中图分类号： TP311

论 文 编 号 ： 10006SY0814226

北京航空航天大學

硕 士 学 位 论 文

软件安全性需求分析方法研究与应用

作 者 姓 名 张一凡

学 科 专 业 控制科学与工程

指 导 教 师 鲍晓红 副教授

培 养 院 系 可靠性与系统工程学院

Research and Application of Software Safety Requirements Analysis Method

A Dissertation Submitted for the Degree of Master

Candidate：Zhang Yifan

Supervisor：Bao Xiaohong

School of Reliability & System Engineering

Beihang University, Beijing, China

中图分类号： **TP311**

论 文 编 号 ： **10006SY0814226**

硕 士 学 位 论 文

软件安全性需求分析方法研究与应用

作者姓名	张一凡	申请学位级别	工学硕士
指导教师姓名	鲍晓红	职　　称	副教授
学科专业	控制科学与工程	研究方向	软件安全性
学习时间自	2008 年 9 月 18 日起	至 2011 年 1 月 20 日止	
论文提交日期	2010 年 12 月 17 日	论文答辩日期	2010 年 12 月 28 日
学位授予单位	北京航空航天大学	学位授予日期	2011 年 1 月　日

关于学位论文的独创性声明

本人郑重声明:所呈交的论文是本人在指导教师的指导下独立进行研究工作所取得的成果，论文中有关资料和数据是实事求是的。尽我所知，除文中已经加以标注和致谢外，本论文不包含其他人已经发表或撰写的研究成果，也不包含本人或他人为获得北京航空航天大学或其它教育机构的学位或学历证书而使用过的材料。与我一同工作的同志对研究所做的任何贡献均已在论文中做出了明确的说明。

若有不实之处，本人愿意承担相关法律责任。

学位论文作者签名: yufan zhang　　日期：2010 年 12 月 17 日

学位论文使用授权书

本人完全同意北京航空航天大学有权使用本学位论文（包括但不限于其印刷版和电子版），使用方式包括但不限于：保留学位论文，按规定向国家有关部门（机构）送交学位论文，以学术交流为目的赠送和交换学位论文，允许学位论文被查阅、借阅和复印，将学位论文的全部或部分内容编入有关数据库进行检索，采用影印、缩印或其它复制手段保存学位论文。

保密学位论文在解密后的使用授权同上。

学位论文作者签名：yufan zhang　　日期：2010 年 12 月 17 日

指导教师签名：_____　　日期：　　年　　月　　日

摘　　要

随着计算机软件的应用范围不断扩大，其地位和重要性也逐渐提升和突出，尤其是在具有高可靠性、安全性要求的航空航天领域，如何保证软件的质量已经成为目前工作关注的难点和热点。需求阶段作为真正意义上软件工作的开端，与软件其他开发阶段在过程上的联系最为密切。而需求提取作为需求工程的基础性工作和重要组成部分，其质量直接影响并决定了软件设计的质量，进而影响并决定了软件代码质量，直至整个系统的最终质量。目前安全关键系统开发领域存在许多安全相关标准。尽管标准的数量众多，我们却很难从中找出一套成熟的方案来指导我们在软件需求阶段开展安全性相关工作。

本文研究的目标是提出一个有效并且可操作的，与现有软件工程过程紧密结合的软件安全性需求获取方法，软件开发机构可以应用此方法获取全面的软件安全性需求以用来为接下来的开发阶段服务。为解决这个问题，本文首先提出了软件安全性需求工作框架，明确了软件安全性需求分析工作在软件开发不同过程所应用的基本策略和具体工作环节。随后针对软件安全性需求分析工作的核心部分软件安全性需求获取，本文从软件安全性需求获取思路及获取方法两方面开展深入研究。在获取通用机载软件安全性需求方面，本文开发了通用软件安全性需求清单，并给出了明确的通用软件安全性需求裁剪步骤；在获取特定软件安全性需求方面，本文考虑从自顶向下的软件安全性需求分析及自底向上的软件安全性需求分析两方面进行，并分别给出了分析工作的具体思路及实施步骤。此外，针对软件安全性需求获取工作中应用到的重点分析方法，本文从方法的原理、目的、步骤等方面进行了详细介绍。

最后，本文将整套思路和方法应用于某型发动机控制系统控制软件，对其进行了详细建模及分析，并根据分析过程生成了通用及特定的软件安全性需求，从而验证了此方法的正确性及有效性。

关键词：需求获取、需求分析、需求工程、软件安全性、安全关键、安全性分析、机载系统

Abstract

With the scope of applying computer software expanding, its status and importance are gradually enhanced and prominent, especially in the aerospace field which has high reliability and safety requirement, how to ensure the quality of software has become the focus of the current works. Requirements phase, as the true sense of the beginning of the software development work, are most closely to other software development processes. And the requirements elicitation as the basis work and important part of the requirements engineering, its quality directly affects the quality of software design and then influence and determine the quality of software code, Until the final quality of the whole system. There are many safety-related standards existing for developing safety-critical systems. Despite the high number, we can hardly find a mature way to guide us to carry out safety-related work during the software requirements phase.

The objective of this research is to propose an effective and operable framework which combines with the existing software engineering process well, software organizations can generate and classify software safety requirements to guide their following development process. In order to solve this problem, at first this paper provides a framework for software safety requirements analysis work and identifies the basic strategy and specific work of software safety requirements analysis in different software development processes. Then for software safety requirements elicitation, the core part of software safety requirement analysis work, this paper makes in-depth study from two aspects: idea and methods of software safety requirements analysis. In the aspect of achieve generic aviation software safety requirements, this paper develops the list of generic software safety requirements and provides clear cutting steps of generic software safety requirements; in the aspect of achieve specific software safety requirements, this paper considers both software safety requirements flow-down analysis and software safety influence analysis and provides the specific ideas and implication steps. Besides, for the key analysis methods applied in the software safety requirements elicitation work, this paper describes in details from the aspects of principle, aim, steps and so on.

Finally, we apply this set of ideas and methods to engine control system control software, modeling and analysis in detail and generate generic and specific software safety requirement

based on the results of analysis, which verifies the correctness and validity of this method.

Key Words: Requirements Elicitation, Requirements Analysis, Requirement Engineering, Software safety, Safety-Critical, Safety analysis, Airworthiness

目　录

图目录

表目录

第一章 绪论

1.1. 论文选题背景及意义

1.1.1. 论文选题背景

自 1986 年 Nancy Leveson 在计算机科学领域引入“软件安全性”的概念并奠定这一研究领域的基础以后，软件安全性相关研究取得了很大进展，并且产生了很多富有挑战性的问题[1]。随着近年来计算机系统在航空、航天、核能、军事等领域广泛的应用于监视和控制复杂的实时物理过程和机械设备，软件在这些领域起到的作用越来越重要，软件的规模、复杂度及其在整个系统中的功能比重急剧上升。这也使软件的可靠性与安全性问题日益突出。

在军事、航空、航天、医疗等领域，核心控制软件的失效可能造成巨大的损失甚至威胁人的生命。 1985 年 6 月至 1987 年 1 月．Therac-25 治疗机发生 6 起超大剂量辐射事故．其中 3 起导致病人死亡。1991 年海湾战争．爱国者导弹在拦截飞毛腿导弹中几次拦截失败．其直接原因为软件系统未能及时消除计时累计误差。1996 年阿里亚娜 5 型运载火箭由于控制软件数据转换溢出起飞 40 秒后爆炸。造成经济损失达 5 亿美元；1999 年大力神 4B 运载火箭由于软件问题飞行 9 秒后偏离航向。造成卫星未进入预定轨道。不断发生的软件失效和事故使人们逐渐认识到：在系统复杂性较高的情况下，常规的软件工程方法和软件评测手段并不能解决软件可靠性与安全性设计深层次的问题。软件系统的安全性问题通过专业技术予以保证。

而在对于众多事故的调查当中研究人员发现，系统开发过程中软件需求的缺失、歧义是主要的错误来源，大部分引入软件的问题都可以直接追踪到软件需求上。同样，Leveson 的研究表明在空间工程领域，相当数量的软件相关故障都与有缺陷的需求或错误理解软件目标有关[2][3]。因此，软件安全性需求工作是整个软件安全性工作的重要一环，通过改进传统需求工程活动来确保软件安全并成功的执行系统分配的任务是十分必要的。

作为软件工程子领域的需求工程在 80 年代中期逐步形成。需求工程是发现、记录和管理计算机系统需求的过程。其目标是尽可能产生一组完整的、一致的、相关的能够反映出客户真实需求的系统需求。其中需求获取被认为是软件开发中最为关键的知识密集型活动[4]，在传统的软件开发过程中扮演了重要角色[5]。随着软件系统规模的扩大，

需求获取与定义在整个软件生命周期中的作用越来越重要，甚至直接关系到软件开发的成功与否。人们逐渐认识到需求获取活动不再仅限于软件开发的最初阶段，它贯穿于系统开发的整个生命周期。好的需求获取活动可以避免分析人员在确定和理解功能和接口需求过程中产生错误，而这些错误经常与安全相关的软件错误有关。因此软件安全性需求获取的结果是整个软件系统开发的基础，关系到工程的成败和软件产品的质量。

1.1.2. 论文选题意义

目前我国航空工作发展正处于关键时期，许多方面均取得了很大进步，但在先进作战飞机、航空发动机和大型飞机研制及其基础科学研究等方面，与世界先进水平相比还存在很大差距。特别在航空机载软件的开发方面，大型飞机的研制目标对软件安全性工作挑出了更高的要求，然而在实际工作中软件安全性工作的开展面临着很大的问题。

国内航空机载软件开发部门目前主要面临着缺少具体有效的技术方法、实践环节缺少指南和方法实践不足三个方面的问题。

首先在软件安全性需求获取工作的具体技术方法上，目前国内对于软件安全性需求获取技术的研究还处于起步阶段，很多软件安全性分析方法在实际应用上还不成熟，而国外应用多年的技术方法又缺少细致的描述，使得航空机载软件开发单位对于软件安全性需求获取技术方法的掌握上存在着巨大的障碍。

其次在实践环节的指导方面，目前软件安全性标准的应用是航空机载软件的开发过程的重要一环，国内外相应安全性标准对于在软件开发过程各阶段开展软件安全性分析工作均进行了定义。然而在实际使用过程中，各标准指导范围的局限性以及标准本身对于使用方法与技术的描述程度使得我们无法依靠单一标准完成软件安全性需求获取工作，同时，各组织之间标准定义的差异，描述方式的不同以及标准背景的不一致使得各标准在相互结合使用的过程中存在着很大的障碍，这些问题的存在使得在实际项目中开展软件安全性需求获取工作遇到了很大的困难。

此外，软件安全性工作的开展在国内尚处于起步阶段，目前许多技术方法仅在航天领域得到了应用。在航空机载软件开发过程中系统、综合的应用软件安全性分析技术所进行的实践很少，这也在很大程度上限制了软件安全性需求获取工作的开展。

最后，我国机载软件开发水平相对国外落后的现状也决定了我们在机载软件开发过程中开展软件安全性工作时不能一味的照搬国外经验与技术。面对当前国内机载系统开发领域对于开展软件安全性工作的迫切要求，我们应该从自身软件工程开展的实际情况

出发，提炼软件安全性需求获取工作的核心内容，结合国内外相关标准与文献，尽快形成一套软件安全性需求获取过程的流程规范，并给出技术或方法的具体指导和实例，对机载软件安全性需求获取工作提供帮助

1.2. 国内外研究现状

人们对于软件安全性的日益关注，使软件安全性相关工作在实践中得到了飞速发展。目前对于软件安全性领域的研究在安全性标准的制订及推广、安全性相关技术研究和应用等多个方面取得了一系列成果，总结起来可以分为以下几个方面：

1.2.1. 软件安全性相应标准和规范的发展与应用

随着航空领域机载系统的复杂程度不断增加，软硬件密切耦合，软件的规模、复杂度及其在整个系统中的功能比重急剧上升，越来越多由硬件实现的功能现在转由软件完成。航空领域机载系统中所包含安全关键软件的数量不断增加。作为开发安全关键软件的一个部分，软件安全性标准的应用被认为是任何安全性项目中的重要元素[6]。目前随着软件安全性实践的发展，陆续有相应的标准和规范制定出来，以满足日益增长的军用与民用需要。根据 FAA 的一项研究显示，截止到 1991 年底全世界共有 146 个不同的与软件安全性相关的标准[7]。这些软件安全性标准代表了软件安全性领域“最佳实践”的总结，各国政府、国际性组织或者相关公司制定了多项关于软件安全性的标准和指南，如 NASA 很早就重视系统安全性及软件安全性，制定了多项标准及配套的技术指南，对于航天领域系统安全性及软件安全性工作做了全面的阐述。这些标准与指南经过多年的实践应用，不断的升级完善，这充分说明了 NASA 对于软件安全在飞行任务中的重要性的认识。同时，欧洲太空部署、美国军方等机构也在积极推出软件安全性相关标准与指南，而这其中的很多标准正在广泛的应用与军用或民用领域，对于实现软件安全性目标起到了十分重要的作用。

1.2.1.1 软件安全性标准和规范总揽

尽管标准的数量还在不断增长，但大多数的标准均采纳了相似的途径来解决安全性问题：通过建议或描述一系列的开发过程和技术方法，使软件对于系统风险的贡献降低到一个可接受的程度。这些建议通常分解进一系列不同的严酷度水平中，最高等级的水平对应着最严重的失效后果。当经过从系统级到软件级的一系列迭代分析之后，目标软

件被分配到一个特定水平，合适的设计和保证活动随后实施以保证软件获得需要的完整性。软件安全完整性指软件在规定条件和规定时间中，在系统中成功完成其安全功能的可能性的度量[8]。根据这个方法，安全性软件由于具有较高的完整性水平，因此需要在开发过程中应用较多的设计和保证活动，达到更多的目标要求。因此安全性软件的开发需要更高的成本，这些成本包括资源、方法及技术等。

经总结，目前工程领域一些比较重要的软件安全性标准如下表 1 所示：

表 1　各国主要安全性标准或指南

序号	年份	标准代号	标准名称
1	1969 年	MIL-STD-882	系统安全性大纲要求
2	1977 年	MIL-STD-882A	系统安全性大纲要求
3	1984 年	MIL-STD-882B	系统安全性大纲要求
4	1993 年	MIL-STD-882C	系统安全性大纲要求
5	2000 年	MIL-STD-882D	系统安全性大纲要求
6	2005 年	MIL-STD-882E	系统安全性大纲要求
7	2005 年	MIL-HDBK-516B	适航合格审定规范
8	1999 年	Joint Software System Safety Committee	软件系统安全性手册
9	1996 年	NASA-STD-8719.13A	软件安全性
10	2004 年	NASA-STD-8719.13B	软件安全性
11	1996 年	NASA-GB-1740.13	软件安全性指南
12	2004 年	NASA-GB-8719.13	软件安全性指南
13	2001 年	EN 50128	铁路应用－通信、信号和处理系统－信号的安全相关电子处理系统
14	1997 年	DEF Stan 00-55	防御设备安全相关软件要求
15	2007 年	DEF Stan 00-56	防御系统的安全性管理要求
16	1999 年	IEC61508	功能安全国际标准及安全性分析
17	1994 年	IEEE 1228	软件安全性计划
18	1992 年	RTCA DO-178B	机载系统和设备合格审定中的软件考虑
19	1996 年	ARP4754	高度整合或复杂航空器系统合格审定考虑
20	1996 年	ARP4761	民用航空系统和设备中开展安全性评估的指南和方法
21	1990 年	GJB 900-1990	系统安全性通用大纲
22	1997 年	GJB/Z 99-1997	系统安全工程手册
23	1997 年	GJB/Z102-9	软件可靠性和安全性设计准则
24	2004 年	GJB/Z 142-2004	军用软件安全性分析指南

软件安全性标准一般并不定义“单一”的软件开发过程，而是使用不同的推荐过程和技术来得到不同等级的安全性。大部分标准基于安全完整性等级（SIL），如 IEC, MOD, ADoD 等， 然而面向民用航空领域的标准倾向于依照开发保证水平(DAL)，如 RTCA。

安全完整性等级（SIL）是规定安全相关软件安全功能的安全完整性要求的基础[9]，IEC 61508 “功能安全国际标准及安全性分析”[10]确定了 4 个安全完整性等级（SIL），作为安全功能的性能指标。SIL1 最低，SIL4 最高。SIL 以失效概率的形式反应了系统的安全关键性，关键系统应的低失效概率，开发者应根据整个系统的需求为软件选择 SIL 级别。

IEC 61508 对于 SIL 的定义如下表 2 所示：

表 2　IEC 61508 中 SIL 定义

	安全关键系统的失效目标测试	
SIL 级别	低频率运行方式 每次请求失效概率	连续运行模式 每小时失效概率
4	$>=10^{-5}$ to $<10^{-4}$	$>=10^{-9}$ to $<10^{-8}$
3	$>=10^{-4}$ to $<10^{-3}$	$>=10^{-8}$ to $<10^{-7}$
2	$>=10^{-3}$ to $<10^{-2}$	$>=10^{-7}$ to $<10^{-6}$
1	$>=10^{-2}$ to $<10^{-1}$	$>=10^{-6}$ to $<10^{-5}$

安全完整性等级虽然为安全关键软件的分类提供了依据，但是在实际应用过程中，软件开发人员住住很难分析出软件的失效概率。

开发保证水平（DAL）类似于安全完整性等级（SIL）。ARP4754[11]和 ARP4761[12]所标识的每个功能失效条件都根据功能危险评估中所标识的失效条件的影响的严重性被指定一个 DAL。提供的 DALs 带有等效的数值的失效率，因此可以进行定量的风险评估。可是，目前大部分标准公认特定设计策略的有效性并不总能量化，因此常常需要定性的判断，特别是从不试图用概率术语来解释软件的保证等级。

RTCA DO-178B “机载系统和设备合格审定中的软件考虑”[13]对于开发保证水平（DAL）的定义如下表 3 所示：

表 3 RTCA DO-178B 示例 DAL

软件等级	失效状态类别	描述
A 级	灾难性的	其异常状态会导致或促使系统功能失效，从而引起航空器灾难性失效状态的软件
B 级	危险的/严重的	其异常状态会导致或促使系统功能失效，从而引起航空器危险的/严重的失效状态的软件
C 级	重大的	其异常状态会导致或促使系统功能失效，从而引起航空器重大的失效状态的软件
D 级	轻微的	其异常状态会导致或促使系统功能失效，从而引起航空器轻微的失效状态的软件
E 级	无影响的	其异常状态会导致或促使系统功能失效，但不会影响航空器的运行能力或驾驶员工作量的软件。

软件系统开发保证等级的确定需要在系统开发的早期进行，这也严重的影响了整个软件寿命周期过程中需要审定的文档的类型和数量（A 级系统需要 66 个“目标”而 D 级系统仅需要 28 个“目标”）。合适的等级由系统安全性评估过程决定。一般的，绝大多数的飞行器系统需要至少被确定为 D 级或更高来保证飞行安全。

1.2.1.2 国内外主要标准中软件安全性需求获取工作概述

软件安全性需求获取既包括建立软件的安全性需求，又包括在安全性方面对形成中的软件需求进行评价。考虑到目前国内软件安全性现状及软件工程开展的实际情况，我们选择以下几部国内外标准作为重点参考对象，标准对于软件安全性需求获取活动简介如下所示：

1.2.1.2.1 GJB/Z 142-2004 军用软件安全性分析指南

本指导性技术文件给出了在软件生存周期中实施软件安全性分析的指南，适用于安全相关软件的获取、供应、开发、运行和维护。此标准给出了软件的安全性分析的一般方法，并以此满足不同复杂程度软件的需要。在具体应用过程中可对本指导性技术文件进行剪裁。

GJB/Z 142 军用软件安全性分析指南主要应用安全完整性级别作为剪裁的主要依据，对于高安全完整性级别的软件应依本指导性技术文件进行充分的安全性分析。而低安全完整性级别的软件，在得到需方认可的情况下，可省略或合并部分安全性分析任务。标准所描述的软件寿命周期安全性工作如下图 1 所示：

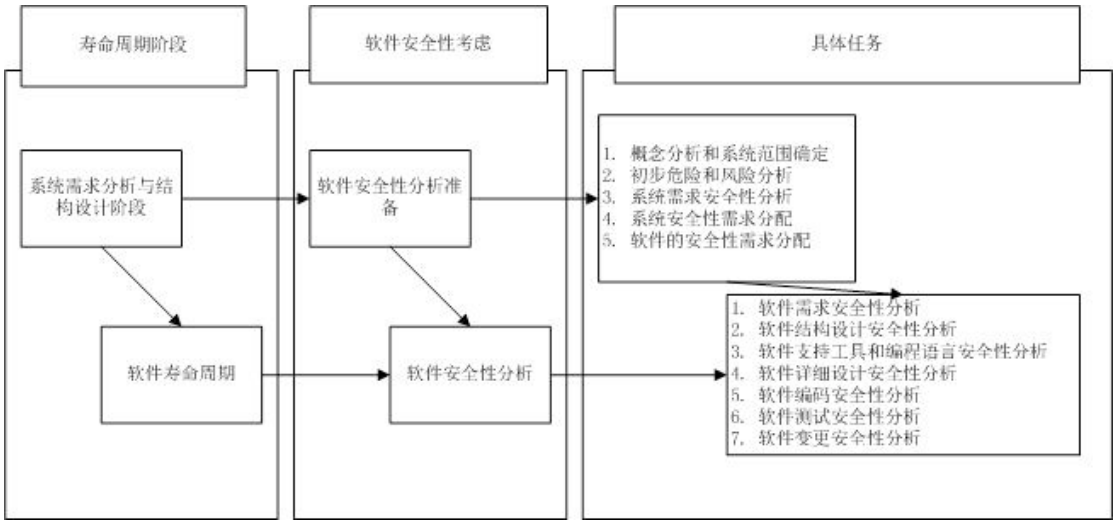

图 1　GJB/Z 142 软件安全性分析工作流程

在 GJB/Z 142 军用软件安全性分析指南中，软件安全性需求获取主要包括下列任务：

1) 系统安全性需求映射

2) 安全相关性分析

3) 软件需求的安全性评价

GJB/Z142 中定义软件需求安全性分析既包括建立软件的安全性需求，又包括在安全性方面对形成中的软件需求进行评价。以下针对具体的任务给予详细的说明：

1. 系统安全性需求映射

在此活动中应分析分配给软件的系统安全性需求，并以将之转换为软件的安全性需求。系统安全性需求映射包括下列工作项目：

1) 检查软件安全性分析准备中产生的信息以确保充分满足要求。应特别考虑整体系统需求安全性分析和整体安全性需求分配

2) 为确保软件安全性，如果没有在系统的安全性需求中说明受控设备的所有相关运行模式，应在软件安全性需求中进行说明。

3) 软件安全性需求应规定以下内容：

1) 使受控设备获得或维护安全状态的功能

2) 与检测、通告和管理控制部件中硬件错误有关的功能

3) 与检测、通告和管理传感器和执行器错误有关的功能

4) 与检测、通告和管理软件本身中的错误有关的功能

5) 与定期测试在线安全功能有关的功能

6) 与定期测试离线安全功能有关的功能

7) 与安全地修改可编程电子系统有关的功能

8) 安全功能与非安全相关功能之间的界面

9) 容量和响应时间等性能

10) 软件与可编程电子系统之间的接口

4) 软件安全性需求描述应阐明所有与安全相关的软件和硬件之间的约束条件。软件安全性需求描述中需考虑下列与硬件结构设计有关的内容：

1) 软件自检

2) 监控可编程电子硬件、传感器和执行器

3) 在系统运行时定期对安全功能进行检测

4) 当可编程电子系统在操作时，能够对安全功能进行测试

5) 分析拟定的软件的安全性需求，如果发现存在不可行的系统需求、接口需求和不适宜软件实现的安全功能，可以考虑进行系统安全性需求的重新分配，以便借助软件以外的风险降低手段（即外部安全设施或系统内其它的安全性技术手段），实现系统总的安全性需求。

2. 安全相关性分析

安全相关性分析的目的是分析所有的软件需求及其之间的关系，确定和安全相关的软件需求，并明确地标明它们。这些软件安全性需求可以是自顶向下即从系统需求传递下来的，也可以是自底向上分析推导出来的。安全相关性分析包括下列工作项目：

1) 将系统安全性需求直接映射的软件需求标识为软件安全性需求

2) 进行自底向上的分析，识别与系统需求不一致，或系统需求所未阐述的安全性需求。自底向上的分析还可能揭示非期望的达到危险和不安全状态的路径（例如潜通

路）。必要时应通过增加或修改系统需求加以解决所识别出的潜在危险，并在将它们再传递给软件需求

3) 应分析系统分配给软件的外部设计约束，包括时序、吞吐量和规模等，确定它们是否是安全相关的

4) 当要求安全相关软件执行非安全相关功能时，软件安全性需求应清楚地标明这些功能

5) 以安全完整性级别的形式规定软件安全功能的安全完整性要求，并且标识出目标失效测度的运行模式。

3. 软件需求的安全性评价

应对形成中的软件需求在安全性方面进行评价，评价的工作项目包括：

1) 分析软件安全性需求是否能溯源到软件安全性需求获取的输入，也就是系统安全性需求和安全性需求分配，显示对于系统需求的依从性

2) 分析软件需求规格说明，对应每种识别出的危险，保证已经完整地规定了分配给软件的系统安全性需求，并且已经转化为恰当的软件需求

3) 分析软件安全性需求与系统安全性需求、软件安全性需求和其它软件需求的外部一致性，以及软件安全性需求的内部一致性

4) 根据安全完整性级别的要求，软件安全性需求的规定要求应这样表示和构成：

1) 清楚、一致、准确、可验证并与安全完整性级别相当

2) 不使用含糊的或文档受众所不理解的术语和描述

5) 应分析安全性需求的可测试性，提出对后续测试需求的建议

6) 应分析软件安全性需求设计和实现的可行性，确定软件安全性需求的规定是否足够细致以便软件的设计和实现达到要求的安全完整性，进而对软件设计提出建议

7) 应提出与安全性相关的必要、提倡、不提倡和禁止使用的软件设计、编码和测试技术列表

GJB/Z 142 军用软件安全性分析指南中给出了一些可用于软件安全性需求获取的主要技术手段，如检查单和交叉参照、层次分析、控制流分析、信息流分析、功能模拟、

约束分析等，并给出了各种主要技术手段的简单说明。

总结看来，GJB/Z 142 军用软件安全性分析指南虽然给出了软件安全性需求获取活动的主要任务，并给出了相应的任务说明和主要技术手段，但仍然存在如下不足之处：

1) 标准提出的软件安全性分析任务分配基于 V 型软件开发过程，在使用范围上受限

2) 标准给出的主要技术手段只有简单介绍，没有具体实施步骤及案例说明，无法满足实际分析过程的需要

3) 标准并没有给出一个系统的开展软件安全性分析工作的流程，各部分任务说明相对孤立，过程之间交互体现不足

4) 标准主要应用安全完整性级别作为剪裁的主要依据，但对于软件安全完整性级别的确定并没有明确给出方法，只提出了一般遵循的原则。

1.2.1.2.2 RTCA DO-178B 机载系统和设备合格审定中的软件考虑

在七十年代末期与八十年代前期，个人电脑的增长与规模经济的作用，计算机系统的成本不断下降。同时系统性能的大幅度增长使得机裁设备开发商大量的使用软件来取代或增强现存的机载硬件设备功能。在安全关键应用中软件和计算机系统应用的增长促使了最初的 DO-178B 标准的出现[14]。1982 年，RTCA 和 EUROCAE 正式发布了 DO-178B，这是民用航空机载软件开发中安全保证的一个里程碑。这个标准在美国(RTCA)被称为 DO-178，在欧洲（EUROCAE）被称为 ED-12。在接下来的几年中，依据 DO-178 进行开发和论证的经验表明，DO-178 还不太完善，需要修订。1985 年，新的版本 DO-178A 问世，与之等价的欧洲版本为 ED-12A。

DO-178A 标准有一个很大的特点，它是面向软件开发技术与开发方法。但是，软件的开发技术更新很快，新的技术和方法层出不穷，日新月异。这样，DO-178A 很快就显得跟不上步伐。为了解决这个问题，RTCA 决定再次修订标准。为了避免重蹈覆辙，RTCA 和 EUROCAE 的专家们改变了制订标准的原则，从原来的“面向开发技术和方法”改成“面向目标”和“面向过程”。这样一个相对稳定的版本 DO-178B 于 1992 年问世，至今还在应用中。

DO-178B 主要应用软件开发保证水平（DAL）作为软件安全性目标分配的主要依据，越高开发保证水平的软件需要满足的“目标”越多。DO-178B 标准中软件需求过程主

要任务如下图 2 所示：

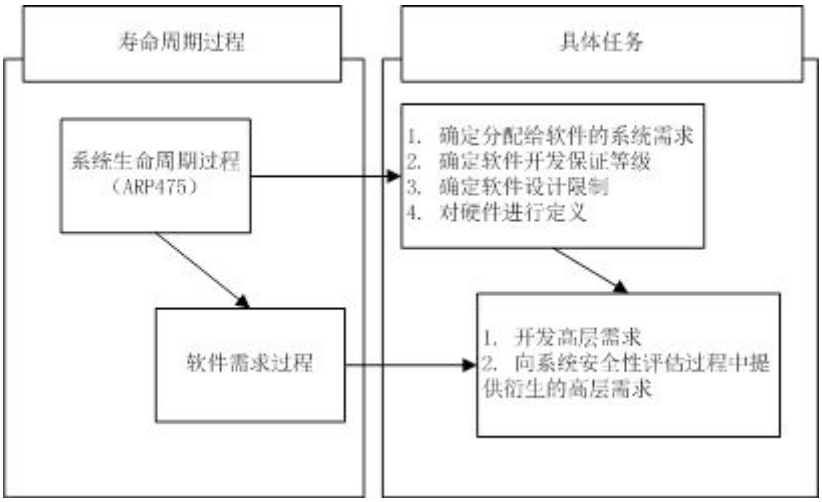

图 2　RTCA DO-178B 软件需求过程主要任务

在 DO-178B 标准中，软件需求过程被定义为使用系统生命周期过程的输出来开发软件的高层需求。这些高层需求包括功能需求、性能需求、接口需求和安全性需求。软件需求过程的目标包括两个方面：

1) 开发高层需求

2) 向系统安全性评估过程中提供衍生的高层需求

软件需求过程中输入包括来自系统生命周期过程的系统需求、硬件接口和系统架构（如果需求中未包括），以及来自软件计划过程的软件开发计划和软件需求标准。当制订的转换准则满足时，使用这些输入来开发软件的高层需求。该过程的主要输出为软件需求资料。当软件需求过程的目标和相应整体过程的目标满足时，软件需求过程就完成了。

DO-178B 标准给出了软件需求过程的指南，包括如下几点：

1) 对不明确的、不一致的和未定义的状态，要分析分配给软件的系统功能和接口要求

2) 要报告软件需求过程检测到的不合适的或不正确的输入，并反馈到输入的源过程以澄清或纠正

3) 要在高级需求中规定分配给软件的每一个系统需求

4) 表明分配给软件以减小系统危害性的系统需求的高级需求要加以定义

5) 高级需求要符合软件需求标准，并且是可验证的和一致的

6) 若适用，高级需求要用容差的定量术语来说明

7) 除了规定的和合理的设计限制外，高级需求不应详细描述设计和验证细节

8) 分配给软件的每一个系统需求要追溯到一个或多个软件高级需求

9) 除派生的需求外，每一个高级需求要追溯到一个或多个系统需求

10) 要为系统安全性评估过程提供派生的高级需求

标准中虽然给出了软件需求阶段的工作指南和相应目标，但仍然存在以下不足：

1) 标准中所规定所有任务均是在获得完备的系统安全性需求前提下进行，而标准对于系统阶段工作只有简单描述

2) 标准面向目标，只给出了软件需求阶段工作应该达到的要求，而没有给出具体的任务

1.2.1.2.3 NASA 软件安全性指南（Software Safety Guidebook）

NASA 软件安全性指南[15]关注安全关键软件，包括固件（例：不可擦写内存中的软件，如 ROM,EPROM,EEPROM 或闪存等）和可编程逻辑的分析、开发和保证工作。这分文档同时考虑了承包商开发软件。文档提供了在整个软件开发、管理、风险控制和保证过程中怎样解决安全关键软件的开发和安全保证。基于可得到的信息的数量，通过这个指南我们可以得到在不同的等级中需要的不同技术与分析方式。此指南不仅是一个开发技术和分析方法的收集。此指南的目标是鼓励开发者从安全的角度来思考软件问题，帮助安全关键软件开发和保证的组织。软件开发者个人可以从指南中得到对于不同技术和分析方法的介绍等。

NASA 软件安全性指南基于风险控制的思想，通过确定软件安全性努力程度来为不同努力程度的软件分配不同任务，以最终满足相应的安全性目标。在软件需求阶段 NASA 软件安全性手册提出的主要任务如下图 3 所示：

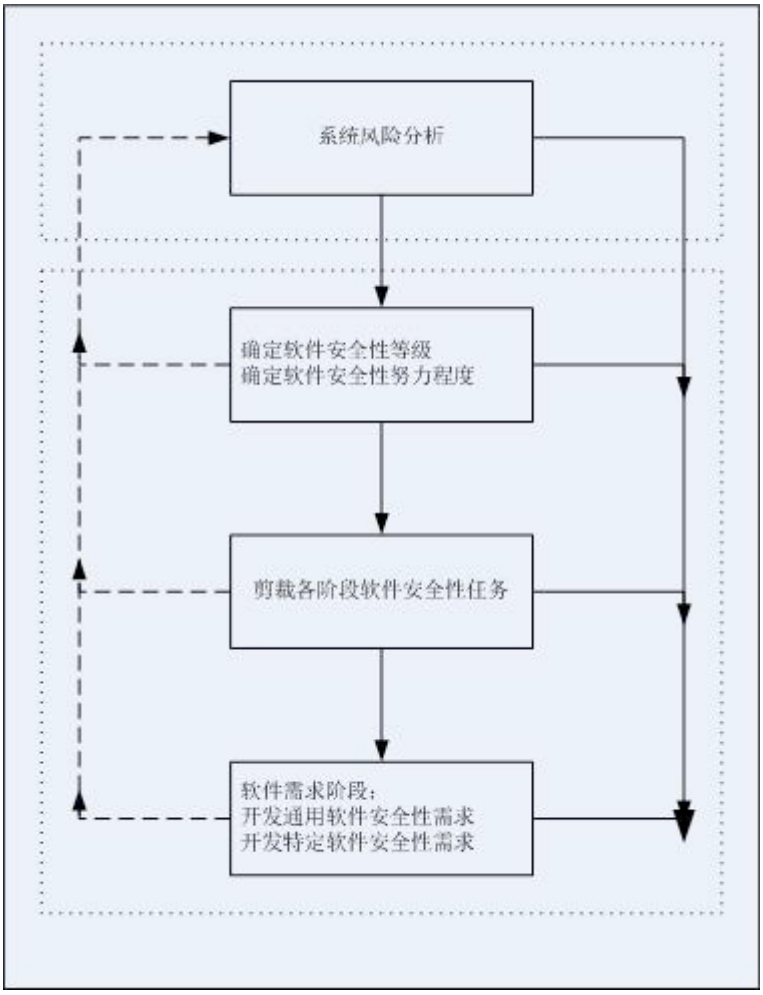

图 3　NASA 软件安全性指南软件安全性需求获取方法

1. 开发通用软件安全性需求

通过剪裁现有通用需求，参考现有标准等方法，主要参考以下标准：

1) NSTS 19943, Command Requirements and Guidelines for NSTS Customers.

2) STANAG 4404 (Draft), NATO Standardization Agreement (STANAG) Safety Design Requirements and Guidelines for Munition Related Safety-Critical Computing Systems.

3) EWRR 127-1, Range Safety Requirements - Western Space and Missile Center, Attachment-3, Software System Design Requirements. See Section 3.16 Safety-Critical Computing System Software Design Requirements.

4) AFISC SSH 1-1, System Safety Handbook - Software System Safety, Headquarters Air Force Inspection and Safety Center.

5) EIA Bulletin SEB6, A System Safety Engineering in Software Development

(Electrical Industries Association).

6) Underwriters Laboratory - UL 1998, Standard for Safety - Safety-Related Software, January 4th, 1994.

7) NUREG/CR-6263 MTR 94W0000114, High Integrity Software for Nuclear Power Plants, The MITRE Corporation, for the U.S. Nuclear Regulatory Commission.

2. 开发特定软件安全性需求

要全面的确定所有的特定软件安全性需求，需要做以下全部三个方法：

1) 自顶向下分析系统设计需求和规格说明。

系统需求可能在最开始识别出了系统危险，具体说明了系统的哪个功能是安全关键的，或者一个故障树分析可能完全的识别出安全关键功能。软件安全性团队把这些需求映射到软件

2) 从初步危险分析（PHA）得到：

PHA从系统危险的角度深入系统，初步危险的原因被映射或与软件相互作用。软件危险控制特点被识别，描述成需求。

3) 通过自下向上的设计数据分析（如流图、FMECA）

NASA 软件安全性指南在软件安全性需求开发部分给出了剪裁通用软件安全性需求可参考的相关文献，并且提出应该在故障和失效容错、危险命令、时间、范围和吞吐量等几个方面对目标软件进行深入考虑以开发针对具体软件的特定软件安全性需求，同时推荐使用 FTA、FMECA 等分析方法，但仍然存在一些缺点如下：

1) 手册中给出的大部分技术方法缺少详细描述

2) 手册中给出的各阶段剪裁任务数量众多，与我国实际软件开发水平存在差距，完全依照手册中任务执行困难

3) 手册只是指出了每个过程中任务的注意方面，而没有说明具体的工作流程。

1.2.1.2.4 软件系统安全性手册

软件系统安全性手册[16]的目标是提供管理和工程的指南，以便能在一个合理的水平上保证软件将以可接受的安全性风险水平在系统环境中执行。此手册是一项联合的工作成果，美国陆军、海军、空军和海岸防卫队安全中心，与联邦航空局（FAA）、国家航空航天局（NASA）、国防工业承包商、学术界合作，广泛的采纳了与软件系统安全性

项目管理及安全关键软件设计有关的“最佳实践”。手册将这些贡献合并统一为一个单一的用户有好的资源。它帮助系统开发团队理解他们的软件系统安全性职责。

软件系统安全性手册提出一种技术性和管理性的团队方法解决软件开发过程中的安全性相关问题。在软件开发过程的需求获取方面主要工作如下表 4 所示：

表 4　软件需求获取阶段安全性相关工作

软件开发过程主要活动	安全性相关工作
系统要求分析	软件安全性策划
	软件安全性项目管理
	初始安全性风险评估
软件需求获取	初步危险清单开发
	通用安全关键要求的剪裁
	初步危险分析（PHA）
	推导系统的安全关键软件需求

在软件安全性需求获取阶段，手册将主要工作分为剪裁通用安全关键软件需求清单与推导特定安全性的软件需求两大部分，各部分的主要工作如下所示：

1. 剪裁通用安全关键软件需求清单

对于通用安全关键软件需求清单的剪裁部分，手册给出了一个可供考虑的通用安全性要求清单，包括了 STANAG 4404,NATO 标准化协议，安全关键计算机系统相关的军火安全性设计要求和指南，Mitre(Ada)清单，以及其它语言专用要求。

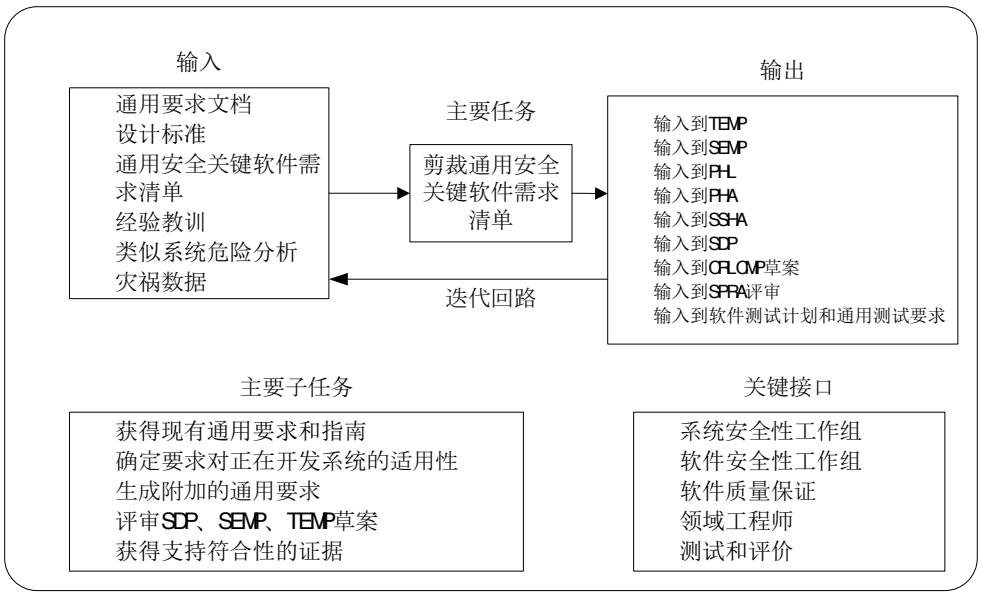

图 4　通用安全性要求的剪裁

2. 推导系统的安全关键软件需求

系统特定软件需求的标识是全部危险分析方法学的直接结果。特定软件安全性需求要从四个来源进行推导：包含通用清单、系统功能性分析（安全性设计要求）、原因因素分析和危险控制的实现四个方面。

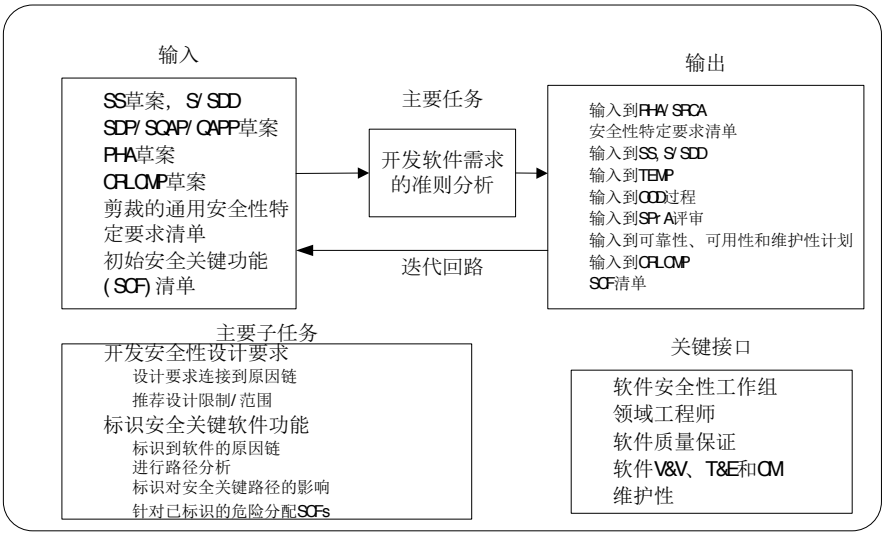

图 5　推导特定安全性的软件需求

但手册旨在帮助系统开发团队理解他们的软件系统安全性职责，强调各学科整合对于保证软件系统安全性的重要性，对于具体的技术细节并没有过多关注。

总结目前多数软件安全性标准可以得出，现在对于软件需求阶段的工作均是建立在前期进行系统安全性相关工作的基础上，两者之间是不可分割循序渐进的过程。各部分主要工作由以下图所示：

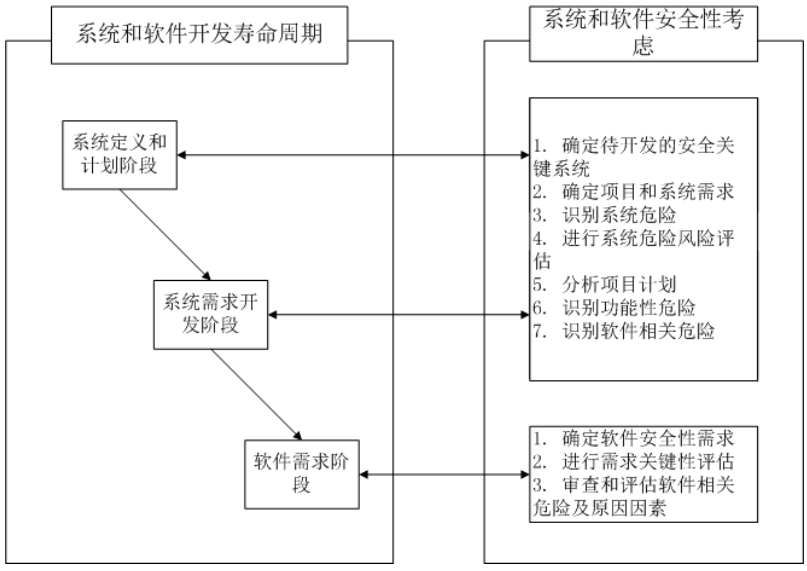

图 6　软件需求阶段软件安全性工作

对于具体各阶段工作采取的方法与技术，各软件安全性标准针对不同等级的软件提出了不同的推荐。

1.2.2. 软件安全性需求获取方法发展与应用

需求工程的主要目的是给出待开发软件系统一个清晰、完整、一致、精确且无二义的模型，该模型以软件需求规格说明书的形式定义待开发软件系统的所有外部特征。

需求获取是软件需求工程过程的首个阶段。在这个阶段，需求工程师的任务包括发现目标系统所处应用领域的信息；认识待解决的特定问题；需要目标系统参与的业务及目标系统需求相关者的特殊需要。需求抽取的基本目标是获取客户、用户和其他利益相关者对待开发系统的需求。在此过程中，需求工程师和需求提供者一起工作，找出要解决的问题，系统要提供的服务，系统要达到的性能，硬件约束等。

在需求工程中处理安全问题的最常见办法是把注意力集中在确定安全性需求。安全性需求一般是功能、数据或接口需求，如果要避免或将危险及其相关的事故危害减少到最低限度，这些需求必须加以适当执行。需求工程领域针对安全性需求的获取方法进行了一些研究，但是多数是针对软硬件结合的系统综合分析，很少有单独对于软件安全性需求获取方法的研究。

随着软件需求工程方法的发展，软件需求抽取方法也在不断改进和发展，并逐渐成为更加系统化的方法体系。目前需求获取技术主要有两个方面：

1.2.2.1 面向目标的需求获取技术

面向目标的需求工程是指利用系统目标来进行需求的启发、求精、结构化、规约、分析、协商、建档并随时对需求进行修订的过程。这里系统是指需要实现的软件及其运行环境。目标则是指正在考虑的系统需要实现的属性，并以事先约定的形式进行描述，它一般包含系统的功能属性和非功能属性。功能目标是所期望的服务，非功能目标是指服务的质量，例如：保密性、安全性、准确性、可执行性、费用和可使用性等,也可以指开发的质量，例如:适应性、互操作性和可重用性等[17]。

面向目标的需求获取是近年来兴起的一种方法。1987年初Yue K首先提出目标概念，他不仅分析了“What”和“How”需求,还在对“Why”问题理解的基础上，将目标作为判断需求完整性的依据：即需求如果能满足当前的目标,则说明需求是完整的[18]。

面向目标的需求获取方法是一种有效的确认需求的方式[19][20]。在需求工程领域,出现了许多面向目标的需求获取方法：如 KAOS(knowledge acquisition in automated specification)方法[21][22]，I*(distributed intention)框架[23][24], GONFR(goal-oriented non-function requirement)[25]等方法。其中基于自动规约的需求抽取方法(knowledge acquisition in automated specification, KAOS)[26]以目标为中心获取需求规约，支持由时序逻辑表示的高层目标生成候选系统体系结构设计。此外，Bubenko 等用目标把组织和组织环境同软件需求相联系[27]。面向目标的需求获取方法从分析最原始的需求材料开始，将最初目标进行分解和提炼，逐步获取更具体的目标并构造出目标模型，进而完成需求建模[28]。

面向目标的需求获取技术目前主要存在的缺点有：

1）初始系统目标的获取是面向目标的需求获取方法的一个关键点，但分析员通常无法完全正确的获得系统全局的目标

2）很多面向目标的解决方案缺乏对于方案正确性的形式化证明

1.2.2.2 基于场景的需求获取技术

多数需求获取方法都将重点放在描述系统功能上。然而需求获取活动也是“一个多方协作的，反复迭代的学习过程”，因此需求获取过程涉及多个需求相关者间的交流和理解。这些需求相关者大都不具有与需求工程技术相关的知识，对需求描述语言和模型不熟悉。这也造成了用户和需求工程师间知识和表达方法的鸿沟。针对上述情况，需求获取研究将主要关注点从软件技术转移到用户身上。基于情景(scenario-based)的需求获取方法即属于一种面向用户的需求获取方法。

场景是为了完成特定任务而按照时间顺序排列的一系列对象间的交互[29]；是对象交互的特定时序；是完成信息系统的某个需要[30]。在一个场景中一系列动作的组合描述了一条唯一的路径。场景可以用初始状态和终止状态作为其区别特征。一个场景的初始状态定义了触发这个场景的前提条件，终止状态则定义了这个场景结束时的状态。

场景可以用不同的符号(语言)表示，可以是非形式化语言，半形式化语言以及形式化语言。非形式化的场景使用自然语言，图像或者描述表达，适用于不愿意或者不能处理形式化符号的用户群。半形式化的场景使用结构化的符号例如表和场景脚本来捕捉实际的动作。形式化的场景用基于规范文法的建模语言或者状态图来表示。

对于基于场景的方法而言,它抓住事例进行分析，有利于人们弄清复杂系统的要求。较之自然语言，场景可以更为精确地刻画系统行为。相比形式化描述技术，场景概念则更为直觉，这也使场景概念受到广泛重视。然而,场景即要描述事例，也要进行分析，但由于场景仅仅提供有限的需求描述，完整需求的获得需要需求获取人员的采用其它方法进行分析。由于不同的人对场景的理解的侧重点不同，所以目前描述场景的方法多达60多种[31]。研究较多的如时序图(event trace diagram)[32]、use case[33]、UML活动图[34][35]、合作图、交互图等方法。这些方法所描述的场景主要用于表现系统行为和系统与环境之间的交互、产生需求规约说明、驱动设计和系统演化等。场景描述由内容、目的、生命周期和表现形式组成。其中，场景的表示形式有静态、动态和交互式[36]。场景既可通过分支和耦合的方式和其它场景相结合，也可以与其它需求获取方法相结合使用，如快速原型法、渐增式方法等[37]。

基于场景的需求获取技术主要缺点有:

1）场景不适合于描述非功能需求。

2）由一系列场景合成出系统整体行为是一个非常困难的问题[38]

3）当前基于场景的设计方法都尚不成熟，有待进一步的研究。

总结这两种技术看来，和面向场景的需求获取方法相比，目标在需求工程早期精确刻画系统行为上比场景要弱，但是，同场景描述相比，目标描述更加容易被形式化，因此也就容易证明其完整性和一致性的问题。

1.2.2.3 需求工程方法在软件安全性需求获取领域的应用

传统需求工程中的很多技术与方法在软件安全性需求获取过程中都可以继续得到应用[39]。在这里我们总结了一下软件安全性需求获取领域目前的主要障碍来自与以下几个方面：

表 5 软件安全性需求获取领域主要障碍

方面	障碍
技术	需求获取技术不充足
	系统组件和接口的表示模型不充足
	缺少安全性途径
	缺少工具
过程	缺少过程模型
	不稳定的安全性途径
	只有部分过程得到检测
	交流不够充分
人员	不同的系统和软件文化背景
	观点片面
	管理效率低下
	对于问题领域的理解有限

目前在针对安全性需求获取的研究领域，Elena Navarro, Pedro Sanchez, Patricio Letelier, Juan A. Pastor and Isidro Ramos等人提出一个面向目标的框架来识别和提取安全性需求。并在机器人系统中进行了初步应用[40]。Letier et al等人[41,42]在面向目标的基础上提出了基于把不预期的行为作为一组障碍,否定这些障碍就成为了满足需求的先决条件的思想，应用KAOS方法进行安全性需求获取研究，取得了不错的进展。Du Junwei, Xu Zhongwei, Mei Meng和Du Junwei等人提出一个基于场景的安全性需求验证技术[43]。

目前国内在软件安全性方面的研究主要集中在航天领域，初步形成了一套适用于航天工程的安全性分析方法，其中以周新蕾、牛爱民等人的研究比较有代表性[44,45,46,47]。周新蕾结合航天型号软件研制工作的特点对软件安全性分析问题进行了较为深入的研究，指出了软件安全性分析在项目实施过程中的基本策略、分析思路和技术要点[48]，特别提出软件安全性分析的关键是从系统角度进行与软件相关的危险分析以及从软件设计角度进行软件对系统的影响分析，并对个别软件安全性分析技术进行了介绍[49]，但对于软件安全性需求获取工作并没有进行深入探讨。

在航空工程领域目前尚无系统的开展过软件安全性分析工作，大部分研究都集中在针对某一具体软件应用单一安全性分析方法上面，如应用 WL_Net 进行导弹飞行控制软件安全性分析[50]，基于时间 Petri 网的软件系统安全性分析[51,52,53,54,55]等，在实际项目中取得了初步成效。

1.2.2.4 软件安全性需求获取方法现状总结

通过以上的论述，我们可以发现目前的软件安全性标准在实践过程中暴露出很多问题[56,57]，在软件安全性需求获取工作方面，目前国内外主要软件安全性标准均提出了各自的解决方案，但也存在很多问题。对各主要软件安全性标准的总结如下表6所示：

表 6　各主要软件安全性标准目前存在问题

国内外研究	解决方案	主要存在问题
GJB/Z 142 军用软件安全性分析指南	主要从系统安全性需求映射及安全相关性分析两方面得到软件安全性需求。给出了一些可用于软件安全性需求获取的技术手段，如检查单和交叉参照、层次分析等	标准给出的主要技术手段只有简单介绍，没有具体实施步骤及案例说明，无法满足实际分析过程的需要；没有给出一个系统的开展软件安全性分析工作的流程，各部分任务说明相对孤立
RTCA DO-178B 机载系统和设备合格审定中的软件考虑	明确了软件需求过程的目标：开发高层需求及向系统安全性评估过程提供衍生的高层需求。面向目标给出了软件需求过程的简单指南	标准中所规定所有任务均是在获得完备的系统安全性需求前提下进行，而标准对于系统阶段工作只有简单描述；只给出了软件需求阶段工作应该达到的要求
NASA软件安全性指南	明确要全面确定所有软件安全性需求需要从系统需求映射、自上而下的系统危险分析及自下向上的设计数据分析得到。推荐使用FTA、PHA、FMECA、模型检查等方法	大部分技术缺少详述；各阶段剪裁任务数量众多，与我国实际软件开发水平存在差距，完全依照手册中任务执行困难；只是指出了每个过程中任务的注意方面，而没有说明具体的工作流程
软件系统安全性手册	在软件安全性需求获取阶段，手册将主要工作分为剪裁通用安全关键软件需求清单与推导特定安全性的软件需求两大部分。给出了各环节的主要任务和输入输出内容	手册旨在帮助系统开发团队理解他们的软件系统安全性职责，强调各学科整合对于保证软件系统安全性的重要性，对于具体的技术细节并没有过多关注

从软件安全性标准的发展趋势我们可以看出，软件安全性标准越来越多的将如何实施具体工作等细节留给软件开发商自己完成，而不再详细规定软件开发过程各环节的具体安全性工作及使用技术。

同样目前国内对于软件安全性分析工作的研究还处于起步阶段，现有的研究大多集中在系统安全性分析领域，重点关注软件硬件结合的系统安全性分析工作，并没有针对不同阶段的软件安全性分析工作给出具体的可操作的指导。很多研究中软件安全性分析方法的应用过于单一，考虑不够全面且多数并没有体现出航空机载软件的特点，尚未形成具体、有效、可操作的软件安全性分析方法。

因此为满足现有航空软件开发的需要，我们应该对标准中已定义的过程进行裁剪、

补充和完善，结合自己的组织和开发过程，参考国内外软件安全性领域的现有研究成果，不断实践和探索，最终形成适合自己的软件安全性需求获取过程。

1.3. 论文研究内容

针对机载安全关键软件特点，结合软件工程过程与软件安全性的相关理论，本论文的主要研究内容如下：

1. 国内外软件安全性需求获取工作开展情况与应用技术调研
    1) 国内外机载系统安全关键软件开发过程与安全性工作开展情况调研
    2) 总结国内外各主要标准中软件安全性需求获取工作流程步骤
    3) 梳理国内外软件安全性工作开开过程中具体使用的主要软件安全性需求获取方法及技术
2. 提出符合现有软件开发水平和开发模式的软件安全性需求获取方法
    1) 结合国内外目前开展软件安全性分析工作的要求， 提炼软件安全性需求获取工作核心内容
    2) 考虑我国机载软件开发单位的实际情况，给出结合现有软件开发过程的软件安全性需求获取工作流程
3. 软件安全性需求获取方法应用研究
    1) 基于航空机载软件的初步危险分析（PHA）方法应用研究
    2) 基于航空机载软件的功能危险分析（FHA）方法应用研究
    3) 基于航空机载软件的软件数据流图分析（DFA）方法应用研究
4. 应用具体航空机载软件项目对提出的软件安全性需求获取流程和应用的具体技术方法进行实例验证
    1) 通过实施过程反馈信息来改善下一步的执行过程
    2) 进一步完善该获取方法，验证其有效性

1.4. 论文结构安排

本论文围绕软件安全性需求获取方法对相关的各个方面进行了探讨与研究。全文正文共分为五个章节，结构安排如下：

第一章、 介绍本课题的研究背景及研究意义，对当前国内外在这一领域的研究情况进行了分析、对比和总结。

第二章、 介绍了软件安全性需求获取的基本概念及相关原理，详细总结了安全关键软件，特别是航空机载软件的特点并明确了软件安全性需求相关概念。

第三章、 首先从软件寿命周期全过程的角度提出软件安全性需求分析工作框架，并明确了各主要工作环节的分析策略。随后重点具体介绍了本文提出的软件安全性需求获取方法。方法从获取通用软件安全性需求及获取特定软件安全性需求两方面展开，详细介绍了方法的分析思路及实施流程

第四章、 针对软件安全性需求获取工作中应用的重要分析方法，从自顶向下的软件安全性需求分析和自底向上的软件安全性需求分析两方面详细介绍了分析过程中应用到的分析方法的原理、目的、操作步骤等。

第五章、 将提出的软件安全性需求获取方法应用于某型航空发动机数控系统控制软件，详细介绍了实例的实施过程、思路与方法，并对最终得到的结果进行了分析，验证了该软件安全性需求获取方法的正确性。

第六章、 概括本文研究的主要内容，阐述取得的成果，指出尚存的问题，提出对未来工作的展望。

1.5. 小结

本节作为文章的总论部分，对文章主要内容进行了简要概述。主要阐述了软件安全性需求获取的重要性，介绍了需求获取技术在国内外的发展现状，指出了目前航空机载软件需求开发过程中存在的瓶颈问题。介绍了本文的研究内容，给出了论文的整体框架和组织结构。

第二章 软件安全性需求获取基本概念及相关原理

安全性（Safety）是指“使伤害或损害的风险限制在可接受的水平内的状态”[58]。通俗的讲，安全是指系统不可能或至少不大可能导致灾难性事故的发生。通常认为灾难有三类或四类，如人员伤亡、重大财产损失、环境污染，或航天武器装备的任务失败。

软件安全性（Software Safety）是指软件运行不引起系统事故的能力[59]。Nancy Leveson 在 1986 年的文献[1]中首次把软件安全性引入到更加宽广的计算机科学领域，建立了这个研究领域的基础。软件没有硬件所具有的物理和化学属性，因此它对人类和社会没有直接威胁，不会造成直接的损害。但是当软件用于过程监测和实时控制时，如果软件中存在错误，则这些错误有可能通过硬、软件的接口使硬件发生误动或失效，造成严重的安全事故。软件的安全性问题不仅在实时控制系统中存在，在其他类型的软件中也可能存在。如果软件储存的或提供的数据是有关安全的重大决策的依据，那么软件错误同样会给人类和社会造成严重的损害。

2.1. 安全关键系统及软件

许多航天飞行器，如航天飞机、空间站、运载火箭和卫星等，其失效可能带来灾难性后果和重大经济损失，所以是安全关键系统。另外其它领域的飞机自动驾驶系统、航空电子系统、导弹指挥系统、核反应控制系统和电厂调度等系统都有一定的安全性要求。安全关键系统出现危险是不可避免的事情，因此采取的办法是在某种程度上接受这些风险，并尽量的想办法来避免这些危险的发生。

2.1.1. 安全关键软件及特点

众所周知，软件本身不会给客观世界带来任何危害，但软件运行时借助载体（如 CPU 或其它执行单元）对现实世界的信息进行获取、存储、加工并通过硬件产生输出。那些控制关键硬件或者向关键硬件提供重要信息的软件、监视关键硬件的软件以及监视系统关键条件和（或）状态的软件，它们的错误可能引起系统的错误操作，从而造成灾难性事故，这一类软件被称为安全关键性软件。可见，软件虽然没有硬件所具有的物理和化学属性，不会对人和社会造成直接的威胁，但是软件一旦出现错误，这些错误可能通过其控制的硬件产生错误或者失效，造成严重的安全事故。

安全关键软件指能直接产生或控制危险的软件，也包括能够对危险软件产生影响的

所有的相关软件。安全关键软件一般具有以下特征：

1) 控制危险和安全关键性的硬件
2) 检测安全关键性的硬件
3) 产生重要数据提供其它软件使用
4) 对危险操作提供决策信息
5) 阻止安全关键性硬件恢复正常

2.1.2. 航空机载软件特点

机载设备作为飞机大系统的重要组成部分，正在发挥越来越重要的作用。大型飞机机载设备主要由航电设备/系统和机电设备/系统两大部分组成，航电系统主要包括飞行控制、飞行管理、座舱显示、导航、数据与语音通讯、监视与告警、机内通话、客舱娱乐等主要功能系统；机电系统主要包括电力系统、环控系统、燃油系统、液压系统、救生系统、辅助动力装置、机轮刹车系统、照明和生活设施等功能系统。

航空机载软件应用的特殊性使其具有一些与通常软件相区别的特点。

1) 航空机载系统通常结构复杂、航空机载软件功能覆盖面广

航电系统的特殊功能及其担负的特殊任务，使其具有极强的领域特性，对环境要求严格，复杂性高。另外，航电系统的综合性也使得航电系统软件的应用范围从数据管理、数值计算到实时操作系统内核等，功能覆盖面广。

2) 高可靠性、高安全性要求

航电系统软件通常都具有较高的质量属性要求。这意味着此类软件不仅要实现其功能性需求，还需要满足若干非功能性需求，如硬件限制、硬实时行为限制、系统资源限制等。而且非功能性需求的实现与否也极大地影响软件质量。

3) 很多航电系统都包含大量实时嵌入式软件

航电系统软件，如航空型号软件中承担飞行控制、系统指挥和数据处理等关键任务的软件都包含大量实时嵌入式软件，具有极强的专用外部设备处理要求和实时性要求，通常运行于特定或具有特殊条件的环境中。

2.2. 软件安全性需求

2.2.1. 软件安全性需求定义

IEEE 软件需求规格[60]中软件安全性需求定义为：识别和使用产品时可能存在的损

失、破坏或者不利相关的需求，定义针对性的保护措施，同时阻止一些行为的发生。参考所有标准和规范中对产品设计和使用的安全性提议，确定满足所有的安全性审定。

NASA 软件安全性手册[61]认为软件安全性需求从系统安全性需求中分解而来，保证系统维持在一个安全状态，同时可以对潜在失效做出充分的反应。软件安全性需求不仅要屏蔽不安全的行为，还可以用来预先监控系统、分析关键数据、进入危险状态之前的信号。所以软件安全性需求必须包括体现这些行为的、主动的、反应式的和系统及软件要求它们必须是有效的需求。

国军标 438B[62]的《软件需求规格说明》关键性要求中对安全性的说明如下：

1) 明确安全性要求等；
2) 规定软件其他安全性要求，如关键功能至少要由两个独立的程序模块共同完成，“监视时钟”的设置要求，软件多余物的处理，程序块的隔离，内存未用空间和未采用中断的处理，对关键数据、变量的保护和校核等；
3) 规定安全性关键功能软件的标识、控制、检测和故障识别；
4) 规定软件失控、加电检测控制顺序出现异常造成的可接受的最低安全性水平；
5) 规定系统的故障模式和软件的故障对策要求。

2.2.2. 软件安全性需求获取

根据 GJB/Z 142-2004 军用软件安全性分析指南，软件安全性分析是指对和软件安全性相关的特定信息进行的系统而有序的获取和评价过程。其目的是通过获取和评估软件安全性相关信息，保证系统安全性质量。

软件安全性分析是系统安全性分析的组成部分，应与系统安全性分析密切结合，协调一致。软件安全性分析在软件开发过程中是迭代进行的。随着项目开发的细化，软件安全性分析也应不断地深入完善。如果出现与较早生存周期活动有关的变化，则应对该变化对安全性的影响进行评估，必要时重复与较早的生存周期活动和随后的活动相关的安全性分析。

软件安全性分析主要包括如下任务：

1) 软件需求安全性分析
2) 软件结构设计安全性分析
3) 软件支持工具和编程语言安全性分析

4) 软件详细设计安全性分析

5) 软件编码安全性分析

6) 软件测试安全性分析

7) 软件变更安全性分析

软件安全性分析任务包含于软件生存周期过程的若干活动中，是针对软件的安全性质量，对于这些活动的补充。在软件安全性分析的众多任务中，本课题重点关注软件安全性需求获取环节。

软件安全性需求获取的目的是要对分配给软件的系统级安全性需求进行分析，规定软件的安全性需求，保证规定必要的软件安全功能和软件安全完整性。软件安全性需求获取的主要输入是软件安全性分析准备的结果和系统的初步结构设计文档，包括系统分配的软件需求、接口需求。软件安全性需求获取开发出的软件安全性需求应成为软件需求规格说明的一部分。软件安全性需求获取可产生后续软件安全性分析的输入信息，和对后续的软件设计和测试的建议。应在软件需求评审中评审软件安全性需求，并将之反馈给进行中的系统安全性分析活动。

2.3. 小结

本章首先明确了安全性及软件安全性概念的定义及内涵，明确了研究对象和范围。其次，对安全关键系统的应用领域探讨，分析并考虑了安全关键系统中安全关键软件的特点，并重点介绍了航空机载软件由于应用领域的特殊性而具有的自身特点。最后，本章给出了关于软件安全性需求的多种定义，并着重剖析了软件安全性需求获取工作的主要任务及目的要求，为后续研究工作奠定了理论基础

第三章 软件安全性需求获取思路

3.1. 软件安全性需求分析工作框架

软件安全性工作，作为系统安全性和软件开发工作的一个重要组成部分，不能独立于总的工作单独进行。软件安全性需求获取工作是整个软件安全性需求分析工作的一部分，是在软硬件功能、逻辑、时序、交互与接口结合的层次上进行的系统、全面的分析获取软件安全性需求的过程。为保证软件安全性分析人员能够全面准确的分析获取软件安全性需求，本文首先从整个软件开发过程的角度出发，进行总体规划，提出航空机载软件安全性需求分析框架，以此为基础介绍软件安全性需求分析工作在软件开发不同过程所应用的基本策略，使软件安全性分析人员明确软件安全性需求分析工作在各阶段所使用的分析思路。

在寿命周期阶段的划分方面，考虑到目前航空领域机载软件系统开发过程大多遵从GJB2786A 武器系统软件开发标准，本框架采用如下过程划分：

1) 系统需求和设计阶段

2) 软件需求阶段

3) 软件设计阶段

4) 软件编码阶段

5) 软件测试阶段

软件安全性需求分析工作作为整个软件安全性工作的开端，其质量直接影响并决定了软件设计的质量，进而影响并决定了软件代码质量，直至整个系统的最终质量。本框架重点考虑了与软件安全性需求分析密切相关的系统需求和设计阶段及软件需求阶段工作内容。

软件安全性需求分析框架如下图 7 所示：

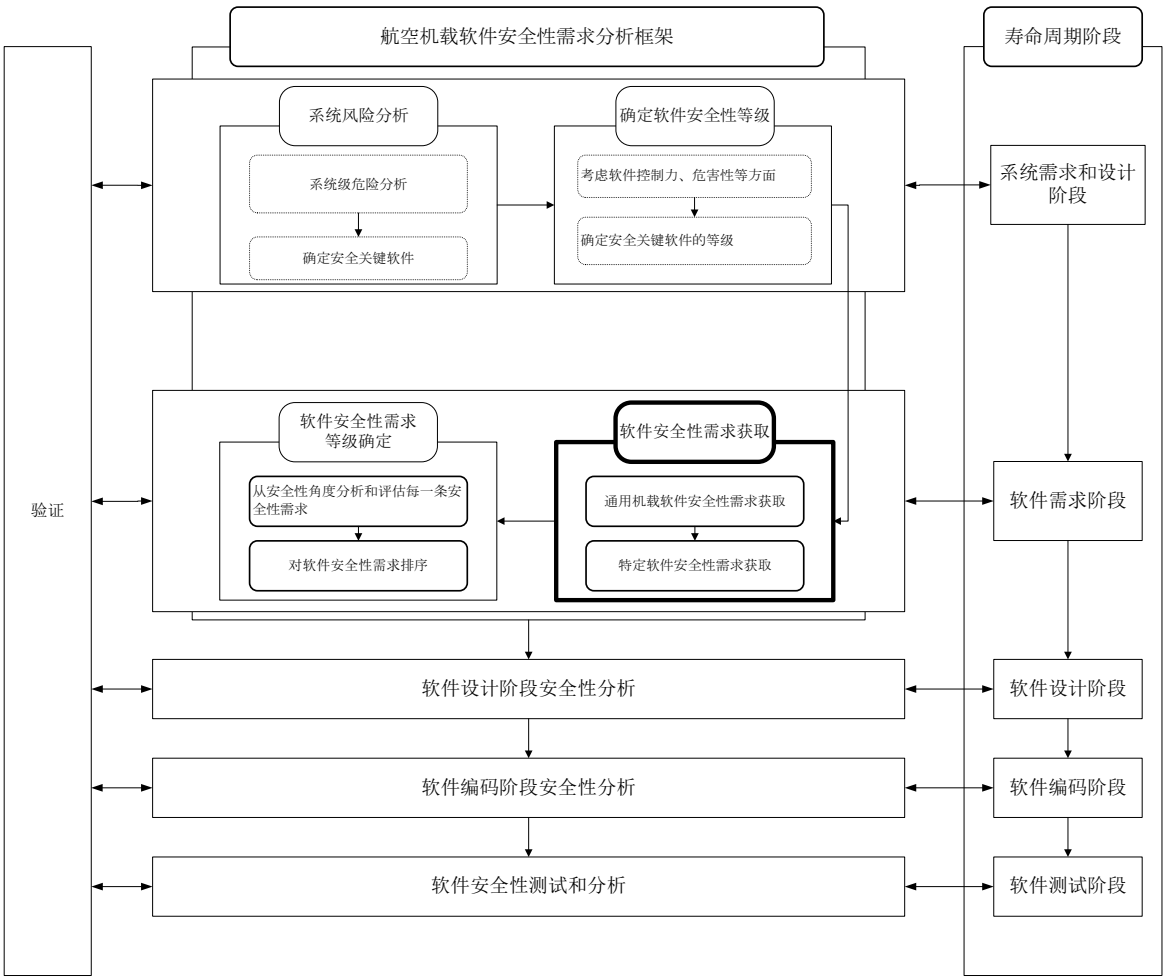

图 7　软件安全性需求分析框架

软件安全性需求分析框架中的主要工作环节如下表 7 所示：

表 7　软件安全性需求分析主要工作环节

序号	工作描述	基本策略
1	系统风险分析	根据系统初步危险分析工作的结果判断软件在系统危险监测、缓解与控制中的作用，确定安全性关键软件
2	确定软件安全性等级	根据与软件相关危险事件可能后果的严重性，参照相关标准要求确定软件安全性关键等级
3	软件安全性需求获取	根据软件安全性需求来源的不同，从通用安全性需求和特定安全性需求两方面分析获取。通用软件安全性需求来自于安全性标准规范、设计准则和前人的工程经验。特定软件安全性需求来自软件所属系统的安全性要求、系统约束和相关危险分析结果
4	软件安全性需求等级确定	按照系统的安全性要求确定危险可能性等级和危险严重性等级，然后根据软件安全性需求对应的软件功能数量、功能失效导致危险事件的严重性等级、可能性等级给出评价，确定软件安全性需求的等级

以下是针对软件安全性需求分析框架中每一个具体工作环节的详细说明：

3.1.1. 系统风险分析

软件安全性需求分析工作由实施系统的初始安全性风险评估来启动。这一部分的工作可以通过开展初步危险分析（PHA）和功能危险分析（FHA）两种方式展开。PHA通常在系统开发的早期进行，标识正在开发系统中的危险及其原因因素，并将它们按优先级排序。FHA 主要分析评估系统内或设备中潜在的故障及所发生故障的影响，通过分析来确定系统操作中可能发生的故障，并判断故障对人员等的危险。

经过初步危险分析后，当系统被确定为安全关键系统，应当分析软件在系统危险的产生、监测、缓解或控制中的作用，确定安全性关键软件。

满足下述准则之一的软件应当被确定为安全关键软件：

1. 可导致危险或危险产生的条件之一；
2. 控制或缓解危险；
3. 控制安全性关键功能；
4. 处理安全性关键命令或数据；
5. 对系统是否达到特定危险状态，进行检测、报告或者采取纠正措施；
6. 危险发生时，减少其损失；
7. 与安全性关键软件在同一处理器内运行的软件；
8. 进行危险趋势分析或数据处理，且其结果直接用于安全性决策；

3.1.2. 确定软件安全性等级

衡量软件安全性等级主要从以下几个方面进行：

（1） 控制程度。软件在系统危险控制上的参与度

（2） 复杂度。越复杂的软件越危险，随着软件用于控制危险的安全性相关的需求增加，软件也随之复杂。

（3） 时效危害度。控制危险的软件实时性是一个关键因素。

对所分析的安全关键软件进行初步软件控制分类，然后考虑软件本身的复杂度等因素，再次进行分类，随后得到软件风险矩阵，这样就最终确定了软件安全性等级

表 8 是在考虑了软件控制分类、软件复杂度、实时性之后得到的软件控制分类描述：

表 8　更新的软件控制分类

软件控制分类	描述
IA（系统风险指数 2）	软件部分或全部自主控制安全关键性功能
	多子系统的，多交互并行处理器的，多接口的复杂系统
	部分或全部的安全关键性功能都是时间关键性的
IIA、IIB（系统风险指数 3）	控制危险过程中，专门的安全系统能够降低危险程度。能够检测出危险，同时能够提示操作人员采取相应安全性措施
	只有几个子系统，几个接口，没有并行处理器的中等复杂系统
	一些危险控制功能是时间关键性的，但是足够操作人员或自动控制系统做出反应
IIIA、IIIB（系统风险指数 4）	软件功能失效时有几个备用方案防止危险发生。大量的安全关键性信息来源。
	比较复杂的系统，少数几个接口
	系统能够在任意时间段做出响应
IV（系统风险指数 5）	没有危险硬件的控制系统。操作人员不会产生安全关键性数据
	只有 2-3 个子系统，少数几个接口的简单系统
	没有时间关键性

3.1.3. 软件安全性需求获取

软件安全性需求通常指那些与阻止或缓解危险发生有关的功能或性能需求。根据软件安全性需求获取来源的不同分为如下两类：通用软件安全性需求和特定软件安全性需求。通用软件安全性需求是指那些可以应用在不同的项目及环境中解决共同的软件安全性问题的需求。而特定软件安全性需求指通过应用危险分约束或功能能力等。

软件安全性需求提取作为需求阶段软件安全性核心工作，其结果直接影响了后续软件安全性工作的开展效果，并且获得正确及完备的软件安全性需求对于从早期根本上消除导致系统危险发生的缺陷，降低软件安全性风险水平有着十分重要的意义。

事实上，大多数危险有不止一项设计或风险降低的相应软件安全性需求，以保证软件系统达到最终的安全性要求。在开发软件安全性需求过程中，对于不同层次安全性设计要求的选取，需要重点考虑软件安全性等级。对不同安全性等级的软件提出不同层次的安全性设计要求，以便在软件开发过程中合理的分配资源，降低软件安全性工作成本。

此外，为全面考虑软件安全性需求，同时要关注软件系统的特征。并不是所有的软

件特征都包含在软件需求中。有些特征是设计的结果，设计可能通过多种不同的方式满足需求。重要的是安全关键的特征得到识别并明确的包含在软件安全性需求中。软件安全性需求分析的结果要反馈到系统需求和系统安全性分析中去。对于所有识别出的差异，系统需求应当因为不完备或不正确得到修改，软件需求也应该调整使得与系统需求符合。

3.1.4. 软件安全性需求等级确定

此部分的主要目标是通过逐步深入的分析将软件安全性需求与系统风险联系起来，从软件相应安全性目标的角度评估每一条软件安全性需求，确定软件安全性需求的等级并对软件安全性需求进行分类。

软件安全性需求通常均与软件实现的功能相关。通过对软件需求规格说明书的理解，软件安全性分析人员已经明确了软件系统的功能，结合系统初步危险分析阶段的工作成果，在此根据功能失效后可能导致的危险严酷度等级将软件功能进行安全性排序，进而联系到软件安全性需求，确定软件安全性需求的等级。

软件安全性需求等级确定流程如下：

1) 分析软件需求规格说明、软件任务书等文档资料，明确与软件系统相关的功能模块。
2) 联系软件安全性需求与对应的软件功能，通常情况下，一个软件安全性需求对应不只一项软件功能模块。
3) 根据软件安全性需求对应软件功能的数量及相应软件功能失效后果的严重性确定软件安全性需求的等级。

在确定软件安全性需求等级过程中，应用矩阵方式来进行是一个很有效的方式，通过确定软件安全性需求等级，软件开发人员应优先实现高优先级的软件安全性需求。

明确软件安全性需求分析工作的总体思路后，针对软件安全性分析工作中的核心部分软件安全性需求获取，本文从获取通用软件安全性需求及获取特定软件安全性需求两部分重点展开进行研究。

3.2. 获取通用机载软件安全性需求

通用软件安全性需求是指那些可以应用在不同的程序及环境中解决通用的软件安全性问题的需求，这些通用需求均是从包括安全关键软件的系统征集得到的设计特征、

设计限制、开发过程、最佳实践方法、编码标准和技术以及其它的通用要求等。这些要求本身并不是安全性专用的（即不与特定系统危险相联系），但它们都根据以往发的导致灾祸或潜在灾祸的失效或错误的系统经验教训总结得到。

机载软件的研制过程中，相似的处理器、平台、软件功能和使用环境可能引起相似的安全性需求。这些软件安全性需求实际上是在不同项目和环境中解决共同的软件安全性问题时的最佳实践的集合，为开发者提供了非常有价值的通用的安全性需求来源，可以通过复用这些通用的安全性需求和已经过实践证明的满足这些需求的解决方法来代替采用新方法以减少代价，避免重蹈覆辙。

对于通用软件安全性需求的获取拟采用如下方法进行：

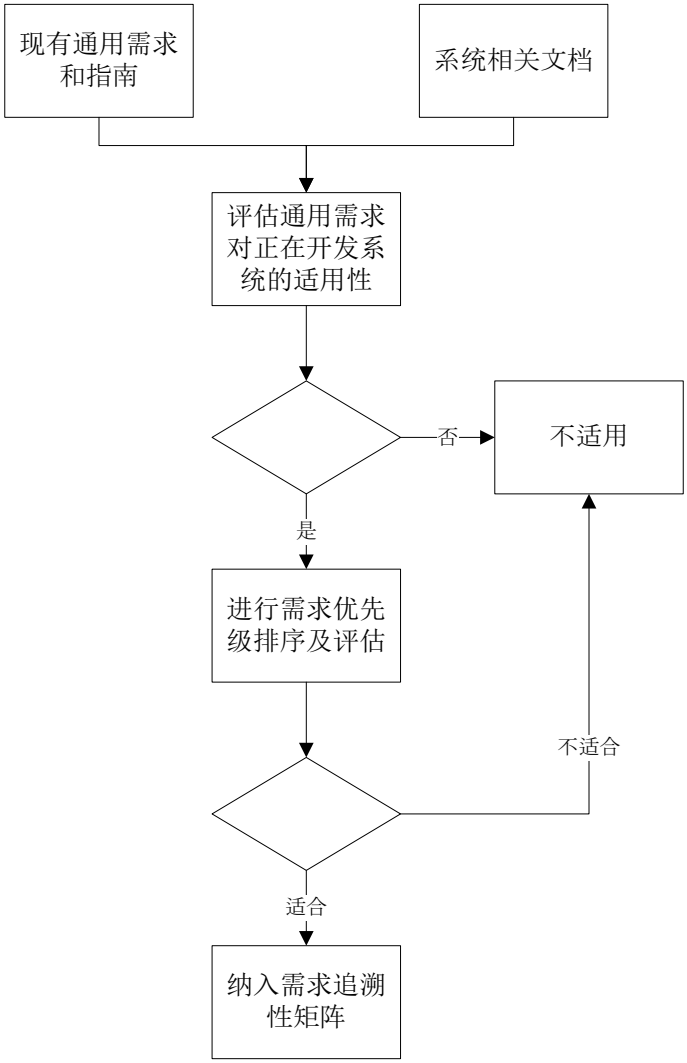

图 8　获取通用软件安全性需求

提取具体步骤详细描述如下所示：

1) 参阅相关文档：初步系统规格说明、初步产品规格说明、经验教训、类似系统的

分析（包括安全性分析）、参照的设计准则和标准（如GJB、航标、所标等），了解系统的物理和功能要求、对系统方案设计有初步了解。

2) 结合已有的通用软件安全性需求清单 ，逐条考虑对所分析软件系统的适用性，进行相应剪裁，选择适用于该开发工作的软件安全性需求

3) 考虑软件开发项目的资源要求，评估实现人员，设计、编码和测试活动的预算以及项目的进度要求等方面，对初步得到的软件安全性需求进行优先级排序，评估选取对软件开发成本和产生效益影响可接受的软件安全性需求

4) 针对所分析软件系统，将剪裁得到的通用软件安全性需求进行细化，纳入需求追溯性矩阵。

3.2.1. 开发通用软件安全性需求清单

目前主要参考的通用软件安全性需求来源如下表 9 所示：

表 9 通用软件安全性需求来源清单

序号	标准号	标准中文名	标准英文名
1	Joint Software System Safety Committee	软件系统安全性手册	Software system safety handbook
2	ESD-TR-86-278	用户接口软件设计指南	Guideline For Designing User Interface Software
3	NASA-GB-8719.13	软件安全性指南	NASA Software Safety Guidebook
4	FAA	系统安全性手册	System safety Handbook
5	SSP 50021	安全性需求文档	Safety Requirements Document
6	NSTS 19943	北约客户命令要求和指南	Command Requirements and Guidelines for NSTS Customers
7	STANAG 4404	监视相关的安全关键计算系统安全性设计需求和指南	NATO Standardization Agreement (STANAG) Safety Design Requirements and Guidelines for Monition Related Safety-Critical Computing Systems
8	EWRR 127-1	雷达安全性需求	Range Safety Requirements - Western Space and Missile Center, Attachment-3, Software System Design Requirements
9	AFISC SSH 1-1	系统安全性手册-软件系统安全性	System Safety Handbook - Software System Safety
10	EIA Bulletin SEB6	软件开发中的系统安全性工程	A System Safety Engineering in Software Development

综合以上各文档内容，考虑机载软件特点，最终整理得到通用软件安全性需求 110 条，按照通用需求描述的方面不同，分类整理如下：

表 10　通用软件安全性需求分类统计

类别	数量
危险命令相关	9
初始化相关	5
数据处理相关	31
人机交互相关	17
软件自身功能相关	41
其它	7

具体通用软件安全性需求示例如下表 11：

表 11　通用机载软件安全性需求清单

编号	类别	需求描述
1.	危险命令相关	危险命令必须只能由控制应用程序、机组人员、地面或者控制执行主管提出
2.		在执行一个已标识的危险命令之前，应满足安全执行该命令的前提条件（如正确的模式、正确的配置、组件可用、合适的顺序和参数在有效范围之内）
3.		撤销或取消命令需要经过多个操作步骤
4.		执行危险命令的软件必须通知发起者、地面操作员、被授权的控制执行者或者提供执行失败的原因
5.		所有和危险命令关联的软件约束只能有唯一的检验器（检验准则）
6.		每一个和危险命令关联软件约束命令在使用规则和合法值时必须是一致的
7.		在解除了一个与危险命令有关的软件禁止机制之后，软件应能恢复对一个已禁止操作的控制
8.		对于一个确定的危险命令执行前应当满足安全执行的前提条件
9.		危险命令应仅由单一的控制软件功能发出
10.	初始化相关	软件必须设计成在上电时进行系统级检查，以便在对安全关键功能包括由软件控制的硬件通电之前验证该系统是安全的并正确的运行。必须用软件进行定期测试以监视系统的安全状态。
11.		在任何可替代单元或组成部件中使用的软件（包括固件）加电自检应局限于由该可替代单元或组成部件控制的单一系统处理
12.		用于任何可替换单元或部件的软件（包括固件）上电自检，必须结束在安全状态。
13.		软件必须能够初始化、开始和重启可替代单位到一个安全的状态
14.		软件开机自检（POST）时，须限于由接受的POST组件控制的单一的系

		统进程
15.		确保一个用户只需对特定数据输入一次，之后计算机可以根据相同或不同任务的需要存取这些数据
16.		当数据输入对于用户任务来说非常重要时，输入的数据应当显示在用户的主显示器上
17.		对数据输入过程中的所有用户活动提供显示的反馈；一步一步显示输入的数据
18.		确保计算机快速响应数据输入活动，这样用户不会因计算机响应的延迟延缓进度；对于正常的操作，显示的反馈延迟不能超过0.2秒
19.		设计数据输入处理和显示，使得一个用户可以使用一种方法输入而不用转换到其它方法
20.		数据输入时只允许在电子显示器确定区域显示，如填写进表格，提供明确的输入区域的定义
21.		关键数据输入时，始终允许用户在必要时（包括显示的缺省值）使用删除和插入方法更改先前的输入；如果数据改变在某些时候通过字符替换（改写）完成，则该方法也应该始终可用
22.		允许用户按自己的节奏输入数据，而不是通过计算机进程或是外部活动来控制节奏
23.		始终要求一个用户执行一个明确的输入活动来开始输入数据的过程；不要以其它活动的副作用的方式开始这个过程
24.		明确的标记输入键来表明他的功能
25.	数据处理相关	需要一个用户执行一个明确的活动来取消一个数据的输入；数据取消不应该以其它活动副作用的方式完成
26.		确保计算机可以通过一个确认消息对数据输入处理的完成进行响应，如果数据输入是正确的，或者产生一个错误消息
27.		对一个由持续处理完成的反复数据输入，需要通过在显示屏重新生成输入的数据，并自动清除为下一次输入做准备来表明成功完成一次输入
28.		如果一个用户需要改变（或删除）一个现在没有被显示的数据项目，在确认改变之前提供给用户显示该数值的选项
29.		对于编码数据，数字等，保持数据输入简短，这样一个单独项目的长度不会超过5-7个字符
30.		当必须输入一个长数据项目时，长数据项目应当为了输入和显示分割成较短的符号组
31.		允许专家用户使用可选择的长数据项目的缩写来减化关键数据输入，当这样做不会产生歧义时
32.		当定义缩写或其它的代码来缩短输入数据时，选择有区别的那些，以避免相似的两者之前产生混淆
33.		当定义缩写时，遵从某些简单的缩写规则，确保用户者可以理解这些规则
34.		只在为了清楚的需要时使用特殊的缩写 （例如，那些没有形成一致的规则）
35.		当一个缩写必须与一贯的规则偏差时，尽量减少偏差程度
36.		使缩写长度相同，在能确保缩写独一无二时，缩写尽可能短

37.		当电脑不能识别一个缩写的数据输入时，必要时询问用户解决任何不明确问题
38.		提供数据输入的需要格式和可接受值的提示
39.		允许用户通过单按一个适当标记的键输入一个数据项目的每一个字符
40.		设计数据输入的处理以最小化转换按键的需要
41.		对于编码数据的输入，等价对待大小写字母
42.		允许在一个整数的结尾可选择的输入或省略一个小数点作为等效替代
43.		对于通常的数字数据，允许可选的输入或省略起始零作为等效替代
44.		在数据输入时单一或多个空格等效对待；不要要求用户计算空格数量
45.		如果一个用户必须输入分层数据，在某些数据将会从属于其它数据的地方，提供计算机辅助来帮助用户明确分层结构里的关系
46.		软件必须在实施自动的危险或安全处理后通知机务、地面操作员和授权的控制者
47.		当软件被告知或者探测到能够引起系统失效硬件错误活软件故障，或者配置和当前的运行状态不一致时，软件必须通知机务、地面操作员和授权的控制者
48.		自动的恢复动作必须通知机务、地面或者控制执行人员，同时没有必要对机务、地面操作员继续执行恢复动作做出响应
49.		如果先决条件未被满足，软件必须拒绝执行命令同时向机务、地面操作员、被授权的执行控制者告警
50.		软件应给机务、地面操作者或控制执行者提供可获得的软件可控制约束。
51.		软件必须接受和处理机务、地面操作员、被授权的控制执行者执行激活或解除的命令
52.		软件必须提供警告和报警情况给机务、地面操作员、被授权的控制执行者
53.	人机交互相关	软件必须为机务或地面提供强制执行的一些自动化安全、隔离或者切换的功能
54.		软件必须为机务和地面提供强制结束的一些自动化安全、隔离或者切换的功能
55.		软件必须为机务和地面提供一些自动化安全、隔离或者切换到之前状态或配置的功能
56.		软件必须为机务和地面提供取消的一些自动化安全、隔离或者切换的功能
57.		需要操作的软件功能的访问必须要得到授权认可
58.		软件必须设计成使操作员可以用单一的行动删除当前处理，并将系统回复到一个已设计的安全状态
59.		软件应当提供给机组人员和地面人员与危险命令有关的软件抑制状态
60.		软件必须能检测不正确的操作员录入或操作，并防止由于该差错的结果而执行安全关键功能
61.		重写命令需要至少两个独立的操作人员
62.		软件应为机组人员和地面人员提供自动终止或禁用自动防护功能
63.	软件自身功能	安全关键软件功能应该能够被检测、隔离和恢复以防止灾难或严重级危

	相关	险事件的发生
64.		软件必须为已知的安全关键功能，视危险程度24小时执行自动的失效检测、隔离和恢复
65.		FDIR的切换软件必须存在于可用的、无失效的控制平台上，这个平台和被监视功能的平台是区别开的
66.		软件应该在导致严重级危险事件的时间之内处理必要命令
67.		软件必须提供独立的且唯一的命令来控制每个软件可控约束
68.		软件必须同时具备识别和上报出每个和危险命令相关的软件约束
69.		软件必须使得当前情况可用于处理和软件约束有联系的对机务、地面操作员和其他控制执行者的危险命令
70.		如果一个自动的序列在软件危险命令关联的约束激活之前已经开始执行，那么它必须在软件约束得到执行之前全部完成
71.		在取消操作之后，软件约束的状态必须保持不变
72.		软件必须提供支持安全关键功能的出错处理
73.		软件必须提供故障包容（容错）机制以防止错误在可替代的单元接口交叉传播
74.		危险的负载必须给核心软件系统提供失效情形和数据，监控状态和报告失效
75.		对于系统只使用软件来减轻危险的情况，软件必须需要两个独立的来自命令系统的可能导致危急或灾难性危险的通知
76.		软件必须需要两个独立的操作动作去初始或者终止可能导致关键危险的系统功能
77.		软件必须需要三个独立的操作动作去初始或者终止可能导致灾难性危险的系统功能
78.		软件必须提供正确的安全关键命令的处理顺序（包括时间）
79.		软件必须终止在安全的系统状态
80.		在硬件失效时，软件故障引起系统失效，或者软件探测到配置和当前的运行状态不一致时，软件必须有能力将系统置于安全状态
81.		视关键等级而定的危险和安全的实时处理不适合人为的干预，必须是自动化的
82.		无用或者非正式的代码不能造成关键或灾难性危险
83.		所有的安全关键元素（需求、设计、代码和接口）都要标识为“安全关键”
84.		检测到不安全条件时软件必须将在软件控制下的硬件子系统项返回到某个指定的安全状态
85.		完成测试和/或训练以后，在测试或训练期间被去除、禁止使用或旁路的安全互锁的恢复必须在能重新开始正常运行之前由软件进行验证。
86.		软件必须保证记录所有检测到的系统错误。安全关键例程中的错误必须被突出，并必须使它们出现之后尽快地引起操作员注意。
87.		关键功能的软件控制必须具有反馈机制，该机制给出该功能出现的正向指示。
88.		系统和软件必须设计成确保在峰值负载条件下设计安全性要求不会被违反。

89.		软件必须设计成在电源故障或断电情况下保证安全，有序地关闭系统，使得不会产生潜在不安全状态。
90.		软件必须设计成能防止未被授权的系统或子系统交互启动或继续安全关键功能指令序列。
91.		系统设计必须防止未被授权或无意的存取或修改软件和目标代码，包括防止代码的自修改。
92.		软件设计中必须考虑已知的部件失效模式，并将检查手段设计到软件中以检测失效。
93.		安全关键计算系统功能中的判定语句必须不依靠全一或全零的输入，特别当这个信息得自外部传感器时。
94.		要求两个或多个来自软件的安全关键信号的外部功能必须不从单个输入/输出寄存器或缓存器接受全部必须的信号。
95.		软件的满刻度和零表示必须都与任何数字到模拟、模拟到数字、数字到同步、和同步到数字转换器完全兼容。
96.		必须要求两个或多个不同的操作员动作来启动任何潜在危险的功能或功能序列。该要求的动作必须设计成使无意动作的可能性最小，并必须检查顺序是否正确。
97.		软件必须能鉴别有效和无效（即虚假）的外部和/或内部中断
98.		存储安全关键数据所用文件必须是唯一的并有单一的目的
99.		用来存储或传输安全关键信息的文件必须在使用之前和之后被初始化到一个已知状态。数据传输和数据存储必须尽可能加以审核，使能实现系统运行的追溯性
100.		倘先决条件未得到满足，软件应当拒绝执行该命令
101.		被一个覆盖忽略或改变的软件抑制应恢复到原来的状态
102.		实时性的危险进程和防护进程，人为干预可能无法及时提供安全处理，应自动化进行
103.		软件应提供错误处理以支持关键功能
104.	其它	系统设计必须不允许检测到的不安全状态被回避
105.		输入/输出寄存器和端口决不能既用于安全关键功能又用于非关键功能，除非同样的安全性设计准则都适用于非关键功能
106.		系统必须设计成将能够检测安全性内核（如果实现）的失效，并将系统返回到指定的安全状态
107.		系统必须设计成包括在系统部件失效事件情况下低效运行和恢复到一个已设计好的降低系统功能能力的安全状态
108.		系统必须设计成在某个安全状态中上电
109.		对于所有模拟和数字输入和输出，必须在按照这些值执行安全关键功能之前进行范围和合理性检查，包括时间范围、依从关系和合理性检查。不得根据不能验证的安全关键的模拟或数字输入执行任何安全关键功能
110.		安全关键功能必须有一个且只有一个导致其执行的可能路径

3.2.2. 通用软件安全性需求裁剪

在得到通用软件安全性需求清单后，软件安全性分析人员需要与系统安全性分析人员，软件质量保证人员，领域专家，软件开发人员等共同确定每一项通用软件安全性需求针对所开发系统的适用性，裁剪出适合该项目的软件安全性需求。在这里可以采用头脑风暴的方式，按照以下两步骤进行：

1) 初步分析所开发软件系统的特点，确定需要重点考虑的安全性方面

通用软件安全性需求主要可以分为危险命令、初始化、数据处理、人机交互、软件自身功能等相关方面。很多情况下软件安全性问题的发生是多方面原因共同作用的结果，解决安全性问题需要安全性分析人员对系统有一个全面的考虑。同时考虑到不同软件自身的独有特点，安全性分析人员在逐条确定通用软件安全性需求适用性时，可以针对不同特点的软件侧重不同的方面：

如针对软件人机交互功能，类似机载导航软件与飞行员信息交互密切，需要重点关注；针对数据处理相关方面，类似软件飞行控制软件等涉及到复杂的运行控制率及数据采取处理等，需要重点关注；针对初始化相关方面，发动机控制软件等对上电自检要求严格的软件系统需要重点关注。

2) 逐条确定通用软件安全性需求的适合程度

所有的软件功能都有其成本。绝大多数项目没有足够的时间或资源实现功能性的每个细节。决定哪些特性是必要的，哪些是重要的，是需求开发的主要部分。针对具体软件开发项目，某些通用软件安全性需求可能在技术上行不通，或者实现它要付出极高的代价，而某些需求试图达到在操作环境中不可能达到的性能，或试图得到一些根本得不到的数据。在时间和资源限制下，软件分析人员应当与软件开发人员及客户进行协商，提供有关每个需求的花费和风险的信息，确定软件安全性需求在具体软件开发项目中的适用性

软件安全性分析人员需要考虑现有的系统设计约束，软硬件配置情况及相似系统设计经验，据此确定通用软件安全性需求中每一项的适用性。对于每一项通用软件安全性需求，分析人员应当讨论明确适用及不适用的原因以及此软件安全性需求在系统中实现的物理位置等。

需求适用程度定性分为以下三级：

表 12　需求适用度等级

等级	含义
适合的（高）	一个关键的安全性需求，必须在此版本实现；只有在这些需求上达成一致意见，软件才会被接受
条件的（中）	完全实现这些需求将增强产品的功能或质量，但如果部分实现，产品也是可以被接受的；
不适合的（低）	功能或质量上的增强，如果资源允许的话，实现这些需求会使产品更完美；在此项目上可以不实现

依据以上分析的结果，分析人员从中选择适合具体软件开发项目的通用软件安全性需求，并根据具体软件开发项目的实际情况进行限定及补充。

3.3. 获取特定软件安全性需求

与通用软件安全性需求不同，特定软件安全性需求是通过应用危险分析技术得到的系统独一无二的功能能力或约束。由于系统故障的检测、隔离与恢复、系统危险的缓解、监测与控制等功能常常由软件完成，软件安全性需求获取不能只从软件本身出发，必须从系统角度进行分析，考虑软件使用过程中软件、硬件和操作人员的相互作用，分析软件可能的工作时序、适用条件、逻辑缺陷及其可能造成的不利影响。凡是与软件相关的硬件、操作人员、时序、接口和模型等方面的缺陷，包括软件本身的缺陷而导致的安全性问题，均属于软件安全性需求获取的范畴。因此，特定的软件安全性需求通常来自软件所属系统的安全性设计要求、系统约束和相关的危险分析结果。

在特定软件安全性需求的具体获取过程中，软件安全性分析人员主要应从系统角度进行的与软件相关的危险分析以及从软件功能设计角度进行的软件对系统安全性的影响分析两个方面进行考虑：

3.3.1. 自顶向下的软件安全性需求分析

根据 NASA 软件安全性手册中的要求，特定软件安全性需求的获取应考虑通过对系统设计需求和规格说明进行自顶向下的分析，从系统危险的角度分析系统，将初步的危险原因映射到软件或者与软件进行交互作用，识别出软件危险控制特征并将其描述为安全性需求。同样，GJB/Z 142 军用软件安全性分析指南中也提出软件安全性需求包括自顶向下从系统需求传递而来的部分，通过系统安全性需求映射工作刚分配给软件的系统安全性需求分析转换为软件的安全性需求；此外考虑周新蕾等的研究，软件安全性分

析应从系统顶级开始，自顶向下逐层逐级分析到软件功能层次，获得的安全性需求。

综合以上思路及其它相关研究，本文提出的自顶向下的软件安全性需求分析工作考虑从系统角度出发，根据软件以上的系统层次安全性分析结果和系统设计要求，进行自顶向下的软件安全性需求分析。通过进行系统和软件上层分系统的危险分析，获得与软件相关的系统危险模式，在此基础上进一步分析由于软件输入、软件运行支撑环境的硬件故障模式所可能导致的危险事件及其影响、采取的相关措施。

分析过程如下图所示：

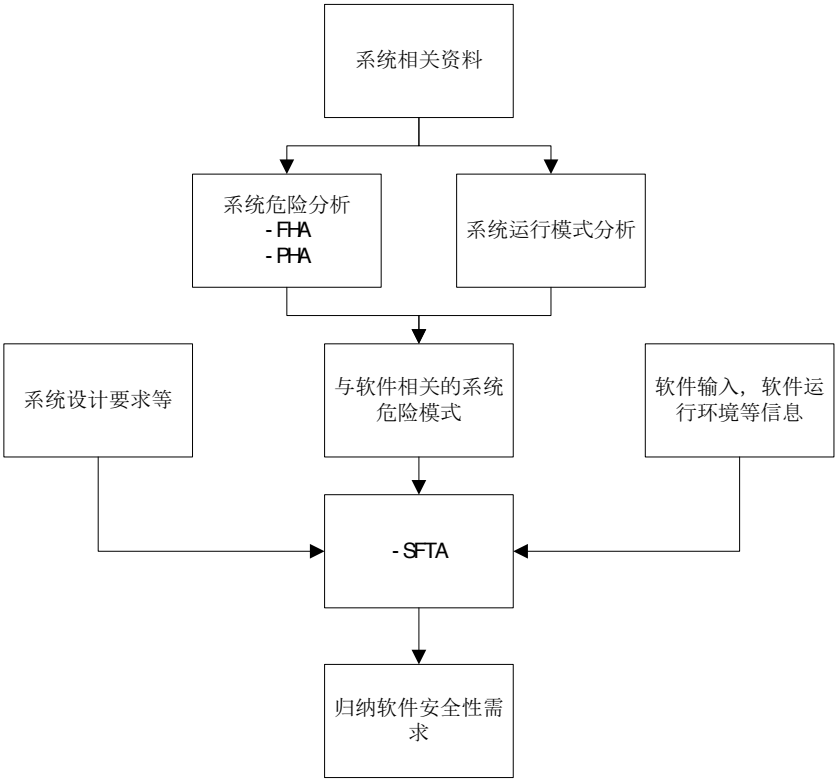

图 9　自顶向下的软件安全性需求分析流程

自顶向下的软件安全性需求分析具体步骤如下：

1) 软件特定安全性需求的获取是从系统开发工作早期系统需求和设计阶段的系统危险分析工作开始。软件作为系统组成的一部分，在不同的运行阶段实现不同的功能满足系统不同的要求。因此，了解系统运行模式对于软件安全性分析人员开展安全性分析工作十分重要。

2) 结合系统运行模式，软件安全性分析人员可以运用不同的危险分析方法（如初步危险分析（PHA），功能危险分析（FHA）等），通过分析与系统规格说明、

经验教训、类似系统的安全性分析结果、常识以及相关系统信息，识别与软件相关的系统危险模式，并在系统级层次提出相应的危险控制考虑。

3) 随后在此基础上，软件危险分析可采用如软件故障树分析（SFTA）及软件故障模式影响和危害性分析（SFMEA）等方法，考虑系统初步的安全性设计要求，软件输入、软件运行环境等因素，进行深入的危险原因因素分析。危险原因可能由硬件（或硬件部件）、软件输入（或没有软件输入）、人员错误等产生。危险可以由于某个特定原因产生，也可以由许多原因的任何组合产生。无论何种原因，软件安全性分析人员必须为软件设计和开发人员标识和定义危险控制需求，以影响软件概要设计活动。

4) 通过一系列的危险分析过程，软件安全性分析人员应确立追踪软件安全性需求到测试的方法（如采用软件需求追溯性矩阵），将由系统安全性要求转换、细化，以及由软件安全性危险分析得到的特定软件安全性需求纳入软件需求追溯性矩阵，以便在随后的开发过程中对每条软件安全性需求的实现进行追踪。

3.3.2. 自底向上的软件安全性需求分析

根据 NASA 软件安全性手册的要求，软件安全性需求获取重点还应包括通过对软件设计数据进行的自底向上的分析，对软件的设计实现进行分析标识新的危险原因，最终通过提出软件安全性需求来规定软件危险控制；GJB/Z142 军用软件安全性分析指南中提出通过进行自底向上的分析，识别与系统需求不一致，或系统需求所未阐述的安全性需求，揭示出不安全状态的路径等情况来对软件安全性需求进行补充；此外周新蕾等人的研究中也指出软件安全性分析应从软件的设计角度出发，分析软件设计缺陷对系统的不利影响 ，自下而上地归纳分析结果进行补充。

综合以上思路及其它相关研究，本文提出的自底向上的软件安全性需求分析拟从软件功能设计角度出发，建立软件功能模型，并依据软件功能模型进行软件危险分析，确定软件安全性关键功能，安全性关键功能的输入、输出，分析可能影响软件运行的软件功能设计缺陷，提出相应措施

分析过程如下图所示：

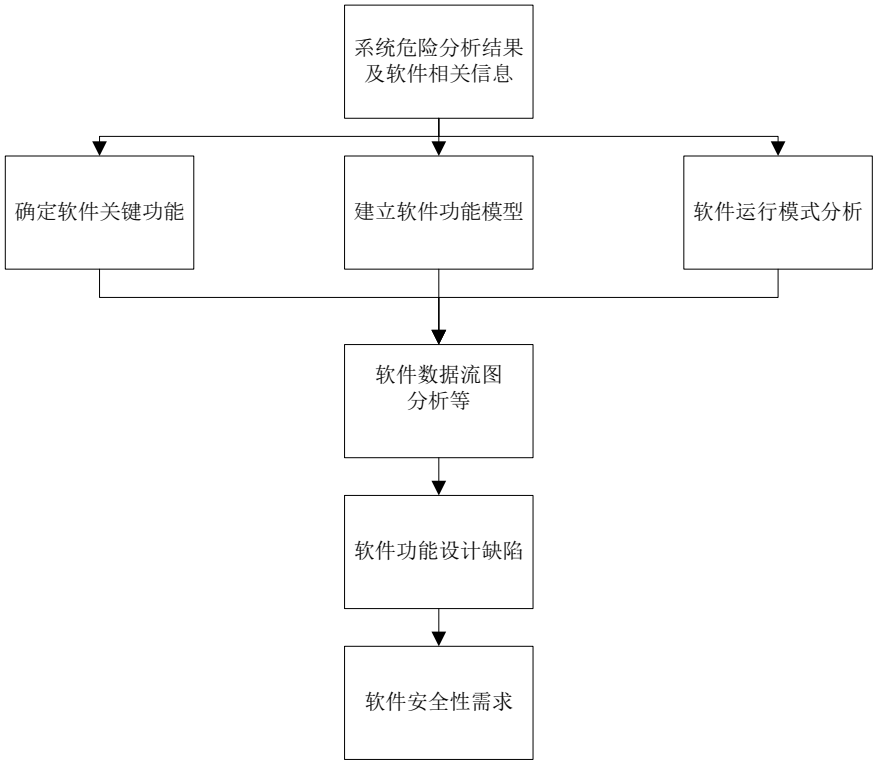

图 10　自底向上的软件安全性需求分析流程

自底向上的软件安全性需求分析具体步骤如下

1) 进入软件需求获取阶段，软件分析人员需要深入描述软件的功能和性能，确定软件设计的约束和软件同其它系统元素的接口细节，在此基础上使用自顶向下，逐层分解的方法将外部需求赋予软件的各个功能成分，定义软件成分的内部功能，并标定它们之间的接口，建立软件功能模型，作为软件危险分析的基础。

2) 与此同时，软件安全性分析人员需要根据系统初步危险分析的结果及系统功能的逐层分解识别确定安全关键的软件功能，并根据系统设计说明、软件需求规格说明等文档的信息，分析确定软件在系统环境中的运行模式。

3) 在软件功能模型的基础上，软件安全性分析人员可采用结构化的分析方法，如软件数据流图分析（SDFD）、软件控制流图分析（SCFD）等，重点关注安全关键功能有关的数据传递、变换关系，开展面向数据流及控制流的软件安全性需求获取，识别软件功能设计缺陷。

4) 针对识别的软件功能设计缺陷，考虑功能设计缺陷可能导致的危险事件及其可能的影响，包括软件设计缺陷直接导致危险发生，软件设计缺陷诱发人员

错误操作，软件设计缺陷影响其它系统级危险的缓解等方面，提出相应的解决措施，细化为特定的软件安全性需求。

5) 将由软件安全性危险分析得到的特定软件安全性需求纳入软件需求追溯性矩阵，以便在随后的开发过程中对每条软件安全性需求的实现进行追踪。

3.3.3. 软件安全性需求获取特点

软件安全性分析人员在分析获取软件安全性需求过程中，应当充分考虑软件与硬件之间，安全性需求与一般需求之间的差异，重点包括以下几个方面：

1. 软件安全性需求获取，应当与系统危险分析结合进行。系统危险分析过程为软件安全性需求获取提供输入信息，而软件分析的结果，应当反馈给系统危险分析人员，用于在系统层次进行更深入细致的安全性分析

2. 软件与硬件的故障模式不同，因此分析的侧重点也不同。针对机载软件系统的特点，分析人员应重点关注危险命令、故障和失效容错、采样率、动态内存分配、控制系统设计等方面。

3. 由于软件的逻辑、数据、时序等设计缺陷，与软件相关的硬件故障和状态等都有可能引起软件失效，导致系统进入危险状态。因此分析时必须对软件进行全方位的分析，包括软件的静态、动态、逻辑和物理模型。

4. 软件安全性分析以人工分析为主，且需要各领域的专业人员共同工作，通过不断的归纳、分析、修改，不断细化最终获得完整的满足工程要求的软件安全性需求

3.4. 小结

本章首先分析了软件安全需求分析工作的重要性，提出软件安全性需求分析工作框架，并在此基础上了介绍软件安全性需求分析工作在软件开发不同过程所应用的基本策略。 随后本章重点展开描述了作为软件安全性需求分析工作核心部分的软件安全性需求获取工作。针对软件安全性需求获取工作的两个主要部分，本章依次给出了获取工作的实施思路，具体步骤，对流程的每个环节进行了具体说明并形成了完整、具体、可操作的分析过程。

第四章 软件安全性需求获取方法

采用适当的软件安全性分析技术对于软件安全性需求获取工作的顺利开展十分重要。许多软件安全性相关标准和研究文献中都提出了适用于软件安全性需求获取过程的安全性分析技术，在这里针对目前航空机载软件开发现状及本文软件安全性需求获取工作的分析思路，分别从自顶向下的软件安全性需求分析及自底向上的软件安全性需求分析两方面详细介绍所应用的主要分析方法。

4.1. 自顶向下的软件安全性需求获取主要分析方法介绍

自顶向下的软件安全性需求分析过程可采取的主要方法包括初步危险分析（PHA）、功能危险分析（FHA）、系统运行模式分析、事件链分析、软件故障树分析（SFTA）及层次分析等。这里选择介绍几种主要应用方法如下：

4.1.1. 系统运行模式分析

系统运行模式是指在不同的外在环境、操作指令等条件下，系统不同的运行状态和组合方式。航空机载系统的高度复杂性决定了在一个任务周期中系统经常需要切换不同的运行模式。明确系统运行模式，可以使软件安全性分析人员针对不同的运行模式分类不同的系统功能失效，进而对软件系统进行全面的分析。

系统运行模式分析具有如下步骤：

1) 参考初步系统规格说明、初步产品规格说明，明确系统组成。
2) 参考系统需求规格说明、系统设计文档等信息，明确系统的功能、运行过程、实现的任务要求等
3) 分析系统任务周期全过程，根据过程的阶段特点，分析系统不同的运行方式及其有效组合，最终得到全部的系统运行模式。

4.1.2. 初步危险分析（PHA）

作为系统寿命周期中最先应用的危险分析，PHA一般在系统研制初期缺乏详细设计资料时予以去用。由于PHA是针对整个系统，因此，其分析的范围是相当广泛的，除PHL所列的危险外，还应包括故障、接口、环境、使用、试验、维修、规程、人为因素等方方面面的可能的危险分析。其分析结果将作为其它安全性分析的基础。因此，根据初步的系统设计方案，在识别出各种危险的基础上（可能要对PHL进行增补），要尽可能鉴

别出由其导致的事故场景，包括潜在条件与引发事件，相应的事变及后果事件。这些场景可能还不完整、详细，尽管还缺乏系统设计的细节，但应最广泛彻底地鉴别危险及其可能的相应场景。此处应特别注意事故场景的源头是危险，而不是某个恶化事件及其状态，否则可能导致场景的不完整而忽视了真实的事故原因。

分析人员通过PHA，可全面识别系统及其使用、运行环境中存在的危险因素、相关的危险特征和危险状态，确定可能的事故。PHA应尽可能早的进行，并随着设计和研制工作的开展不断的改进。

PHA既是一个危险分析工作项目，又是一种具体的分析方法。作为一种具体的方法应用时，PHA具有以下工作步骤：

1) 明确安全性对象的范围，即可能遭受危害的对象，如人员、设备、生产能力、作战能力、环境等
2) 确定可接受的风险水平（指标）。有可能对每一类安全性目标都要确定一个风险接受水平。
3) 定义系统边界和事件阶段（即系统状态），确定分析的范围
4) 具体分析潜在的事故场景
    a) 确定具体分析对象（系统、分系统或设备）
    b) 确定系统事件阶段
    c) 确定相关危险因素
    d) 分析危险条件
    e) 初别初始事件
    f) 分析事故发展过程和事故后果
5) 确定每一个事故发生的可能性和严重性，定性评价其风险等级
6) 确定事故风险是否可接受，对于不可接受风险的事故，提出相应的控制措施，并分析控制措施是否会导致新的事故。

初步危险分析工作的分析表格的内容和格式可以根据分析要求的不同而不同。以下是一个初步危险分析示例：

表 13 初步危险分析

系统：				PHA			分析人员： 日期：年 月 日
NO	危险	危险描述	阶段	初步危险评估	危险原因	初步危险控制	验证方法
1	小车不受系统控制	导致系统小车撞车、翻车以及设备损坏等	任务执行阶段	高	行进装置机械故障、采集的数据未经校验	采用高可靠的设备、优化算法对数据进行校验	测试和进行进一步分析
…	……	……	……	……	……	……	……

4.1.3. 功能危险分析（FHA）

功能危险分析是系统地、综合地按层次检查产品的各种功能，以确定不但发生故障时，而且在其正常工作时可能产生或促使诱发产生的潜在危险及其后果。为了安全性设计，必须确定危险状态，以便在产品整个寿命周期内消除它们或使它们得到控制 。

功能危险分析的主要目的就是发现潜在危险或突变故障模式，以控制或避免可能危险后果的发生。功能危险分析是自上而下评估系统可能失效的所有“通道”以及每个失效通道对系统和整机功能的影响。它与系统的具体构型或组成无关，因为它是从系统功能角度提出。功能危险分析给出各种危险后果评估，推导或者确认系统安全性设计准则，提出系统安全性要求，并推荐可能的控制措施，并为其它危险性分析建立框架。

功能危险分析主要包括以下步骤：

1) 确定与分析系统层次相关的所有功能及完成功能的有关环境条件。
2) 危险性说明。确定并描述各功能的失效情况或故障
3) 确定该失效情况或故障出现的飞行阶段
4) 确定失效情况对其它系统产生的影响，该现象的出现导致其它系统的失效或故障
5) 确定危险情况。失效情况对飞机或人员的影响
6) 确定影响等级。根据故障情况对飞机的影响进行分类
7) 提出进一步分析的方法或技术建议

功能危险分析工作的核心是填写分析表格。分析表格的内容和格式可以根据分析要

求的不同而不同。以下是一个功能危险分析分析表格示例：

表 14　功能危险分析示例

功能编号	功能	失效说明（危险描述）	工作状态	对发动机影响	等级	措施
1	冷运转逻辑控制	提前结束或无法自动停止	地面	冷运转失败	4	无
……	……	……	……	……	……	……

FHA分析过程中注意要点：

1. FHA分析中最常见的错误是分析时过多地涉及系统的的硬件机构。这将导致FMEA式的思路，而不适合FHA。如果FHA要通过评估功能故障状态的严重性而有效的制订出设计准则的话，则它不允许成为具体设计的反应。具体设计的评估由后续的FMEA、FTA等完成。因此，FMEA式的思路和分析不能出现在FHA中。
2. 在功能还未全面列出前就开始分析风险和功能故障，很容易导致FHA难以分析全面。保证FHA分析完整的唯一途径就是采用系统的分析流程，首先建立所分析系统完成的所有功能的完整清单，然后随着分析的进展而不断补充完善。
3. FHA分析的结论应采用系统的方式编辑整理，以便随后审核。
4. 必需防止以风险是极不可能或仅能由多重故障或极小情况引起为由而不予考虑。FHA不考虑故障状态的发生概率，只假定故障状态并评估其严重性，以便确定其最大允许概率。
5. 后备或警告功能的考虑。在风险分析中单独考虑后备或警告功能是较困难的。因为通常一个后备或警告功能自身的功能丧失是4类事件，除非基本功能出现故障，否则后后备或警告功能不起作用。而在风险分析中评估一个基本功能时，若明显需要后备或警告功能时，则后备或警告功能将构成基本功能设计方案的一部分。因此，这样的次要功能可不在风险分析中处理，而作为对基本功能风险的一种设计结果处理。

4.2. 自底向上的软件安全性需求获取主要分析方法介绍

此软件安全性需求分析过程可采取的主要方法包括软件运行模式分析、软件安全关键功能分析、数据流分析（DFA）、控制流分析、信息流分析及软件功能层次上的故障模式影响及危害度分析（SFMECA）等。介绍主要应用方法如下：

4.2.1. 软件安全关键功能分析

软件安全性需求通常与安全关键的软件功能有密切关系。安全关键的软件功能指一个软件功能，功能正确的执行对于减轻危险的风险是必须的；或者如果功能执行不正确，顺序不适当或没有执行可能导致危险发生，影响安全缓解功能的效果或导致安全性水平降级。

软件安全关键功能分析的目的就是确定所分析软件系统的安全关键软件功能，确定软件安全性分析的范围。软件安全性分析人员可在此基础上重点针对逐项安全关键功能开展进一步的深入分析，找出与安全关键功能相关的软件设计缺陷。

软件安全关键功能分析主要包括以下步骤：

1. 确定软件功能与顶层危险的对应关系

   结合软件需求规格说明、软件任务书等文档信息明确软件系统的功能层次，同时考虑自顶向下的软件安全性需求分析的分析结果，通过逐层分析将系统危险与具体软件功能联系起来。

2. 考虑软件安全关键功能确认准则，通常情况下满足以下准则的软件功能被认为是安全关键的。

   软件安全关键功能确认准则：

1) 处于安全关键软件或系统中；
2) 至少符合以下各项中的一项：

   a) 引起一个危险或对危险的产生有贡献
   b) 提供控制或缓解危险的功能
   c) 控制安全关键功能
   d) 一旦系统到达一个特定的危险状态，进行检测、报告或纠正
   e) 一旦危险发生缓解损害
   f) 在相似系统中认定为安全关键
   g) 处理数据和分析，可直接影响安全性决策
   h) 对安全关键系统进行全部或部分的验证，包括硬件或软件子系统

4.2.2. 软件数据流图分析

结构化分析方法源于数据处理应用，适合于数据处理类型软件的需求获取，具体说来，结构化分析方法就是用抽象模型的概念，按照软件内部数据传递、变换的关系，自

顶向下逐层分解进行分析，直到满足软件系统功能要求为止，具有较好的分割，抽象能力。八十年代初期 Page-Jones，Gane 等人提出结构化分析方法的一些变种，用于信息系统的开发，八十年代中期 Ward,Mellor,Hatiy,Pirbhai 等人对结构化分析进行扩充支持实时、控制和嵌入式系统的开发。

软件数据流图分析即是一种结构化软件需求获取方法。它采用层次结构的数据流图，按照系统的层次结构进行逐步分解，表达并分析数据处理过程的数据加工情况，以分层的数据流图反映这种结构关系，它表示了系统内部信息的流向，并表示了系统的逻辑处理的功能。

1. 数据流图特点：

   1) 数据流图描述数据流和加工

   2) 数据流图用图形符号表示数据流、加工、数据源及外部实体

   3) 数据流图具有层次结构，支持问题分析、逐步求精的分析方法

   4) 数据驱动的数据流图即既可以表示基于计算机的系统，也可以表示软件

2. 数据流图标记：

   数据流图中的各种元素如下图所示：

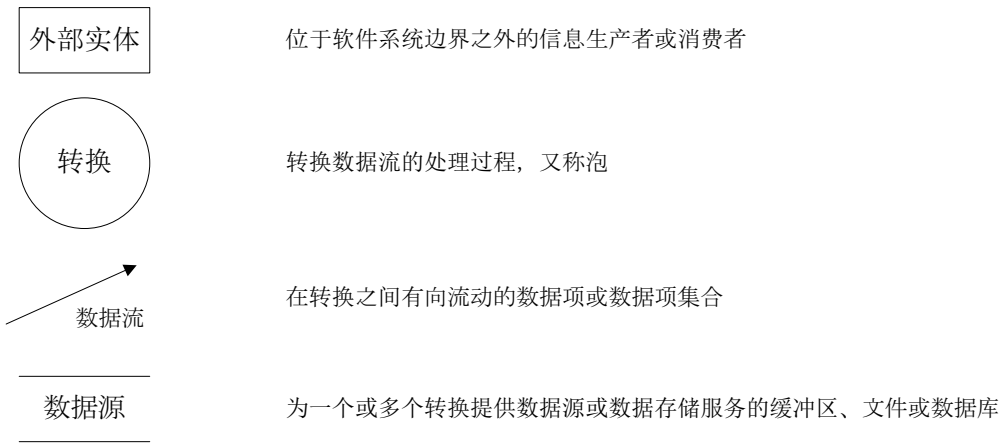

图 11　数据流图元素

3. 数据流图的构建：

   1) 各层数据流图

随着需求获取活动的深入，较高抽象级别的复杂加工逐步精化为一系列相互关联的数据流和子加工

图 12　顶层数据流图

在多层数据流图中，顶层流图仅包括一个加工，它代表被开发系统。它的输入流是该系统的输入数据，输出流是系统所输出数据。

底层流图是指其加工不需再做分解的数据流图，它处在最底层。

中间层流图则表示对其上层父图的细化。它的每一加工可能继续细化，形成子图。

2) 数据流图的精化与平衡

a) 逐层精化必须保持数据流图的平衡，数据流与加工精华必须保持一致

任何一个数据流子图必须与它上一层的一个加工对应，两者的输入数据流和输出数据流必须一致

b) 需求获取活动只求对问题全面、清晰的理解，不考虑软件设计细节

4. 数据流图的优缺点

1) 总体概念强，每一层都明确强调“干什么”，“需要什么”，“给出什么”。

2) 可以反映出数据的流向和处理过程。

3) 由于自顶向下分析，容易及早发现系统各部分的逻辑错误，也容易修正。

4) 容易与计算机处理相对照。

5) 不直观，一般都要在作业流程分析的基础上加以概括、抽象、修正来得到。

6) 如果没有计算机系统帮助的话，人工绘制太麻烦，工作量较大。

4.3. 软件安全性需求获取方法应用策略

尽管目前软件安全性需求获取工作中可以借鉴许多成熟的软件分析方法，但软件安全性需求获取工作的特点决定了我们在应用传统成熟方法的过程中既需要借鉴分析方法的传统分析思路，又要特别考虑各软件分析方法在软件安全性需求获取工作过程中的应用目的，明确各软件分析方法解决的范围和对象，使软件安全性分析人员能够更好的利用现有软件分析方法，发挥成熟方法的优势解决软件安全性需求获取工作中的问题。

表 15 软件安全性需求获取方法应用策略

编号	主要方法	分析思路	应用目的
1	初步危险分析	PHA 是针对整个系统分析的范围相当广泛，通过 PHA，可全面识别系统及其使用、运行环境中存在的危险因素、相关的危险特征和危险状态，确定可能的事故	对各种系统级危险进行分析，提出系统安全性需求，分析结果将作为其它安全性分析的基础
2	功能危险分析	功能危险分析的主要目的就是发现潜在危险或突变故障模式，以控制或避免可能危险后果的发生	从系统功能角度提出。功能危险分析给出各种危险后果评估，推导或者确认系统安全性设计准则，提出系统安全性要求，分析结果将作为其它安全性分析的基础
3	系统运行模式分析	系统运行模式是指在不同的外在环境、操作指令等条件下，系统不同的运行状态和组合方式	明确系统运行模式，可以使软件安全性分析人员针对不同的运行模式分类不同的系统功能失效，进而对软件系统进行全面的分析
4	软件故障树分析	软件故障树分析方法是源于硬件的故障树分析，与通常的故障树分析方法没有本质的区别。软件安全性分析人员可以从系统的角度评估软件失效的影响	应用软件故障树进行深入的危险原因因素分析，分析得以软件相关的危险原因，标识危险控制需求
5	软件安全关键功能分析	软件安全关键功能分析的目的就是确定所分析软件系统的安全关键软件功能，确定软件安全性分析的范围	软件安全性分析人员可在此基础上重点针对逐项安全关键功能开展进一步的深入分析，找出与安全关键功能相关的软件设计缺陷
6	数据流图分析	软件数据流图分析即是一种结构化软件需求获取方法。它采用层次结构的数据流图，按照系统的层次结构进行逐步分解表示系统内部信息的流向，并表示系统的逻辑处理的功能	通过结构化的分解工作明确软件执行功能中的关键节点，并从节点的数据输入、数据处理、数据输出三方面进行软件安全性相关考虑

4.4. 小结

本章针对软件安全性需求获取过程中应用的主要分析方法给予了详细的介绍和说明，包括自顶向下的软件安全性需求分析方法介绍及自底向上的软件安全性需求分析方法介绍两部分。其中重点阐述了初步危险分析（PHA）、功能危险分析（FHA）、数据流图分析（DFA）等目前系统安全性分析工作中经常采用及本文分析思路中选取的成熟、可操作的分析方法。

第五章 某型发动机数控系统控制软件安全性需求获取

先进飞机的航空发动机要求高推重比、低油耗、长使用寿命、大灵活性。发动机性能的充分发挥主要依靠控制系统来实现和保证。目前，航空发动机控制系统正处于从传统的机械液压式控制向数字式电子控制的转变，并且经历了从单个部件到整体、从模拟式到数字式、从有限功能到全权控制的发展过程。这也导致了发动机控制系统的复杂性不断增加。

发动机电子控制系统应用在整个飞行包线范围内，在发动机所有稳态和过渡态工作时实现各种控制功能，及参数的极限限制、防喘、消喘、故障诊断、隔离与重构、状态监视等功能，并且可与飞机系统进行通讯，传送发动机工作参数和状态信息。其中发动机控制软件是整个发动机控制系统的核心，它对完成发动机的复杂控制功能、提升发动机的控制品质、缩短发动机的研制周期都起着极其重要的作用。

目前，国内对于航空发动机数控系统的研究主要集中在发动机数学模型及模拟数字混合计算机实时仿真研究[63]、发动机故障诊断技术[64]、及先进的控制理论方面均有比较深入的研究，但在软件冗余技术、核心芯片技术、软件可靠性与安全性技术等方面与国外还存在很大的差距[65,66]。

其中在软件可靠性方面，西北工业大学樊丁等人[67]从航空发动机数控系统控制软件设计角度出发，总结了在航空发动机控制系统软件中较常发生的软件故障类型，但总结主要依赖经验积累，并没有进一步提出具体有效的分析方法。软件安全性方面，中国民航大学孙春林等人从民机适航角度出发，简述了ARP4761标准整机安全性评估过程及相应技术，但在应用于发动机控制系统时提出对于航空发动机数控系统控制软件分析，系统安全性评估可以应用但十分困难，显得力不从心[68]。

由以上分析可以发现目前在航空发动机数控系统控制软件安全性工作方面，现有的国内研究主要依赖于经验经累及照搬国外相关标准，并没有形成系统的综合的分析体系，无法满足现有航空发动机控制系统要求。因此，本文针对某型发动机数控系统控制软件开展软件安全性需求获取工作具有很强的现实意义。

5.1. 航空发动机控制系统相关介绍

5.1.1 航空发动机控制系统介绍

某型航空发动机是带矢量喷管的加力涡轮风扇发动机，由三级风扇、六级压气机、环形燃烧室、一级高压涡轮、一级低压涡轮、加力燃烧室和轴对称矢量喷管及相关系统组成。该发动机的控制系统主要由电子控制器、数控系统控制软件、传感器、液压机械装置和相应电气系统等组成。该系统主要完成主燃油控制、加力燃油控制、风扇导叶控制、压气机导叶控制、喷管喉道面积控制、轴对称矢量喷口控制、参数限制控制和发动机异常报警等功能。

控制系统工作原理图见图13：

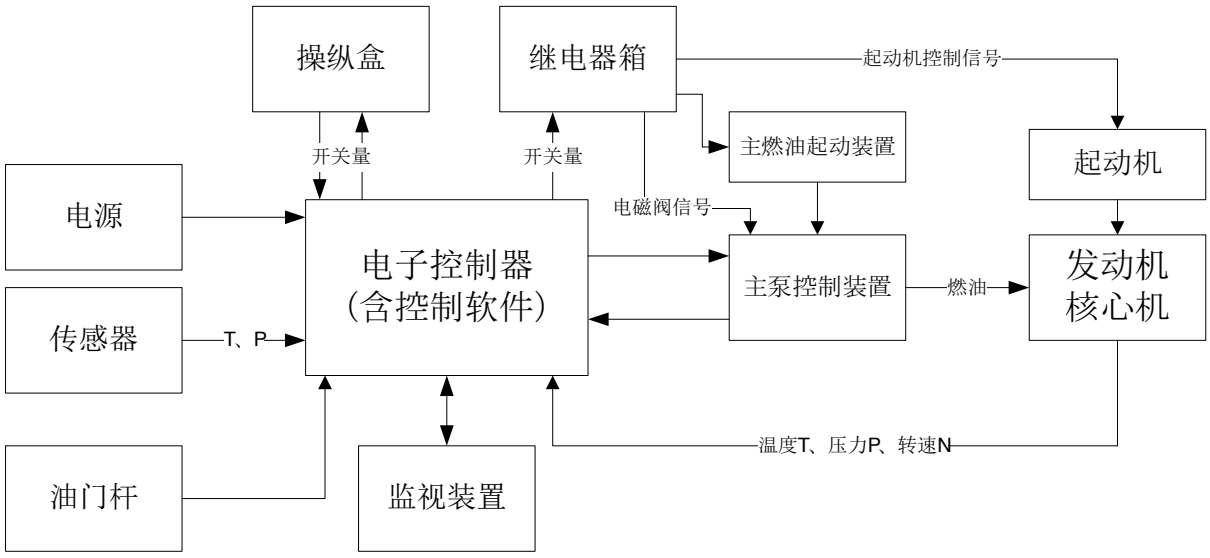

图 13　控制系统工作原理图

电子控制器接收来自操纵盒的开关量输入指令、油门杆信号、发动机进口温度、发动机进口压力、压气机出口压力、压气机转速N、涡轮出口温度T、主燃油计量活等信号，经控制器处理后，由控制软件进行采集和转换，并进行故障诊断与处理，然后进行控制逻辑处理与控制算法处理，控制逻辑处理产生的报警信号经开关量输出装置输出给操纵盒显示面板进行报警显示，控制逻辑处理产生的控制信号经开关量输出装置输出给继电器箱，然后由继电器箱输出给主泵控制装置、主燃油起动装置和起动机以控制电磁阀的开关和起动机的运转，再由起动机带动发动机起动，同时由控制软件控制算法处理得到的主燃油电液伺服阀控制信号经电子控制器DA 转换装置处理后输出给主泵控制装置，控制主泵控制装置的主燃油计量活门位移和压气机导叶作动筒位移，由主燃油计

量控制装置输出主燃油流量来控制发动机的转速，由压气机导叶控制装置驱动压气机导叶作动筒来控制压气机导叶角度，通过油门杆信号的改变来改变发动机的状态，控制系统的各个信号经过串口通讯给监视装置进行显示和记录。

5.1.2 航空发动机控制系统电子控制器介绍

发动机控制系统的核心部分电子控制器由两个功能完全相同的软件非相似双余度数控通道和一个液压机械备份通道组成。当数控通道工作时，一个通道主控，另一个通道处于热备份状态，当主控通道发生故障时，能自动无扰动地切换到热备份通道工作；当数控双通道均失效后转为液压机械备份通道。其原理如图所示：

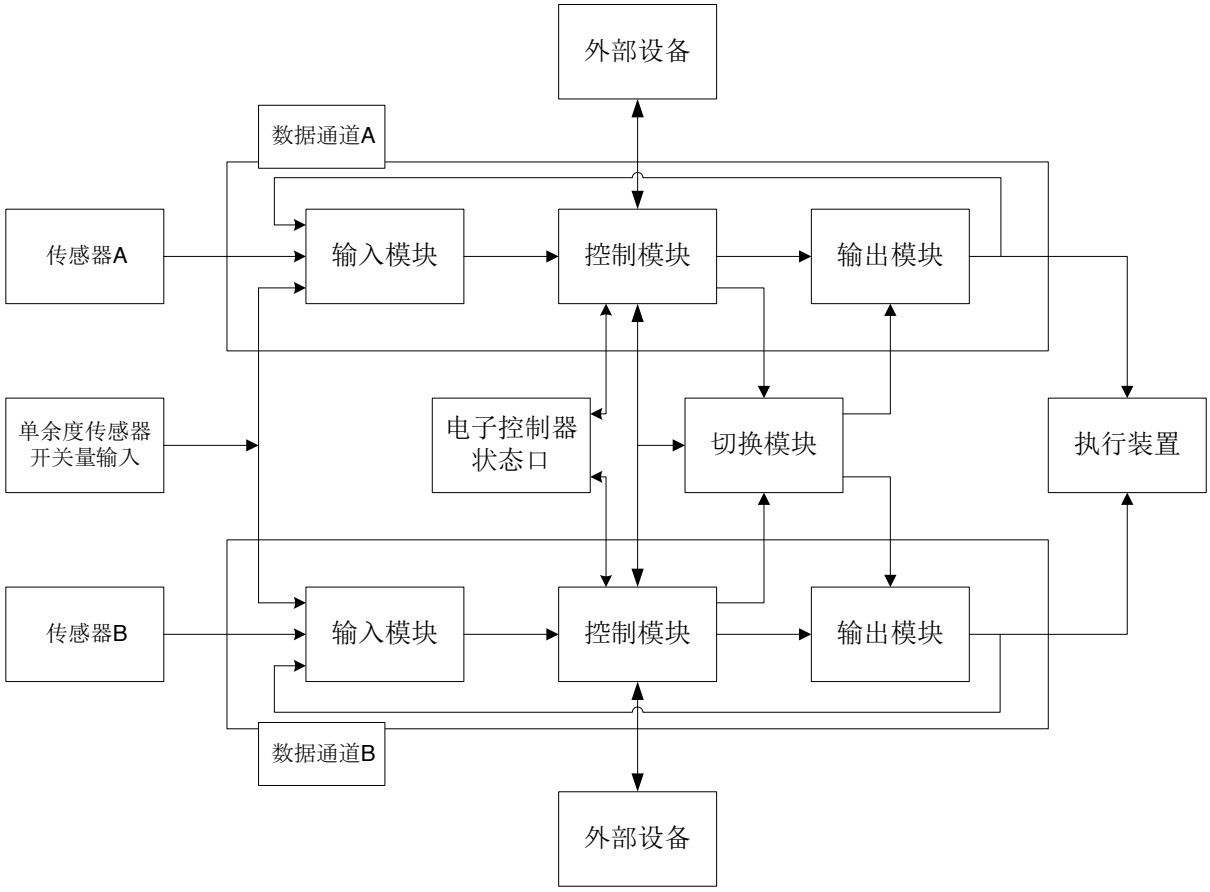

图 14 电子控制器工作原理图

模拟量传感器信号由模拟量输入信号处理电路进行处理，并同时进行硬件BIT 检测，BIT 检测信号由模拟量输入信号处理电路进行处理，压气机转速传感器信号由频率量输入信号处理电路进行处理，处理后的模拟量和频率量信号采集后输入到CPU 指定的I/O 空间，供控制程序读取。开关量输入信号和开关量输出BIT 检测信号经开关量输入信号处理电路处理后，直接送到CPU 指定的I/O 空间，供控制程序读取。控制程序根

据输入信号进行逻辑和算术运算，得到的开关量输出信号经开关量输出信号处理电路处理后输出开关量信号给相应电磁阀或指示灯，得到的主燃油、压气机导叶电液伺服阀控制电流信号经模拟量输出信号处理电路处理后输出控制电流给相应的电液伺服阀，以实现发动机的电气检测、起动、停车、加减速、稳态、加力控制，对转速、温度和压力进行限制保护的功能，同时传输信号给飞机机载装置，用于显示、告警和记录。

5.2. 软件安全性需求获取

针对此航空发动机控制系统控制软件，本文从通用软件安全性需求及特定软件安全性需求两部分进行安全性需求获取工作。

5.2.1 通用软件安全性需求获取

在进行通用软件安全性需求获取工作前，软件分析人员通过对系统概要设计、软件产品规格说明、软件开发计划、软件任务书等文档资料的理解和与系统设计人员，软件开发人员的沟通，明确了数控系统控制软件的模块组成、外部接口及实现功能。

经初步分析，数控系统控制软件的主要功能包括以下三个方面：

1. 初始化功能

   完成电子控制器硬、软件初始化，上电自检测功能

2. 实时控制功能

   包括信号采集、信号处理、状态监控、控制算法处理、通讯等功能

3. 非实时控制功能

   包括飞行前机内自检、维护功能等

5.2.1.1 通用软件安全性需求清单剪裁

由软件主要实现功能及自身特点，通用软件安全性需求的剪裁主要从初始化、危险命令及软件自身功能相关方面考虑。

初步剪裁得到的通用软件安全性需求如下表 16 所示：

表 16 初步剪裁得到的通用软件安全性需求

类别	编号	描述
初始化	1	软件必须设计成在上电时进行系统级检查，以便在对安全关键功能包括由软件控制的硬件通电之前验证该系统是安全的并正确的运行。必须用软件进行定期测试以监视系统的安全状态
	2	软件（包括固件）的上电自检利用到的任何可替换的单元或者组件必须只能用于这个单独系统的处理过程
危险命令	3	在安全执行被认为是危险命令之前，必须满足执行的先决条件（正确的模式、正确的配置、组件可用、合适的顺序和参数在范围之内）
	4	撤销或取消命令需要经过多个操作步骤
软件自身功能	5	在硬件失效时，软件故障引起系统失效，或者软件探测到配置和当前的运行状态不一致时，软件必须有能力将系统置于安全状态
	6	软件必须提供支持安全关键功能的出错处理
其它	7	系统必须设计成在某个安全状态中上电

5.2.1.2 通用软件安全性需求评估

针对初步剪裁得到的通用软件安全性需求，分析人员通过考虑考虑需求实现的技术可行性、费用分析、时间分析等方面，与软件设计人员、开发人员协商，结合现有系统的设计约束，逐条对剪裁出的软件安全性需求进行评估，确实软件安全性需求的优先级。

评估后的通用软件安全性需求如下表 17 所示：

表 17 评估后的通用软件安全性需求

类别	编号	描述	优先级	理由
初始化	1	软件必须设计成在上电时进行系统级检查，以便在对安全关键功能包括由软件控制的硬件通电之前验证该系统是安全的并正确的运行。必须用软件进行定期测试以监视系统的安全状态	条件的	考虑到航空发动机控制系统的复杂程度，传感器数量众多，控制软件容量限制，只对于重要传感器的故障判断，进行多周期的确认
	2	软件（包括固件）的上电自检利用到的任何可替换的单元或者组件必须只能用于这个单独系统的处理过程	不适用	考虑到系统设计的限制，部分传感器如发动机舱压传感器，涡轮总压传感器等设计为单余度传感器，供双通道共同使用
危险命令	3	在安全执行被认为是危险命令之前，必须满足执行的先决条件（正确的模式、正确的配置、组件可用、合适的顺序和参数在范围之内）	适用的	

	4	撤销或取消命令需要经过多个操作步骤	不适用	在系统正常情况下，对操作员的操作是无约束的。通过对操作员进行发动机控制专业知识及相关飞行知识的培训解决
软件自身功能	5	在硬件失效时，软件故障引起系统失效，或者软件探测到配置和当前的运行状态不一致时，软件必须有能力将系统置于安全状态	适用的	
	6	软件必须提供支持安全关键功能的出错处理	适用的	
其它	7	系统必须设计成在某个安全状态中上电	适用的	

5.2.1.3 确定通用软件安全性需求

由通用软件安全性需求评估的结果，最终确定采纳的软件安全性需求如下：

表 18 最终采纳的软件安全性需求

	编号	描述	补充
基本的	1	在安全执行被认为是危险命令之前，必须满足执行的先决条件（正确的模式、正确的配置、组件可用、合适的顺序和参数在范围之内）	
	2	在硬件失效时，软件故障引起系统失效，或者软件探测到配置和当前的运行状态不一致时，软件必须有能力将系统置于安全状态	
	3	软件必须提供支持安全关键功能的出错处理	
	4	系统必须设计成在某个安全状态中上电	
条件的	5	软件必须设计成在上电时进行系统级检查，以便在对安全关键功能包括由软件控制的硬件通电之前验证该系统是安全的并正确的运行。必须用软件进行定期测试以监视系统的安全状态	只针对重要传感器

5.2.2 特定软件安全性需求获取

考虑到安全关键软件的复杂性，仅仅依靠通用软件安全性需求无法保证软件安全性，因此针对数控系统控制软件的特定软件安全性需求分析十分重要。特定软件安全性需求获取过程主要包括以下两个环节：

5.2.2.1 自顶向下的数控系统控制软件安全性需求分析

1. 系统运行模式分析

结合系统任务书，系统概要设计，系统需求规格说明书等文档信息，与系统设计人员进行沟通，对此航空发动机控制系统开展系统运行模式分析，得到系统运行模式如下：

表 19 发动机系统运行模式

编号	系统运行模式	运行子模式
1	发动机地面起动模式	冷运转模式
		油封/启封模式
		假开车模式
		起动模式
2	发动机工作模式	慢车模式
		节流模式
		中间模式
3	发动机工作过渡模式	加速模式
		减速模式

2. 系统级危险分析

a) 确定初步的系统危险清单

表 20 初步危险清单

编号	危险状态
1	发动机悬停
2	发动机空中停车
3	发动机超温
4	发动机喘振
5	发动机超转
6	发动机超压
7	发动机空中起动不成功
8	发动机失速
9	起动失败
10	发动机推力反向
11	发动机失去 N2 机械限转功能
12	部分功能失效
13	发动机控制出错
14	发动机失去座舱显示与告警功能
15	发动机失去燃油监视功能
16	发动机失去振动监视功能
17	发动机失去滑油监视功能
18	发动机无法提供反推力

b) 确定系统功能清单

表 21 发动机数控系统功能清单

顶层功能	展开一层	展开二层	功能编号
发动机起动逻辑控制	地面起动逻辑控制	冷运转逻辑控制	0101
		油封/启封逻辑控制	0102
		假开车逻辑控制	0103
		起动逻辑控制	0104
	空中起动逻辑控制	油门杆空中起动逻辑控制	0105
		起动机辅助的风车起动	0106
		空中点火电门起动	0107
		惯性自动起动	0108
发动机燃油流量控制	起动过程燃油控制	点火供油控制	0201
		起动加速供油控制	0202
	发动机燃油流量控制		0203
	加减速供油控制	加速供油控制	0204
		减速供油控制	0205
停车控制			0301
VBV 控制			0401
VSV 控制			0501
限制控制			0601
	高压转子转速限制		0602
	发动机排气温度限制		0603
	高压压气机后压力限制		0604
	低压转子换算转速限制		0605
主动间隙控制	五级引气控制		0701
	九级引气控制		0702
反推力控制			0801
对滑油系统的冷却功能	对发动机滑油系统的冷却		0901
	对飞机交流发电机滑油的冷却		0902
对发动机 N2 转速进行机械限转			1001
故障诊断与处理			1101
发动机状态监控	座舱显示与告警信号		1201
	燃油监视		1202
	振动监视		1203
	滑油监视		1204
数据储存			1301
与其他系统的通讯			1401

c) 进行系统级 FHA 分析

表 22　发动机数控系统系统级 FHA

功能编号	功能	失效说明（危险描述）	工作状态	对发动机影响	等级	控制或处置情况
0101	冷运转逻辑控制	提前结束或无法自动停止	地面	冷运转失败	4	无
0102	油封/启封逻辑控制	提前结束或无法自动停止	地面	油封/启封失败	4	无
0103	假开车逻辑控制	提前结束或无法自动停止	地面	假开车失败	4	无
0104	起动逻辑控制	不点火	地面	无	4	无
		点火系统延迟	地面	爆燃	4	无
		供油时序异常	地面	起动失败	4	无
		点火器未断开	地面	点火系统烧坏	2	告警，设计采取措施
		起动机未正常终止工作	地面	无	2	停车
		发动机超温而未终止起动	地面	发动机超温	2	停车
0105	油门杆空中起动逻辑控制	不点火	飞行中	空中起动不成功	1	设计采取措施
		起动不供油	飞行中	空中起动不成功	1	设计采取措施
		起动超温	飞行中	空中起动不成功	1	设计采取措施
0106	起动机辅助的风车起动	不点火	飞行中	空中起动不成功	1	设计采取措施
		起动不供油	飞行中	空中起动不成功	1	设计采取措施
		起动超温	飞行中	空中起动不成功	1	设计采取措施
		无法接通空气涡轮起动机	飞行中	空中起动不成功	1	设计采取措施
0107	空中点火电门起动	起动不供油	飞行中	空中起动不成功	1	设计采取措施
		起动超温	飞行中	空中起动不成功	1	设计采取措施
0108	惯性自动起动	不点火	飞行中	空中起动不成功	1	设计采取措施
		起动不供油	飞行中	空中起动不成功	1	设计采取措施
		起动超温	飞行中	空中起动不成功	1	设计采取措施
0201	点火供油控制	供油多	地面	起动失败或爆燃	4	设计采取措施
			飞行中	空中起动不成功	1	设计采取措施
		供油少	地面	起动失败	4	设计采取措施

			飞行中	空中起动不成功	1	设计采取措施
0202	起动加速供油控制	供油多	地面	发动机失速或起动超温	3	设计采取措施
			飞行中	发动机失速或起动超温	1	设计采取措施
		供油少	地面	发动机悬停	4	设计采取措施
			飞行中	发动机悬停	1	设计采取措施
0203	发动机燃油流量控制	转速摆动	地面、起飞、飞行中、着落	推力脉动、发动机可能出现间歇超温	3	设计采取措施
		转速静差	地面、起飞、飞行中、着落	推力提供不合适、发动机可能不稳定工作（达不到慢车）	3	设计采取措施
		燃油流量失控（向大的方向移动）	地面、起飞、飞行中、着落	推力失控、发动机超温	1	停车
		燃油流量失控（向小的方向移动）	地面、起飞、飞行中、着落	推力失控、空中停车	1	停车
0204	加速供油控制	供油多	地面、起飞、飞行中、着落	转速超调、发动机超温、喘振	2	设计采取措施
		供油少	地面、起飞、飞行中、着落	加速时间长或者转速悬挂	2	设计采取措施
0205	减速供油控制	供油多	地面、起飞、飞行中、着落	减速时间长、转速悬挂	2	设计采取措施
		供油少	地面、起飞、飞行中、着落	转速超调、空中停车	3	设计采取措施
0301	停车控制	失效	地面、起飞、飞行中、着落	发动机无法正常停车	1	人工控制保证自动停车
0401	VBV 控制	失控使 VBV 在全开位置	地面、起飞、飞行中、着落	无	4	人工控制保证 VBV 在全关位置

		失控使 VBV 在全关位置	地面、起飞、飞行中、着落	发动机喘振	1	人工控制保证 VBV 在全开位置
		摆动	地面、起飞、飞行中、着落	发动机喘振	1	设计采取措施
0501	VSV 控制	失控使 VSV 在全开位置	地面、起飞、飞行中、着落	发动机喘振	1	人工控制保证 VSV 在全关位置
		失控使 VSV 在全关位置	地面、起飞、飞行中、着落	影响发动机的推力	2	人工控制保证 VSV 在全开位置
		摆动	地面、起飞、飞行中、着落	发动机喘振	1	设计采取措施
		控制滞后	地面、起飞、飞行中、着落	减速过程发动机喘振	1	设计采取措施
		静差	地面、起飞、飞行中、着落	发动机喘振	1	设计采取措施
0601	限制控制	失控	地面、起飞、飞行中、着落	发动机超转	2	报警，软件采取相应措施
0602	高压转子转速限制	失控	地面、起飞、飞行中、着落	发动机超转	2	机械液压式转速限制器起作用
0603	发动机排气温度限制	失控	地面、起飞、飞行中、着落	发动机超温	2	报警，软件采取相应措施
0604	高压压气机后压力限制	失控	地面、起飞、飞行中、着落	发动机超压	2	报警，软件采取相应措施
0605	低压转子换算转速限制	失控	地面、起飞、飞行中、着落	发动机超转	2	报警，软件采取相应措施
0701	五级引气控制	失控使五级引气位于全开位置	地面、起飞、飞行中、着落	有可能使发动机涡轮叶片与机匣碰磨	2	设计采取措施
0702	九级引气控制	失控使九级引气位于全关位置	地面、起飞、飞行中、着落	有可能使发动机涡轮叶片与机匣碰磨	2	设计采取措施

0801	反推力控制	反推力意外打开	地面、起飞、飞行中、着落	发动机推力反向	1	设计采取措施
		反推力控制失效	着陆	发动机无法提供反推力	3	设计采取措施
0901	对发动机滑油系统的冷却	失效	地面、起飞、飞行中、着落	发动机滑油温度上升	3	报警
0902	对飞机交流发电机滑油的冷却	失效	地面、起飞、飞行中、着落	交流发电机无法正常工作	3	报警
1001	对发动机 N2 转速进行机械限转	失效	地面、起飞、飞行中、着落	发动机失去 N2 机械限转功能	1	停车
1101	故障诊断与处理	误判	地面、起飞、飞行中、着落	部分功能失效	2	降工作状态
				发动机停车	2	降工作状态
		漏判	地面、起飞、飞行中、着落	发动机控制可能出错	2	设计采取措施
1201	座舱显示与告警信号	失效	地面、起飞、飞行中、着落	发动机失去座舱显示与告警功能	3	报告故障信息
1202	燃油监视	失效	地面、起飞、飞行中、着落	发动机失去燃油监视功能	4	报告故障信息
1203	振动监视	失效	地面、起飞、飞行中、着落	发动机失去振动监视功能	4	报告故障信息
1204	滑油监视	失效	地面、起飞、飞行中、着落	发动机失去滑油监视功能	4	报告故障信息
1301	数据储存	失效	地面、起飞、飞行中、着落	无	4	报告故障信息
1401	与其他系统的通讯	失效	地面、起飞、飞行中、着落	无	3	报告通讯故障

3. 软件故障树(SFTA)分析

进行初步的系统危险分析之后，软件安全性分析人员对整个数控系统相关的危险模式，各功能对应的失效情况有了初步了解。随后软件安全性分析人员针对安全性等级高的危险，考虑系统初步的安全性设计要求，软件输入、软件运行环境等因素，进行深入的危险原因因素分析。危险原因可能由硬件（或硬件部件）、软件输入（或没有软件输入）、人员错误等产生。危险可以由于某个特定原因产生，也可以由许多原因的任何组合产生。无论何种原因，软件安全性分析人员必须为软件设计和开发人员标识和定义危险控制需求，以影响软件概要设计活动。

考虑控制系统控制软件的实现功能及功能失效的影响范围，在这里我们主要选择发动机反推力功能失效、发动机喘振两个危险状态进行进一步的软件故障树分析工作。

a) 发动机反推力失效软件故障树分析

反推力装置是一种改变发动机推力方向的装置，主要起减速和缩短滑跑距离的作用。在军用方面，不仅能缩短飞机着陆滑跑距离，而且能大大提高飞机的作战效能；在民用方面，对运输机更具有较高的经济价值和实用价值，被公认为是现代和未来高性能运输飞机必不可少的常设装置。

针对数控系统数控软件反推力功能失效的软件故障树分析如图 15 所示，其中深色代表软件相关原因因素：

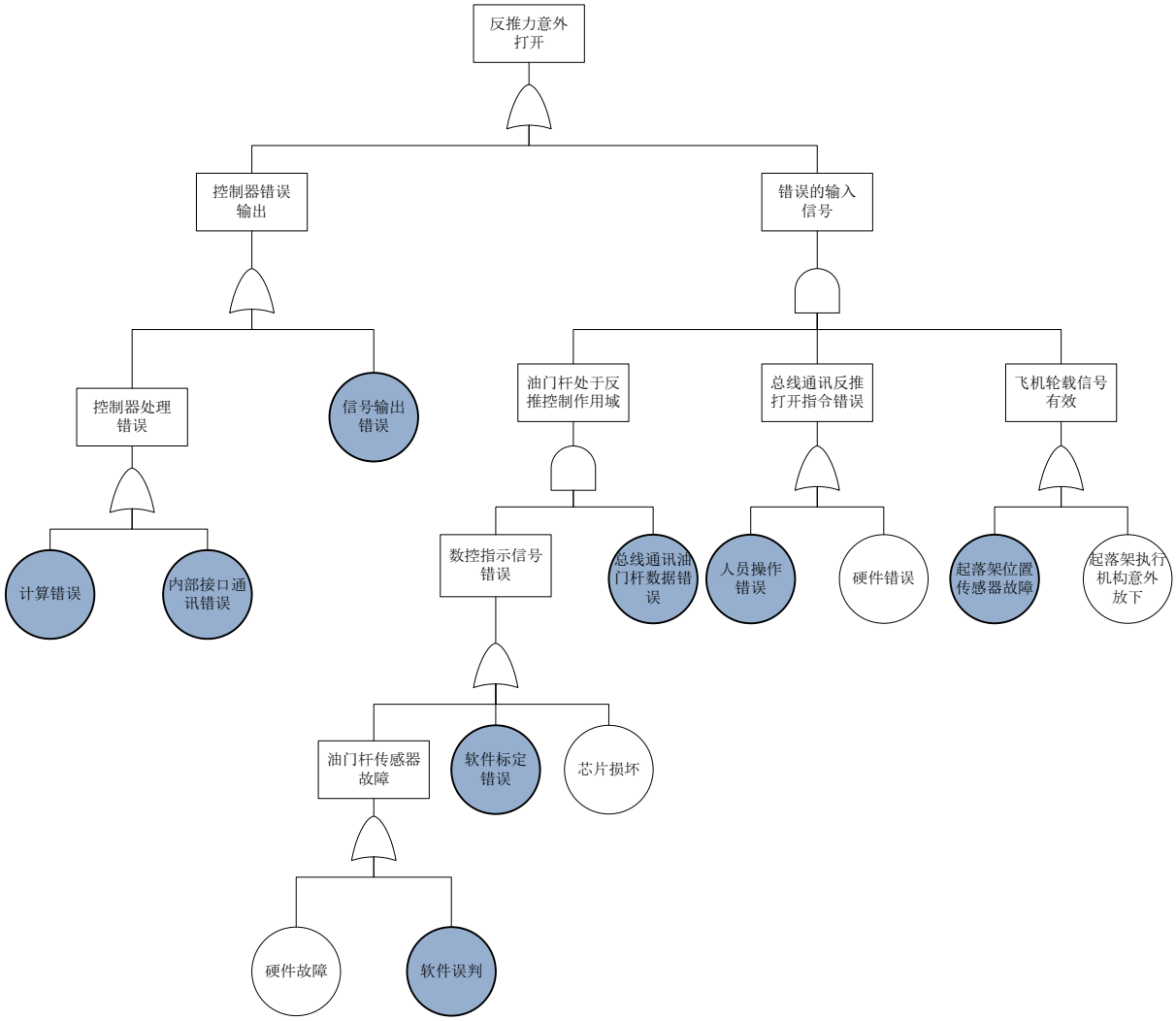

图 15　反推力功能失效软件故障树

针对以上分析结果，软件安全性分析人员应结合数控软件特点，具体明确原因的具体情况，标识和定义危险控制需求。

由此分析得到数控软件特定安全性需求如下所示：

表 23　数控软件特定安全性需求

编号	原因因素	危险控制需求
1	计算错误	软件算法确定前应进行确认；软件处理输入输出数据类型应保持一致；特征值初始化和设置应进行验证；
2	内部接口通讯错误	软件内部接口参数应集中声明定义；应明确内部程序调用关系；
3	信号输出错误	软件应能过滤重复输出的信号；在一个周期任务完成前，软件应能保存输出信号，防止输出信号丢失
4	软件误判	软件判断传感器故障情况时应进行多周期确认
5	软件标定错误	明确软件输出信号与外部控制设备对应关系；指示信号更

		新响应时间在规定范围内；
6	总线通讯油门杆数据错误	软件判断传感器故障情况时应进行多周期确认；软件应防止读取错误的外部存储区；软件应能处理油门杆快速的状态改变
7	人员操作错误	软件应能过滤错误操作；操作错误时软件应给出明显的警告信号
8	起落架位置传感器故障	软件判断传感器故障情况时应进行多周期确认；软件应防止读取错误的外部存储区；软件应具有数据溢出保护功能

b) 发动机喘振软件故障树分析

发动机喘振软件故障树分析如下图 16 所示，其中深色代表软件相关原因因素：

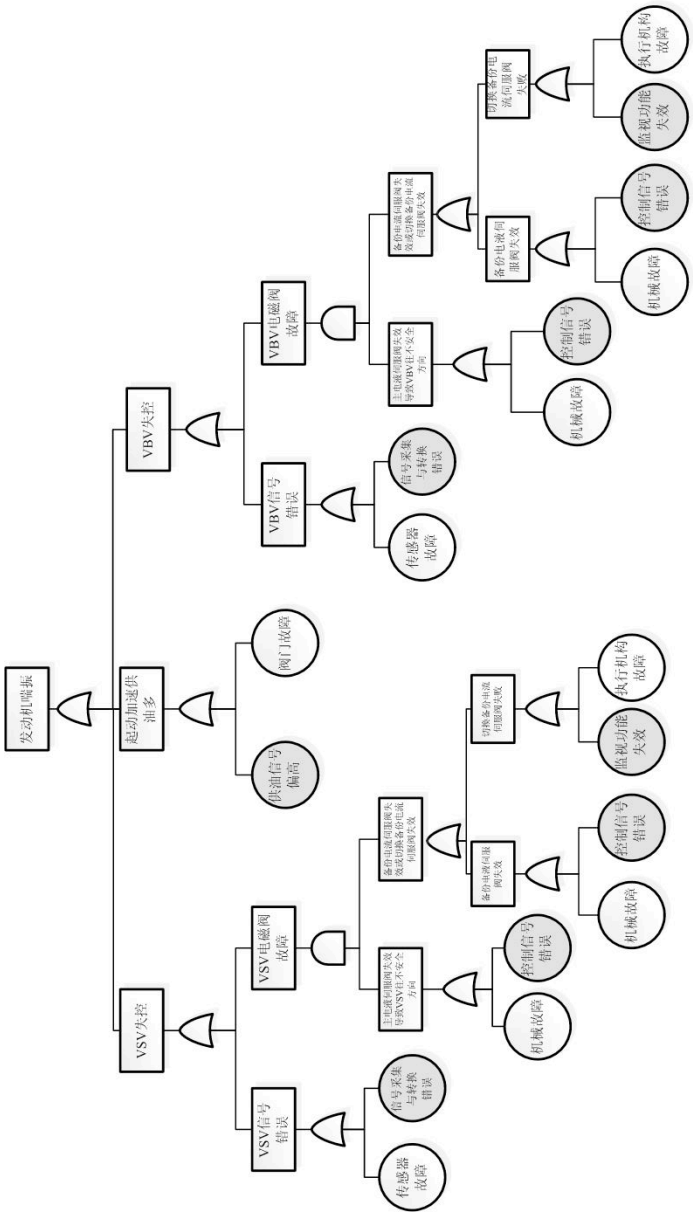

图 16　发动机喘振软件故障树分析

根据以上软件故障树分析的结果，分析得到数控软件特定安全性需求如下所示：

表 24　数控软件特定安全性需求

编号	原因因素	危险控制需求
1	信号采集与转换错误	软件判断传感器故障情况时应进行多周期确认；软件应防止读取错误的外部存储区；
2	控制信号错误	软件应能过滤重复输出的信号；在一个周期任务完成前，软件应能保存输出信号，防止输出信号丢失
3	监视功能失效	软件部分模块出现故障情况时，软件应自动屏蔽此模块，保护故障前数据
4	供油信号偏高	软件供油计算模型需进行验证；

5.2.2.2　自底向上的数控系统控制软件安全性需求分析

随着软件开发过程的进行，在软件需求阶段软件需求规格说明，软件接口需求规格说明等文档的建立使软件安全性分析人员可以进行更为深入的安全性分析，通过建立软件模型，从软件功能设计角度进行软件对系统安全性的影响分析。

1. 软件运行模式分析

根据软件需求规格说明书的定义，此航空发动机数字控制系统软件的运行模式如下所示：

1)　5ms 控制任务

主要包括输出控制回路反馈信号采集

2)　25ms 控制任务

主要包括部分模拟量信号采集、开关量采集、信号处理、故障诊断与处理、控制算法选择与计算

3)　通讯任务

完成双通道通讯及串口通讯功能

2. 确定软件接口关系

发动机控制软件通过各种信号输入、输出通道与外部形成接口，主要的信号包括：模拟量输入信号、开关量输入信号、模拟量输出信号、开关量输出信号及通道间通讯和与监视设备之间的通讯，软件接口关系如下图 17 所示：

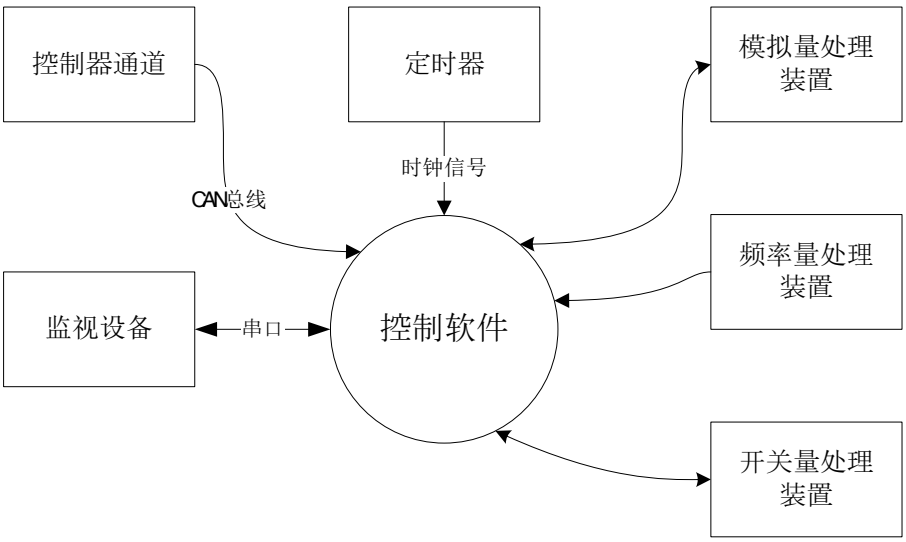

图 17　软件外部接口关系图

3. 软件功能模型建立

软件功能建模的思想就是运用抽象模型的概念，按照软件内部数据传递、变换的关系，自顶向下逐层分解，将外部需求赋予软件的各个功能部分，定义软件的内部功能，并标定它们之间的接口关系，同时包括对模型内部数据间的限制。在这里选择采用数据流图建立软件功能模型。

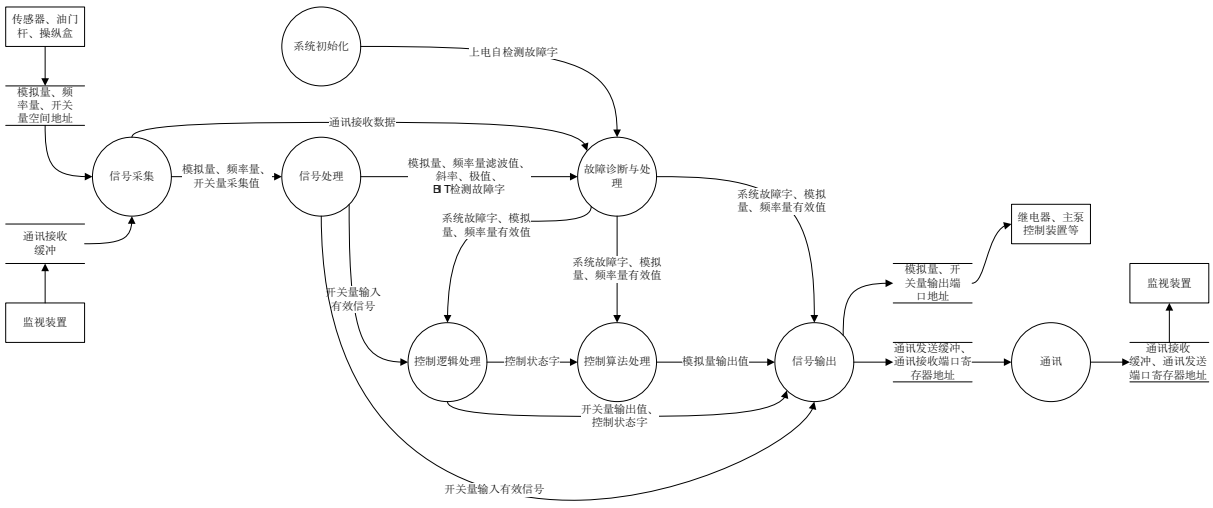

图 18　发动机数控系统控制软件功能模型

4. 确定软件安全关键功能

通过前述自顶向下的软件性需求分析过程，安全性分析人员可以通过系统功能的逐层分析，深入确定软件功能与顶层危险的对应关系，同时根据软件安全关键功能的确定准则，结合软件需求规格说明、软件任务书等文档信息，软件安全性分析人员可确定此数控系统数控软件的安全关键功能如下：

- 发动机起动逻辑控制
- 发动机故障诊断与处理
- 发动机可调放气活门（VBV）控制
- 发动机冷却水温温控控制阀（VSV）控制
- 发动机反推力控制

5. 数控软件数据流图分析

针对以上安全关键功能，软件安全性分析人员应从软件功能模型出发，结合软件需求规格说明、软件接口需求规格说明确定的约束，采用基于数据流图的软件需求结构化分析方法，考虑安全关键功能的实现路径，重点关注安全关键功能有关的数据传递、变换关系，进行软件安全性需求获取工作，识别软件功能设计缺陷并提出相应的危险控制考虑

发动机故障诊断就是借助一定的有效方式对与发动机各系统紧密相关的各种参数实施监测，根据监测的数据对各系统的工作状态及其发展趋势做出有价值的判断，即做出故障诊断结论。以下为针对故障诊断与处理功能模块进行的软件数据流图分析：

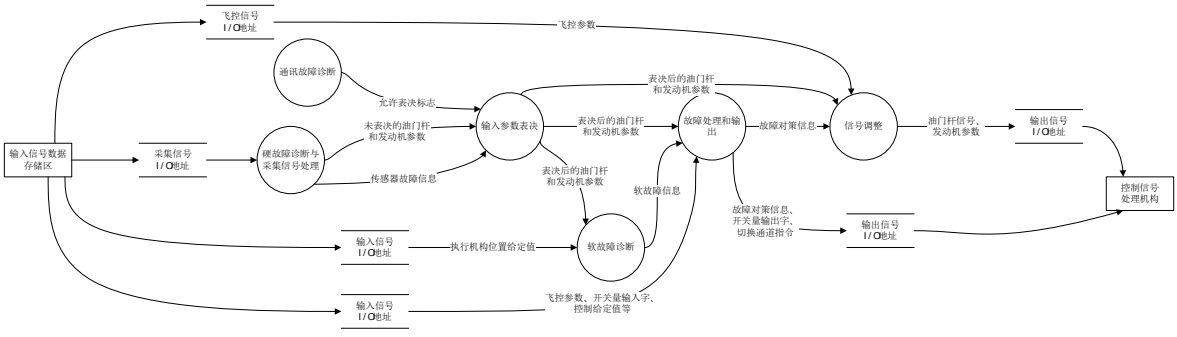

图 19　故障诊断与处理功能模块一层数据流图

结合软件需求规格说明与故障诊断与处理功能数据流图的描述，此数控系统数控软件故障诊断与处理模块中故障诊断功能分为硬故障诊断与软故障诊断两类。软件通过故障检测来判断信号是否存在故障，若有故障进行故障诊断并通过故障处理功能进行报警或给出控制指令使系统在故障状态仍能进行稳定控制。

明确故障诊断与处模块的主要功能后，软件安全性分析人员重点从硬故障诊断、软故障诊断及故障处理和输出三个结点的输入输出进行数据流图安全性分析。

a) 硬故障诊断

硬故障诊断即将直接采集到的发动机数据进行故障检测，包括斜率判断、极值判断等。

软件安全性分析如下表 25 所示：

表 25 硬故障诊断软件安全性分析

软件主要实现方式	安全性相关考虑	确定软件安全性需求
采集值处于门槛值外则认为传感器在本采样周期内出现故障	未考虑数据丢失情况；未考虑读取错误类型数据的情况	软件应考虑输入数据为空或输入数据类型错误的处理准则
采样值与上一正确周期采样值差的绝对值大于最大差额，则认为采样值出现异常	未考虑数据丢失情况；未考虑读取错误类型数据的情况	
存在极值硬故障字时不进行斜率判断	软件需求未明两种故障诊断方式的优先级；	软件应当明确两种判断模式的执行顺序
处理信号出现故障的累计次数达到 3 次后，软件认为该传感器通道出现故障，置硬故障字	未明确置故障字后计数器运行模式，可能数据溢出；未明确计数器累计次数连续或不连续，对间歇性故障处理可能会有问题	在程序中使用计数器应考虑状态变化时计数器是否清零，以防止重新进入该状态时对计数的影响
故障消失累计达到 3 次后认为故障取消，消去故障字	未明确置故障字后计数器运行模式，可能数据溢出；	

b) 软故障诊断

软故障诊断部分二层数据流图如下所示：

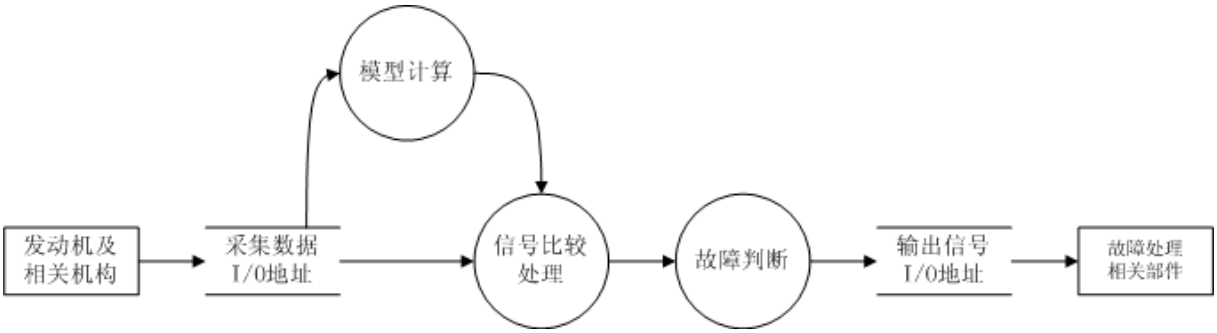

图 20　软故障诊断二层数据流图

软故障诊断通过模型参数对比的方式进行检测，将某一采集的数据作为简易实时发动机模型的输入，通过模型计算，得出一系列某发动机状态的估计参数，然后将实际采集到的数据与估计参数进行对比来进行参数的故障检测。

软件安全性分析如下表 26 所示：

表 26　软故障诊断软件安全性分析

软件主要实现方式	安全性相关考虑	确定软件安全性需求
实测值与给定值差的绝对值超出允许值则认为有故障	未考虑数据丢失情况；未考虑读取错误类型数据的情况	软件应考虑输入数据为空或输入数据类型错误的处理准则
与软故障诊断相关的传感器发生硬故障，则不进行软故障诊断	未考虑软故障诊断确认故障而硬故障诊断正常的情况	软件应明确软故障诊断确认故障而硬故障诊断正常时的处理情况
数据计算依据简化的离散化方程式	未明确规定算法的精度	软件应明确规定算法的精度、迭代次数等

c)　故障处理和输出

故障处理和输出部分根据故障判断准则，判定数控系统是否处于完好状态，判定转换后的采样信号是否故障并采取相应措施

软件安全性分析如下表 27 所示：

表 27　故障处理和输出软件安全性分析

软件主要实现方式	安全性相关考虑	确定软件安全性需求
本通道出现故障的传感器较多而对方通道较少时，切换至对方通道；对故障处理准则未规定的硬件故障，只要本通道出现故障则直接切换至对方通道	对切换情况考虑不全面，可能会出现切换失效	软件应明确各种切换条件的优先级
故障处理按故障处理准则进行	故障处理准则中未明确优先级	软件应明确不同故障处理时的顺序

5.2.2.3 确定特定软件安全性需求

经过自顶向下及自底向上的数控软件安全性需求分析，总结分析结果，最终得到如下软件安全性需求：

表 28 最终得到的软件安全性需求

编号	软件安全性需求描述
1	软件判断传感器故障情况时应进行多周期确认；
2	软件应防止读取错误的外部存储区；
3	软件应考虑输入数据为空或输入数据类型错误的处理准则
4	软件应能过滤重复输出的信号；
5	在一个周期任务完成前，软件应能保存输出信号，防止输出信号丢失
6	在程序中使用计数器应考虑状态变化时计数器是否清零，以防止重新进入该状态时对计数的影响
7	软件部分模块出现故障情况时，软件应自动屏蔽此模块，保护故障前数据
8	软件应明确各种切换条件的优先级
9	软件应明确不同故障处理时的顺序
10	软件应当明确两种判断模式的执行顺序
11	软件应明确软故障诊断确认故障而硬故障诊断正常时的处理情况
12	软件应明确规定算法的精度、迭代次数等

5.3. 小结

本章介绍了软件安全性需求获取方法在实例中的应用情况。首先对安全性需求获取工作的分析对象——某型航空发动机控制系统及其中的电子控制器部分的工作原理、结构等方面进行了简要介绍。然后应用前文提出的软件安全性需求获取方法进行了具体的软件安全性需求获取工作，其间包括通用软件安全性需求获取及特定软件安全性需求获取两部分。最后通过分析过程获得的软件安全性需求被证实对此发动机控制系统数控软件安全性隐患的消除起到了积极作用。从此实例中我们证实了应用此全面考虑、综合多种分析技术的软件安全性需求获取方法能够弥补单一技术的缺陷，使得分析过程更全面、更准确，同时分析方法应用时思路明确，可以较好的减少人为遗漏并且效率较高。工程实践过程是一个不断积累经验的过程，因此，我们还需要大量的案例补充进来。在今后的实践中，我们要充分依据软件开发过程的特色，灵活应用各种软件安全性需求分析技术更好地达到我们所期望的目标。

结论与展望

本文的工作

本论文围绕软件安全性需求获取方法这个核心研究内容，对软件安全性需求分析框架、通用软件安全性需求裁剪、获取特定软件安全性需求及软件安全性需求分析技术等多方面进行了深入的探讨和研究。主要完成了以下方面的研究工作：

(一)在总结分析国内外相关软件安全性标准及研究成果的基础上，结合国内软件开发现状，提出了软件安全性需求分析框架，并针对框架中的主要环节给出了基本的分析策略

(二)提出了通用软件安全性需求获取方法，详细阐述了该分析过程的思路、实施步骤。同时开发了分类整理的航空机载软件通用软件安全性需求清单，并确定了通用软件安全性需求的裁剪原则。

(三)提出了特定软件安全性需求获取方法，包括自顶向下的软件安全性需求分析及自底向上的软件安全性需求分析两个部分。随后本文从两个方面详细阐述了该分析过程的思路、实施步骤，最后进一步总结了软件安全性需求获取工作的特点。

(四)对软件安全性需求获取工作中应用的主要安全性分析技术进行了详细的原理介绍及操作说明。

(五)将本文所提出的方法进行了实例应用，选取某型航空发动机控制系统数控软件，对其进行软件安全性需求获取工作，获取的软件安全性需求对现有软件需求规格进行了很好的补充，暴露了此数控软件的很多安全性缺陷并提出了有效的修正措施，进而验证了本文所提方法的正确性和有效性。

本文的创新点

本文的创新性主要体现在以下方面：

(一)提出了与现有软件开发过程紧密结合的软件安全性需求分析工作框架，该框架从阶段划分、分析思路及应用方法上充分考虑了国内机载软件开发现状，具有较强的通用性及操作性，适用于多种领域的安全关键软件开发。

(二)提出了通用机载软件安全性需求获取方法，方法以通用软件安全性需求清单剪

裁为核心，详细开发了航空机载软件通用安全性需求清单，该清单参考了国内外多部安全性相关标准及指南，涵盖了危险命令、初始化、数据处理、人机交互、软件自身功能等多个方面，尽可能做到了对各种安全性情况的全面考虑。

(三)提出了特定软件安全性需求获取方法，该方法以全过程危险分析的思想为基础，从系统角度及软件功能设计角度出发进行综合分析，在分析过程中突出了软件安全性分析特点，并给出了各安全性需求分析方法结合应用的详细指导。

(四)在某型发动机控制系统数控软件上实施了过程完整、考虑全面的软件安全性需求获取工作，对软件安全性需求获取工作在实际项目中的应用进行了很好的探索，获得宝贵的分析经验

不足与展望

本文还存在一些不足之处，需要在以下几个方面展开进一步的工作：

(一)安全关键软件特别是航空机载软件复杂程度高、时实性强，分析人员考虑情况复杂，分析难度大。本文提出的软件安全性需求获取方法并非是最全面、最高效的，还应在此基础上进一步的改进和完善，补充更多体现软件特点的需求分析技术，使整个软件安全性需求获取工作能在最大程度上满足要求

(二)本文提出的软件安全性需求获取方法以人工分析为基础，没有深入研究如何切实实现计算机辅助。此方法应用在大型安全关键软件时可能会面临分析工作量大、分析效率低等问题，今后还需要在软件安全性需求获取自动化工具研究方面继续努力

(三)目前方法原理还只限于定性分析阶段，今后需要增加工程实践的应用，在应用中积累经验数据，为软件领域的综合分析方法的定量化打下基础

参考文献

[1] N.G.Leveson. Software safety: Why, what, and how?[J] ACM Computing Surveys, 18(2), June 1986.

[2] N.G.Leveson. "A New Approach to System Safety Engineering"[M], Aeronautics and Astronautics, Massachusetts, Institute of Technology, Draft of New Book, 2005.

[3] [3] N.G.Leveson. "The Role of Software in Spacecraft Accidents"[J], AIAA Journal of Spacecraft and Rockets, Vol. 41, No. 4, July 2004

[4] Gottesdeiner, E., Requirements by Collaboration[M], Addison-Wesley, 2002.

[5] Samuel Renault, Xavier Franch, Carme Quer. PABRE: Pattern-Based Requirements Elicitation, Research Challenges in Information Science[J], 2009, 81-92.

[6] Matthew John Squair. Issues in the Application of Software Safety Standards[J]

[7] Bowen,J. & Stavidrou,V., Safety-Critical Systems,Formal Methods and Standards[J], In IEE/BCS Software Engineering Journal, Volume8 No.4, pp189-209,1992.

[8] Atchison,B., Wabenhorst,A., A Survey of International Safety Standards[J], Software Verification Research Centre (SVRC), SVRC Technical Report 99-30, The University of Queensland QLD, Australia, 1999.

[9] GJB/Z 142-2004 Guide for military software safety analysis[S]

[10] IEC 61508 Functional safety of electrical/electronic/programmable electronic safety-related systems[S]

[11] ARP4754 Certification Considerations for Highly-Integrated Or Complex Aircraft Systems[S]

[12] ARP4761 Guidelines and Methods for Conducting the Safety Assessment Process on Civil Airborne Systems and Equipment[S]

[13] RTCA DO-178B Software Considerations in Airborne Systems and Equipment Certification[S]

[14] Dima Zemskyy Safety and Reliability Considerations in DO 178B[C]

[15] NASA-GB-8719.13 NASA Software Safety Guidebook[S]

[16] Joint Software System Safety Committee SOFTWARE SYSTEM SAFETY HANDBOOK[S]

[17] Li Yonghua Requirement Engineering Based on Combining Goal with Scenarios[C]

[18] Yue K.What Does It Mean to Say that a Specification is Complete?[J] In: Proceedings of the IEEE International Workshop on Software Specifications and Design,Monterey:IEEE Computer Society Press,1987.42-49.

[19] Lamsweerde AV. Goal-Oriented Requirements Engineering:A Guided Tour.[J] Proceedings of the Fifth IEEE International Symposium on Requirements Engineering.Los Alamitos:IEEE Computer Society Press,2001.249-262.

[20] Dardenne A,Lamsweerde AV and Fickas S.Goal-Directed Requirements Acquisition[J].Science of Computer Programming,1993,20(1-2):3-50.

[21] Lamsweerde AV,Dardenne A,Delcourt B,Dubisy F.The KAOS Project: Knowledge Acquisition in Automated Specification of Software[J].In: Proceedings AAAI Spring Symposium Series,Stanford University:American Association for Artificial Intelligence,1991.59-62.

[22] Darimont R,Delor E,Massonet P,Lamsweerde AV.GRAIL/KAOS:An Environment for Goal-Driven Requirements Engineering[C].In:Proc.ICSE'98-20th Intl.Conf.on Software Engineering,Kyoto:ACM Press,1998.58-62.

[23] Yu E.Modelling Organizations for Information Systems Requirements Engineering[C].In:Proc.RE'93-1st Intl Symp.on Requirements Engineering, San Diego:IEEE Computer Society Press,1993.34-41.

[24] Yu E.Towards Modeling and Reasoning Support for Early-Phase Requirements Engineering[C].In:Proc.RE-97-3rd Int.Symp.on Requirements Engineering, Annapolis:IEEE Computer Society Press,1997.226-235.

[25] Mylopoulos J,Chung L,Nixon B.Representing and Using Nonfunctional Requirements:A Process-Oriented Approach[J].IEEE Transactions on Software Engineering,1992,6(18):483-497.

[26] Dardenne A, van Lamsweerde A, Fickas S., Goal-directed Requirements acquisition[J]. Science of Computer Programming, 20(1, 2). 3-50.

[27] Bubenko, et al. Software Requirements Acquisition through Enterprise Modeling[C]. Software Engineering and Knowledge Engineering (SEKE'94). Jurmala, Latvia, 1994.

[28] Dardenne A,Fickas S,Lamsweerde AV.Goal-Directed Concept Acquisition in Requirements Elicitation[C].In:Proc.IWSSD-6-6thIntl.Workshop on Software Specification and Design,Como:IEEE Computer Society Press,1991.14-21.

[29] Desharnais J,Frappier M,Khédri R,Mili A.Integration of sequential scenarios[C]. In:Proceedings of the 6th European conference held jointly with the 5th ACM SIGSOFT international symposium on Foundations of software engineering, Zurich:Springer-Verlag,1997.310-326.

[30] Chin G,Rosson MB.Progressive design:staged evolution of scenarios in the design of a collaborative science learning environment[C].In:Proceedings of the SIGCHI conference on Human factors in computing systems,Los Angeles: ACM Press,1998.611-618.

[31] Sutcliffer A.Scenario-Based Requirements Engineering[C].In:Proceedings of the 11th IEEE International Requirements Engineering Conference.Los Alamitos: IEEE Computer Society Press,2003.320-329.

[32] Rumbaugh J,Blaha M,eds.Object-Oriented Modelling and Design[M],New Jersey:Prentice Hall,1991.

[33] Billard EA.system scenarios as Use Case Maps[C].In:Proceedings of the 4th international workshop on Software and performance,Redwood Shores:ACM Press,2004.266-277.

[34] Fowler M.UML Distilled[M].2nd edition,Addison-Wesley,1997.

[35] Jger D,Schleicher A,Westfechtel B.Using UML for software process modeling[C].In:Proceedings of the 7th European software engineering conference held jointly with the 7th ACM SIGSOFT international symposium on Foundations of software engineering,Toulouse:Springer-Verlag,1999. 91-108.

[36] Young RM,Barnard P.The use of scenarios in human-computer interaction research:turbocharging the tortoise of cumulative science[C].In:Proceedings of the SIGCHI/GI conference on Human factors in computing systems and graphics interface,Toronto:ACM Press,1986.291-296.

[37] Carroll J,Rosson MB,McInerney P.Scenarios in practice[C].In:CHI'03 extended abstracts on Human factors in computing systems,Ft.Lauderdale: ACM Press,2003.1046-1047.

[38] Fickas S,Johnson L,Karat J,Potts C.Using scenarios to elicit user requirements[C].In:Conference companion on Human factors in computing systems,Boston:ACM Press,1994.467

[39] Lahoz C.H.N, Camargo Jr.J.B, Abdala, M.A.D, Burgareli L.A, A Software Safety Requirements Elicitation Study On Critical Computer Systems[C]

[40] Elena Navarro†, Pedro Sánchez‡, Patricio Letelier, Juan A. Pastor‡ and Isidro Ramos A Goal-Oriented Approach for Safety Requirements Specification[C]

[41] E. Letier and A. van Lamsweerde, “High Assurance Requires Goal Orientation”[C], Proceedings of International Workshop on Requirements for High Assurance Systems, Essen, September 2002.

[42] S. Kelly, K. Lyytinen, M. Rossi: “METAEDIT+ A fully configurable Multi-User and Multi-tool CASE and CAME Environment”[C]. Proceedings of 8th International Conference on Advances Information System Engineering, LNCS1080, Springer-Verlag, 1996, 1-21.

[43] Du Junwei, Xu Zhongwei, Mei Meng, Du Junwei Verification of Scenario-Based Safety Requirement Specification on Components Composition[C]

[44] 周新蕾，刘正高 航天软件可靠性安全性技术应用发展趋势[J]. 质量与可靠性，2006

[45] 周新蕾 软件安全性分析技术及应用[J]. 质量与可靠性，2005

[46] 周新蕾，缪峥红 安全性关键软件的可靠性测试[J]. 载人航天，2005

[47] 牛爱民，叶东升 软件安全性技术在工程中的应用[J]. 计算机工程与设计，2007

[48] 周新蕾，宋星，林佳，杜杠 软件安全性分析在运载火箭上的应用[C]，第7届国际可靠性、维修性、安全性学术会议，2007

[49] 林佳，杜杠，程华彦，周新蕾 基于功能节点识别和路径追踪的软件潜在分析 质量与可靠性，2007

[50] 韩翔宇，石柱 WL_Net在导弹飞行控制软件安全性分析中的应用[J]. 航天控制，2008

[51] 代彬，陆刚，韩可琦 基于时间Petri网的实时嵌入式软件系统安全性分析[J]. 现代计算机，2001

[52] 宋晓秋 软件安全性分析的Petri网方法[J]. 质量与可靠性，1998

[53] 宋晓秋 软件安全性分析的Petri网方法续一[J]. 质量与可靠性，1998

[54] 宋晓秋 软件安全性分析的Petri网方法续二[J]. 质量与可靠性，1998

[55] 宋晓秋 软件安全性分析的Petri网方法续三[J]. 质量与可靠性，1998

[56] 张鲁峰，黄敏桓，张剑波 软件安全性相关标准浅析[C]. 第十三届全国抗恶劣环境计算机学术年会，2003

[57] 洪益群 变与不变——美军标改革刍议[J]. 航空标准化与质量，1997

[58] ISO8402: 1994 - Quality management and quality assurance[S]

[59] GJB102-1997 软件可靠性和安全性设计准则[S]

[60] IEEE830-1998 Recommended Practice for Software Requirements Specifications[S]

[61] NASA-STD-8719.13B. NASA Software Safety Standard[S]

[62] GJB438B-2009 军用软件开发文档通用要求[S]

[63] 祁新杰，郭迎清，王海泉. 航空发动机控制系统集成仿真平台[J]. 科学技术与工程, 2009, 9(10)

[64] 陈毅. 航空发动机控制系统传感器故障诊断研究[D]. 南京航空航天大学,2007

[65] 周永权. 基于实时操作系统的航空发动机数字控制器软件设计[D]. 南京航空航天大学,2008

[66] 荣莉，刘德宏，孙志岩，梁春华 航空发动机全权限数字式电子控制系统在中国的发展与展望[J]. 航空发动机，2007

[67] 石斌 航空发动机高可靠性FADEC软件系统技术研究[D]. 西北工业大学，2004

[68] 韩小琦 航空发动机控制系统安全性评估研究[D]. 中国民航大学，2009

攻读硕士学位期间取得的学术成果

Zhang Yifan, Bao Xiaohong, Li Zhen A framework for airborne aviation software safety requirements analysis. International Symposium on Aircraft Airworthiness 2009(EI,ISTP)

致　谢

光阴似箭，岁月如梭，不知不觉我即将走完两年半的研究生生活，回想这一路走来的日子，我收获的不仅仅是愈加丰厚的知识，更重要的是在实践中所培养出来的思维方式、表达能力和广阔视野。很庆幸这些年来我遇到了许多恩师益友，无论在学习上、生活上还是工作上都给予了我无私的帮助和热心的照顾，让我在诸多方面都有所成长。感恩之情难以用语言量度，谨以最朴实的话语致以最崇高的敬意。

首先，我要衷心感谢我的研究生指导老师鲍晓红老师。在学术上鲍老师以严谨求实的治学态度和勤勉的工作态度教导了我踏实求学，在学术探讨中她又以对问题高屋建瓴的专业见地使我茅塞顿开，逐渐掌握了许多研究方法和研究思路；在生活上，鲍老师更是平易近人，亲人般无微不至的关怀，常带给我许多温暖和感动。可以说无论是我在学业上的进步还是个人的成长，都离不开她的关怀。每思及恩师教诲和为此付出的辛劳，常自责未能达到恩师期望，唯有在以后的道路上更加勤勉努力，望能不负师恩。

同时由衷感谢教研室里各位老师，钟德明、李国旗、吴玉美、张虹、曾福萍等老师在软件安全性课题组的多次交流中给我很多指导和热情帮助；陆民燕教授和刘斌教授在百忙之中给予我求学道路上的无私帮助与热忱鼓励；王轶辰、杨顺昆、李秋英等老师在研究生学习过程中给予我的专业指导；各位老师道德与学术并重，宽容博大的胸襟、谦逊朴素的为人，令我如沐春风，倍感温馨。我在这里各位老师鞠躬致谢！

随后我要感谢李震师兄对我的论文提出了诸多宝贵的意见和建议，对学长的帮助表示真挚的感谢；感谢徐小杰、徐燕同学在研究课题上开诚布公的讨论和各方面的热情帮助，；感谢 2008 级软件教研室的所有的同窗好友，在同大家的交往中我学到很多，你们使我的求学生涯充满了欢乐，让我在开心的时候能与人分享，让我在烦恼的时候能找人倾诉，使我在北航的生活才能如此丰富而充实。

最后，我要感谢我最亲爱的父母。你们数十年含辛茹苦、无私的关爱和奉献，让我在漫长的求学道路上不感到孤单，让我在拼搏和奋斗的历程中不感到疲倦，你们是我永远的牵挂和眷念。

在今后新的征程中，无论面临多大的困难，我也将怀抱着感激、怀抱着情谊、怀抱着责任、怀抱着期望和梦想，坚定、自信地走下去