载人航天工程软件工程化技术标准若干技术探讨

刘　杰[1]　叶东升[1]　牛爱民[2]

（1 北京计算机应用与仿真技术研究所　2 中国载人航天工程办公室）

摘　要　根据载人航天软件系统的新要求、新特点，针对载人航天工程软件工程化技术标准，从工程应用角度探讨软件系统设计、软件系统验证、软件安全性等技术问题，为标准的执行和进一步完善提供参考。

关键词　载人航天　软件工程　标准　技术

1 引　言

为了进一步加强载人航天软件工程标准化工作，满足下一步任务要求，载人航天工程总体颁布了载人航天工程软件工程软件化技术标准（以下简称“技术标准”）。

“技术标准”是对载人航天软件工程技术和工程管理中的概念、过程、程序和方法等内容规定统一要求的标准。“技术标准”为了满足载人航天工程的总体要求，促进载人航天工程技术的发展，从载人航天工程全局出发，为载人航天软件研制工作提供了行之有效的技术指导。

目前，软件系统设计、软件系统验证、软件安全性等技术不仅是软件工程领域研究的重大技术，也是研制载人航天软件系统的关键技术，这些软件技术不仅将直接影响软件的功能、性能、可靠性、安全性，也将直接影响系统的成败，其研究、应用和标准化工作备受各国政府和军方的重视。工程技术人员需要逐步积累在载人航天软件系统研制中应用软件系统设计、软件系统验证、软件安全性技术的实践经验，以满足载人航天任务的需要。本文试图从应用角度对上述技术进行探讨，供工程技术人员在实际工作中参考。

2 软件系统设计

2.1 基本概念

软件系统设计是指，在系统（分系统）分析和设计阶段，同步分析等至超前分析为达到系统（分系统）目标软件所应承担的任务、软件系统与所属系统的关系以及运行环境，设计软件系统的功能、性能、体系结构、配置项以及相互关系，提出软件系统的设计约束、质量保证、进度、测试、验收和交付等研制要求，从而确定软件系统的开发过程。

软件系统设计工作主要包括软件系统分析、设计、以及对分析、设计的早期验证，所以，本文所述的软件系统设计，可称为软件系统分析和设计。一般认为，软件系统设计具有以下主要特点：

（1）强调软件是所属系统的一个重要组成部分，软件将影响系统总体结构，其设计应当在工程的顶层方案设计阶段、系统要求分析和设计阶段同步进行；

（2）强调主动分析软件为达到系统总体目标所应承担的任务，而不是被动地接受系统分配给软件的任务；

（3）软件工程是系统工程的一个重要组成部分，应当与系统工程相互适应，软件工程计划需要在系统（分系统）分析和设计阶段同步制定；

（4）为了保障系统的顺利研制，应当采用有效的方法进行软件系统设计，验证所属系统对软件系统的全部关键性要求；

（5）在系统（分系统）分析和设计阶段，软件分析人员应当主动参加系统分析和设计任务，发挥积极的作用。

2.2 用途

在载人航天工程系统的研制中，软件系统设计

不仅是软件开发的重要依据，而且对整个系统生命周期产生重要影响，一般认为，软件系统设计具有(但不限于)以下主要用途：

(1)为软件系统研制提供依据和指南；

(2)为制定软件系统开发计划和预算提供依据；

(3)为软件系统研制过程中各方人员交流提供依据；

(4)为测试、分析、评估软件系统提供依据；

(5)为软件系统再设计打下基础；

(6)为软件系统重用提供依据和指南。

2.3 设计方法

目前，结构化方法是航天软件系统设计经常使用的方法，已在航天软件系统研制中取得许多成功经验。结构化分析方法使用数据流图（Data Flow Diagram，DFD）、控制流图（Control Flow Diagram，CFD）、状态图(State Diagram，SD)及数据字典(Data Dictionary，DD)，按照“由系统外部向系统内部、从系统顶层向系统下层”原则，逐层分解系统的信息流程和信息变换过程，逐层分解系统的行为，建立系统的逻辑模型。

图 1 为结构化方法用于航天软件系统设计的示意图，“载人航天系统数据流/控制流图”定义载人航天飞行任务由“飞行控制中心”和“航天器”组成，定义“飞行控制中心”向“航天器”发送的“飞行控制命令”，以及“航天器”向“飞行控制中心”发送的“运行状态报告”。

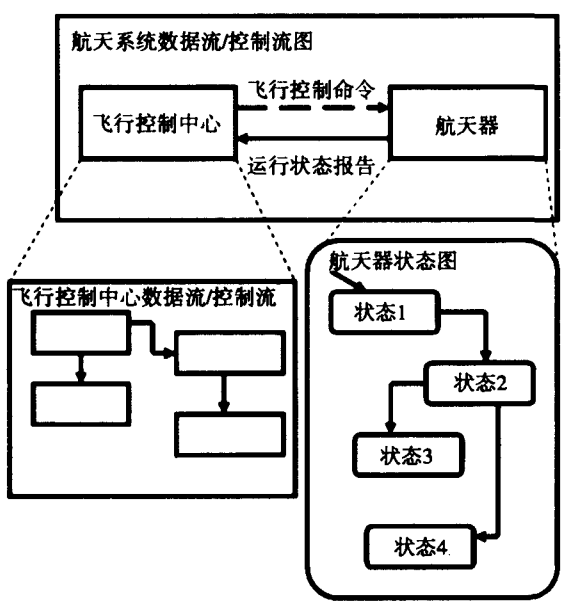

图 1 结构化方法用于航天软件系统设计示意图

“飞行控制中心数据流/控制流图”用于定义“飞行控制中心”的组成部件，是对“载人航天飞行任务数据流/控制流图”中“飞行控制中心”的分解和细化，图中示意其由四个部件组成。

“航天器状态图”用于定义“载人航天飞行任务数据流/控制流图”中“航天器”的状态及变化，是对“航天器”行为的说明，图中示意其由四个状态及其转移组成。

一些支持结构化方法的软件工具可辅助工程技术人员规范地建立一个软件系统模型，并对模型进行一定程度上的检查、仿真，以早期验证软件系统分析和设计结果，乃至根据模型自动生成源程序。但是，这些软件工具可能不适合以图形的方式描述大型复杂的软件系统，此类系统的模型将因其过于复杂，导致难于验证，应用领域专家难于阅读和评审。

结构化方法不适用于所有复杂软件系统，若一个系统(分系统)或其中一个部件不适用于结构化方法进行分析和设计，则可选择其它方法，例如，CSP (Communicating Sequential Processes)是一种形式化方法，较结构化方法易于描述和验证通信系统。

形式化方法是一种用于规范、设计和验证计算机系统的基于数学的方法，包括各种语言、技术和工具等，形式化方法已在许多领域得到成功应用，也积累了一些经验和教训，形式化方法并不能保证一个软件系统达到一个可靠性水平、保证一个软件系统是绝对安全的，但是若正确应用，则会有助于提高软件系统的可靠性和安全性，所以，应当根据实际需要决定是否使用形式化方法以及在何种程度上使用形式化方法，在应用形式化方法前，需要充分认识到它的局限性。

一般认为，不同应用和要求的软件系统（分系统）及配置项(或部件)，需要采用与之相适应的软件系统设计方法，方能得到预期的设计结果。选择软件系统设计方法应考虑的因素有方法的特定目标、方法的主要应用领域、应用方法的人员、应用方法将有何益处、应用方法必备的技能、方法的执行过程、方法如何描述软件系统、方法如何支持系统的可变性、是否有工具支持、是否得到实践工程的检验、如何验证方法输出结果等一系列问题。

航天任务不断发展变化，结构化方法、形式化方法未能满足所有航天软件系统设计的要求，直至 20 世纪 90 年代，美国国家航空航天局(NASA)完成一个航天任务需要花费多年时间，通常将系统分解为飞行控制软件 — 地面软件 — 测试、设计 — 测试 — 运行、工程 — 科学、空间 — 地面、导航 — 能源 — 推进 — 通信等子系统进行研制，缺少统一软件体系结构，不同航天任务软件系统之间也难以实现重用。

为了改善这种局面，NASA 开展了一系列有关软件系统设计的研究和应用，其中之一是开始于 1998 年由 NASA 喷气推进实验室(JPL)负责的任务数据系统(MDS)项目，以改进现有的航天软件系统设计方法、规范软件体系结构，提高航天器的自治能力和可靠性，更快、更好、更经济地研制航天任务软件，达到以下主要目标：

· 为不断变化的航天任务应用，建立一个高度可重用的核心软件体系结构；

· 加强载人航天系统工程和软件工程之间的相互促进作用；

· 建立一个可改进的开发生命周期，以适应更加可靠的航天任务软件；

· 减少软件研制费用和时间；

· 减少人员操作，增强载人航天系统的自治性。

MDS 是一个由状态分析方法、软件体系结构、基于构件体系结构组成的框架，其中，状态分析方法用于分析航天软件系统，建立 MDS 软件体系结构；软件体系结构为航天任务定义了飞行、地面、测试系统一体化的概念模型(如图 2 所示)；基于构件体系结构用于实现 MDS 软件体系结构，为软件系统的实现提供了有效途径。

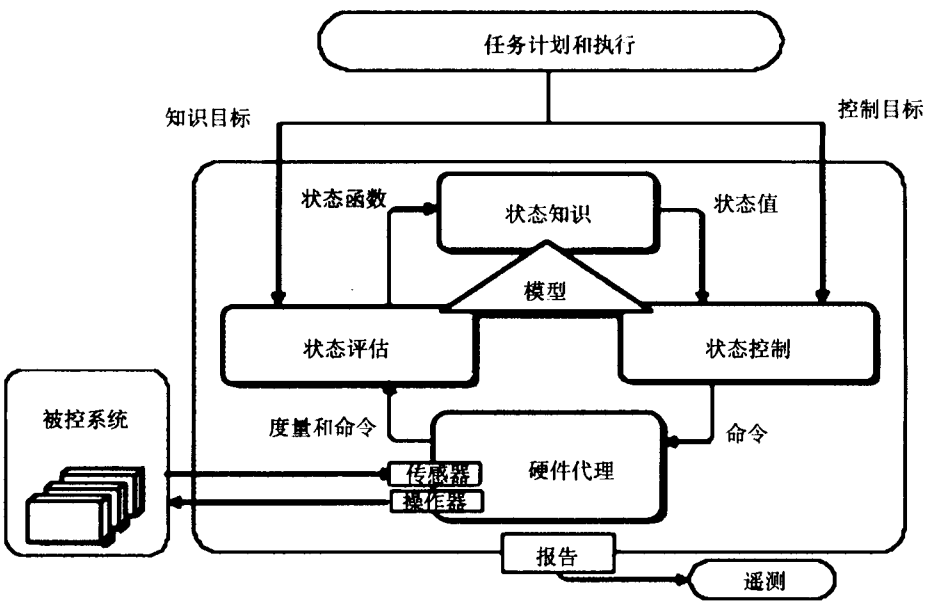

图 2　MDS 软件体系结构示意图

MDS 模型数据的应用和管理与传统方法不同，MDS 不将分析结果按照文档分为功能需求、失效模式和影响分析、数据字典、硬件和软件接口规格说明等进行管理和应用，而是按照状态分析方法过程将获取的所有信息存放在状态数据库，工程人员共享状态数据库，支持图形、文本等多种工具访问，生成相关的系统需求文件(例如：目标说明、度量和控制算法、软件构件接口等相关文档)，这是 MDS 框架称为任务数据系统的缘由(如图 3 所示)。

NASA 认为载人航天系统 MDS 软件体系结构具有许多特点，归纳为：目标指导操作、地面控制转移到航天器控制、闭环控制、实时资源管理、分别定义状态度量模型和状态控制模型、故障保护、承认不确定状态、区分数据管理和数据传送、导航与姿态控制合并等。

MDS 项目得到了美国国家科学基金、NASA 等机构连续多年的支持。Golden Gate 是由喷气推动试验室、卡内基梅隆大学、SUN 公司共同承担的一个

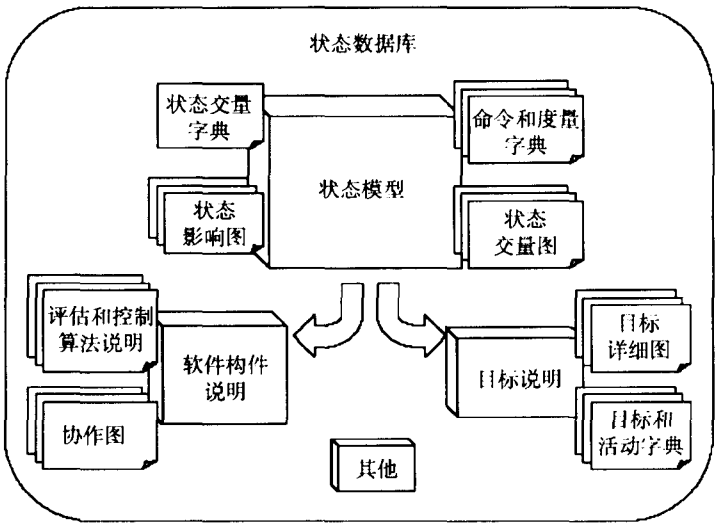

图 3　MDS 状态数据库示意图

MDS 试验工程，已按照 MDS 状态分析方法建立了 Rocky7(岩石 7)航天器的软件体系结构，将在火星科学实验室 (Mars Science Laboratory) 中得到验证，Golden Gate 的主要成果如下。

· Rocky7 MDS 软件体系结构

Rocky7 MDS 软件体系结构包括：构件和连接子、数据目录、历史值、数学库、物理库、资源库、状态知识、希望目标、目标网络、构件调度、硬件适配器、状态类型、实用程序等。

· Rocky7 MDS 软件体系结构的软件包

用 Real Time Java 实现的软件包包括 629 个驱动程序、发动机控制软件、发动机仿真软件、位置和方向控制软件，地面遥测软件、轮子控制软件等。

2.4 早期验证

早期验证是指在软件系统设计中对设计方案进行检查、试验和确认的过程。目前，人们还难以凭借经验清楚地推定大型复杂航天软件系统的行为模式和故障模式，推定一个完整的软件系统设计方案，所以，早期验证是软件系统设计的一个重要环节。

自然语言陈述是软件系统设计的一种表达方式，由于自然语言陈述没有严谨规范的语法和语义，所以，软件工具难于对其进行自动检查，也难于对其进行仿真测试。形式化模型是软件系统设计的一种表达方式，不管是采用何种方法进行设计，一般可用软件工具对其模型进行相应的检查与仿真测试，验证该设计是否达到了系统的要求。仿真模型是软件系统设计的一种表达方式，不管是采用何种方法进行设计，不管设计模型是否形式化，都可以通过系统仿真测试与分析，验证该设计是否达到了系统的要求。目前，软件工具可对有限状态图模型验证以下内容：

· 模型一致性完整性检查；
· 故障树分析；
· 操作错误分析；
· 模型仿真与测试；
· 非正常分析(Deviation Analysis)；
· 测试覆盖分析；
· 危险状态分析；
· 状态可达性分析；
· 不确定性分析；
· 资源竞争分析。

一般来讲，模型验证方式可分为以下两类：

(1)静态验证

静态验证不需要系统模型或目标原型执行，主要方法有模型静态语法检查方法、模型静态语义检查方法、失效模式及影响分析方法、故障树分析方法、初步危险分析方法等。

(2)动态验证

动态验证的主要特点是在系统模型或目标原型执行过程中，测试软件的功能和性能，主要方法有模型仿真测试方法等。

软件系统设计模型将会直接影响设计的可验证性，模型规模越大、关系越复杂，模型的可验证程度将会越低，设计方案可信程度也随之降低，可以认为，当系统模型大且复杂时，模型仿真测试将需要使

用更多的测试用例及测试资源(时间、人力、设备、经费),同样,当系统模型大且复杂时,模型检查(主要指符号模型检查、形式证明)也受到很大的影响,甚至会得不到检查结果。

模型仿真测试方法和模型静态语法检查方法是比较成熟的方法,模型静态语义检查方法尚未成熟,主要研究内容包括:形式证明自动化技术、大型复杂模型的符号模型检查算法、大型复杂模型抽象与复合的自动化技术,大型构件/体系结构形式化验证技术等。

3 软件系统验证

软件系统验证是试验验证阶段可采用的综合技术措施之一,软件系统验证主要用于验证软件系统(分系统)是否达到软件系统设计方案(软件系统设计说明)的要求,验证软件系统是否满足系统任务书的要求。软件系统验证主要包括软件系统测试和系统试验验证。

3.1 软件系统测试

载人航天软件系统一般由多个软件配置项组成,每个软件配置项通过测试,并不意味着整个软件系统可通过测试,所以,应按照软件系统设计方案及软件交互失效模式,针对软件系统与外部环境在动态交互过程中可能出现的各种问题及危险失效情况,进行充分的软件系统测试。

软件系统测试主要包括系统功能测试、系统性能测试、系统接口测试、系统余量测试,以及系统强度测试、系统可靠性测试、系统安全性测试、系统恢复性测试、系统边界测试、系统敏感性测试等。

载人航天软件系统中软件的种类多,可分为逻辑控制类软件、模型计算类软件、硬件控制类软件、信息处理类软件、板级驱动类软件、网络通讯类软件、人机界面类软件等,将不同类别的软件集成在一起进行软件系统测试,应充分研究和分析对测试环境的要求,若真实系统环境难以满足软件系统测试的要求,则可选择系统仿真环境。

以下系统仿真环境及其组合可构成软件系统测试环境。

(1)半实物仿真环境

半实物仿真环境通常由多配置项目标机(包括地面、箭载、船载、空间站载)、外部仿真环境、测试监测和记录设备组成,目标机运行被测软件,外部仿真环境用于模拟嵌入式系统外部环境信息的输入/输出。某运载火箭控制计算机的半实物仿真环境如图 4 所示[2]。

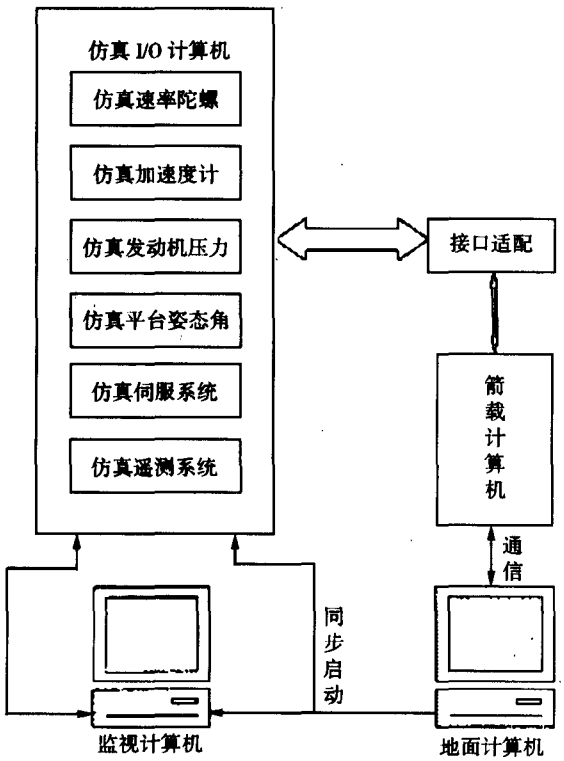

图 4　运载火箭控制计算机的半实物仿真环境

(2)全数字仿真环境

全数字仿真环境是指在宿主机上建立由多个配置项目标机仿真模型(包括地面、箭载、船载、空间站载)、外部环境仿真模型组成的软件系统运行环境,以实现对软件系统的运行与测试。全数字仿真环境可以是集中式的,也可以是分布式的,较好的灵活性、可控性是全数字仿真环境所具有的优势与特点。

由于大型复杂系统全数字仿真环境在真实性(如:仿真多重硬件中断 CPU、仿真各种外部环境状况)等方面与实际系统可能存在差异,这些问题在实际工作中应当给予注意。

3.2 系统试验验证

系统试验验证应在真实系统环境中进行,软件系统将与硬件系统一起进行系统验证。无论每一次系统试验验证是否重点针对软件系统,都将为验证软件系统提供宝贵的时机。

由系统总体人员主持的系统试验验证,除按照系统任务书的要求对系统进行试验验证外,还应从软件角度关注以下问题。

(1)软件系统设计要求的验证

试验验证内容应全面覆盖软件系统设计中的要求，并从软件角度对系统测试报告进行评审。

(2)飞行软件验证

对于飞行软件，应注重软件及计算机系统抗空间环境影响的验证。

(3)实时软件验证

对于实时软件，应注重软件系统时限、时序、流量、并发、强度、余量等性能的验证。

(4)嵌入式软件验证

对于嵌入式软件，应注重软件及计算机系统资源、余量、强度、实时等性能的验证。

(5)容错软件验证

对于容错软件，应注重软件容错设计的验证。

(6)软件最大处理能力的验证

验证软件及计算机系统在"崩溃"之前的最大处理能力，包括：外存容量、CPU 处理速度、I/O 响应时间、内存页面调度频率、内存利用率、可用内存、网络等。

(7)长期运行软件敏感性验证

对于长期运行软件，验证软件系统在长期运行中对事件序列、间歇性错误数据、内存泄漏等情况反应的敏感程度。

(8)针对性验证

根据在系统试验验证中出现的问题，有针对性地加强对特定软件要求及问题的验证。

4 软件安全性

安全相关软件是指在安全相关系统中用于实现或危及安全功能的软件。软件安全性是指软件具有的不导致事故发生的能力[3]。

4.1 安全性工作

软件安全性工作可以包括安全性分析与开发和安全性验证，并贯穿于软件生命周期。在安全相关软件研制过程各个阶段，应开展相应的软件安全性工作。

安全性分析与开发的主要任务是确定软件安全性需求、开展软件安全性设计、编程，其目的是全面、正确、合理地确定软件安全性需求，并通过一系列软件安全性设计、编程方法和准则，在软件设计和实现阶段实现软件安全性需求。

安全性验证的主要任务是通过采用相应的软件安全性分析方法，验证软件安全性需求的一致性、完整性和正确性，验证软件安全性设计和编码的一致性、完整性和正确性，验证软件安全性测试的充分性，最终目的是验证软件安全性需求是否通过软件安全性设计、编程已经全部正确实现。此外，通过软件安全性分析，发现软件安全性开发工作的不足，进一步改进和完善软件安全性开发工作。

4.2 技术应用

保证软件安全性即是一个新的理论问题，亦是一个复杂的工程问题，目前还不存在一种通用的技术和方法可以解决整个载人航天软件系统的安全性问题，因此，系统地运用各种软件安全性技术以及相关领域技术是保证软件安全性的关键，也是"技术标准"规范载人航天软件安全性工作的基本指导思想。在工程应用实践中需关注以下技术应用问题。

(1)系统地进行安全性设计

在软件系统设计中，注重分析与设计软件系统与外部运行环境之间的相互作用，注重分析与设计软件系统中各分系统之间的相互作用，尽量避免工程技术人员在软件系统设计中对外部信息处理及其相关时序的遗漏和曲解，只有将软件安全性与所在系统安全性一同设计和分析，软件安全性才有意义。

(2)异常运行环境的安全性设计

注重分析与设计软件在异常运行环境中的安全性能力，包括：硬件失效、操作人员错误、外部软件错误等。

(3)容错设计

注重软件硬件容错分析与设计，包括软件多版本冗余、多版本和重计算结合、多机单版本、多机多版本、恢复块结构、软件自动检错纠错等，并全面分析由于冗余导致系统复杂程度增加带来的新问题。

(4)安全性的验证途径

加强软件分析技术(包括形式化方法)和测试技术的综合应用，对于可在正常条件下运行的功能，则以实际测试为主，分析方法为辅；对于在异常条件下运行的功能，则可仿真环境中测试与分析方法并重；对小概率的软件危险失效则以分析方法为主，仿真环境中测试方法为辅，以减少测试充分性问题、测试高成本问题对安全相关软件验证的影响。

(5)设计与分析技术的选择

"技术标准"推荐了许多用于软件安全性设计和分析的技术与措施，并不意味着排除其它技术，由于影响软件系统安全性的因素很多，应当根据实际问题选用这些技术，或综合应用这些技术。

(6)一般与特殊安全性需求

软件一般安全性需求从系统安全的角度，规定了安全相关软件控制和执行的基本要求，规定了安全相关软件失效检测、隔离、恢复故障的基本要求，规定了安全相关软件与人员、外部硬件设备、外部软件之间的基本协同工作关系，安全相关软件应不违反这些基本要求。

软件一般安全性需求不包括载人航天系统对软件的特殊、具体的安全性要求，所以，软件一般安全性需求不是一份完整的安全性要求。仅实现软件一般安全性需求，不能全面保证软件系统安全性。

(7)安全性保证及证据

安全性保证是指提供方对软件消除和控制危险的措施的说明和对软件安全性质量特性的保证，包括软件的安全功能、软件安全功能的安全完整性等级、其它利于安全性的软件质量特性。

软件提供方应提出证据证明安全保证的可信性、合理性、充分性，安全性证据可分为过程证据、历史记录证据、测试证据、分析证据四类。

一般认为，不管采用何种技术、何种过程进行安全相关软件分析与开发，安全性证据均是软件安全性验证的依据，必须认真寻求和采集证据，以下证据将从技术角度更加有力地证明软件的安全性：

· 将风险降至可接受水平的安全性设计的可度量证据；

· 在软件系统测试中安全相关软件错误严重性和出现率的减少过程；

· 与实际应用相符合的安全相关软件测试的数据和用例；

· 第三方(独立)安全相关软件测试策划、测试设计和实现、测试执行和测试总结。

(8)可靠性和软件安全性

软件可靠性和软件安全性是安全相关软件系统的两种不同的重要属性，它们密切相关，但是，软件系统具有可靠性并不等于具有安全性，在有些情况下，安全性可降低可靠性，反之亦然。一般认为，提高软件可靠性和安全性有以下基本途径：

· 从软件可靠性要求、安全性要求、软件复杂程度和规模、开发周期和资源投人、开发工具和环境等诸多方面综合考虑，制定和优化软件安全性工作和可靠性工作的计划；

· 适度控制软件系统的复杂程度和规模，控制软件任务的安全关键性程度，减小软件系统中配置项之间(包括构件之间)接口关系的复杂程度；

· 在研制早期对较高安全级别的软件系统应用形式化方法，不但有助于对软件进行早期验证，评估软件的安全性和可靠性，也有助于不断促进软件开发过程的改进和完善。

5 标准的实施和维护

载人航天软件系统研制是一项复杂的系统工程，涉及多科学、多技术领域，需要众多单位协作，以“技术标准”为纽带，把各方面的工作有机地联系和组织起来，具有特别重要的意义。

在“技术标准”实施过程中，应积极开展技术培训、咨询和服务工作，并通过对相关文档、过程及其结果的检查，监督“技术标准”的执行情况。针对本文探讨的若干技术，建议加强以下方面工作。

第一，针对软件系统的特点、复杂程度，针对软件安全性、可靠性、质量要求，针对软件系统验证要求等多方面因素，选择有效的方法开展软件系统设计，为后续软件研制工作打下可靠的基础。

第二，由于缺乏在大型复杂的软件密集型系统研制中开展软件安全性工作的工程经验，可有目的的加强工程技术的试点和应用，并搜集软件安全性工作有关信息，为丰富和完善标准内容，提供更多的案例和经验。

第三，针对载人航天任务要求，研究和构建系统仿真环境，进行软件系统测试，以进一步加强软件系统测试工作。

第四，进行“技术标准”宣传和培训，加强工程技术人员对“技术标准”的理解，提高执行、维护和完善“技术标准”的意识和技术能力。

在“技术标准”实施过程中，注意搜集标准的贯彻执行情况，分析“技术标准”存在的问题，并结合技术发展和载人航天系统要求的发展变化，提出对“技术标准”改进和处理意见及建议。做好标准的维护和完善工作对保证“技术标准”的适用性和先进性，提高载人航天软件工程化水平，具有重要的意义。 ◇

参考文献

[1]胡世祥，张庆伟.中国载人航天工程 — 成功实践系统工程的典范.载人航天，2004，4

[2]于正伦.空间计算机嵌入软件的开发和测试环境的建立.小型微型计算机系统，2003，3，24

[3]中国人民解放军总装备部.军用软件安全性指南，GJB/Z 142-2004