软件安全性的开发与分析

中国国防科技信息中心
《载人航天》编辑部

一九九八年十一月

目　录

美国航宇局软件安全性标准

NSS 1740.13

王　纬(译)　　何新贵(校)
(航天工业总公司第二〇四研究所)　　(北京系统工程研究所)

前　言

本标准提供了美国航宇局型号项目的软件安全方法学，介绍了确保将安全性设计引入美国航宇局采购或开发的软件中去所进行的必要活动。所有型号/项目经理都要评估其项目软件内存在的安全性风险，并鼓励在本标准框架内相应地改进他们的软件安全性活动。

本标准发展了 NMI2410.10“美国航宇局软件管理保证和工程策略”、NHB1700.1(V1)“美国航宇局安全策略和要求文档”与 NASA－STD－2201－93“软件保证标准”的要求。

如果不能满足强制性（用 Shall，Will，Must 等（必须）表示限定的）要求，那么必须按照 NHB1700.1(V1)的规定准备一组偏差或放弃申明文件。美国航宇局要求对这些文件申明的偏差/放弃要求至少要经主管装备或型号的安全官员批准。对本标准所包含的安全性要求进行的全部改动必须每半年向安全与任务保证部通报一次，并且必须得到装备或型号一级机关的批准。

对本标准的意见和疑问请提交美国航宇局总部、局长、安全与风险管理部、安全与任务保证部副部长办公室。

本文件由安全与任务保证部副部长弗雷德里克·格雷戈里签署。

生效日期：1996 年 2 月 12 日

1. 引　言

1.1 范围

本标准提供美国航宇局型号项目的软件安全方法学。

1.2 目的

本标准的目的是提供把软件安全性作为整个系统安全性大纲的一个组成部分，以实现系统化方法的要求。它描述确保将安全性设计引入美国航宇局采购或开发软件的必要活动中去，并在整个软件寿命周期保证安全性所进行的必要活动。

1.3 适用性

本标准适用于美国航宇局所采购或开发的软件，这些软件是有可能直接或间接造成人员伤亡或外部财产损失的系统的组成部分。美国航宇局采购软件时，本标准适用于合同条款或谅解备忘录中规定的安全级别。当软件由美国航宇局开发时，本标准适用于型号计划、软件管理计划或其他控制文档中规定的安全级别。

本标准仅适用于以下特定的软件元素：经过系统危险分析被定为可能引起或促使系统趋于某种特定的危险状态的软件；用来检查系统是否已达到某种特定的危险状态，并在达到时采取纠正措施的软件；用来在发生事故时减轻损失的软件。

1.3.1 政府提供的设备(GFE)、重用和购

置的软件

对于要求采用本标准的系统来说，本标准将适用于政府提供的软件、购置的软件(包括现货软件）以及任何其他的系统中重用的软件。如果本标准的某些要求由于软件和文档的特性而不可行，那么开发者应得到美国航宇局系统采办者的认可，才能放弃。

1.3.2 固件

就本标准的目的而言，必须把固件当作软件对待。

1.4 改进

对本标准进行改进需由型号/项目/任务管理者与美国航宇局相应中心的安全与任务保证部门协商进行。改进工作必须包括确定可接受的风险级别、哪个软件应作为安全关键软件，以及与软件相关的安全风险级别是否需要正式安全认证。

2. 参考文献

2.1 参照的文档

下列参考文献被用于本标准的制定：

航宇局软件获取寿命周期，SMAP/ 版本 4.0 1989

DOD－STD－2167A　军用标准，防务系统软件开发，1988.2.29

MIL－STD－882B　系统安全性大纲要求，1987.7.01. 通告 1

MIL－STD－882C　系统安全性大纲要求，1993.1.19

NASA－GB－A302　软件正式审查指南，1993.8

NASA－STD－2100－91　航宇局软件文档标准软件工程大纲，1991.7.29

NASA－STD－2201－93　软件保证标准，1992.11.10

NASA－STD－2202－93　软件审查标准，1993.4

NHB 1700.1(V1－B)　航宇局安全方针和要求文档，1993.6

NMI 2410.10　航宇局软件管理保证和工程策略，1993.4.20

SMAP－GB－A201　软件保证指南，1989.9

SMAP－GB－A301　软件质量保证审计指南，1990.11

2.2 术语汇编

附录 A 给出了与软件安全性有关的术语汇编。

2.3 缩略语

附录 B 给出了缩略语清单。

3. 要　求

3.1 一般要求

软件安全性活动的目的是确保软件不引起或不促使系统趋于危险状态；如果系统达到危险状态，在检测或进行纠正时软件不失效；如果发生了事故，在减轻损失时软件不失效。

软件安全性过程必须做到：

(1)确保系统/子系统安全性分析时标识出哪个软件是安全性的关键。在安全性分析中，确定为有可能引起危险或被要求控制危险的软件是安全性关键软件。

(2)确保系统/子系统安全性分析时明确标识出软件需求规格说明的关键输入(例如，标识危险的命令、界限、界限的相互关系、事件顺序、计时限制、表决逻辑、失效容限等)。

(3)确保开发的软件需求规格说明包含软件安全性分析已标识出的软件安全性要求。

(4)确保软件设计和实现正确考虑了软件安全性要求。

(5)确保制定适当的验证与确认要求，以保证正确实现软件安全性要求。

(6)确保测试计划和规程满足软件安全性验证要求的意图。

(7)确保软件安全性验证工作的结果令人满意。

3.2 系统安全性分析

当软件的作用被确定时，在系统需求阶段进行的初步系统安全性分析(例如,初步危险分析，PHA)，开始识别与具体设计方案和(或)操作有关的危险。这些初步分析，以及随后的系统与软件安全性分析，将识别可能引起危险或将被用于控制危险发生的软件。这些软件必须归入安全性关键软件类并经受软件安全性分析。典型的安全性关键软件一般是这样的软件:如果它不执行,或者执行得不正确，无意或不按规定顺序执行就可能导致危险或使危险状态存在。例如：①具有潜在危险的功能模块和(或)硬件用于直接指挥与控制的软件;②监视关键硬件部件的软件;③对可能的系统关键条件和(或)状态进行监视的软件。

系统安全性分析是识别软件安全性要求的第一步，这些要求对于拟订软件需求规格说明是必要的。这些要求必须提供给开发者以便纳入软件需求文档中。软件安全性要求的例子包括界限(例如红线、边界值)、事件顺序、计时限制、界限的相互关系、表决逻辑、危险硬件失效判别、失效容限、警告和报警界面、危险指令等等。

系统安全性分析在整个项目寿命周期中连续进行。软件安全性分析需要连续进行，以便评审系统分析的结果，保证在必要时把系统级的变化和发现的问题综合到软件中。此外，软件安全性分析是整个系统安全性分析的一个特殊部分,而且不是孤立进行的。

3.3 软件安全性

软件安全性必须是整个系统安全性和软件开发工作的一个组成部分。软件安全性工作的目的在于确保在整个软件寿命周期内考虑安全性。因此，软件安全性活动在系统和软件开发期间从概念阶段开始到运行和维护的各个阶段都要进行。在软件开发期间从上到下都参与安全性分析，分析所发现的安全性问题并做出报告。这样做有助于及时地并以较少的代价解决问题。

软件安全性工作需要所有参与开发该软件的部门协同努力,其中包括型号项目经理、硬件和软件设计者、安全分析员、质量保证和操作人员。从事软件安全性工作的人员还必须与可靠性、保密性、独立验证与确认(如果有的话)以及人因素等学科的人员接触。

3.3.1 寿命周期各阶段的软件安全性任务

下面几节说明，在采用瀑布模型作为基本寿命周期的方法时，适于软件开发各阶段的软件安全性任务。许多安全性分析在软件开发期间是迭代进行的（即一个阶段的结果提供给下个阶段进行分析)。随着项目的细化,安全性分析也不断地深入、完善。

3.3.1.1 软件安全性计划

对于应用本标准的每项软件的获取都必须制定软件安全性计划。该计划必须记入要求的软件管理计划或安全性管理计划。软件管理计划必须按照 NASA－STD－2100－91 “航宇局软件文档标准”中的软件管理计划编制。安全性管理计划必须按照 NHB1700.1 (V1－B) 的 306 节写成文档。如果安全性计划记载在多个计划中，那么每个计划都必须包含一个与其他计划中安全性活动对照的交叉参照表。计划必须说明如何实施本标准的活动。计划必须规定要进行的活动、进行这些活动的日程表和将产生的产品。计划必须阐明系统安全性分析、软件安全性分析和软件开发工作之间的关系。计划必须特别强调产生、实现、追踪和验证安全性关键需求的机制。

3.3.1.2 软件需求规格说明拟订

在软件需求规格说明拟订期间，要对分配给软件的系统级要求进行分析并作为软件需求记入文档。开始制定测试计划时，要确定验证每项要求的方法并记入初步测试计划中，要识别风险和建立风险管理控制机制。在软件开发寿命周期阶段必须完成两项软件

安全性任务：

(1)拟订软件安全性需求；

(2)分析软件需求中为防止潜在危险而采取的措施。

成功地为软件需求规格说明拟订安全性要求，对于开发安全性软件并在寿命周期早期就能将安全性引入软件中（这样做所需代价较低）是十分重要的。拟订软件安全性需求的过程包含分析系统安全性需求、硬件、软件和用户的接口，以及下述系统性能方面，其中软件是一种潜在危险因素，或是对由系统安全性分析识别出的危险进行控制的部件。为了拟订确保有关危险得到适当解决所必需的软件需求，必须分析这些系统要求、接口和性能。拟订的软件安全性要求必须成为软件需求规格说明的一部分。软件安全性要求必须与 NHB1700.1(V1－B) 102 节规定的优先次序及 108 节规定的风险评估方法一致。

软件安全性要求分析必须遵循软件需求安全性分析规定的要求。

3.3.1.3 软件概要设计

软件概要设计过程将进行实现软件需求的高层设计。作为这个过程的一部分，必须将“软件需求规格说明拟定”一节拟订的所有软件安全性要求综合到高层软件设计中。该设计过程必须包括在整个软件中。为了实现软件安全性要求所用的安全性设计特征和方法(例如，禁止、陷阱、互锁和断言)，还要制定安全性专用编码标准。这些标准将识别安全性关键代码注解的需求和对可能降低软件安全性的某些语言特征的使用限制。软件安全性要求定位到软件设计之后，必须确定部件层的安全性关键计算机软件部件(SCCSCs)。这些就是实现软件安全性要求的全部软件部件或可能影响它们输出的与 SCCSCs 接口的部件。

在这个阶段，必须制定用来验证软件安全性要求是否得到正确实现的测试计划。该计划必须确定验证全部软件安全性要求和评价对潜在危险的正确响应所采用的测试过程。安全性活动必须评审该计划，以便协调一致。

必须按照“软件概要设计安全性分析”一节的要求分析结构设计和测试计划。

3.3.1.4 软件详细设计

软件详细设计过程对实现软件需求和高层设计的软件单元进行低层设计。作为该过程的一部分，必须把部件级 SCCSCs 的高层设计，包括安全性设计特征和方法开发到低层单元设计中。开发了详细设计之后，必须确定单元层 SCCSCs。这些 SCCSCs 是实现软件安全性要求的全部软件单元或可能影响它们输出的与 SCCSCs 接口的单元。

在这个阶段，必须将与安全性有关的信息综合到所有用户手册中。这些信息包括警告、报警、处理安全性有关过程的规程和危险。

在这个阶段，必须开发验证软件安全性要求的测试过程。这些与安全性有关的过程必须包含(但不限于)反面(Negative)、拟制危险 (No－Go)，越界 (Off－norminnal) 和强度(Stress)测试，以确保软件正确地对危险作出响应，且不再引起任何危险。这些测试过程必须支持计算机软件配置项 (CSCI) 测试、系统测试和验收测试。安全性活动必须评审这些测试过程，以便协调一致。

必须对详细设计进行分析以确认潜在危险与测试过程的一致性，确保按照“软件详细设计安全性分析”一节的要求加入与安全性有关的测试。

3.3.1.5 软件实现

软件实现是用选定的编程语言将详细设计转换成代码。代码必须实现设计过程中开发的安全性设计特征和方法。必须对安全性关键代码加以注解，以使未来因代码更改而引起危险状态的可能性降低。必须分析代码，并按照“代码安全性分析”一节确定潜在的危险。

必须验证单元层软件安全性要求在

SCCSCs 中是否得到正确实现，以确保准确符合软件工程及安全性详细设计的要求。每个单元的代码验证必须在将该单元综合到主代码包之前完成。

验证软件安全性要求的集成测试和验收测试过程必须在这个阶段编制完成。安全性活动必须评审已完成的测试过程，以便协调一致。

必须分析测试过程，以验证按照“代码安全性分析”一节的要求而引入的与安全性有关的测试。

3.3.1.6 软件集成和验收测试

必须进行测试以验证是否正确地综合了软件安全性要求。测试必须显示危险已被消除或已被控制到可接受的风险水平。测试期间识别出的另外的危险状态必须在软件交付或使用之前进行彻底分析。集成和验收测试必须包括 SCCSCs 的软件安全性测试。验收测试必须验证 SCCSCs 和系统硬件及操作员能否一起正确地运行，必须验证在强度测试条件下和有系统故障时软件能否正确运行。

必须按照“软件测试安全性分析”一节的要求分析测试结果。

3.3.1.7 软件运行和维护

进行更改时必须采用本章内容为详细说明、开发、分析和测试 SCCSCs 规定的各种软件安全性作出处理。必须包括以下活动：进行危险分析；更新软件安全性要求；识别新的 SCCSCs；更新 SCCSCs 的规格说明、设计和操作员文档；更新并增添对安全性关键代码的注解；测试 SCCSCs。测试必须包括回归测试，以验证是否正确地实现了有关的软件安全性要求。

3.3.2 独立于阶段的任务

下面几小节说明在整个寿命周期都要完成的软件安全性任务。

3.3.2.1 安全性要求可追踪性

必须采用一个系统来追踪设计、实现和测试过程的软件安全性要求，追踪系统还必须反映软件安全性要求与系统危险报告之间的关系。

3.3.2.2 偏差报告和跟踪

必须采用一个系统来对已入基线的软件产品的安全性有关的偏差、问题和失效进行闭环跟踪。必须评审所有偏差报告的安全性影响，使得直到提出该安全性有关偏差报告为止的安全性活动协调一致。必须按照“软件更改分析”一节的要求分析为纠正偏差所进行的更改。

3.3.2.3 软件更改控制

必须对 SCCSCs 要求、设计、代码、系统、设备、测试计划、过程或准则所做的更改、改进和修补进行评估，以确定所建议的更改对系统/子系统安全性的影响。必须按照“软件更改分析”一节的要求进行分析。

3.3.2.4 安全性大纲评审

必须对安全性大纲（Safety program）进行评审，以确保对危险进行了足够的安全性控制。软件安全性活动必须支持系统安全性评审过程。

3.4 软件安全性分析

下面几节将对软件安全性分析提供一些系统的方法。这些方法支持“软件安全性”一节所叙述的对软件安全性的处理。

3.4.1 软件需求安全性分析

软件需求安全性分析（SSRA）必须进行并写成文档。必须用系统级 PHA 和系统方案设计作为 SSRA 的输入。SSRA 必须检查系统级的软件需求、接口控制文档和正在进行的软件需求规格说明的拟订工作，以便：

(1)识别安全性关键的软件需求；

(2)确保高层安全性要求的分析正确、完备；

(3)提出与设计和测试过程安全性有关的建议。

必须分析所有的软件需求，以便识别系统分析未包括的附加危险，识别不正确地分配给软件的系统要求或接口要求。然后必须通过增加或更改接口、系统要求和(或)软件

需求来解决识别出的危险。SSRA 必须考虑特殊要求与特殊限制范围：顺序不对事件的保护要求（例如“if－then”语句）；计时器；互相依赖界限的关系逻辑；表决逻辑；危险命令处理要求；故障检测、隔离和恢复；以及容失效能力的切换逻辑。

SSRA 的输出必须用作后续软件安全性分析的输入。必须在软件需求评审（SRR）/软件规格说明评审（SSR）以及系统级安全性评审时说明 SSRA。必须将 SSRA 的结果提供给进行中的系统安全性分析活动。

3.4.2 软件概要设计安全性分析

软件概要设计安全性分析（SADA）必须进行并写成文档。必须用概要设计、SSRA 结果和系统危险分析作为 SADA 的输入。SADA 必须检查软件需求规格说明、测试计划和正在进行中的概要设计，以便：

（1）将被 SSRA 确定为实现软件安全性要求的那些软件部件标识为 SCCSCs，而且影响 SCCSCs 输出的那些软件部件也必须标识为 SCCSCs；

（2）确保与软件安全性要求和安全性设计建议相关的概要设计正确、完备；

（3）为详细设计提出有关安全性的建议；

（4）确保软件安全性要求的测试覆盖率，并对测试过程提出建议。

SADA 的输出必须用作后续软件安全性分析的输入。必须在软件概要设计评审（PDR）和系统级安全性评审时说明 SADA。必须将 SADA 的结果提供给进行中的系统安全性分析活动。

3.4.3 软件详细设计安全性分析

软件详细设计安全性分析（SDDA）必须进行并写成文档。必须用详细设计、SSRA 和 SADA 的结果，以及系统危险分析作为 SDDA 的输入。SDDA 必须检查软件需求规格说明、测试过程和正在进行的详细设计，以便：

（1）将识别出的 SCCSCs 细化为实现 SSRA 规定的软件安全性要求的单元级软件部件，影响 SCCSCs 输出的那些单元级软件部件也必须标识为 SCCSCs；

（2）确保与软件安全性要求、概要设计，以及与安全性设计建议有关的详细设计的正确、完备；

（3）为代码的实现提供与安全性有关的建议；

（4）开发与安全性有关的信息，以便写入用户指南和其他有关的文档中。

SDDA 的输出必须用作后续软件安全性分析的输入。在软件关键设计评审（CDR）和系统级安全性评审时必须说明 SDDA。必须将 SDDA 的结果提供给进行中的系统安全性分析活动。

3.4.4 代码安全性分析

代码安全性分析（CSA）必须进行并写成文档。必须用代码、SSRA、SADA 和 SDDA 的结果，以及系统危险分析作为 CSA 的输入。CSA 必须检查软件需求规格说明、测试过程和正在进行的代码开发，以便：

（1）确保与软件安全性要求、详细设计，以及与安全性编码建议有关的编码的正确、完备；

（2）识别由输入/输出计时、多重事件、顺序不对事件、事件失效、有害环境、死锁、错误事件、不合适的量、不正确的极性，以及硬件失效敏感性等引起的潜在的不安全状态；

（3）确保充分地注解 SCCSCs；

（4）确保软件安全性要求的测试覆盖率；

（5）校正与安全性有关的信息并写入用户指南和其他有关文档。

在准备测试评审（TRR）时必须介绍软件安全性分析的情况。必须将 SCA 的结果提供给进行中的系统安全性分析活动。

3.4.5 软件测试安全性分析

必须分析测试结果以验证所有的安全性要求是否均已满足。该分析还必须验证所有识别出的危险是否均已消除或已控制到可接受的风险水平。必须将测试安全性分析的结

果提供给进行中的系统安全性分析活动。

3.4.6 软件更改分析

软件更改分析必须评价所建议的更改是否能引起危险状态,影响危险控制,增加危险状态的可能性,对安全性关键软件有不利的影响,或改变软件部件的关键性。该分析还必须确保任何受到影响的文档均被更新,以反映已进行的与安全性有关的一切更改。

4. 质量保证条款

质量保证(QA)必须做到:

(1)软件安全性计划得到执行、批准和实现;

(2) 软件安全性活动提出的技术建议由更改控制机构评审和考虑,并适时加以实现;

(3)评审和审计了软件安全性关注点、要求和指南;

(4) 软件安全性处理过程、产品标准,以及相关过程得到遵守和满足;

(5) QA 必须监督所执行任务的关键要素,以保证批准的计划和过程已得到适当的执行。

5. 包装

这一章不适用于本标准。

6. 附加信息

这一章不适用于本标准。

附录 A 术语汇编

断言(Assertions)

规定必须存在的某种程序状态或规定在程序执行过程中某特定点上程序变量必须满足的一组条件的逻辑表达式。断言类型有输入断言、循环断言和输出断言等三种。(IEEE 标准 610.12 - 1990)

代码安全性分析(Code Safety Analysis(SCA))

对程序代码和系统接口中可能引起或促使影响安全性的意外事件的那些事件、故障和条件进行的分析。

命令(Command)

引起接受方执行某种行动的任何信息。

计算机软件配置项(Computer Software Configuration Item (CSCI)

指定要进行配置管理并在配置管理过程中作为单一实体对待的软件集合。(IEEE 标准 610.12-1990)

概念(方案)/概念(方案)的(Concept/Conceptual)

软件开发周期中的一段时间。在这一期间,通过文档(例如:与项目有关的需求说明、先期计划报告、项目起动备忘录、可行性研究、系统定义、文档、条例、过程或与计划有关的政策来叙述并评价用户的需要。

关键设计评审(Critical Design Review(CDR))

为了验证一个或多个配置项是否满足规定的需求所进行的评审:包括确立配置项与其他的设备、设施、软件和人员项的适应性;评估每个配置项的风险区域;在应用时,评估可生产性分析结果;评审初步的硬件产品规格说明;鉴定初步的测试计划和初步的使用和保障文档的充分性。(IEEE 标准 610.12 - 1990)

对于计算机软件配置项 (CSCIs) 来说,这种评审将注意力集中在确定详细设计的可接受性、性能和设计解法的测试特性上,并关注使用和保障文档的充分性。

死锁(Deadlock)

计算机的处理过程因两个或更多设备或进程各自正在等待分派给其他设备的资源而被挂起的状态。(IEEE 标准 610.12 - 1990)

失效(Failure)

系统或部件不能在规定的性能要求范围内完成所需要的功能。

容失效能力(Failure Tolerance)

系统或分系统,在其硬件、固件和软件出现失效

时，完成其功能或控制危险发生的能力。

故障(Fault)

被认为是处于异常并有理由采取某种纠正措施产品的状态的任何改变。例如机内检测(BIT)/机内检测设备(BITE)所报告的设备差错，传感器值超限状态，与一些设备的通信丢失，设备断电，总线传输事务中通信错误，软件异常(例如被零除、文件找不到)，已舍弃的命令，测量的性能值超出控制值或预期值，计算机程序中的一个错误步(Step)、错误进程或错误的数据定义等。故障是失效可能发生的预先(Preliminary)迹象。

故障检测(Fault Detection)

揭示故障或为了揭示故障而设计的一个过程；确定故障已经发生的过程。

故障隔离(Fault Isolation)

确定故障位置或故障源的过程。

故障恢复(Fault Recovery)

消除未产生永久性重构故障的过程。

固件(Firmware)

装在存储器中，并在处理过程中不能被计算机动态修改的计算机程序和数据。

危险(Hazard)

存在或潜在的可能引起或促使事故发生的状态。

危险命令(Hazardous command)

在执行(包括无意、顺序不对或不正确的执行)中可能导致某种确定为关键的或灾难性危险的命令，或在其执行时能降低对危险的控制(包括降低针对某种危险的失效容限能力或消除针对某种危险的抑制)的命令。

独立验证与确认(Independent Verification and Validation(IV&V))

代表软件需方并完全独立于供方的一个组织，对软件开发周期各阶段的产品独立地进行评审、验证和确认的过程。

禁止(Inhibit)

一种设计特征。它提供在能源和功能之间的实际阻断(例如，电池和烟火启动器之间的继电器或晶体管、推进剂储箱和推进器之间的掣子阀等)。

互锁(Interlock)

在特定条件下，阻止继续操作的硬件功能或软件功能。

寿命周期(Life Cycle)

从开始设想开发软件产品到该软件不再提供使用为止的时间周期。软件寿命周期通常有八个阶段：概念研究和起始、需求分析、概要设计、详细设计、实现、集成和测试、验收和交付、维护工程和使用。

事故(Mishap)

导致人员死亡、受伤、职业病，设备受损或丧失，或环境受到破坏的意外事件或一系列事件；也可能是一种偶然事故。

反面测试(Negative Testing)

在响应超限或非法输入时，确保软件不进入危险状态或产生会引起系统危险的输出所进行的软件安全性测试。

抑制危险测试(No Go Testing)

在响应特定危险状态时，确保软件能执行规定的处理并进入规定的安全状态所进行的测试。

概要设计评审(Preliminary Design Review (PDR))

一种评审活动。它将评价一个或多个软件配置项选用的设计途径的进展、技术恰当性和风险解决方法；评价每个配置项的设计与其需求的适应性；评价与所选用的制造方法和过程相关的确定程度并评估相关的技术风险；确定配置项与其他设备项、设施、软件和人员之间的物理接口和功能接口的适应性；适当时也可评价初步的使用和保障文档。(IEEE标准 610.12－1990)

对于 CSCIs 来说，这种评审将集中在：(1)评价所选的概要设计和测试途径的进展、一致性和技术恰当性；(2)软件需求和概要设计的适应性；(3)使用和保障文档的初步版本。

初步危险分析(Preliminary Hazard Analysis(PHA))

在系统级进行的分析，旨在识别安全性关键区域，提供初步的危险评估和确定必要的危险控制和后续活动。

风险(Risk)

当它应用于安全性时，风险是指暴露到受伤害或受损失中的概率。它是可能发生不希望事件的频率、造成后果的潜在严重性，以及与该频率及严重性有关的不确定性的一个函数。

安全性分析(Safety Analysis)

为获得和评价与系统安全性有关的特定信息的系统化的和有序的处理过程。

概要设计安全性分析(Safety Architectural Design Analysis (SADA))

对高层设计进行的分析，旨在验证安全性要求

是否正确引入，并分析安全性关键的计算机软件部件(SCCSCs)。

安全性关键的(Safety – Critical)

是指一些软件操作，如果这些操作不执行、不按规定顺序执行或执行得不正确，就可能导致不正确的控制功能（或缺少系统正确操作所要求的控制功能）。这些不正确的控制功能可能直接或间接地引起某种危险状态或允许某种危险状态存在。

安全性关键的计算机软件部件(Safety – Critical Computer Software Component (SCCSC))

其错误(无意的或非授权出现、要求出现时不出现、不按规定顺序出现、与其他功能组合在一起出现、错误的值)能导致潜在的危险或丧失系统的可预测性或丧失系统控制的那些计算机软件部件（过程、模块、函数、值或计算机程序状态）。

安全性关键的软件(Safety – Critical Software)

包括下述三种软件：(1)在硬件部件的条件或状态范围内直接进行指挥与控制，并且如果不执行、不按规定顺序执行或执行得不正确就可能导致不正确的控制功能（或缺少系统正确操作所要求的控制功能），从而可能引起危险或允许危险条件存在；(2)监视硬件部件状态，并且，如果不执行、不按规定顺序执行或执行得不正确就可能导致操作员或伴随系统错误地判定，从而可能引起危险或允许危险条件存在；(3)在硬件部件的条件或状态范围内直接进行指挥与控制，并且，如果不执行、不按规定顺序执行或执行得不正确就可能与其他人员、硬件或环境故障一起引起危险或允许危险条件存在。

详细设计安全性分析(Safety Detailed Design Analysis (SDDA))

对安全性关键的计算机软件部件进行的分析，以验证是否正确综合了安全性要求并识别出其他的危险条件。

软件需求评审(Software Requirements Review(SRR))

对一个或多个软件配置项的规定需求进行的评审，旨在评价它们与系统要求的对应性和对系统要求的说明，并确定它们是否已构成进入配置项概要设计的令人满意的基础。(IEEE 标准 610.12 – 1990)

与 DOD – STD – 2167A 的软件规格说明评审相同。

软件需求规格说明(Software Requirements Specification (SRS))

对软件及其外部接口的基本要求（功能、性能、设计限制和属性)的文档。(IEEE 标准 610.12 – 1990)

软件安全性需求分析(Software Safety Requirements Analysis(SSRA))

为了考察系统与软件的需求与方案设计，和识别解决方案中的不安全模式（例如，顺序不对、错误事件、死锁和不能控制)而进行的分析。

软件规格说明评审(Software Specification Review (CSSR))

与软件需求评审同义。

软件安全性(Software Safety)

在整个软件寿命周期内，应用系统安全性工程技术，来确保软件采用能提高系统安全性的有效措施，并确保那些可能降低系统安全性的错误均已被排除或控制在可接受的风险水平。

系统安全性(System Safety)

在系统寿命周期的各个阶段，应用工程和管理原理、准则和技术，以便在使用效果、时间和费用的约束范围内使安全性最优并降低风险。

测试准备评审(Test Readiness Review(TRR))

一种评审活动。其目的是评价一个或多个配置项的初步测试结果；验证每个配置项的测试过程是否完备，是否符合测试计划和测试说明并满足测试要求；验证项目是否已准备好可继续进行配置项的正式测试。(IEEE 标准 610.12 – 1990)

陷阱(Trap)

为在正常程序逻辑之外提供附加检查而设计的，用于监视程序执行和关键信号的软件特征。陷阱对未检测出的软件错误、硬件故障和预料之外的危险条件提供保护。

确认(Validation)

(1)为了确保必要功能的完备并可追踪，而采用的支持或证实安全性要求的评价技术。

(2)在软件开发过程结束时，评价软件的过程，以确保符合软件需求。

验证(Verification)

(1)确定软件开发周期某个阶段的产品是否达到前一阶段制定的要求的过程(还可参见确认)。

(2)程序修正的正式证明。

(3)进行评审、审查、测试、检查、审计等活动，或以其他方式确定并以文档记载产品项、过程、服务或文档是否符合规定的要求。

放弃(Waiver)

当采用替代方法来降低风险，或当提高的风险

水平已被管理部门接受时，由当权者作出的准予偏离某个特定安全性要求的变化。

附录 B　缩略语

BIT	机内检测
BITE	机内检测设备
CDR	关键设计评审
COTS	现货商品
CSA	代码安全性分析
CSCI	计算机软件配置项
DID	数据项说明
DOD	国防部
FDIR	故障检测、隔离和恢复
GFE	政府提供的设备
IEEE	电气和电子工程师协会
IV&V	独立验证和确认
MIL－STD	军用标准
NHB	航宇局手册
NMI	航宇局管理指令
QA	质量保证
PDR	概要设计评审
PHA	初步危险分析
SADA	概要设计安全性分析
SCCSC	安全性关键的计算机软件部件
SDDA	详细设计安全性分析
SSRA	软件需求安全性分析
SRR	软件需求评审
SRS	软件需求规格说明
SSR	软件规格说明评审
TRR	测试准备评审

美国航宇局安全关键软件的分析和开发指南

NASA－GB－1704.13－96

王 纬(译)　　何新贵(校)
(航天工业总公司第二〇四研究所)　　(北京系统工程研究所)

本文件是一个适用于全航宇局（NASA）范围的软件大纲，用来促进全航宇局范围内软件工程的不断改进。1995 年 7 月 13 日制定的航宇局软件战略计划描述了这个大纲的目标和战略。更多的信息可从万维网上的软件 IV & V 设施中得到，网址是 http: // www.ivv.nasa.gov

1. 引　　言

本航宇局安全关键软件分析和开发指南，是航宇局刘易斯研究中心安全性和任务保证办公室根据航宇局的研究题目(RTOP)的任务编制的。航宇局总部制定的航宇局安全性标准 NSS 1740.13[1]阐述了软件安全性分析“由谁做、做什么、何时做和为何做”。本软件安全性分析指南则阐述“如何去做”。

1.1 范围

本文件的重点放在安全关键软件的分析和开发上。本指南也可以用于分析和开发那些在非易失存储器(例如，ROM 或 EPROM)中驻留着软件的固件。

本指南描述分析每项任务所需的资源数据、进行分析所用的方法学和工具，以及输出的产品。它还说明在总的风险管理活动中如何使用这些产品。

然后，本指南更详细地描述了技术和规程。为使本指南比较切实可行，其中包含了一些分析示例和分析过程中可能遇到的问题和陷井。本指南并不包含新的分析技术，它是一些已在航宇局和工业界使用的技术的协同集。关于各种技术的有效性有些意见很不相同，本指南试图介绍这些意见而不判定其适合程度。在大多数情况下，很少或不存在可用来定量地评价或比较这些技术的“度量”。

本文件参考了许多现有指出各种分析技术的文档，并引用了在别处很好描述过的一些技术。如果有规定某个特定文档格式和/或内容的航宇局标准或指南，该标准或指南便被引用，用户应按照那个文档的说明执行。

除了现有文献中的一些技术外，介绍了一些已经成功地开发和应用于系统级自顶向下软件危险分析的实用方法。这些实用方法与 NSTS 13830 “航宇局有效载荷系统安全性要求的实现规程”[2]类似。

本文收集、评价并比较了公开文献中许多不同的自底向上的技术，阐述了每项技术对总的软件开发和保证目标带来的价值与成本的对比关系。

期望读者对航宇局的系统安全性分析和/或软件开发方法学有一定的了解。不过，并不要求也不假定读者具有这两方面的任何经验。对航宇局软件开发和系统安全性方法学完全不了解的读者读本指南某些部分时可能有困难。附件 A 包含本指南所用的缩略语

和术语定义。

“软件成份”这个术语用来在一般意义上表示重要的软件开发产品，如软件需求、软件设计、软件代码或程序集、软件测试，等等。

1.2 目的

本指南的目的是帮助参与安全关键软件开发和保证的组织。（即软件开发者、软件产品保证组织、系统安全性和软件安全性组织）。

1.3 感谢

本指南中所介绍的许多材料直接或间接地来源于各种组织（航宇局、政府机构、技术文献），以它们的有关文件为基础，并包含一些以前未用文档记载的原始材料。这些来源太多，这里无法一一列举。需要特别感谢加利福利亚洲 Pasadena NASA/Caltech 喷气发动机实验室，本指南几乎完全采用了他的《软件系统安全性手册》[4]，只有几章稍有改动。

我们感谢下列参加评审本指南草案的航宇局工程师和承包商，包括：

评审者	航宇局	城　市	承包商
Gilbert White	NASA HQ Code - QS	Washington D.C.	
Paul Senger	NASA HQ Code - QS	Washington D.C.	Vitro Corp
Ricky Butler	NASA Langley	Hampton, VA	
Michael Holloway	NASA Langley	Hampton, VA	
James Watson	NASA Langley	Hampton, VA	
Judith Grigory	NASA Marshall	Huntsville, AL	
Bonnie Pfitzer	NASA Marshall	Huntsville, AL	Madison Research Corp
Scott Seyl	Nasa Johnson	Houston, Tx	
David Tadlock	NASA Johnson	Houston, Tx	
Dr. Robyn Lutz	JPL Caltech	Pasadena, CA	University of Iowa
George A BENDROTH	NASA Lewis	Cleveland, OH	
Michael Ciancone	NASA Lewis	Cleveland, OH	
Sell James	NASA Lewis	Cleveland, OH	
David Marchese	NASA Lewis	Cleveland, OH	
Susan Cole	NASA Lewis	Cleveland, OH	Raytheon Corp
James Roupp	NASA Lewis	Cleveland, OH	Raytheon Corp
Michele D. Smith	NASA Lewis	Cleveland, OH	Raytheon Corp
William Stuckey	NASA Lewis	Cleveland, OH	Raytheon Corp

我们还要感谢美国安全性工程师协会允许复制 Gowen Lon D 和 Collofello James S 的论文《安全关键软件系统的设计阶段考虑》。（《专业安全性》的一部分，1995.4.）

2. 系统安全性大纲

系统安全性大纲是进行安全关键软件分析和开发的先决条件。这个大纲勾画出贯穿开发周期进行一系列系统级和子系统级分析的类型和日程表。它还阐述应如何处理安全性分析的结果以及结束批准过程。

系统安全性大纲的第一个活动是初步危险分析（PHA），在 2.1 节讨论。这一步的结果是一个危险原因和潜在危险控制清单，要传递给进行安全性需求开发活动，这个活动在 2.2 节讨论。

一个航天系统通常包含三个要素：硬件、软件和一个或多个操作人员（例如，地面控制人员、任务专家或飞行器驾驶员）。其中每一个要素又可进一步分解成子系统和部件。尽管单个要素、子系统或部件分别考虑时常是不危险的，但组合成的系统可能展示各种安全性风险或危险。对于软件尤其如此。

虽然常常宣布“软件不能引起危险”，这只在软件驻留在非危险平台上且不与任何危险硬件或操作人员接口或交互作用时才正确。与此类似，一个硬件部件，如一个电器开

关或一个液体阀门，作为一个单独部件也许是不危险的，但当用作系统中控制危险的一个抑制部件时就可能变成危险的或安全关键的。因此，本指南始终把软件看成一个更大系统中的一个系统子集(即一个子系统)。

在可以对用于危险系统或环境中的软件进行分析或开发之前，必须进行系统的PHA。一旦初始的系统PHA结果可用，安全性需求便向下传递，子系统和部件的危险分析就能开始。系统安全性大纲分析遵循系统和软件开发工作的生存周期，从初步危险分析(PHA)开始。

2.1 初步危险分析(PHA)

PHA是"特定的"软件安全性需求(即对于具体系统结构来说是独特的)的第一个来源。它是进行任何软件安全性分析的先决条件。

PHA是一系列系统级危险分析的第一步，系统级危险分析的范围和方法学，在NASA NHB 1700系列文档[1]和NSTS 13830"航宇局有效载荷系统安全性需求实现规程[2]"中描述。解释或重复其它航宇局文档中描述过的系统危险分析过程，不是本文档的目的。然而，为了软件开发者、管理者以及其他不熟悉航宇局系统安全性的人，下一节概括了这些过程并说明如何将PHA的输出传递给软件安全性分析过程。

2.1.1 PHA方法

下述内容摘自NHB 1700.1(V1-B)附录-H:

初步危险分析"用文档记载哪些一般危险(见表2-1，一般危险检查表用作一般危险检查单的一个样本)是与设计和运行概念相关联的。这个文档提供一个初始框架，作为要求在程序设计和开发过程中进行跟踪和解决的危险和相关风险的主表(或危险目录)。PHA可用来识别安全关键系统，这些安全关键系统将要求在设计期间运用失效模式影响分析(FMEA)和进一步危险分析。大纲必须要求进行PHA并用文档记载下来，以便得到系统方案的初始风险因素表。PHA工作必须在大纲的概念探索阶段或生存周期早期阶段就开始。PHA考虑硬件、软件、以及运行概念。对PHA所识别出的危险，将根据从类似系统的最适用的数据、已得到的其它教训、以及与所提出的设计或功能相关联的危险来评估其风险。关于意外事故和已得到的教训的信息可从意外事故报告和纠正措施系统(MR/CAS)和已有教训信息系统(LLIS)得到。根据PHA进行的风险评估将用于：确保在进行设计方案的权衡研究时考虑安全性；为大纲和设计规格说明开发安全性要求，包括安全关键的监控软件；以及定义运行条件和约束条件。"

2.1.1.1 识别危险

整个系统的初步危险分析自顶向下地进行，以识别危险和危险条件。目标是把所有的可信危险识别出来。起初分析是硬件驱动的，考虑硬件执行机构、末端操纵装置和能源、以及可能产生的危险。对于每一个识别出的危险，PHA指出危险原因和备选的控制方法。将这些危险和危险原因对应到系统功能和它们的失效模式。大多数关键功能都与一个或多个系统控制相关联。这些控制功能控制该部分系统的操作、监视和/或保安，而安全性评估必须考虑该系统的所有有关的各种子系统，包括硬件/软件和操作员。

为了确保充分覆盖功能安全性的所有方面，将系统功能分为如下两种类型可能是有益的：

(1)必须运行的功能(MWF)

(2)必须不运行的功能(MNWF)

系统规格说明起初常定义某些系统功能的关键性(例如，安全关键的)，但可能不完全。这种关键性通常只用必须运行的系统功能特性这样的术语来表达，往往忽略必须不运行功能的关键性。PHA应定义所有危险的MNWF(必须不运行的功能)和MWF(必须运行的功能)。示例：

表 2－1　一般危险检查表

1.污染/腐蚀	A.化学分解 B.化学取代/化合 C.潮气 D.氧化 E.有机的(霉菌/细菌,等等) F.粒子 G.无机的(包括石棉)
2.释放电/电击	A.外部电击 B.内部电击 C.静电释放 D.电晕 E.短路
3.环境/气候	A.雾 B.闪电 C.沉降(雾/雨/雪/冻雨/冰雹) D.沙/灰尘 E.真空 F.风 G.温度极值
4.火灾/爆炸	A.化学变化(放热的/吸热的) B.燃料和氧化剂在有压力和点火源的状态 C.压力释放/挤压 D.高热源
5.撞迫/碰撞	A.加速度(包括重力) B.分离的设备 C.机械冲击/振动/音响 D.流星体/陨石 E.移动/旋转的设备
6. 丧失可居住的环境*	A.污染 B.高压 C.氧含量低 D.低压 E.毒性 F.低温 G.高温
7.病理/ 生理/心理	A.加速度/冲击/撞击/振动 B.大气压力(高、低、快变) C.湿度 D.疾病 E.噪声 F.尖锐的棱边(sharp edges) G.睡眠,缺少 H 可见性(强光,窗户/防护帽发雾) I.温度 J.工作负荷,超额 K.高处(可能跌落)
8.辐射	A.电磁 B.电离辐射(包括氡) C 非电离辐射
9.温度极值	A.高 B.低 C 变化

* 健康问题需要与职业保健人员协作。

(1) 某个科学实验按其系统功能而言可能具有标志为3的系统关键性（非关键的），因为丧失基本的科学数据并不出现危险。不过该实验仍然可能由于在维护期间无意地激活电源而产生电击之类的危险。在维护期间电源的激活是一个 MNWF。

(2) 一个实验，如果不保持负压(真空)，可能释放毒气。保持负压是一个 MWF。

(3) 空中交通管制系统和飞机飞行控制系统都被设计成能预防两架飞机在邻近飞行时发生碰撞。碰撞避免是一个 MWF。

(4) 航天飞机的火箭发动机在 STS 货仓中时可能无意地点火。在那个时刻发动机点火是一个 MNWF。显然，当需要点火时，它就变成一个 MWF。

如果 PHA 中识别的功能没有被包括在系统规格说明中，就应该把它补上，以便阐述对这些功能的控制。

2.1.1.2 风险评估

危险由系统安全性组织按危险严重性和发生的可能性进行排序，如表 2－2(危险排序－系统风险指标)中所示。

表 2－2　危险排序－系统风险指标

严重性等级	发生可能性			
	可能	偶尔	极少	不可能
灾难性的	1	1	2	3
关键的	1	2	3	4
临界的	2	3	4	5
可忽略的	3	4	5	5

1＝最高优先级（最高系统风险），5＝最低优先级（最低系统风险）。

下列危险严重性等级的定义来自 NASA NHB 1700.1.

灾难性的：整个系统丧失，或丧失生命或永久丧失工作能力；

关键的：重大系统损坏，严重受伤或暂时丧失工作能力；