动断开通道进行维修）的决定，除了增加对软件功能的要求外，必然还会使硬件接口增加。

6.30. 选择什么样的介质储存软件逻辑（例如随机存储器（RAM）还是只读存储器（ROM）），将对为确保计算机系统的持续完整性而应该实施的联机或定期脱机检查产生影响。

评估对通告和操纵员接口的要求

6.31. 人因工程师应当确定（例如通过人因任务分析）通告的性质和提供通告的设备（例如通告是基于警告窗口还是基于图像显示设备）。他们还应确定怎样最好地为操作人员和维护人员提供使他们能够随时充分了解核动力厂状态所需的信息，以及在必要时通过操作进行调整的手段。这些决定可能使增加硬件接口和软件功能成为必然。

6.32. 尽管应当将计算机系统的所有方面都保持在适当的配置控制水平之下，但还应当考虑系统的某些部件（例如设定点）是否应当比其他部件更容易调整。倘若如此，应当规定对实现此目的合适手段的要求（参见第15章）。

对硬件和软件接口的分析

6.33. 应当对所有的内部接口（诸如与驱动程序和通信协议有关的接口）进行分析，并记录在案，以证明所有的接口特性都已明确规定。此事应当包括通过接口传输的信息的定义、考虑同步和异步协议的特性以及选择传输速率等项。

6.34. 应当确定需要加以证明的安全特性（例如“活跃度”或“信息争用”和不发生死锁），并通过接口特性分析来证实这些安全特性已得到满足。

输入-输出信号处理和数据操作与存储

6.35. 为使核动力厂的过程数据可供计算机系统使用，信号可以在计算机的硬件或软件中处理，或在计算机系统以外的设备中处理。在仔细考虑计算机系统的特性对目的的适合性后，应当在有关计算机系统设计的文件中说明有关的设计决定（带有有关此种选择的合理性证明）以及存储数据与随后的读取数据用的物理方法（例如存储器或磁盘）和逻辑方法（例如数据库）的特性。

计算机系统的危害分析

6.36. 应当对计算机系统的体系结构及系统内的功能进行危害分析，以确定任何可能危及安全功能的具体危害，并因此指出改变体系结构或增加功能（例如自检）以减轻危害效应的必要性。此类分析应当被包括进安全证明中。

计算机系统的安全结构

6.37. 为了响应在计算机系统危害分析中确定的特定风险，需增加一些减轻危害的功能，作为对这些新增功能的补充，程控时钟、程序序列校验、变量的重新初始化及其他故障检测机制之类的设施，都是良好的做法。由于系统需要简明性，这些设施的增加应以不使软件变得不必要的复杂为前提。

保安方面的考虑

6.38. 作为需要对计算机系统保持严格的配置控制的一部分，计算机系统设计应当确定怎样预防计算机系统的功能有意无意地变差(例如由未经授权的访问、未经授权的代码或病毒引起的) [12、13]。其中应当包括有关怎样对系统进行变更并验证这些变更，以及怎样阻止未经授权的变更的程序细节或其他控制措施的细节。应当分析对保安的威胁有哪些以及有待实现的保安水平是否合理。

故障安全机制的范围

6.39. 有关计算机系统设计文件应当详细说明为满足计算机系统非功能要求而规定的故障安全机制的实施范围。文件还应当说明与可靠性要求相关的任何限制。

系统集成要求

6.40. 有关计算机系统设计文件应当载有下列各项集成要求：

- 已有或现有软件或固件（例如操作系统）；
- 硬件或专用集成电路的应用；
- 通过配置预先开发的软件来开发或生产的软件。

这些系统集成要求源自于硬件和软件的接口分析（第6.33和6.34条），并应当涵盖硬件-软件接口的定义和相关例外的处理等内容。这些信息是制订用于验证该集成系统的完整的测试规范所必需的。

文件格式

6.41. 与计算机系统设计有关的文件，应当是从系统要求的规范开始的一系列文件的一部分。与计算机系统有关的那部分系统要求被转变为计算机系统要求。有时，系统要求中所列的这些功能要求和性能要求在转变时并未发生改变，但它们更经常被细化成较低的抽象层次，以反映与计算机系统有关的细节或已做出的决定。

6.42. 在计算机系统设计的文件中，应当明确而完整地将这些要求作为一个整体进行介绍，并介绍其中的每个组成元素，还应当可追溯到其内容的理论基础。这应当与将系统的功能要求形成文件时所使用的办法是一致的，只不过是较低的抽象层次而已。

7. 软件要求

7.1. 软件要求是最终将以计算机程序的形式被实现的计算机系统要求的子集。所有计算机系统要求都应当完全而准确地转变为软件要求或硬件要求。软件要求将包括根据选择的特定计算机系统设计得出的要求。在许可证审批过程中，验证这些软件要求是否满足上一层次的要求是重要的一步。因此，开发人员和并未参与制订这些要求的人员（审查人员和监管人员）都可能会追溯这些软件要求的起源，并进行验证。

7.2. 草拟软件要求是一个与设计计算机系统密切相关的过程。软件要求描述软件在选定的计算机及其相关外围设备上运行时必须做些什么以保证总的系统要求得到满足。在设计计算机系统时，可能按不同的方式分配软件和硬件的功能，以便在诸如性能、费用、尺寸、简明性和兼容性等相关因素之间保持适当的平衡。这将确定软件要求或硬件要求究竟是什么。规范制订者还应当了解计算机、工具和类似的已有系统的能力，以便确信在软件设计和软件实现中实现这些软件要求是可行的。

7.3. 软件要求在系统层与计算机能力之间建立了链接。因此，通过计算机代码在外部环境中的处理和通过计算机代码提供的运算，这些软件要求都应当是可以理解的。

7.4. 如果计算机系统要求是足够详细的、其文件是足够正式的，且如果计算机系统设计及其编码部分是借助于工具产生的，则或许没有必要为这些部分编写单独的软件要求文件。但是，编码从中得以产生或重用的那部分计算机系统

要求,应当被看作是随后的代码对照软件要求进行验证的那些软件要求的说明。此外,代码发生器涵盖的任何单独编译的模块,都应当得到单独的软件要求文件的支持。

建 议

概 要

7.5. 软件要求应当包括把系统要求分配给了软件的描述,并要关注安全要求和潜在的故障条件、在每种运行模式下的功能要求和操作要求、性能标准、时限和限制因素、故障检测、自监、安全监测要求和保安要求。

7.6. 应当把软件要求正确地形成文件,且应当是容易理解的,以便有利于软件设计人员、计算机系统工程师、监管人员以及在必要时过程工程师和安全工程师使用。它们的书写格式应当与将要适用的特定的设计和实现无关。为了有助于理解这些要求和有利于向较高层次文件的可追溯性,应该对这些要求进行分组分层。

7.7. 应当对软件要求进行分析,看看有无模棱两可、前后矛盾之处和定义不明确的条件。应当鉴别出、弄清楚和修正各种缺陷。软件要求应当是可验证的和前后一致的。功能要求应当尽可能采用数学术语陈述。最小精确度和准确性、时间特性、执行线程以及故障安全特性等非功能软件要求,也应当明确并在需要时采用定量术语陈述。

有待执行的功能

7.8. 软件在实现基于计算机的系统的功能要求方面将起较大的作用。软件要求应当是将那些功能要求转变为从数字编码输入到数字编码输出的特殊变换。所要求的从输入到输出的变换,可以采用逻辑表达式、数学函数或数学关系式等形式,或在算法中的某些特定步骤是这些要求的一部分时采用算法(运算的顺序)的形式。为了取得更好的可追溯性和易理解性,输入、输出和内变量的命名和编码应当尽可能在该系统的环境中是有意义的。

安 全

7.9. 对系统的安全来说是重要的那些功能，同样应当在有关软件要求的文件中指出。对每个相关的输出，应当指出一旦发生可检测的但不可恢复的故障时能够使用的安全状态。计算机系统设计还可能引入与安全相关的软件输入和输出间、或各种输出间的附加关系。应当将这些内容作为附加安全要求加以描述。

可靠性和可用性

7.10. 计算机系统的要求包括可靠性或可用性要求，且这些要求应当转变为软件和硬件的要求。鉴于基于计算机的系统所需的可靠性，以及可从硬件预期的可靠性，应当纳入一些软件要求，以便使系统能够满足其可靠性目标（参见第6章）。

7.11. 可以规定一个总的软件可靠性目标，但是应当清楚，相对于满足其它类型的要求来说，软件可靠性目标的达到与否是较难得到证明的。证明定量的软件可靠性要求是否已得到满足是十分困难的。现行可以使用的方法不能提供使人相信安全最重要系统所需要的可靠性水平已经达到的结果，因此本安全导则没有提供有关使用软件可靠性模型的指导性意见。如果申请人建议为认证或调试使用软件可靠性模型，在认证或调试计划中应当介绍该模型的理论基础，并获得监管机构的同意。

接 口

7.12. 应当提供有关软件与操作人员、仪器（传感器和驱动器）、计算机硬件、其他系统及可能在同一硬件上使用的其它软件的接口的说明。这将定义软件边界。

设计限制

7.13. 在编写软件要求前，一定已经作出了许多选择，它们必然会限制软件的设计和软件的实现。这些选择应当被记录在案或在必要时作为软件要求的一部分供参考。例如，计算机硬件的选择将限制编译程序和操作系统的选择，多样性这一要求必将限制软件体系结构。应当证明此类设计限制的合理性，且它们应当是可追溯的。

模 型

7.14. 软件要求可以建立在有待实现的系统模型之上(第5章和参考文献[4]的第5章)。在这种情况下,应当仔细定义这个模型及其应用并形成文件详细说明这些要求。例如,控制软件有时是使用有限状态机模型描述的。应当对有限状态机模型的使用情况加以说明,以便使特定状态所特有的状态转换要求与功能要求能得到正确的理解。

时限性能

7.15. 应当提供对软件完成转换所需的时限的说明。其中可以包括输入与相应输出之间、或相继的相关输出之间的最小和最大时间;或预期的时限特性和要求软件处理的输入的采样速率(例如取决于某些输入到达频率的特性)。从总体上说,可以要求软件在规定时间内检测到失效或故障并进行修复。

计算精度

7.16. 应当规定由软件处理的数字信息的计算精度。鉴于处理过程是在字长有限的数字计算机上进行的,因此应当认识到计算期间会产生一些误差。应当明确规定允许误差水平,以指导软件设计人员选择数字数据的软件表示(即精度)和计算所需功能的特殊方法。

保 安

7.17. 计算机系统的某些保安要求一定要转换成软件要求(例如有关输入、已存储数据乃至程序本身的有效性检查)。保安要求还可以指明一些由软件处理的信息最好应当有特许存取权,例如安全有关的设定点。

文 件

内 容

7.18. 有关软件要求的文件的主要目的是形成软件开发和取证的基础。因此,这些文件可能载有与软件设计、申请许可证(例如风险方面的考虑、有关功能或专设安全设施的推荐意见)以及为特殊要求提供背景资料的其他事项。

文件格式

7.19. 应当按照标准草拟有关软件要求的文件，其形式不应妨碍可读性。有关软件要求的文件应当是可验证和可维护的。使用形式说明语言可能有助于表明软件要求的一致性和完整性。

7.20. 有关软件要求的文件的编制和格式的其他详细指导还可以从有关供核动力厂安全系统使用的高可靠性软件的国际标准[5]中找到。

7.21. 有关文件的一般建议见第3.34—3.44条。

8. 软件设计

8.1. 软件设计是将软件划分为一组相互作用的模块并对这些模块进行详细说明。使软件设计结构化及易于实现人员、维护人员、测试人员和监管人员理解，都是十分重要的。设计应当可证明地包括所有软件要求，并不应包含任何不安全的内容。设计通常要处理软件的体系结构以及在此体系结构内的详细设计。

8.2. 软件体系结构是指如何将软件组织成模块。这些模块包括处理过程、抽象数据的类型、通信路径、数据结构以及显示模板。将软件分解成模块的方法众多，每种方法各自侧重于某些特定类型的模块。体系结构的选择对决定模块的简明性及它们之间的相互作用的简明性十分重要，这进一步又决定它们的技术条件以及验证和确认任务的简明性。

8.3. 即使在单个的设计中，也能以一种以上的方法来描述模块的组合方式（软件体系结构）。本安全导则不推荐任何特定的表示软件体系结构的方法。

建 议

8.4. 至少推荐2个层次的设计：软件体系结构及其详细设计（更详细的建议见参考文献[4]的第5.1.1条以及有关软件工程的标准和专门书籍）。软件设计要求具备的属性在第8.5—8.12条讨论。

避免复杂性

8.5. 在安全重要系统中，所有设计层次中都应当避免不必要的复杂性。设计越简明，实现并证明所有其他属性就越容易。此外，这也使人们更能相信软件得到了充分的理解。

安 全

8.6. 软件设计阶段应当解决在先前分析（参见第5.21—5.23条的计算机系统分析和第6.36条的计算机系统危害分析）中确定的危害，并处理已被确定为安全重要的要求。这些要求应当包括必要的自监功能，以检测执行时可能发生的硬件故障。软件还应当检查其自己的控制流和控制数据。能够通过诸如看门狗技术等故障检测机制来监测硬件-软件模块的活跃特性。以失效检测为基础，应当从复原、中断各种操作和发出错信息等方面采取相应的行动。所有错误，不管是永久的还是暂时的，都应当做好记录。编码存取和数据存取的监督和完整性，如果光靠硬件不能完全确保的话，尤其应当通过合适的软件和编码技术加以确保，例如将代码和数据保存在可擦写可编程的存储器中。

可理解的和可修改的结构

8.7. 为了便于审查，软件体系结构的组织应当形成一种层次结构，以提供抽象层次。应当使用信息隐藏（参见参考文献[4]第3.3.4条），以使分段审查和验证成为可能，并便于修改。使用图解法可有助于理解。最好是采用形式语言（带有定义明确的语法和语义）描述此设计，并以自然语言作出说明（参见参考文献[4]的第5.1.3.7、5.2.3和5.2.4条，但应当注意，还存在着其他的形式方法）。

8.8. 在单个模块或少量模块内，接口应当简明。并对预期更改进行隔离。

可追溯性

8.9. 为了便于追溯各种要求，每个设计单元（诸如模块、程序、子程序或文件）都应当有惟一的标识符。

可预测性

8.10. 选定的体系结构应当是确定论的。应选择使软件的运行从对输入的响应和产生响应的时间角度来讲是可预测的那种设计。通常可使用固定的、重复的运算顺序（例如查询），而不是中断。通信协议应当是确定论的，并应该与外部系统的正确操作无关。

一致性和完整性

8.11. 设计中不应当包含矛盾和模糊之处。模块间接口的说明应当完整。模块间接口的两侧应当匹配，尽可能具有一致的顺序并在模块的输入输出接口之间使用变量名字。

可验证性和可测试性

8.12. 应当使设计及其说明做得有可能证明每条软件要求都已得到满足，并验证有关详细设计的实现是否正确。

文件

8.13. 有关软件设计的文件资料，应当提供有关软件总体体系结构和有关所有软件模块的详细设计的技术信息。还应当规定相关的实现限制。

8.14. 应当提供有文件为证的证明，证明该软件设计注意到了先前分析中确定的危害和已经被定为安全重要的要求。

软件体系结构

8.15. 应当很好地描述将软件分解为模块的情况，描述将这种分解映射到各种要求上，并要妥善地证明其合理性。该体系结构还应当考虑在该系统的生命周期中不可避免地会发生的变更，以便于软件的维护和升级。

8.16. 应在编制文件时标出和规定各种软件模块间以及软件和外部环境（在软件要求文件中有规定）间的接口。

8.17. 如果系统包括多个处理器，且软件分布在这些处理器中间，那么软件体系结构说明应当规定哪些过程在哪个处理器上运行，文件和显示数据又放在哪里，等等。

8.18. 该体系结构应当考虑因使用多样性的决策而可能引起的对模块和接口的限制。

实现过程的限制

8.19. 在软件开发的设计阶段，可能需要选择技术属性。此类实现限制的例子是需要确保程序设计语言、编译程序、子程序库及其它支持工具的多样性和所需的属性。应当在软件文件中提供这些信息。应当证明实现限制是合理的，或者是可追溯到较高层次的要求或限制的。

8.20. 应当提供已形成文件的证据，证明检测到故障时，为恢复、停止过程和错误信息而采取的行动能使系统保持在安全状态。

详细设计

8.21. 应当在详细设计中描述软件体系结构中列出的每个软件模块。此种说明的特定内容取决于模块类型。总之，模块说明应当完整地定义其与其他模块的接口，并应完整地定义模块功能及其在整套软件中的作用。

文件格式

8.22. 有关文件的一般建议见第3.34—3.44条。

8.23. 只要图表中的要素的含意有很好的定义，就应当使用图表和流程图。用于描述设计的其他常用方法包括数据流程图、结构图或图示法（例如，参见参考文献[4]的第5.1.3.7条）。

9. 软件实现

9.1. 这个阶段的输入包括内部设计规范和软件模块接口规范。输出是包含源代码与可执行码以及单元测试结果与模块接口测试结果的清单。

9.2. 能够从系统规范开始产生编码的方法有多种，这些方法在本质上是两种截然不同的方案的组合：一种是如第5—8章详细介绍过的经由规范和设计等阶段进行的经典开发过程；正如第5章提到过的，另一种方案是使用代码生成工具，该代码生成工具以采用高级语言书写的面向应用的系统描述以输入。在这两种方案中究竟选择哪一种，取决于该项目的当事人能够得到的工具和资源，并尤其应当考虑设计和证明工具可信性之间的权衡。本章的建议适用于这两种方案的一切可能的组合。

建 议

可验证性和可审查性

9.3. 能够被证明是软件规范正确实现的代码的生成是一个较大的问题。所生成的代码应当是可以对照这些软件规范进行验证的。如果靠人工检查进行验证，则编码应当是可读的、注释充分的和易懂的。为了有利于这一编码验证过程，应当使用确认过的工具（见第10章）。

变更和版本管理

9.4. 在第一版中，通常不能生产出正确的编码。应当谨慎控制有关在实现中进行的变更和修改的请求，应当保持所产生的相继版本之间的连贯性。

9.5. 实现可以暴露软件设计规范或计算机系统要求中的疏漏和不一致。因此，请求正式变更和控制修改的制度应当在实现阶段建立，以处理这些疏漏和不一致。应当做好这些变更的记录，并可供监管人员使用。保持将要产生的这些模块的不同版本之间的连贯性（见第4和15章）以及保证测试充分覆盖这些变更的制度也应当建立。

程序设计语言

9.6. 对于安全系统，程序设计语言（或使用的子集）应当有严格定义且有文件记录的语法和语义。

9.7. 程序设计语言（或使用的子集）应当具有足够的表达力。模块规范能够被细化成实现的容易程度，极大地取决于程序设计语言的表达力。

9.8. 应用软件应当使用面向应用而不是面向机器的语言。汇编语言应当仅用于规模和功能有限的模块,且只有实时性或兼容性限制得到充分证明时才使用。否则,该语言应当可足够的方便地用于支持模块的程序设计和用于利用具有与其编码主体截然不同的规范和调入顺序的过程、例行子程序或子程序构筑代码。

9.9. 程序设计语言及其译码器在设计上不应妨碍使用限制出错结构、译码时间类型检查、运行时间类型和数组边界检查,以及参数检查的使用。但是,如果这种实现方式是经过充分考虑的和完全值得信赖的,或如果在运行时占用过多的处理器能力,则可以不进行运行时间检查。

工具的使用

9.10. 应当使用经过确认的工具,因为它们能够使程序员从易于产生手工错误的代码生产过程这一任务中解脱出来,例如编程和验证。

9.11. 应当选择一套合适的工具供实现之用,其中可以包括译码器和编码发生器、调试器、连接器,以及测试和其他验证工具。

9.12. 应当使用在系统的整个寿期内都可供使用和可维护的、并彻底测试过的译码器。如果使用没有彻底测试过的译码器,则应当通过手工或使用其他工具进行补充分析和验证(参见第10章),以证明译码是正确的。

9.13. 这些工具应当与在开发工作的其他阶段使用的另外一些工具兼容。

多样性

9.14. “软件多样性”是借助下述开发技术实现的。具体做法是,让不同的程序员或程序设计组从相同的规范出发,开发2个以上版本的功能全同的程序(异体),旨在提供错误检测、提高可靠性或减少编程或转换器错误影响输出结果的概率。尽管使用软件多样性能够降低共因故障的发生概率,但不要认为它可以纠正所有的错误。仍然可能有剩余的重合错误,剩余的共因故障的概率仍然难以估计。应当考虑使用“功能多样性”,因为它可以显著降低剩余错误。为了实现功能多样性,就会想出各种不同的手段以实现相同的安全目标。当软件中有2种以上的实现方法在执行时,仍应当使用为了使软件的多样性达到可接受的水平而推荐的这些技术(第6.15和6.16条)。

9.15. 可使用不同技术来实现软件多样性,其中使用恢复程序块和N-变量技术是最常见的。还可将软件多样性概念用于软件实现的不同方面,包括测试。例

如，可以使用独立程序设计组、不同算法，以及不同的工作环境、工具和运行时间支持系统（操作系统、编译器）。如果打算采用软件多样性，那么应当在作出这一选择前考虑好所有的这些技术和技术。应当用适当的文件记录并保存这一选择及其合理性，并着重关注这些技术带来的额外复杂性。

9.16. 表决逻辑可以引发复杂的设计问题，这是应当考虑的。但是，如同冗余系统一样，应当尽可能在每个单独的子系统的决策链的下端才进行表决，以便使子系统的多样性最大。例如，在打算使用2种以上的测量方法来指出是否需要反应堆事故停堆的地方，应当在将每个信号与其事故停堆设定值比较后再实施表决，即应当根据二进制的状态信号进行表决，而不是根据测量值进行表决。还应该考虑3取2之类的表决策略，以减少误动作的可能性。需要考虑的与表决有关的其他内容包括硬件在测量过程中的偏差（包括数据位数）、允许进行比较的带宽（扭曲度）以及表决点之间的距离。

9.17. 具有联机错误检测的能力是软件多样性的积极方面，这是应当考虑的。就此环境中的未经测试的输入向量或意外的变更而言，不同的变量很可能有不同的输出，因此很可能更需要检测错误的行为。

9.18. 任何类型的多样性都意味着一定程度的独立性。如果使用了设备、软件或功能的多样性，那么应当证明独立性水平与所声称的一样。

模块编程

9.19. 在开始给模块进行编程前，必须保证其设计规范及其接口规范都是完整的和可供使用的。

编码简明性

9.20. 每个模块程序的编码，无论是它的总体结构，还是它的细节，都应当简明且易于理解。应当避免使用递归结构、编码压缩、最优化以及暗藏“机关”等隐藏编码的功能和使程序员很容易损害程序可读性的技术。

数据结构的协调一致性

9.21. 应当在整个系统中使用统一的数据结构及其命名规则。每个数据结构标识符都应当始终如一地反映其类型（阵列、变量、常量等等）、范围（局部、共享等等）和性质（输入、输出、内部等等）。

避免语言的不安全

9.22. 大部分程序设计语言都受到不安全的困扰，这种不安全使得很难甚至不可能通过编译器或者通过分析程序文本来检测是否违反了语言规则。如果不能通过放弃一些语言特点或通过增加补充的静态语义规则来消除这种不安全，那么应当避免使用这些语言，或至少应当受到限制、鉴别和彻底验证。

编码规则

9.23. 应当在经批准的一组详细的编码规则中规定推荐的程序设计技术，并在验证软件模块期间确定偏离这些规则之处。有关编码规则的指导见有关供核动力厂安全系统使用的高可靠性软件的国际标准（例如[S]）。

自监和自检

9.24. 软件设计包括必要的自监特点，以检测在执行时可能发生的硬件故障。软件还应当监督其自己的控制流和控制数据。这些监督特点在软件设计要求（参见第8章）中也许并没有被全部预料到。在这种情况下，应当请求对这些软件设计要求进行修改。

操作系统

9.25. 应当仅使用经过彻底而令人满意地测试过的操作系统。对于安全系统，应当仅使用遵循本安全导则的建议的那些操作系统。应当把操作系统的使用限制在不可缺少的功能上。应当标出这些不可缺少的功能，它们应该有定义非常明确的接口。每项特定的功能都应当总是以同样的方式调用。对于这些功能的应用，应当有用文件记录并保存的相关操作经验可供采用。

9.26. 应当采取预防措施，以确保公用的操作系统、其他支持运行的软件或网络通信软件的使用不会危及安全功能和非安全功能的隔离。

可测试性

9.27. 应当有可能有权访问全部的可执行码（包括为确保运行时间支持编码以及硬件故障监督机制的行为正确而进行的访问），以便使用本安全导则提到的

一种技术或几种技术的组合，或者“黑箱”或“白箱”技术（例如，见参考文献[4]第9.2.2条）进行测试。

文 件

9.28. 每个模块程序的编码，连同对照其规范验证该程序的正确性所必需的与上下文有关的信息，应当在某个文件的特定章节中显示出来。

9.29. 所提供的上下文信息，在将该程序与该模块的其他程序隔离以及将该模块与其他模块相隔离的情况下，应当足以构成理解和测试模块程序的基础。有关模块的上下文信息应当重复描述或提及软件设计规范中的相关片段、逻辑断言，或者该模块的每个程序都应当满足的前置条件和后置条件。还应当确定调用或被调用的其他程序和模块、程序的输入输出变量及参数连同它们的有效范围。为了便于审查该编码，该文件的篇幅不应当多于1或2页。

9.30. 应当证明所用程序设计语言的选择是合理的，并记录在案。语言的语法和语义说明应当完整且可供使用。

9.31. 应当证明所用的所有工具的选择是合理的，并记录在案。工具的确认过程也应当有案可查。

9.32. 有关程序的文件和格式的其他详细指导见有关供核动力厂安全系统使用的高可靠性软件的国际标准（例如[5]）。

10. 验证和分析

10.1. 验证是确保每个开发阶段（开发阶段的定义参见第2章）的产品（包括任何现成软件）满足前一阶段的要求的过程。分析是检验每个开发阶段的产品内部是一致的并正确使用了选定的技术的过程。验证工作组的目的应该是，提供与所要求的可靠性相称的全面而透彻的检查。

建 议

概 要

10.2. 有许多种验证和分析技术可供使用(例如见参考文献[4]第8章和参考文献[16])。每种技术都为产品质量提供某种质量保证,但没有一种技术能提供完整的质量保证。因此应当使用各种互相补充的技术。通常可对所有文件实施人工审查和检查。附加的验证和分析形式被认为对于完成质量保证来说是适当的甚至是必不可少的。例如,尽管静态编码分析可以消除反复测试这种昂贵的操作的必要性,但不能认为它可以替代测试。所有形式的静态分析都是以程序特性的模型为基础的;测试对研究这些模型没有涵盖的程序特性的方方面面是不可或缺的。

10.3. 应当充分证明每种验证和分析技术的范围和深度的合理性。

10.4. 验证结果将增加对设计过程的了解。应当保存有关异常之处的数量和类型的记录,应对这些记录进行审查,以确定是否可从中吸取任何教训,并采取相应的措施改进设计过程。在开发特定系统的范畴内实现此类改进可能很难或是不明智的。但这种改进将有益于未来的系统开发项目。说明验证过程和验证技术的文件应当受与计算机系统文件的配置管理截然不同的配置管理的支配。

10.5. 应当将审查、预排、检查或监察等手工技术用于验证生存周期内的所有阶段,而且对于源编码产生之前的那些阶段的验证而言,它们可能是惟一可适用的技术。由于这些是手工方法,因此应当考虑采用什么样的手段来记录此类审查的结果,这一点很重要。可以使用校验表。但是,应当注意校验表的结构,使其使用性优化(例如,所需要的对校验表内各项的回答应当清楚,不需要解释)。验证计划应该陈述验证人员打算记录审查结果的手段以及证明已选定方法的合理性。

10.6. 有关软件设计和软件实现的文件应当在设计软件测试案例之前进行检查。这样就能够显示软件的结构,因而设计测试案例的工作能够获悉检查过程的信息。应当将测试案例规范全面地形成文件,并经过全面审查。应当证明测试范围的合理性。审查还应当检查设计限制和编码限制是否在软件的编制过程中得到了遵守。

静态分析

10.7. 下列静态分析技术可能用来建立对源编码正确性的置信度:

- 验证是否遵守设计、编码和标准的限制；
- 控制流分析；
- 数据的用法和信息流分析；
- 符号执行；
- 正式代码的验证。

10.8. 在对软件要求做出正式规定后，才可能对编码正式进行验证。但是，正式验证通常需要大量的专门知识，因此应当考虑请教有能力的分析员。有关正式验证和程序证明的更多指导见参考文献[4]第5.2.4条。

10.9. 应当对软件的最终版本实施静态分析。

测试策略和范围

10.10. 测试就是对软件进行包括执行结果代码（最好在目标处理器或模拟器上进行）在内的分析。如果进行测试，那么所有的软件、译码工具和硬件元件的正确性都可以在测试期间受到挑战。依靠设计合适的测试案例，能够向正在被检查的软件发起种种挑战。此外，应当设计好测试策略（例如自底向上或自顶向下），允许让整个软件逐步地组合起来。典型的测试程序可以包括从该软件层次中最低的一层开始测试该模块的程序，接着在越来越高的层次上一层一层地测试该模块的程序。采用这种方法，能够将已经测试过的较低层次的程序整合到测试环境，并被用于测试该模块的较高层次的程序。

10.11. 应当在验证计划中规定一套方法，利用它能使这些测试的整个功能覆盖范围得到证明。使用如第3.41条描述的可追溯性矩阵追踪每个测试案例应该是可能的。软件中实现的所有非功能要求（例如数值精度和性能）都应当接受测试。

10.12. 应当在验证计划中将证明非功能要求已得到满足的方法形成文件。性能测试应当涵盖诸如与响应输入的速度有关的所有时限要求、检测和消除故障的时间，以及接受所有规定输入速率的能力。

10.13. 验证计划应当陈述测试的结构覆盖范围的目标值，并证明其合理性。应当证明对该计划中陈述的目标值的任何偏离是合理的，并记录在案。

10.14. 测试时应当特别注重测试各种接口（例如模块-模块、模块的程序-模块的程序、软件-硬件、系统边界处的内部-外部）。应当确保所有数据通过机制和接口协议正在令人满意地执行。

10.15. 对于使异常工况（例如除数为0，超出量程）的测试得以进行的手段，应该予以考虑。这可能需要使用专门化的测试工具（例如故障注入工具、电路内仿真器）。

10.16. 应当测试输入变量的所有变化范围。由于穷举是不切实际的，因此应当考虑使用诸如等价类划分和边值分析等技术，这些技术有助于减少使被测试软件的测试范围足够大所需的测试案例数目。

10.17. 在设计测试案例期间，应考虑所有的系统运行模式。

测试的准备和实施

10.18. 应当在验证计划中规定一套方法，利用它使这些测试的整个覆盖范围得到证明（见第4章）。应当证明对计划中陈述的目标值的任何偏离的合理性，并记录在案。

10.19. 测试计划应当设计成便于进行回归测试，办法是确保测试是可重复的和需要的人为干预最少。

10.20. 应当在测试工具、测试规程和测试技术的使用方面对规划和实施测试的人员进行适当的培训。这些人员应当与开发工作组无关。

10.21. 应当对照可追溯的标准设备对测试设备进行校准。应该在测试规程中专门标出用于实施测试的设备，并做好记录，以确保测试是可重复的和有助于对异常状况进行调查。

10.22. 应当对被测试系统的所有输出进行监测。应当调查对预期结果的任何偏离，并将调查结果记录在案。

10.23. 应当保存测试记录。记录应当可供第三方监查。测试的范围应当非常明显，包括每种测试都可追溯到相应的功能要求。

10.24. 测试进行时出现的任何异常都应接受审查；如果断定有必要对测试程序进行修改，则应当采用相应的变更控制规定，例如咨询测试规程的制订者。

10.25. 如果断定硬件或软件中存在错误，那么应当在早先商定的修改规程的控制下进行必要的变更（参见第15章）。应当分析出错的原因，应当调查为什么这些错误没有在开发过程的早期检测到，并应当实施相应的回归测试（参见第10.19条）。

10.26. 有关测试的其他指导见参考文献[4、5]。

危害分析

10.27. 本安全导则已经指出过，应当在不同的开发阶段对基于计算机的系统及其与核动力厂的接口进行评价，估计它们对核动力厂这一层的危害可能做出的贡献（可能采用的技术参见参考文献[4] 第8.3.9条）。在确定了此类潜在的重要行为后，应当将它们追溯到计算机系统的设计、软件的设计和编码，以便确认哪部分设计或哪部分软件需要采取专门的设计特点。此外，还应当将这些危害追溯到原先确定的要求，并应酌情将这些危害纳入到核动力厂安全分析中。

10.28. 随后应当验证软件设计阶段是否已经满足核动力厂的安全标准，软件实现阶段是否影响了安全或引入了新的危害。鉴于这些验证工作的难度，以及从拟议的变更或附加特点的角度看它们可能会对设计产生的影响，因此应当随着设计的不断推进分阶段进行验证，而不是仅仅在实现完成后再验证。

工具评定

10.29. 软件生产中使用的工具分为以下两大类：

- (1) 软件开发工具，其输出（可能已变换过）是程序执行的一部分，因此能够引入错误。例如编码发生器、编译器和链接器。
- (2) 软件验证工具，它不可能引入错误（但可能没有检测出错误）。例如静态分析用的工具和测试范围监测器。

10.30. 对于准备投入使用的任何一类工具的功能，应当有精确的定义。对于软件开发工具，应当准确地知道其适用范围；对于软件验证工具，应当明确地定义它要进行什么样的分析或检查。

10.31. 在所有情况下，工具应当具有足够的安全可信性，以确保其不会危及最终产品的安全性。因此，将不对其输出再次进行审查的那种软件开发工具，应当具有最高的可信性水平。如果其输出将接受如前文（第10.1—10.28条）所述的验证，或如果将使用如第10.32条所述的反求工程，那这条要求就可以降低。对于软件验证工具，要求也可以多少降低点，因为其输出将接受仔细的检查。

反求工程

10.32. 在有些情况下，有可能使用反求工程技术[19]，以提供对转换（例如从设计说明转化成源编码,或从源编码转换成机器代码）的置信度。这种反求工程涉及到反方向（或者说倒转）应用转换过程。其可行性取决于最初的转换过程是

不是定义明确的、可追溯的、直截了当的以及（单值地）可逆的。对于来源于设计的编码产生器的输出，这种倒转一直在令人满意地进行着，甚至是机械化地进行的。这项技术也一直被成功地用于利用来源于相当低层次的机器代码来重建源编码。这项技术不能严格地适用于重建高层次的语言程序，对于此种程序来说，可执行码的映射是十分复杂的。但是，将源编码和机器代码两者转换为常用的形式语言，然后在机器帮助下对两个版本进行比较的这项技术是有可能的，因此是应当加以考虑的。

运行史的评价

10.33. 除了其他验证方法外，有关软件在役行为的反馈，也可以提供对软件能够发挥其预期作用的能力的置信度。如果打算使用此类信息，那么应当分析其过去的使用情况的适当性，尤其是分析它与拟议中的新环境的兼容性和对于拟议中的新应用的适当性。

文 件

10.34. 应当将验证和分析工作记录在案。这些文件应当提供一组连贯的证据，证明开发的产品是完整的、正确的和一致的。

10.35. 前面已经建议（第4章）在项目的初期就制订验证和确认计划。该计划应当定义一套准备使用的验证和分析技术，应当提供这套技术定能提供足够多证据的理由，并详细规定将要编写的特殊记录、报告和文件。该计划还应当涵盖管理问题，例如每个过程由谁负责，这些过程彼此间以及与开发过程间将怎样协调。

10.36. 由于验证就是把一个阶段的结果与其前一阶段的结果进行比较，因此前一阶段一结束就可以开始为验证做准备。此种准备工作应该作为验证结果的第一部分形成文件。这些文件应当由下列内容组成：

- 校验表和期望的响应；
- 提供证明的义务；
- 测试案例和预期的结果（测试计划）；
- 测试范围目标及理由；
- 工具评定标准。

10.37. 在验证期间，验证人员应当记录有关验证过程的足够多的信息，以便能够有把握地重复该过程，并获得相同的结果。如果使用任何验证工具，应当详细鉴定这些工具（包括版本和配置），并记录使用的任何输入或设定参数。

10.38. 应当记录验证的结果，以证明已覆盖所有准备验证的内容。应当明确地指出哪些结果与预期相符，哪些不符（异常）。应当记录和调查由各种验证活动和分析工作发现的所有异常。应当保存所有此类异常的记录，包括有关解决这些异常所用办法的证据。当断定文件和源编码需要进行变更时，应当适用第4章中描述的变更规程。

10.39. 只要合理可行，测试计划的范围和内容应当尽量广泛而完整。某些建议（例如功能方面的和结构方面的范围）将比其他建议产生更多的测试案例。但是，它们都提供确定应当包含在测试计划中的测试案例的机会。有关产生测试案例的更多指导见参考文献[4]第9章。

10.40. 应当保存测试计划、规程、预期结果和测试报告，并应该可供质量保证监察和第三方评定使用。测试规程应当说明每个测试案例的理论基础，并就如何将测试案例追溯到相关的源文件做出规定。应在进行测试前编写的测试文件中说明预期的测试结果（连同导出这些结果的方法）。

10.41. 应当报告危害分析结果，并评定范围的完全性。

11. 计算机系统的集成

11.1. 这项活动由3部分组成：

- 将所有软件装入硬件系统；
- 测试软件—硬件接口要求是否得到满足；
- 测试所有这些软件能在集成后的软件—硬件环境中运行。

11.2. 在设计人员完成系统集成和测试后，应对这一系统集成进行验证。此阶段的目的应当是获得能够证明这种系统集成已得到完全控制的证据。

建 议

11.3. 硬件集成工作应当在计算机系统集成阶段之前进行（相关指导见参考文献[17]）。

11.4. 应当根据计算机系统设计文件中的系统集成要求制订系统集成计划（参见6.40条）。

11.5. 应当将有案可查的可追溯性分析作为验证活动的组成部分加以实施，以证明系统集成要求相对于计算机系统设计规范来说是完全的。

11.6. 应当确保只有经过验证的硬件和软件模块才能提交给系统集成阶段。这意味着需要对被集成的系统进行严格的配置控制，这种控制包括：

- 所有硬件模块的受控构造清单；
- 从软件库检索出正确的模块版本。

11.7. 供系统集成阶段使用的软件和硬件模块的版本应当是现行的经验证的版本。未经验证的版本也可以使用，条件是有文件足以证明其合理性，即验证之后没有再做过变更，或者如果需要变更则整个集成阶段将重新开始。

11.8. 系统集成的验证意在检查一直受到挑战的所有集成接口（例如硬件—软件之间的接口或软件模块之间的接口）。

11.9. 对安全系统而言，设计和从事系统集成验证测试的队伍应当独立于设计人员和开发人员。测试工作组和设计人员之间与项目相关的通信应当做好记录。对已批准的测试步骤的所有变更应当做好记录并必须重新得到批准。

11.10. 应当遵循第10章中规定的总的测试要求。

11.11. 此阶段中进行的测试应当尽可能基于“白箱”原则进行；

11.12. 应当证明这组测试的合理性。在制订测试案例时，应当考虑以下因素：

- 所有集成要求的范围（明说的和隐含的，例如证明该系统可对所有可能的接口状况安全地作出响应的坚固性测试）；
- 所有硬件—软件接口；
- 满量程范围（包括接口信号的溢出值）；
- 异常情况的处理（例如证明发生硬件故障时的行为是可接受的）；
- 等值分级和边界条件；
- 与时限有关的要求（例如响应时间、输入信号扫描、同步）；
- 准确性。

有关制订测试案例的其他指导见第12章。

11.13. 应当考虑使用除测试以外的验证方法，以及能够对系统集成的验证提供支持的工具。

文 件

11.14. 这个阶段编制的文件应当包括：

- 验证系统集成的计划；
- 可追溯性分析的结果；
- 验证系统集成用方法的合理性；
- 系统正常运行时的干预水平的证明，为了在测试期间观察系统的特性。

11.15. 系统集成的验证报告除其它内容外应当涵盖以下内容：

- 测试范围审查；
- 模块集成和测试审查；
- 硬件—软件集成测试结果审查；
- 内部时限性能的评定；
- 可追溯性审查。

12. 计算机系统的确认

12.1. 确认是证明计算机系统能履行其预定功能的过程，这是指以功能要求和非功能要求名义（参见第5章）定义的那些功能。正如参考文献[6]第2.6条所提到的，这个安全证明过程被认为是不可或缺的，因为目前的分析技术不能使系统的正确性得到充分的证明。更确切地说，确认被认为是测试和评价已集成的基于计算机的系统硬件和软件是否符合其功能要求和非功能要求。确认通常在场区外进行。

12.2. 由于基于计算机的系统是数字性质的，在决定证明其是否符合要求的测试的数量时不能采用外推法和内插法。这种局限性要求确认人员证明相对于所需的系统可靠性，测试的数量是否合理。可以考虑采用等值分级和边界条件。还有一个需要注意的问题是，要保证某些运行模式和核动力厂与基于计算机的系统间的某些相互作用（它们在确认阶段并不是可容易地进行测试的）将来能够在调试阶段得到令人满意的测试。最后，对可接受的证明而言，应当表明输入模拟的水平是足够的。

建 议

12.3. 应当遵循第10章中规定的总体测试要求。测试应当依照在质量保证制度内受到控制的质量保证大纲说明与质量保证规程，以及依照参考文献[5]规范有序地实施。附加的指导见参考文献[6]第2.6条和参考文献[7]。

12.4. 设计确认测试的队伍应当独立于设计人员和开发人员。测试工作组和设计人员之间与项目相关的通信应当做好记录。对已批准的测试步骤的所有变更应当做好记录并重新得到批准。

12.5. 该阶段实施的测试主要是基于计算机系统要求（见第5章）的黑箱测试。但是，应当对在各个设计阶段附加的特性进行分析，以确定哪些特性希望在系统这一层（即通过在系统边界注入合适的测试信号）进行测试，例如确认在检测出质量低劣或溢出的输入信号后能实施正确的故障安全动作。

12.6. 应当证明这组测试的合理性。在制订测试案例时应当考虑以下因素：

- 所有要求的范围（包括坚固性测试和安全特性）；
- 满量程范围（包括输入信号的溢出值）；
- 异常情况的处理（例如证明发生输入故障时的行为是可接受的）；
- 等值分级和边界条件；
- 与时间安排有关的要求（例如响应时间、输入信号扫描、同步）；
- 准确性；
- 所有接口（例如系统集成中的硬件—软件接口和确认期间的外部接口）；
- 应力和负荷试验（例如揭示陡壁边缘效应的试验）；
- 计算机系统的所有运行模式，包括运行模式和电源故障之后的复原间的转换。

12.7. 应当实施可追溯性分析，以证明(有关测试或评价的)确认要求相对于计算机系统要求而言是完全的。

12.8. 接受确认测试的系统硬件，相对于准备安装在场区装置上的基于计算机的系统的最终配置来说，应当是充分典型的，而且系统的软件应当是相同的。

12.9. 系统应当经受各种各样的静态输入和动态输入的测试。重要的是采用涉及所有输入的逼真情景对系统的所有部件进行演习（动态测试）。但是，由于采用有关核动力厂的真实的故事情景来测试安全系统的行为既不合理可行，又不安全，因此应当采用模拟的故事情景来对系统进行测试。有可能使用测试工具来记录结果。

12.10. 这些动态测试应当以对由假设始发事件导致的核动力厂瞬态分析为基础。测试的范围应当是会对计算机系统提出要求的机组参数的预期变化范围的典型代表。已完成测试的数量应当大得足以使人相信系统的可信性。

12.11. 应当考虑使计算机系统经受统计性测试的可能性。其目的是通过使计算机系统经受一系列统计上有效的测试得到其可靠性的估计值（参见参考文献[4]的第9.3.7条）。这些测试应当是从计算机系统的运行输入空间中随机挑选出的。

12.12. 还可以使用完全随机的测试，这意味着这些测试是从输入的一切可能组合中挑选出的，因为它为产生可揭示意外副作用的大量测试案例提供了一种便捷的手段。当这些输入的组合没有被包括在计算机系统的运行输入空间中时，相应的测试被定义为坚固性测试。

12.13. 对于动态的和统计性的测试来说，测试的数量应当与计算机系统的可靠性要求有关。应当提供证据证明误差检测方法是适当的。

12.14. 应当进行测试以确认设定点的准确性和滞后作用。

12.15. 应当尽可能在此阶段对系统的操作和维护手册进行确认。

文 件

12.16. 确认阶段的文件应当包括：

- 可能发生的对确认计划的更新；
- 对各个设计阶段中补充的那些特性的分析；
- 证明（供测试或评价用的）确认要求相对于计算机系统的要求的完整性的可追溯性分析；
- 确认的结果；
- 统计性测试的计划、时间表和结果；
- 由供统计测试使用的假设始发事件导致的核动力厂瞬态的分析；
- 随机测试的计划、时间表及其结果。

13. 安装和调试

13.1. 计算机系统交付给场区之后，需要在核动力厂进行安装。这是一个逐步进行作业过程，涉及到确保各种设备被安装到指定位置、接通电源，以及重复

进行已在制造商的厂房内进行过某些测试，以便证明计算机系统在运输和安装期间未受损。在这个阶段之后是将核动力厂的电缆连接到计算机系统：可以既包括数据通信电缆，也包括核动力厂的信号电缆。在连接工作完成之后，应当证明每根电缆都已正确地与其接线终端相连。

13.2. 调试是使已建成的核动力厂部件和系统投入运行，并验证其是否与设计假设相符以及是否满足性能判据的过程。调试包括非核测试和核测试（俗称“冷试验”和“热试验”）。在这一阶段中，基于计算机的系统将逐步与其他部件以及核动力厂的其它物项组合成一个整体（系统集成）。核动力厂环境内的测试是调试基于计算机的系统的重要组成部分。尤其应当在调试阶段测试各种运行模式和基于计算机的系统与核动力厂之间的种种相互作用，这些都是有可能在确认阶段不易进行测试的。

13.3. 安装和调试活动能够在新核动力厂上进行，或者在核动力厂修改期间进行（例如设备翻新和升级）。第13.5—13.10条给出的建议既适用于新建核动力厂，也适用于已有核动力厂的修改。

13.4. 对照核动力厂安全要求对基于计算机的系统进行确认，是验证和确认过程[2、3]的最后一个阶段。

建 议

13.5. 在核动力厂内安装整套系统的工作应当在计算机系统的集成和确认阶段完成之后再行进行。但是，如果不可能这样做，则应当提供理由，说明在安装完成后进行测试是合适的。系统安装好后应当进行安全证明，达到的水平应该与在制造商厂房内进行的此类证明所达到的水平相同（参见参考文献[4]的第8第9章以及参考文献[5]的第7第8章）。

13.6. 应当考虑部分或全部重复进行第10—12章中概述过的测试，因为设备以及相关的软件在安装期间可能受到损害。尤其应当注意测试系统的外部接口，并确认能与相接的设备一起正确地履行职能。只要可行，调试阶段实施的测试应当确认先前进行的有关接口在正常和例外情况下的响应的正确性的那些测试。

13.7. 基于计算机的系统应当经历一段较长的现场试用期，在此期间，该系统的运行、测试和维护应尽可能与典型的在役工况相同。只要适宜，在试用期间，新系统应当与老系统同时运行，直到对新系统的合适性具有足够的置信度为止。

应当在此阶段完成运行和维护手册的确认。在此期间，该系统应当经历例行的测试和维护活动。应当做好任何已暴露的故障和已采取的纠正措施的记录。

13.8. 应当认识到，随着计算机系统逐步组合进核动力厂中，要求计算机系统履行其安全功能的时刻必定会达到。在到达该时刻之前，计算机系统应当完成支持连续的调试和运行所必需的一切可以进行的测试。此外，系统的维护所必需的一切硬件和软件应当在该系统需要支持其安全功能前就位。

13.9. 进行调试测试的队伍应当独立于基于计算机的安全系统的生产者。应当确保参加调试工作的人员有能力完成分派给他们的任务。

13.10. 在调试期间，应当对计算机系统（硬件和软件）保持严格的配置控制。此阶段所需的任何变更都应当受到正式的有文件为证的变更过程的支配（见有关交付后修改的第15章）。

文 件

13.11. 应当编制足够多的文件，以证明调试工作对于全面地使已安装的基于计算机的安全系统令人满意是充分的。调试队伍应当提供文件，证明包括将测试追溯到源要求（对安全系统的要求）在内的测试范围是充分的。证明调试工作的充分性是合理的文件，应当由核动力厂经营者保存，并应供第三方评定之用。这个阶段编制的文件应当包括说明下列各项的理由：

- 安装后实施的任何系统集成和确认测试是充分的；
- 此阶段打算进行的重复安装前测试（有时称为工厂测试或验收测试）的数量；
- 包括将测试追溯到源要求（对计算机系统的要求）在内的调试测试的范围、所有的调试记录和最终报告以及异常和测试故障的记录（包括其解决方案的说明）；
- 计算机系统在所有调试阶段履行相应安全功能的能力；此外，应当证明支持连续调试和运行所必需的计算机系统的所有可进行的测试已经在需要其履行相应的安全功能之前完成。

14. 运 行

14.1. 基于计算机的系统的运行阶段是在安装、调试并经监管机构批准使用之后开始的。系统已成为核动力厂的一部分，并由许可证持有者负责运行。特定系统的运行阶段将持续到其被拆除或替换（可能被第15章所描述的修改版本替换）为止。

14.2. 基于计算机的系统的运行将涉及一些维护活动，以使设备保持良好的状态以及修理任何已发生故障的部件。如果基于计算机的系统不能保持足够高的完整性，那么用基于计算机的系统支持正在运行的核装置安全系统的这种过程，就有可能危及安全系统发挥其特定功能。

建 议

试运行期

14.3. 就象第13.7条对新系统的建议一样，系统改进后应当有一个试运行期。在此期间，基于计算机的系统的在役测试频率应当增加。

纠正措施

14.4. 在硬件元件发生故障后，纠正措施应当仅限于一对一地置换硬件，并重新安装已有的软件模块；这些行动应当不包括任何改进（见第15章有关交付后改进的内容）。

保 安

14.5. 根据针对基于计算机的系统的环境规定的保安政策，应当执行适当的保安规程——诸如口令管理——以预防未经授权的访问和病毒等。相关内容见有关保安要求的第5.20条和有关计算机系统设计中的保安措施的第6.38条。

14.6. 安全的存储器安排和程序性的控制措施应当确保只有经过授权的软件版本才能被装入核动力厂设备中。在重新投入使用之前，应当证明基于计算机的系统的性能是正确的。

校准数据

14.7. 校准数据应当具有足够高的准确度，不会降低基于计算机的系统的可靠性。对于安全系统来说，此类数据应由按照与基于计算机的系统所用标准相同的标准开发的系统自动产生。如果没有按照该标准开发出的数据生成系统，就应当由独立的小组采用不同的方法核对所产生的校准数据。

14.8. 在进行下一步骤前，应当要求操作人员按照商定的规程确认输入的数据。在安全系统得到恢复前，所有输入数据都应当由独立的一方存档保存和核对。

14.9. 在校准数据变更后，应当对相关子系统进行合适的测试，以证明其运行是正确的。

文 件

14.10. 应当由许可证持有者保存有关系统运行的翔实记录，并可供第三方评定时使用。这些记录应当包括有关所有的维护活动（包括预防性的和纠正性的行动）、在役测试和系统运行期间观察到的异常信息。

14.11. 应当按照为此目的而制订的规程来记录与基于计算机的系统相关的事件和异常（包括运行中的困难和使用维护手册方面的困难）。应当陈述事件调查的结果和所采取的纠正行动。涉及修改软件的纠正措施须受如第15章所描述的变更控制过程的支配。

14.12. 对安全系统而言，如果没有按照与基于计算机的系统所用标准相同的标准开发出的校准数据生成系统（参见第14.7条），就应当说明校准数据的产生过程，并证明该过程是异样的。应当确定进行所需计算工作的小组，并有确保最大多样性的规程。所有计算都应当被记录在案并保存，以供未来监察之用。

15. 交付后修改

15.1. 在运行阶段，必须确保基于计算机的系统在修改期间和修改之后都能保持其功能方面的安全性。就像第4.24条提到过的，因此应当制订专门处理核动力厂安全问题的适当的变更控制规程。此外，应当由来自相应学科的人员对修改的安全影响进行分析，也是十分重要的。在这方面而论，基于计算机的系统的修改与其他的核动力厂修改没有什么不同。但是，还有一些仅适用于基于计算机的系统及其软件的具体问题，本章将论述这些问题。

建 议

15.2. 核动力厂设计人员和经营者应当确保适当的变更控制规程已经就位，包括供审查和核准这些修改中的安全方面用的相应的规程和组织机构。这些规程应当从项目的开始就已就位。

15.3. 所有修改都应当按照其安全重要性加以考虑并分类。安全重要性最高的修改或许需要提交给监管机构，尤其是会改变以前已批准的运行限值和条件的那些修改。

15.4. 规程应当由独立于设计人员的专家制订，修改的开发人员应当评定拟议的修改及其实现的合适性。应当由变更规程涵盖的变更包括软件修改、硬件变更和工具变更。

15.5. 变更规程应当确保给每种修改提供安全性的理由（参见第15.3条）。

15.6. 只有在得到详细理由支持的情况下，才允许在联机运行期间修改计算机系统尤其是修改其软件，且不当为此目的而提供设备。对核动力厂运行期间可能需要变化的参数（例如事故保护停堆设定值和校准常数）的修改，应当使用已证明适合该目的的专设设施。专设设施提供的可变程度应当被局限在核动力厂安全分析中已证明是合理的范围内。

15.7. 在整个变更过程中应当始终保持严格的配置控制，尤其是在解决由同时实施的修改而引起的任何冲突的过程中。在核动力厂设备中应当只安装已经历整个变更过程的那些物项。变更规程应当对下述情况作出规定：当排除故障或测试需要在计算机系统没有联机的情况下作出临时变更时。

15.8. 应当规划和做好将修改过的软件安装到异样子系统的任何冗余子系统中去的工作，以减少系统临时降级的效应。将修改过软件安装到安全系统中去的工作应当一个一个子系统地分阶段进行，以降低共因效应（即每次只安装一个子系统）。

文 件

15.9. 对每条拟议的变更，应当酌情提供下列信息：

- 修改的理由；
- 修改的功能描述；
- 修改的安全评价，证明核动力厂的安全性未受到这些变更的不利影响；

- 设计修改的详细描述，并进行涵盖拟议变更的全部范围的影响分析，包括核动力厂中可能受到影响的所有物项；
- 有关验证和确认以及第三方评定的报告，包括其范围和回归分析的合理性；
- 拟议安装方法的合理性（参见第15.8条）；
- 在现场进行的测试的报告。

15.10. 所有修改文件都应当在软件修改控制清单中署上日期、编号并归档。

15.11. 应当建立和保存序时记录。它应当提供所有修改和变更的详细资料，包括提及修改和变更的请求、影响分析、数据和结果的验证和确认、以及受到修改和变更活动影响的所有文件。

15.12. 变更的文件应当可供指望他批准变更过程并执行变更的监管者使用。

参考文献

- [1] INTERNATIONAL ATOMIC ENERGY AGENCY, Safety of Nuclear Power Plants: Design, Safety Standards Series No. NS-R-1, IAEA, Vienna (2000).
- [2] INTERNATIONAL ATOMIC ENERGY AGENCY, Protection System and Related Features in Nuclear Power Plants, Safety Series No. 50-SG-D3, IAEA, Vienna (1980).
- [3] INTERNATIONAL ATOMIC ENERGY AGENCY, Safety Related Instrumentation and Control Systems for Nuclear Power Plants, Safety Series No. 50-SG-D8, IAEA, Vienna (1984).
- [4] INTERNATIONAL ATOMIC ENERGY AGENCY, Software Important to Safety in Nuclear Power Plants, Technical Reports Series No. 367, IAEA, Vienna (1994).
- [5] INTERNATIONAL ELECTROTECHNICAL COMMISSION, Software for Computers in Safety Systems of Nuclear Power Plants, Standard No. 880, IEC, Geneva (1986).
- [6] EUROPEAN COMMISSION, European Nuclear Regulators' Current Requirements and Practices for the Licensing of Safety Critical Software for Nuclear Regulators, Rep. EUR 18158 EN, Office for Official Publications of the European Communities, Luxembourg (1998).
- [7] ATOMIC ENERGY CONTROL BOARD, CANADA; DIRECTION DE LA SÛRETÉ DES INSTALLATIONS NUCLÉAIRES, INSTITUT DE PROTECTION ET DE SÛRETÉ NUCLÉAIRE, FRANCE; NUCLEAR INSTALLATIONS INSPECTORATE, UNITED KINGDOM; NUCLEAR REGULATORY COMMISSION, UNITED STATES OF AMERICA, Four Party Regulatory Consensus Report on the Safety Case for Computer—Based Systems in Nuclear Power Plants, HMSO, Norwich (1997).
- [8] INTERNATIONAL ATOMIC ENERGY AGENCY, Specification of Requirements for Upgrades Using Digital Instrument and Control Systems, IAEA-TECDOC-1066, Vienna (1999).
- [9] INTERNATIONAL ELECTROTECHNICAL COMMISSION, Instrumentation and Control Systems Important for Safety: Classification, Standard No. 61226, IEC, Geneva (1993).

- [10] INTERNATIONAL NUCLEAR SAFETY ADVISORY GROUP, Basic Safety Principles for Nuclear Power Plants, Safety Series No. 75-INSAG-3 Rev. 1, INSAG-12, IAEA, Vienna (1999).
- [11] INTERNATIONAL ATOMIC ENERGY AGENCY, Application of the Single Failure Criterion, Safety Series No. 50-P-1, IAEA, Vienna (1990).
- [12] BRITISH COMPUTER SOCIETY, Guidelines on Good Security Practice, BCS, Swindon (1990).
- [13] BRITISH STANDARDS INSTITUTION, Code of Practice for Information Security Management, BS 7799, BSI, London (1995).
- [14] INTERNATIONAL ATOMIC ENERGY AGENCY, Quality Assurance for Safety in Nuclear Power Plants and other Nuclear Installations, Code and Safety Guides Q1-Q14, Safety Series No. 50-C/SG-Q, IAEA, Vienna (1996).
- [15] INTERNATIONAL ATOMIC ENERGY AGENCY, Manual on Quality Assurance for Computer Software Related to the Safety of Nuclear Power Plants, Technical Reports Series No. 282, IAEA, Vienna (1988).
- [16] INTERNATIONAL ATOMIC ENERGY AGENCY, Verification and Validation of Software Related to Nuclear Power Plant Instrumentation and Control, Technical Reports Series No. 384, IAEA, Vienna (1999).
- [17] INTERNATIONAL ELECTROTECHNICAL COMMISSION, Programmed Digital Computers Important to Safety Systems of Nuclear Power Plants, Standard No. 987, IEC, Geneva (1989).
- [18] VOGES, U., “Software diversity”, Proc. 9th Annual Conf. on Software Safety, Luxembourg, 1992, Centre for Software Reliability, City Univ., London (1992).
- [19] PAVEY, D.J., WINSBORROW, L.A., Demonstrating equivalence of source code and PROM contents, Computer J. **36** (1993) 654.

附 录

现成软件的使用和确认¹

本附录全部（略有修改）摘自参考文献[A-1]的第1.3条。

基本原理

A-1. 如果引入的方式正确，将现成软件（PSW）的部件纳入应用软件可能不仅有利于提高生产能力，还可提高软件系统的安全水平。这是因为PSW的部件通常已在许多领域得到应用，在评定和描述其主要特点时，可以考虑这些应用的运行经验。可再度使用的这些软件部件可能已被开发得达到其他工业中相当高的标准，被用于安全上高度重要的应用，因此可以在核工业中再度使用。倘若知道该软件已经做过适当的评估，则许可证持有者可以请求采用此类软件。

涉及的问题

A-2. PSW的功能特性和非功能（可信性、性能……）特性经常没有明确地界定和形成文件。

A-3. 有关PSW运行经验的文件和数据，经常不足以提供补偿对产品及其开发过程的不了解所需的证据。

A-4. 由于A-2和A-3中所提及的问题，有关证明PSW能否适合特定应用的验收标准和调查规程可能难以到位。

A-5. 与PSW有关的运行经验可能不完全与未来的应用相对应。因此，该应用可能援引质量未知的软件途径。

¹ ©欧洲共同体。本附录的复制已得到出版者欧洲共同体正式出版物办公室的许可。

普通的见解²

- A-6. 应明确列出必须由PSW部件履行的功能，并应评价这些功能对安全的影响。
- A-7. 应当明确列出想要使用的PSW部件，包括其代码的版本。
- A-8. 应当明确列出并一丝不苟地确认用户或其他软件通过它调用PSW模块的接口，并提供证据证明不可能执行其他调用序列，即使是无意的。
- A-9. PSW必须是已开发好的，并且是按照与其预定用途相适应的良好软件工程实践和质量保证标准进行维护的。
- A-10. 对于安全系统来说，PSW必须接受与为该应用新开发的软件相同的最终产品（不是生产过程）评定（分析和审查）。必要时，必须实施反求工程，以便能够评价PSW的全部规格说明。
- A-11. 如果必须修改PSW部件，则必须备有PSW的设计文件和源编码。
- A-12. 必须备有评价PSW产品及其评定和开发过程的质量所需的信息；这些信息必须足以评定PSW是否满足所要求的质量水平。
- A-13. 对于验收来说，必须采取下列行动：
- A-14. 验证由PSW履行的功能是否满足安全系统要求规范中以及其他适用的软件规范规定的所有要求；
- A-15. 对于安全系统要求规范没有要求的PSW功能，要验证它们不会因诸如不正确的输入、中断和误用而被调用并对所需要的功能产生不利的影响。
- A-16. 对照可适用的标准要求（例如[A-2]），对PSW设计进行遵章分析；
- A-17. 必须通过测试对预定使用的PSW功能进行确认。测试可能包括由供应商进行的测试；
- A-18. 确保安全系统、其他软件或用户不能以不同于已经规定的和测试过的方式（必要时通过实现前提条件、锁定机制或其他保护措施）使用PSW的功能。

² 在本安全导则中，该词的意思是推荐意见。

A-19. 如果在办理许可证的过程中相信反馈经验，则必须备有足够多的运行历史和故障率方面的信息。必须以对运行时间、错误报告和正在运行系统的版本史的分析为基础对反馈经验做出正确的评价。这些反馈经验还必须以在相同的运行方式中使用正在接受评价的PSW为基础。此种运行经验必须以最近的版本为基础，除非恰当的影响分析表明先前获得的基于PSW的未变更部分的经验仍然有效，因为这些部分未受到后来版本的影响。

A-20. 如果已有的有关上述推荐意见A-19所要求的那种类型的资料不充分，则必须分析PSW的故障对安全性的影响（风险评定）。必须特别注意可能产生的副作用和PSW与用户和/或其他软件部件之间的各个层次的接口处或许会发生的故障。

A-21. 验收时必须分析和考虑在确认PSW期间发现的错误。

推荐的做法

A-22. 运行经验可能被看作是具有统计学根据的证据，是对系统软件（操作系统、通信协议、标准的功能[A-2、E3、统计方法]）的确认或验证的补充。

A-23. 应当从场区信息、运行方式、需求率与运行时间、错误报告和版本历史等方面收集供评价能否相信反馈经验用的数据。

场区信息和运行方式数据应当包括：

- PSW的配置；
- 使用的功能；
- 输入信号的类型和特点，包括量程范围和变化率（如果必要）；
- 用户接口；
- 系统数量。

需求率和运行时间数据应当包括：

- 首次启动以来的实耗时间；
- 自最近的PSW版本发布以来的时间；
- 自最近发生严重错误（如果发生了的话）以来的时间；
- 自最近的错误报告（如果有的话）以来的时间；
- 要求执行PSW的需求的类型和数量。

错误报告应当包括：

- 发现错误的日期，严重性；

该出版物已被第 SSG-39 号取代。

— 修复情况。

版本历史应当包括:

- 版本发布日期和版本标识;
- 已修复的故障, 功能的修改或扩充;
- 未解决的问题。

应当将这些数据连同PSW的版本标识和相关配置的标识做好记录。

附录的参考文献

- [A-1] EUROPEAN COMMISSION, European Nuclear Regulators' Current Requirements and Practices for the Licensing of Safety Critical Software for Nuclear Regulators, Rep. EUR 18158 EN, Office for Official Publications of the European Communities, Luxembourg (1998).
- [A-2] INTERNATIONAL ELECTROTECHNICAL COMMISSION, Software for Computers in Safety Systems of Nuclear Power Plants, Standard No. 880, IEC, Geneva (1986).

术 语 表

下列定义适用于本导则。

安全重要的基于计算机的系统 采用内部计算机系统实现系统安全功能的核动力厂安全重要系统。

计算机系统体系结构 计算机系统的硬件（处理器、存储器、输入输出设备）、硬件间的联接、通信系统以及基于这些硬件的软件功能的映射。

计算机系统集成 集成软件和硬件，以产生计算机系统的过程。

计算机系统要求 已在较高设计水平确定的，计算机系统必须和足以满足核动力厂安全要求的功能特征和非功能特征。

可信性 已提供的服务的可信赖性，因此能够将信心合理地建立在这种服务之上。可信赖性、有效性和安全性是可信性的属性。

程序包 被存入一种存储器（例如只读存储器（ROM）），在操作期间不能使用计算机进行动态修改的计算机程序和数据。

功能安全要求 要求提供的安全服务或安全功能。

实现 （1）将设计转变为硬件或软件的过程；（2）第（1）步操作的结果。

非功能要求 必须得到保证的性质或性能。

预先开发软件或现有软件 在基于计算机的系统中使用的由负责系统开发人员管理的、以前没有开发过的软件（成品软件是一种现有软件）。

冗余 提供可供备择的（相同或不同）构筑物、系统或元件，其中的任何一个都能够实施特定功能，无论其他设备的运行状况如何或是否发生故障。

反求工程 根据已完成设计重新产生技术要求的过程，以便与原始技术要求相比较，以确定两者是相容的。

可审查性 有关评价软件元素或可容易确认的项目状况的已计划结果的容许差异的性质。

安全有关系统 不是安全系统但是安全重要系统。

安全系统 用来确保反应堆安全停堆或从堆芯排出余热、或限制预期运行事件和设计基准事故后果的安全重要系统。

软件要求 为了满足系统要求，在选定的计算机及其外部设备上运行软件时，要求软件做什么的陈述。

安全重要系统 属于安全群的系统和/或其失灵或故障将会导致厂区人员或公众受到辐射照射的系统。

系统集成 集成计算机系统和核动力厂系统其他部件的过程。

系统生存周期 从系统的方案形成到最终处置的整个阶段。

时限 软件实现变换所要求的时间限值。

可追溯性 能够在开发过程中的两种产品间建立的联系的程度，尤其是彼此间存在原有物—后继者关系的产品。

确认³ 为确保其符合相关的功能、性能和接口要求，对集成计算机系统（硬件和软件）进行测试与评价的过程。

验证³ 确保系统生存周期的某一阶段满足前一阶段对它的要求的过程。

表决 通过对输出信号的投票来降低系统假动作可能性的策略（例如3个输出信号中的2个）。

³ 为本安全导则之目的，这些有关确认和验证的定义在基于计算机的系统的生存周期范围内适用。

参与起草和审订的人员

Asmis, G.J.K.	加拿大原子能管理委员会
Bafurik, J.	斯洛伐克核管理机构
Beltracchi, L.	美国核管理委员会
Bouard, J.-P.	法国电力公司
Carre, B.	英国Praxis重要系统股份有限公司
Chandra, U.	印度巴巴原子研究中心
Courtois, P.-J.	比利时核安全主管机构AVN
Duong, M.	国际原子能机构
Fandrich, J.	德国西门子公司.
Faya, A.	加拿大核安全委员会
Ficheux, F.	法国电力公司
Geerinck, P.	比利时动力集团
Greenberg, R.	以色列原子能委员会
Hamar, K.	匈牙利原子能委员会
Henry, J.Y.	法国核防护和安全研究所
Hirose, M.	日本核动力工程公司
Hohendorf, R.J.	加拿大安大略水电公司
Hughes, P.	英国核设施检查局
Karpeta, C.	捷克共和国国家核安全办公室
Kersken, M.	德国安全技术有限公司研究院
Kulig, M.J.	国际原子能机构
Lawrence, D.	美国劳伦斯·利弗莫尔国家实验室

Lee, J.-S.	大韩民国原子能研究所
Mandij, D.	斯洛文尼亚克尔什科核电厂
Nechanický, M.	捷克泰梅林核电厂
Pachner, J.	国际原子能机构
Regnier, P.	法国核防护和安全研究所
Roca, J.L.	阿根廷国家核监管局
Saidel, F.	德国联邦辐射防护办公室
Tanaka, T.	日本东京电力公司
Taylor, R.P.	加拿大核安全委员会
Vojtech, J.	斯洛伐克核动力研究所
Voumard, A.	瑞士联邦核安全局
Wainwright, N.	英国核设施检查局
Yates, R.L.	英国核设施检查局
Zambardi, F.	意大利国家环境保护局

认可安全标准的机构

核安全标准委员会

比利时: Govaerts, P. (主席); 巴西: da Silva, A.J.C.; 加拿大: Wigfull, P.; 中国: Lei, Y.; Zhao, Y.; 捷克共和国: Stuller, J.; 芬兰: Salminen, P.; 法国: Saint Raymond, P.; 德国: Wendling, R.D., Sengewein, H., Krüger, W.; 印度: Venkat Raj, V.; 日本: Tobioka, T.; 大韩民国: Moon, P.S.H.; 荷兰: de Munk, P.; Versteeg, J.; 俄罗斯联邦: Baklushin, R.P.; 瑞典: Viktorsson C., Jende, E.; 英国: Willby, C., Pape, R.P.; 美利坚合众国: Morris, B.M.; 国际原子能机构: Lacey, D.J. (协调员); 经济合作与发展组织/核能机构: Frescura, G., Royen, J.

安全标准委员会

阿根廷: Beninson, D.; 澳大利亚: Lokan, K., Burns, P.; 加拿大: Bishop, A. (主席), Duncan, R.M.; 中国: Huang, Q., Zhao, C.; 法国: Lacoste, A.-C., Asty, M.; 德国: Hennenhöfer, G., Wendling, R.D.; 日本: Sumita, K., Sato, K.; 大韩民国: Lim, Y.K.; 斯洛伐克: Lipár, M., Misár, J.; 西班牙: Alonso, A., Trueba, P.; 瑞典: Holm, L-E.; 瑞士: Prêtre, S.; 英国: Williams, L.G., Harbison, S.A.; 美利坚合众国: Travers, W.D., Callan, L.J., Taylor, J.M.; 国际原子能机构: Karbassioun, A. (协调员); 国际辐射防护委员会: Valentin, J.; 经济合作与发展组织/核能机构: Frescura, G.

通过国际标准实现安全

“国际原子能机构的标准已经成为促进有益利用核和辐射相关技术全球安全机制中的一项重要内容。

“国际原子能机构安全标准正在适用于核电生产以及医学、工业、农业、研究和教育，以确保对人类和环境的适当保护。”

国际原子能机构
总干事
穆罕默德·埃尔巴拉迪

国际原子能机构
维也纳
ISBN 92-0-513705-7
ISSN 1020-5853