



普通高校“十一五”规划教材

飞行器研制系统工程

阮 镰 章文晋 编著

北京航空航天大学出版社

内容简介

本书阐述系统工程的原理和方法,特别是工程系统工程的3个主要组成部分,即系统工程过程、工程专业综合、系统分析和控制在现代飞行器研制中的应用。通过系统工程将飞行器研制中的技术和管理有机地结合起来。本书还对我国在下一代先进飞行器研制中的若干问题,如下一代飞行器研制中的软件可靠性、软件系统测试和系统工程管理等问题进行了探讨,并提出了相应的解决方案。

本书可作为航空航天飞行器设计专业、工程系统工程专业的本科高年级学生和研究生教材,也可供广大航空航天行业及其他工程专业的设计人员和管理人员参考。

图书在版编目(CIP)数据

飞行器研制系统工程/阮镰,章文晋编著. —北京:北京航空航天大学出版社,2008.9

ISBN 978-7-81124-414-4

I. 飞… II. ①阮…②章… III. 飞行器—研制—系统科学 IV. V42

中国版本图书馆CIP数据核字(2008)第134180号

飞行器研制系统工程

阮 镰 章文晋 编著

责任编辑 孔祥燮 范仲祥

*

北京航空航天大学出版社出版发行

北京市海淀区学院路37号(100083) 发行部电话:(010)82317024 传真:(010)82328026

<http://www.buaapress.com.cn> E-mail:bhpress@263.net

北京时代华都印刷有限公司印装 各地书店经销

*

开本:787×960 1/16 印张:18 字数:403千字

2008年9月第1版 2008年9月第1次印刷 印数:3000册

ISBN 978-7-81124-414-4 定价:36.00元

作者简介

阮镡 北京航空航天大学工程系统工程系教授,博士生导师。1961年,北京航空学院火箭系本科毕业,1965年,北京航空学院火箭系副博士研究生毕业,曾参加我国若干个飞行器的研制工作。1983—1985年在美国UTSI(田纳西州立大学宇航学院)做访问学者。之后在北京航空航天大学工程系统工程系从事系统工程并行工程的教学和科研,著作有《工程系统的规划和设计》。近年来从事软件工程与软件可靠性工程领域的教学和研究,曾获2001年国防科技进步二等奖。

章文晋 博士,副教授,北京航空航天大学工程系统工程系可靠性工程教研室副主任,北航无人机设计研究所型号可靠性主任设计师。主要从事复杂产品可靠性、维修性、保障性技术研究、教学和型号技术支持工作。主讲《飞行器设计系统工程》和《产品保障性分析技术》课程,参与《飞机设计手册》和《无人机设计手册》的部分章节编写工作。曾获国防科学技术进步二等奖2次。

前 言

笔者于20世纪50年代末在北京航空学院本科飞行器设计专业学习,60年代研读副博士研究生,期间均未上过“飞行器研制系统工程”这门课。当时,连“系统工程”这一名词也未曾听到过。那时,我们研读的是飞机设计、空气动力学、飞行力学、结构力学、热力学、材料学等课程,照样也能把飞行器设计、制造出来。国外的老一代飞行器设计师在五六十年代之前恐怕也未学习过系统工程方面的专业知识。但是,随着科学技术的发展,以及军事和民用领域对飞行器需求的逐步提高,飞行器设计日趋复杂,涉及的专业、学科越来越多,不仅要求飞行器有高的性能,而且要求有高的效能,亦即要求飞行器除了有优良的飞行性能外,还应使用可靠,维修方便,保障经济有效……因此,传统的飞行器设计所包涵的专业学科不够用了,用传统的飞行器设计知识设计出来的飞行器已不能满足军事部门的战备完好性要求和民用部门的飞机出勤率要求。例如美国的第二代飞机F-111、F-4等,在交付空军使用时只解决了57%的问题,还有43%的问题要到交付使用后才逐步解决,严重地影响了飞机的作战使用。有的飞行器因为任务要求分析粗糙,研制到一半才发现该飞行器根本不能满足使用部门的要求,只能半途中止。在经历了一系列的挫折和失败,耗费了数以亿计的金钱之后,人们发现:如果应用了系统工程的原理和方法,那么这些失败本来是可以避免的,或至少是可以减缓的。

自从把系统工程原理和方法引入飞行器研制的全过程后,取得了许多令人瞩目的成果。例如航天领域中的阿波罗登月计划和航天飞机计划的实现;航空领域中的F-16、F-22和波音777等飞机的研制成功。总之,系统工程在现代军事和民用领域中发挥了巨大作用。

对于飞行器研制来说,只有采用系统工程的原理和方法,才能使工程技术人员和管理人员在给定的任务要求和资源约束条件下,合理地确定系统的技术要求,选择出最优的系统方案,全面地管理工程的发展,正确地验证系统的技术性能;并使设计出来的飞行器能顺利地投入生产,使生产出来的飞行器在使用中得到经济而又有效的保障。

飞行器研制系统工程是将系统工程的原理和方法用于飞行器研制,是应用于

现代飞行器研制的系统工程,是组织和管理飞行器研制的规划、研究、设计、发展、试验、生产、使用和保障全过程的系统方法和工程技术。

本书共8章。

第1章是飞行器研制系统工程概述。阐明飞行器设计系统工程的基本概念和飞行器全寿命期的阶段划分。

第2章是系统工程过程。阐述如何将飞行器研制的任务需求,通过一系列规范化的程序和分析——综合的迭代过程,将其转化为飞行器的技术要求及满足这些技术要求的系统最优方案。

第3、4、5章分别是工程专业的综合、综合保障工程、飞行器研制中的软件工程。分别阐述可靠性、维修性、安全性、综合保障工程和软件工程在现代飞行器研制中的应用。只有将这些工程专业有机地在飞行器设计中综合,才能将飞行器从单纯的性能度量扩延为效能的度量。

第6、7章分别是系统分析和控制、风险分析。分别阐述权衡分析、系统体系结构分析、风险分析和技术性能测量等分析方法和控制技术在飞行器设计中的应用。运用这些方法和技术能确保研制出的飞行器满足给定的任务需求。

第8章是并行工程。阐述并行工程的发展背景和概况、定义与特点、实施要素、组织方式、研制过程、环境、管理等。

系统工程,特别是并行工程在飞行器研制中的广泛应用,已在国外发达国家取得了显著的成效,它也必将在我国的现代飞行器研制中发挥重要作用。

本书第3、4章由章文晋副教授执笔,其余各章由阮镡教授撰写,全书由阮镡主编。

阮 镡

2008年5月31日

于北京航空航天大学

目 录

第 1 章 飞行器研制系统工程概述

1.1 飞行器研制系统工程的概念	1
1.1.1 系 统	1
1.1.2 系统工程	2
1.1.3 工程系统工程	2
1.1.4 飞行器研制系统工程	3
1.2 飞行器全寿命期研制阶段的划分	5
1.2.1 零阶段——立项前的酝酿阶段	5
1.2.2 阶段 1——方案探索和系统定义阶段	6
1.2.3 阶段 2——方案确定阶段	7
1.2.4 阶段 3——工程研制阶段	8
1.2.5 阶段 4——生产和部署阶段	9
1.2.6 阶段 5——使用和保障阶段	9
1.2.7 退役处理	10

第 2 章 系统工程过程

2.1 系统工程过程在现代飞行器研制中的作用	11
2.2 系统工程过程的定义和特征	12
2.3 系统工程过程的实施	13
2.3.1 任务要求分析	13
2.3.2 功能分析	18
2.3.3 系统综合	25
2.3.4 系统工程过程的输出	31

第 3 章 工程专业的综合

3.1 概 述	34
3.2 可靠性工程	37
3.2.1 可靠性要求	37
3.2.2 可靠性工程实践	41



3.3 维修性工程	45
3.3.1 维修性要求	46
3.3.2 维修性工程实践	47
3.4 安全性工程	48
3.4.1 安全性评价	48
3.4.2 安全性工程实践	49
3.5 零件工程	52
3.5.1 材料的选择	52
3.5.2 加工过程的选择	53
3.5.3 零件控制	54
3.6 人素工程	55
3.6.1 人素工程的研究方法	56
3.6.2 人素工程研究中的度量尺度	57
3.6.3 人素工程实践	58
3.7 可生产性工程	59
3.7.1 影响可生产性的诸因素	60
3.7.2 可生产性工程实践	61

第4章 综合保障工程

4.1 保障性	64
4.1.1 保障性定义	64
4.1.2 保障性要求	65
4.2 综合保障	67
4.2.1 基本概念	67
4.2.2 装备保障	68
4.2.3 综合保障要素	69
4.2.4 保障系统、保障方案、保障计划	71
4.2.5 综合保障计划与综合保障工作计划	73
4.3 保障性分析	74
4.3.1 保障性分析的概念	75
4.3.2 常用的保障性分析方法	77
4.3.3 保障性分析信息	79
4.4 保障资源研制	80
4.4.1 备件和器材供应	81
4.4.2 保障设备	82

4.3 综合保障试验与评定	84
第5章 飞行器研制中的软件工程	
5.1 概 述	86
5.1.1 软件在现代飞行器中的重要作用	86
5.1.2 软件的质量特性	87
5.1.3 软件的质量模型	88
5.1.4 软件质量的 Pareto 原理	89
5.1.5 软件质量与软件工程化	89
5.2 现代软件开发与管理的三维模型	91
5.2.1 软件生存期的全过程控制	92
5.2.2 软件质量的全方位管理	95
5.2.3 构建多层次的软件开发管理模式	95
5.3 时间维——对软件生存期的全过程控制	96
5.3.1 过程的定义	96
5.3.2 软件过程管理的关键活动	97
5.3.3 过程控制要点	99
5.3.4 软件生存期各阶段的过程控制	99
5.4 空间维——软件关键质量因素的全方位管理	116
5.4.1 软件的分级管理	116
5.4.2 软件文档管理	117
5.4.3 软件需求管理	118
5.4.4 软件评审管理	119
5.4.5 软件配置管理	125
5.4.6 软件测试管理	127
5.4.7 建立软件的失效报告、分析和纠正措施系统	131
5.4.8 对分承制单位的管理	132
5.5 三位一体的软件开发管理模式	132
5.5.1 软件开发者的自我管理——个体软件过程	132
5.5.2 软件开发者的团队管理——小组软件过程	138
5.5.3 软件能力成熟度模型(CMM)	143
5.5.4 从 CMM 到 CMMI	147
5.6 下一代飞行器研制中的软件可靠性问题	149
5.6.1 美国 F-22 的教训	149
5.6.2 F-35 软件可靠性的实施	152

5.6.3 下一代飞行器研制中的软件可靠性工程	154
5.7 下一代飞行器研制中的软件系统测试问题	166
5.7.1 关于嵌入式软件测试环境的思考	166
5.7.2 模型驱动的软件仿真测试技术	168

第6章 系统分析和控制

6.1 权衡分析方法	180
6.1.1 权衡分析的方法和步骤	180
6.1.2 权衡分析的若干方法	185
6.2 工作分解结构	192
6.2.1 系统层次体系	192
6.2.2 工作分解结构的组成和分类	194
6.2.3 工作分解结构的基本功能	195
6.2.4 利用工作分解结构制定进度计划	196
6.2.5 利用工作分解结构估算费用	198
6.2.6 制定飞行器系统费用-进度曲线的方法	201
6.3 技术性能测量	204
6.3.1 技术性能测量的主要任务	204
6.3.2 选择技术性能测量的主要参数	205
6.3.3 技术性能测量的方法和步骤	207
6.3.4 技术性能测量报告	211

第7章 风险分析

7.1 概 述	214
7.2 与风险有关的术语和定义	215
7.2.1 风险和风险因子	215
7.2.2 风险分类	216
7.2.3 等风险曲线	216
7.3 风险的辨识	217
7.4 风险估计	219
7.4.1 技术风险的估计	219
7.4.2 进度风险的估计	222
7.4.3 费用风险的估计	225
7.5 风险管理	227
7.5.1 降低或控制风险的技术	227

7.5.2 减小风险的途径	228
第8章 并行工程	
8.1 并行工程的发展背景和概况	243
8.2 并行工程的定义与特点	244
8.2.1 定 义	244
8.2.2 特 点	244
8.3 实施并行工程的巨大效益	245
8.4 并行工程的实施要素	247
8.5 并行工程的组织方式	248
8.5.1 综合产品研制小组的两个案例	248
8.5.2 典型的“综合产品研制小组”的组织方式	250
8.5.3 IPT 的人员组成及对成员的要求	251
8.5.4 IPT 的指导原则和工作程序	252
8.6 并行工程中确定产品与过程要求的规范化方式	253
8.6.1 运用 QFD 将用户需求转化为产品与过程要求的规范化方法	253
8.6.2 QFD 的规范化流程	254
8.7 并行的研制过程	259
8.7.1 研制过程并行运作方式的特点	259
8.7.2 并行研制过程是一个不断改进的过程	262
8.8 并行工程的环境	263
8.8.1 并行工程环境概述	263
8.8.2 100% 的三维数字化产品设计	264
8.8.3 电子样机	265
8.8.4 计算机辅助可靠性、维修性、保障性分析与设计	266
8.8.5 交互通信与数据传输技术	270
8.9 下一代飞行器研制中的系统工程/并行工程管理问题	271
8.9.1 实施系统工程/并行工程管理的要点	271
8.9.2 系统工程能力成熟度模型	272
参考文献	276

第 1 章 飞行器研制系统工程概述

1.1 飞行器研制系统工程的概念

1.1.1 系 统

据系统论创始人冯·贝培朗菲的定义：系统是处于一定相互联系的、与环境发生关系的、各组成成分的总体。

系统本身又是它所从属的一个更大系统的组成部分。

例如本书讨论的对象是飞行器系统(如飞机系统)，飞机只是飞机系统中的一个组成部分，为了使飞机能正常飞行，还必须有地面保障(使用保障与维修保障)设备、地面设施(如机库)、空勤人员、地勤人员等。缺少这些分系统，飞机是无法正常飞行的。但飞机系统又是一个更大系统(如防空作战系统)的一部分，如图 1.1 所示。这就是系统的概念。

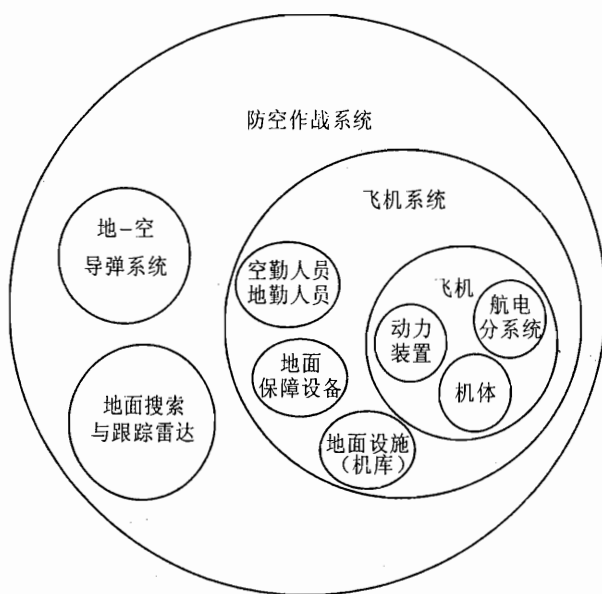


图 1.1 飞机与飞机系统及其他

1.1.2 系统工程

本书不阐述系统工程的起源,以及它的发展历史和各种不同的定义,因为这些内容在其他有关书籍中已多有阐明。这里给出的系统工程定义是参照美国国防部的定义^[1]。

系统工程是对科学和工程成果的应用,以实现:

- 通过反复迭代的定义、分析、综合、设计、试验和鉴定过程,将任务需求转化为对系统性能参数和系统配置的描述;
- 综合相关的技术参数,以保证所有物理的、功能的接口间相容,并优化整个系统设计方案;
- 将可靠性、维修性、安全性、保障性、人素工程、软件工程等工程专业综合到整个工程中去,以达到费用、进度和技术性能的目标。

简言之,系统工程既是一个技术过程,又是一个管理过程,在系统的全寿命期内都必须实施这两个过程。

1.1.3 工程系统工程

根据我国著名学者钱学森同志的意见,系统工程可依其学科的不同分为如表 1.1 所列的 14 个专业。

表 1.1 各类系统工程专业

系统工程的专业	专业的特有学科基础	系统工程的专业	专业的特有学科基础
工程系统工程	工程设计	教育系统工程	教育学
科研系统工程	科学学	社会(系统)工程	社会学、未来学
企业系统工程	生产力经济学	计量系统工程	计量学
信息系统工程	信息学、情报学	标准系统工程	标准学
军事系统工程	军事科学	农业系统工程	农事学
经济系统工程	政治经济学	行政系统工程	行政学
环境系统工程	环境科学	法治系统工程	法学

工程系统工程是应用于工程项目的系统工程,是组织和管理工程系统的规划、研究、设计、发展、试验、生产、使用和保障的科学方法,也是解决工程活动全过程的工程技术。

工程系统工程属于工程技术范畴,并以工程设计作为其特有的学科基础。这是一门极其新颖、尚在发展中的学科。它是以实际应用为目的的一门学科,是把各个领域的各种管理、技术加以综合应用的工程技术。

1.1.4 飞行器研制系统工程

飞行器研制系统工程隶属于工程系统工程范畴,其学科基础是飞行器设计。它是应用于现代飞行器研制的系统工程,是组织和管理飞行器研制的规划、研究、设计、发展、试验、生产、使用和保障全过程的系统方法和工程技术。

飞行器研制系统工程与工程系统工程、系统工程的关系如图 1.2 所示。

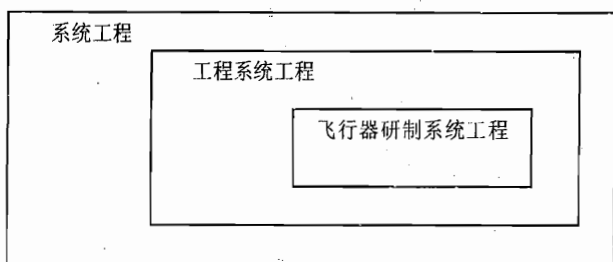


图 1.2 飞行器研制系统工程与各专业类系统工程的关系

1. 飞行器研制系统工程的要点

- 观点和方法：从飞行器系统整体出发,运用分析-综合的迭代方法。
- 应用对象：各类飞行器(包括航天类、航空类、军用、民用飞行器)。
- 应用范围：规划、设计、制造、试验、使用保障的全过程。
- 目的：使各类飞行器研制达到最优化(获取最大效益)。

2. 飞行器研制系统工程的基本内涵

飞行器研制系统工程由系统工程过程、工程专业综合和系统分析和控制这 3 个基本部分组成。

1) 系统工程过程

系统工程过程是一种技术处理过程,由一系列的工程工作和决策的逻辑程序组成。其目的是将飞行器系统的任务需求转化为对系统性能参数的描述,并确定飞行器系统的最优配置。

系统工程过程的主要功能是以一套规范化的方法和程序从任务需求中完整地导出量化的系统技术要求,并确定能满足这些技术要求的系统最优配置。这本质上是一个先导的工作过程,推行这一工作过程就能对整个设计进程起指导作用,并确保设计工作取得成功。

系统工程过程的主要工程活动是系统分析和综合。系统分析是指通过任务要求分析和功能分析把任务要求转化为功能要求,并借助随后进行的技术要求分配进一步转化为系统较低层次上的技术要求,从而使系统、分系统、部件能据此进行设计。而综合则是从系统总体的观点,按给定的详尽程度保持设计的完整性(应当包括组成系统的各个要素:硬件、软件、人员、设施和数据)。在本书第 2 章中将详细地阐述系统工程过程。

2) 工程专业综合

工程专业综合是将与飞行器研制有关的各个工程专业,诸如可靠性、维修性、安全性、人素工程、软件工程、生产工程、后勤工程、价值工程等工程专业的工作协调地纳入到系统的研制过程中,以保证它们能对设计工作施加影响,并最终满足飞行器系统规定的要求。

各个工程专业的设计和发展工作是整个工程研制任务的有机组成部分。除了将上述工程专业综合到系统研制工作中之外,工程系统工程还特别强调在系统研制过程中全面的工程综合。从20世纪80年代末到90年代初,美国针对飞行器系统(和武器装备系统)的技术要求越来越高而可用资源却日益减少的状况,进一步提出了“综合产品研制(Integrated Product Development)”和“并行工程(Concurrent Engineering)”的新概念,强调在整个系统研制中,都应将传统的工程技术(如力学、机械学、空气动力学、电子、电气、动力、液压等)和各工程专业(如可靠性、维修性、安全性、保障性工程等),以及试验工程、生产工程和后勤工程综合到一起,并予以统一协调和管理。采用这种新概念进行工程系统研制,可以大为减少设计与生产、使用、保障间的不协调,从而显著地提高产品的质量和可靠性,并节省了研制时间和费用。

为了实施“综合产品研制”所要求的全面的工程综合,必须在工程研制全过程中的任何阶段,都应由传统的工程设计人员、各工程专业的设计人员(如可靠性工程师、维修性工程师等),以及试验、生产、综合后勤保障部门的工程技术人员共同组成的设计组织来确定工程系统完整的技术要求,并开展研制工作,以确保他们能对设计工作施加影响。这样才能使研制出来的工程系统不仅具有优异的性能,而且有良好的可生产性和保障性。在本书第3~5章中将论述工程专业综合的各个方面,而第8章则阐述并行工程在现代飞行器研制中的应用。

3) 系统分析和控制

系统分析和控制是使飞行器研制过程从输入到输出的可追溯性得以保持,并使其研制过程得到控制,以确保最终研制出的飞行器能符合用户的需求。

系统分析包括权衡分析、系统体系结构分析、风险分析和效能分析。系统分析可用来评价备选飞行器方案及其技术途径、风险是否能恰当地满足系统要求和指标,为选择最终的体系结构和设计方案提供依据。

系统控制包括技术性能测量、技术状态控制和技术评审。其中,技术性能测量是对飞行器系统的技术性能进行连续的验证和控制过程;技术状态控制是对已确立的设计基线进行控制;技术评审则是对飞行器研制各阶段实施里程碑控制的重要手段。在本书第6、7章中将阐述权衡分析、系统体系结构分析、风险分析和技术性能测量等内容。

综上所述,系统工程过程、工程专业综合、系统分析和控制扩延了传统飞行器设计的内涵,并把飞行器的设计与规划、技术与管理融为一体。它们与传统飞行器设计学科的关系如图1.3所示。

正如图1.3中所展示的,系统工程过程将使用部门(或用户)提出的任务需求转化为工程系统量化的技术性能要求,为飞行器系统提供了设计的依据。工程专业综合是将可靠性、维修

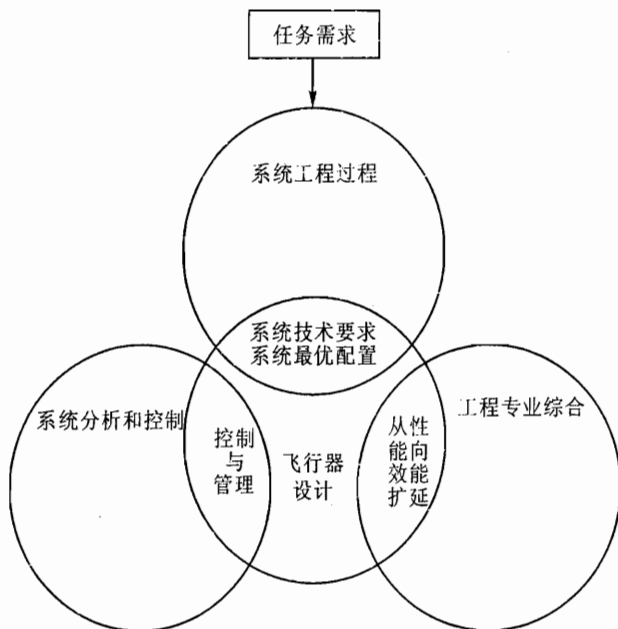


图 1.3 飞行器设计系统工程的 3 个基本组成部分与飞行器设计的关系示意图

性、保障性等与飞行器系统效能密切相关的技术性能纳入到工程设计中，从而扩延了传统工程设计所考虑的系统性能（如重量、速度、精度、功率、机动性等），将性能扩延为系统效能。这对于现代飞行器系统具有极其重要的意义。而系统分析和控制则对飞行器系统的整个研制过程进行监控、实施管理，以确保所研制的飞行器达到预定的任务要求和指标。

1.2 飞行器全寿命期研制阶段的划分

飞行器研制系统工程的一项重要任务是飞行器研制阶段的划分。阶段的划分是为了控制研制过程，并建立研制工作的基线。阶段划分体现了系统的有序性。

事实上，不同类别飞行器的研制阶段不尽相同。例如美国国防部发布的武器系统研制阶段与 NASA 的不同；我国权威部门制定的航天飞行器、人造卫星、航空飞行器的研制阶段程序也都各不相同。本书也不准备为各类飞行器提供一个统一的研制阶段程序。我们只是从飞行器研制系统工程的原理出发，阐明飞行器研制必须遵循的过程。至于对这些过程划分为哪几个阶段，各类飞行器的研制部门可根据飞行器的特点和行业的管理需要予以划分。

1.2.1 零阶段——立项前的酝酿阶段

开始这个阶段的触发因素是：用户的某种需要、外部的某种威胁或新的技术机遇；对一种新的飞行器提出了需求。但对这种新飞行器是否予以立项研制必须进行酝酿。这个阶段有人

称之为飞行器的可行性论证阶段,但笔者认为这种提法似不妥。笔者认为应首先对一个新飞行器的立项进行“不可行性论证”,并对这一论证结果进行“非提倡性评审”。假如,某部门提出要在 20XX 年实现月球载人飞行,于是组织一个班子进行可行性论证。一般,由于这种论证通常是由该部门或该部门的上级主管部门主持的,论证组的成员也往往是这一领域的专家,所以可行性论证的结论多半是可行的。尤其是当这种结论与“长官意志”相符时,更是如此。但当工程进展到一半,或工程已完成相当一段时间后,其后果,甚至是灾难性的后果才显现出来,如我国 XXX 水利工程的例子。因此,应当认真听取反面意见,进行“不可行性论证”:论证这一项目会带来的问题、遭遇的风险、资源的浪费、对环境的影响等。只有在对这些问题给予科学的、实事求是的论证后,才能确定该项目是否真正可行。

零阶段结束时,有两种可能:一是立项被否定,项目不成立;二是立项通过论证,提交项目立项的正式建议书。若该建议书被正式批准,方能启动后面的研制程序。

1.2.2 阶段 1——方案探索和系统定义阶段

1. 阶段 1 的输入

- 用户(或使用部门,下同)的任务需求书(或研制总要求,下同);
- 约束条件(如资金限额、研制周期要求、环境条件等)。

2. 主要任务

本阶段的主要任务是将用户的任务需求,通过一系列规范化的程序转化为飞行器的系统技术要求及满足这些要求的基本系统方案,亦即定义所研制的飞行器系统。

这一阶段的核心工作是确定系统是干什么的,而不是具体地设计系统,亦即不涉及系统是怎样去实现这些(功能)要求的。

在这一阶段应注意的是:

(1) 满足任务需求的系统方案有多个,所以必须进行多方案探索,探索的方案越多,从中选出最优方案的可能性也越大。除非存在极其特殊的情况(如作战的紧迫需要),否则应严禁进行单一方案的设计。

(2) 这一阶段给出的系统要求应是全面的,即不仅要给出性能、功能要求,还应给出使用、保障要求。因此,这一阶段进行的系统方案探索也不应只有技术方案的探索,还应同时进行使用方案和保障方案的探索。即选择的技术方案应经过使用方案的验证。

(3) 方案探索的结果是能提供“足够的工程理由和实用性”,说明这一立项确实可行。“足够的工程理由”是指所选择的方案对给定的技术指标留有余地,而不是方案勉强达到给定指标。“实用性”,是表明所选择的方案是可靠的、实用的,经过恰当的努力是可以实现的,而不是采用过多的新技术,需要边预研、边研制。

3. 阶段 1 的输出

- 飞行器系统技术要求(一般以系统规范或技术要求文件表述);

► 飞行器基本方案。

对该阶段研制工作和程序的详细描述可见本书 2.3.1 小节。

1.2.3 阶段 2——方案确定阶段

1. 阶段 2 的输入

阶段 2 的输入就是阶段 1 的输出,即飞行器系统的技术要求和基本方案。

2. 主要任务

本阶段的主要任务是对初步选定的基本方案进行深入地分析、研究及必要的演示验证,以便最终确定方案。

这个阶段要回答的问题是:系统是如何构成的?即确定系统的配置(configuration,或称为技术状态)。系统的配置(技术状态)确定了,系统方案也就确定了。

为了确定系统的配置,需要将系统级的技术要求按照系统的功能体系结构进行功能分析,将系统级的技术要求下传并分配给各个分系统、子系统、部件……,即进行系统分析。其中包括进行风险分析,辨识、估计并降低选定方案的风险。

在本阶段还要通过系统综合,将系统的功能体系结构映射为物理体系结构,这样才能得出构建系统及其各组成部分的硬件、软件、数据、设备及其接口的配置,才能最终确定一个完整的系统。

在本阶段应特别注意的是:

(1) 被确定的飞行器系统方案的可行性和有效性必须经过充分的、有深度的方案验证,以确信至少有一种技术途径可使其有效。也就是对已确定的系统方案,要经过足够的、有深度的设计工作验证,而不是仅仅通过纸面的分析和设计人员设想的、主观的判断。

鉴于现代飞行器在研制中采用大量的高新技术,因此必须认真探索这些新技术实现的可行性。在飞行器研制中一定要牢记的是“设计不能依赖于一些不可见的、突破的出现”,亦即千万不要把希望寄托在研制过程中会出现不可预见的、突然出现的技术突破。这当然不是说,不要采用高新技术,而是说对采用的高新技术至少应当有一种技术途径可使其有效地实现;否则不应采用。

(2) 对阶段 1 和阶段 2 应给予足够的研制经费的支持。

我国的飞行器研制中,通常在研制初期的经费很少,一般要到设计工作结束,开始投产时才拨发较多的经费。这种在研制前期缺乏经费的现象会产生严重的后果,即使得前期的研制工作做得很不充分,大量是纸面的分析工作,缺乏对研制中要采用的关键技术进行必要的、充分的演示验证。

据系统工程原理和大量的统计结果表明:阶段 1+阶段 2 的研制费用应占研制总费用的 25% 左右。这样做的好处是:

① 使前期工作,尤其是关键技术的演示验证工作做得充分,大大降低型号研制的后期风

险。例如美国的飞机研制,都要求承包商进行原型机验证。这当然会要求投入较多的经费。但是前期经费的投入不仅不会增加研制总经费,反而会降低它。

② 据国外大量的飞行器研制统计数据表明^[2]:

- 如果阶段1+阶段2的研制经费占研制总经费的25%,则最终研制总经费的超支仅为20%。
- 如果阶段1+阶段2的研制经费只占研制总经费的5%(或以下),则最终研制总经费的超支可能达到120%~160%(参见图1.4)。

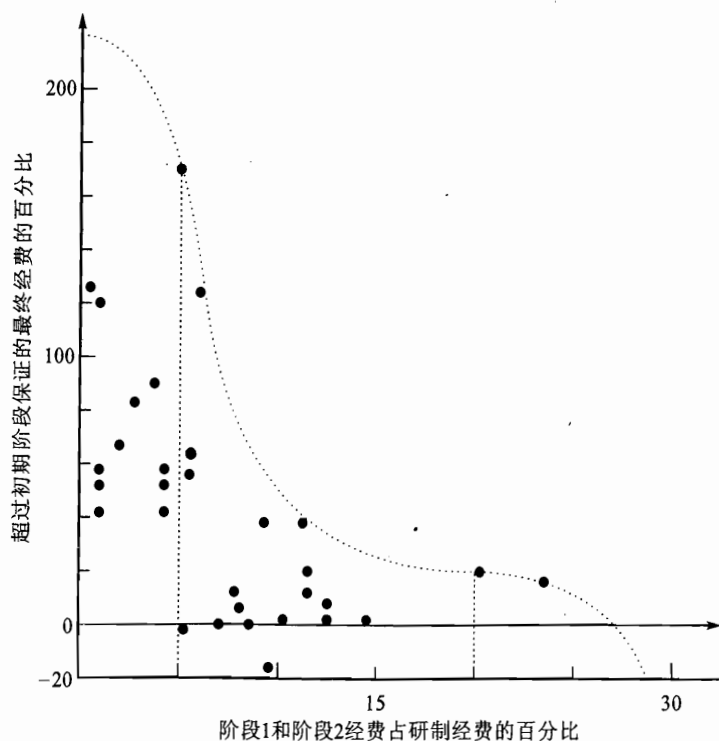


图 1.4 阶段 1 和阶段 2 研制经费对研制总经费超支的影响^[2]

3. 阶段 2 的输出

- 被确定方案的体系结构(功能体系结构、物理体系结构);
- 接口控制文件;
- 系统设计规范(经批准后形成功能基线)。

1.2.4 阶段 3——工程研制阶段

1. 阶段 3 的输入

阶段 2 的输出就是阶段 3 的输入,是开展工程研制工作的依据。

2. 主要任务

本阶段的主要任务是进行飞行器的设计、制造、试验、外场试验(试飞)和定型。具体包括:

(1) 初步设计。在分系统、子系统和部件的层次上,反复执行系统工程过程,从而制定出分系统规范等设计文件。

(2) 详细设计。用设计来实现技术要求,并完成有关零、部件的完整的产品设计。

(3) 生产准备。完成产品规范、材料规范和工艺规范的制定,并根据生产条件和制造技术对修改、完善设计进行反馈。

(4) 试生产。进行小批量试生产,以支持试验和试飞,并继续向设计工作提供反馈信息,进一步修改与完善设计。

(5) 试验与评定。试验与评定一般分为两个阶段。第一阶段:由研制部门进行研制试验与评定,主要试验与评定飞行器性能、功能满足任务需求的状况。第二阶段:在研制试验与评定基本结束后,应转入使用试验与评定。使用试验与评定由使用部门进行,主要试验飞行器的使用适用性(如作战适用性,对使用环境的适用性等)。

(6) 设计定型。工程研制阶段的最后是进行飞行器的设计定型鉴定。只有通过了设计定型的飞行器,方能投入批量生产,并交付用户使用。

3. 阶段3的输出

- 通过设计定型鉴定的飞行器;
- 与飞行器配套的保障系统;
- 成套技术资料与文档。

1.2.5 阶段4——生产和部署阶段

本阶段的工作主要是对设计定型的飞行器投入批量生产,然后陆续交付使用部门予以部署。

生产和部署阶段的主要工作是以合理的费用生产并交付能充分予以保障的飞行器系统。

对于批量很大的飞行器(如飞机、战术导弹),生产阶段最好分两个阶段进行,以降低生产风险:第一阶段先以低速率进行最初几批产品的生产;发现问题后,及时修改生产工艺或材料、设备,然后转入第二阶段的高速率生产。

1.2.6 阶段5——使用和保障阶段

随着飞行器系统的部署,使用和保障阶段就开始了。这一阶段的主要工作是:

- (1) 向系统提供经济而有效的保障;
- (2) 对该阶段的系统技术状态变更进行技术状态控制;
- (3) 根据系统及其相配套保障系统的实际运行情况,测量并评价系统投入实际使用后的适应性、使用可用性及效能;

(4) 根据使用情况和新的需求,提出飞行器系统的改进或改型计划。

系统的使用和保障阶段很长(约为 15~20 年,甚至更长),一直要延续到该系统退设为止,即该系统全寿命期的终止。

上述飞行器寿命期各阶段的划分如表 1.2 所列。

表 1.2 飞行器全寿命期阶段划分

阶段序号	零阶段	阶段 1	阶段 2	阶段 3	阶段 4	阶段 5
阶段名称	立项前	方案探索和系统定义	方案确定	工程研制	生产和部署	使用和保障
里程碑						
输入	- 用户需求 - 外部威胁 - 技术机遇	- 任务需求书 - 约束条件	- 系统技术要求 - 基本方案	- 系统规范 - 接口控制文件 - 系统体系结构	- 生产计划 - 生产工艺	正式交付的飞行器及其保障系统
主要工作	- 不可行性论证 - 非提倡性评审	- 多方案探索 - 确定系统是干什么的	- 基本方案技术途径的演示验证 - 确定系统是怎样构成的(系统配置) - 初步设计 - 详细设计 - 生产准备 - 试生产 - 试验与评定	- 初步设计 - 详细设计 - 生产准备 - 试生产 - 试验与评定	- 批量生产 - 部署	- 使用与保障 - 改进与改型 - 技术状态控制
输出	正式的立项建议书,或不予立项的建议	- 系统技术要求 - 基本方案	- 系统体系结构 - 接口控制文件 - 功能基线	- 分配基线 - 全套设计文档 - 飞行器及其保障系统(原型)	批量生产出的飞行器及相应的保障系统	

1.2.7 退役处理

某些飞行器(如导弹、运载火箭等)使用一次后其寿命即告终止,但一般飞行器有一个较长的使用和维护阶段。当该阶段结束时,已陈旧的飞行器会予以淘汰,即进入退役处理。在退役处理时,飞行器或运至专用的弃置场所长期存放,或予以解体、销毁,从而结束其整个寿命周期。

第 2 章 系统工程过程

在本章中,将阐明下述问题:

- 为什么要在现代飞行器研制中运用系统工程过程?
- 什么是系统工程过程?
- 怎样在飞行器研制中实施系统工程过程?

2.1 系统工程过程在现代飞行器研制中的作用

在现代飞行器的设计中,不能仅对飞行器的功能和性能提出要求,而且在飞行器的总体方案论证中至少应回答下列问题:

- 飞行器(不论航空、航天,也不论军用、民用)将在什么地方使用——飞行器的使用分布或部署。
- 飞行器要执行什么样的飞行任务,如何完成其任务目标——飞行器的任务剖面或场景。
- 飞行器执行飞行任务的成功率或执行任务的效果如何——飞行器的效能。应当着重指出的是,现代飞行器除了应具有先进的飞行性能外,还应具有良好的战备完好性(如出动架次率,或使用可用性)和可信性(如任务成功率)。这样才有可能在任何一个时刻,需要飞行器去执行一项任务时,它能获得高的任务成功概率。
- 飞行器的飞行性能。
- 飞行器能被用户使用多久,并花费多少钱——飞行器的使用寿命和寿命周期费用。
- 飞行器在什么环境中使用等。

要回答上述问题(除了飞行器飞行性能外),只靠传统的飞行器设计学显然是不够的,必须运用系统工程过程,通过任务要求分析、功能分析和系统综合等一系列规范化的逻辑程序,才能最终得到满足系统性能和效能要求(包括硬件、软件、数据、人员、设施在内)的系统体系结构。

系统工程过程在飞行器研制中的作用极其重要。例如,我国与某西方国家合作为我国某型飞机开发一套机载航电设备,我国委托方向国外军方提出的需求只有 2 页纸,但该国军方通过“系统要求分析”这一子过程,就将这 2 页纸的需求,转化成了两本各 100 多页的文件:“技术要求文件 TRD”和“工作说明 SOW”。然后该国承包商以这两份文件为输入,经过系统工程过程的后续子过程(功能分析、系统综合等),输出的是 18 本数千页的设计文件,甚至包括人员培训的飞行员和地勤人员的培训计划(包括培训学时、培训的每节课程内容等)以及将装备(包括飞机)从国外运回我国的包装、运输的流程图等。亦即输出的是包括硬件、软件、数据、人员、



设施五大要素的飞行器系统。

2.2 系统工程过程的定义和特征

系统工程过程是一个技术处理过程,通过这一过程,按规范化的工作和决策的逻辑程序,将用户需求转化为对系统性能参数的描述和经过优选的系统配置。

系统工程过程具有以下特征:

- 系统工程过程是一个系统分析-系统综合的反复迭代、循序渐进的过程;
- 系统工程过程是一个规范化的过程;
- 系统工程过程也是一个并行设计的过程。

系统工程过程的主要目的是把用户需求转化为产品设计。

系统工程过程的输入是:

- 用户需求(包括用户提出的任务需求与使用要求等);
- 约束条件(包括使用环境及可用的技术基础等)。

系统工程过程的逻辑流程如图 2.1 所示。

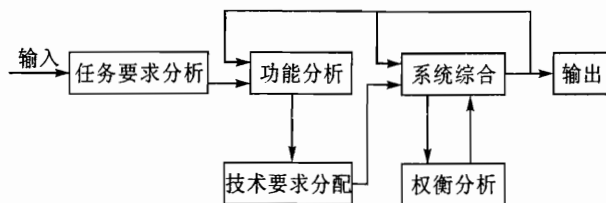


图 2.1 系统工程过程

由图 2.1 可知,系统工程过程包括:

- 任务要求分析——将用户需求转化为系统技术要求的描述。
- 功能分析和技术要求分配——功能分析是建立系统的功能体系结构,该体系结构可作为技术要求向下分配的框架;技术要求分配是依据功能体系结构的框架将系统技术要求分配到下层的分系统及部件和零件。
- 系统综合和权衡分析——系统综合是将功能体系结构转化为物理体系结构,并给出满足飞行器系统所需功能的硬件、软件及其接口;在由功能体系结构转化为物理体系结构过程中,需要通过权衡分析,从众多方案中选择出满足功能和性能要求的最优方案和系统配置。

通过系统工程一系列的技术处理活动,其最终的输出是:

- 系统体系结构;
- 技术状态项目体系结构;
- 系统和分系统规范。

2.3 系统工程过程的实施

2.3.1 任务要求分析

任务要求分析是将用户非常概要和粗略的需求转化为飞行器系统的具体要求,亦即确定其功能、性能、接口和其他要求,明确飞行器系统必须“做什么”,而不是给出解决方案。

任务要求分析是一个形成飞行器系统技术要求的渐进过程(见图 2.2),其输出是系统具体的技术要求。

任务要求分析的流程图如图 2.3 所示。

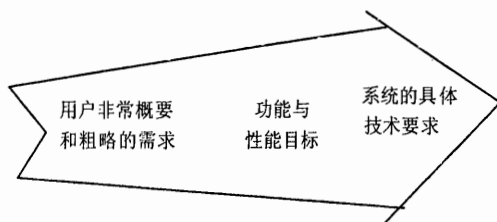


图 2.2 技术要求形成的渐进过程

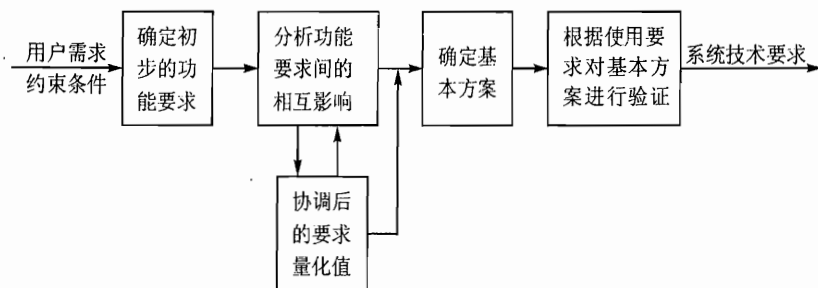


图 2.3 任务要求分析流程图

1. 确定初步的功能要求

在进行任务要求分析时,首先要根据用户需求确定对飞行器系统的功能要求。由于用户需求往往是非常概要和粗略的,因此,系统工程师必须用一组可度量的参数来明确地描述飞行器系统应具有的基本功能特性。

飞行器系统应具有的基本功能特性包括数量、质量、作用范围、时间和可用性/效能。

用来描述这些基本功能特性的可度量参数可以是:

➤ 描述数量的参数:

- 通信信道数/容量(对通信卫星);
- 运载旅客数(对民航客机);
- 运货量(对运输机);
- 发射有效载荷的能力(对运载火箭)。

➤ 描述质量的参数:

- 命中精度(导弹);
- 分辨率(侦察卫星);
- 噪声或误码率(通信卫星);
- 乘坐舒适性(民航客机)。
- 描述作用范围的参数:
 - 射程(导弹);
 - 航程(民航客机);
 - 作战半径(军用飞机);
 - 观察范围(侦察卫星)。
- 描述时间的参数:
 - 数据返回时间(通信卫星);
 - 满足发射窗口的概率(轨道航天器);
 - 从出发至到达城市间的飞行时间(民航客机);
 - 接到命令后的发射响应时间(导弹)。
- 描述可用性/效能的参数:
 - 使用可用度(军用飞机);
 - 飞机出勤率(民航飞机);
 - 发射准备就绪程度(导弹);
 - 可用时间/总时间的比率(载人空间站)。

表 2.1 列出了用可度量参数描述飞行器基本功能特性的几个示例。

表 2.1 用可度量参数描述飞行器基本功能特性示例

基本功能	可度量参数			
	观测卫星	通信卫星	民航飞机	导 弹
数 量	每日帧数、平方公里/日	信道数容量	运载旅客数	达到给定的杀伤概率所需数量
质 量	分辨率(m)	噪声或误码率	乘坐舒适性	圆概率误差(m)
作用范围	地理上的作用范围面积(km ²)	地理上的作用范围面积(km ²)	能提供服务的地理面积(km ²)	射程(km)
时间性	再观察时间、指令时间、处理与输送时间(min)	需用时信道可供使用时间(s)	在主要地点间的飞行时间(min)	需用时的发射响应时间(s)
可用性	需用时可供使用的可能性(%)	99.99%	飞机出勤率	发射准备就绪程度



2. 分析功能要求间的相互影响

飞行器系统的各个功能要求间是相互影响和制约的,在确定系统功能要求时,必须要考虑到这些相互影响和制约关系,并对这些功能要求进行充分的协调与权衡。

例如,在飞机设计中,用户会要求高的飞行性能,如高的飞行速度和爬升率,这就要求设计人员采用高的飞机翼载荷来满足这些需求;但用户又要求飞机的起飞、着落距离尽可能地短,飞机机动性尽可能地高,这就意味着设计人员要采用低翼载荷来满足后面的需求。而这两者之间是有矛盾的,必须充分分析这些功能要求间的相互影响。又如弹道导弹的命中精度和弹头威力两项指标也是相互影响与制约的。对于使用方规定的对目标杀伤效果的要求,在命中精度指标能够提高的情况下,弹头威力就可能小一些,进而又影响到射程指标。如果命中精度低,则势必要求提高弹头威力,这样会使弹头重量增加,对推进器的推力又有影响等。

因此,系统工程师必须运用参数分析方法,定量地描述相互影响的各个系统参数之间的相关关系,并分析各个参数变化时对其他相关参数的影响趋势,最终得到初步满足用户需求的,经协调、权衡后的技术要求及其量化值。

3. 确定基本方案

在初步确定了系统的功能要求及其量化值后,就可按照技术要求确定系统的基本方案。系统的基本方案决定了系统要完成预定任务时所需的各组成要素和工作过程。系统的基本方案一般是以功能框图的形式来描述的。

图 2.4 所示是一个导弹系统的基本方案,即该导弹系统的功能框图。该框图展示了该导弹系统的各组成要素(如导引头、电源、惯性平台等)、各组成要素间的输入与输出关系及其工作流程。

应当注意的是,图 2.4 中展示的各个单元仅是功能单元,即只描述功能,不涉及实现这些功能的硬件、软件 and 具体方法。也就是在本阶段,系统工程师只回答“做什么”,而不涉及“怎样做”。

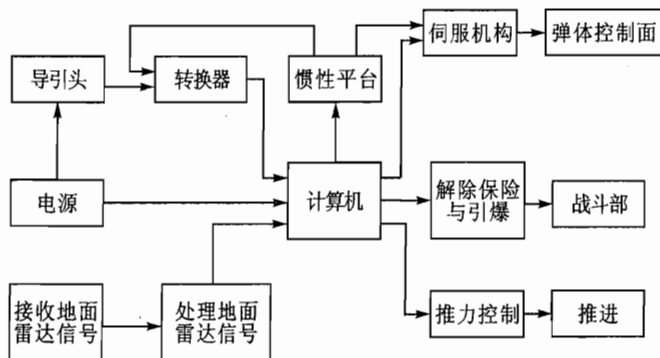


图 2.4 某导弹系统的功能框图

4. 根据使用要求对基本方案进行验证

用功能框图可以简单地描述某个飞行器系统的基本技术方案,但这是不完善和不完整的。它只涉及构成这个飞行器所需的硬件、软件和一些数据。而构成一个完整的飞行器系统必须包括硬件、软件、数据、人员和设施 5 大要素。而这 5 大要素必须经由技术方案、使用方案和保障方案的分析与设计后方可得出。

由 3 大方案得出系统 5 大要素的工作流程图如图 2.5 所示。

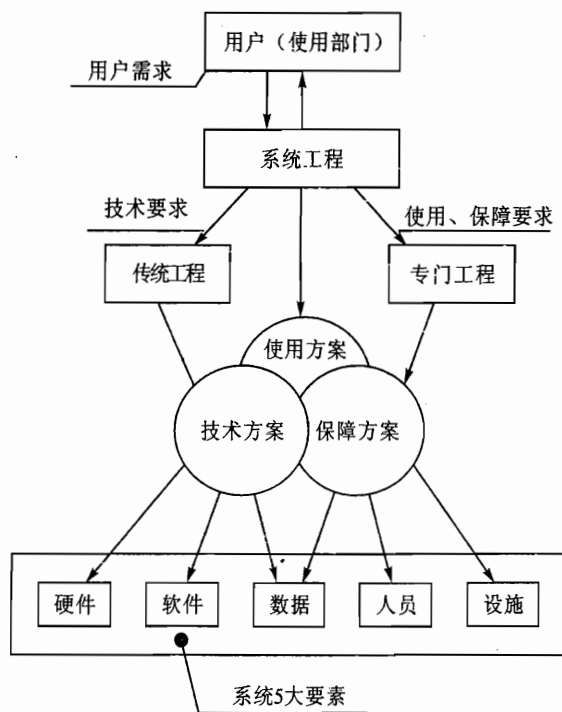


图 2.5 3 大方案与系统 5 大要素

(1) 用户(使用部门)提出需求(如战术、技术要求),但这是非常概要的需求。

(2) 用户或使用部门还应提出(或在研制部门的协助下提出)使用方案。

对军用系统(如军用飞机)来说,一般的使用方案应包括:

① 根据对战争威胁与作战任务的分析结果,说明飞行器应完成的任务及其性能范围,以及该系统的工作方式。

② 预期的作战环境或使用环境条件,包括地面的环境条件和空中的环境。

③ 确定任务剖面及飞机的使用率(包括年使用天数、平均任务时间周期等)。

④ 描述飞行器系统效能的评估准则,即可用性或战备完好性等指标要求。

⑤ 系统的部署计划(布点情况)及应用方案,例如各作战基地的配置:每一基地所需的装



备(飞机)数量、离保障设施的距离、补给线的长度等。

(3) 飞行器研制部门的系统工程师应从用户的需求中分析出技术要求和保障要求,并分别下达给飞行器设计部门中的传统工程设计部门(总体、结构设计部门)和专门工程设计部门(可靠性、维修性、保障性设计部门)。

(4) 传统工程部门应按系统工程部门下达的技术要求设计出飞行器系统的基本技术方案。同时,专门工程部门按照使用、保障需求及使用方提供的使用方案制定飞行器系统的保障方案。

(5) 在对使用方案、技术方案、保障方案综合权衡的基础上,确定系统的硬件、软件、数据、人员、设施等5大要素。

必须指出:鉴于技术方案与保障方案之间有很强的关联关系,要得出一个最终的技术方案和保障方案,并在将来交付飞行器的同时交付一个有效的保障系统,传统工程部门与专门工程部门之间必须进行良好的协调和合作。最好的做法是成立一个由这两个部门设计人员共同组成的综合产品研制小组(Integrated Productive Team, IPT)共同来制定飞行器的技术方案和保障方案。

5. 任务要求分析的输出

任务要求分析的输出是系统技术要求,包括系统的下述要求:

- 功能要求(阐明系统应当做什么);
- 性能要求(描述功能执行的良好程度);
- 接口要求(描述系统内部各功能间的关系和系统与环境的关系);
- 保障要求(阐明系统在使用中如何得到保障);
- 其他要求(质量要求等)。

上述各类要求应满足下述特性:

- 完整性:系统技术要求应包括完整的功能要求、性能要求、接口要求、系统的任务剖面、使用与保障要求及质量要求等内容。
- 可行性:系统技术要求应在给定的费用、研制周期条件下,在技术上是可能达到的。
- 一致性:系统技术要求的表达应当无歧义,要求之间不应相互矛盾。
- 可验证性:系统技术要求必须最终能被验证,因此不准采用像“较高的可靠性”、“有效的保障能力”之类的用词。期望的性能和功能也应当用能被验证的目标来表述。
- 系统技术要求只规定“做什么”,而不应规定“怎样做”。

最终的系统技术要求可用系统技术要求文件或系统规范的方式予以表达,其格式如图2.6所示。

1. 适用范围	3.3.2 接口技术要求
2. 引用的文件	——外部接口描述
3. 系统技术要求	——硬件/软件接口描述
3.1 系统定义	3.3.3 约束条件
3.1.1 系统的任务	——政府提供的设备
3.1.2 系统使用要求和使用场景	——商用现成设备
3.1.3 系统技术状态配置	——计算机芯片、编程语言等
3.1.4 系统效能度量	3.4 质量要求
3.2 系统功能特性	3.4.1 可靠性要求
3.2.1 系统的功能	3.4.2 安全性要求
3.2.2 系统的性能	3.4.3 测试性要求
3.2.3 系统功能间的关系	3.4.4 可用性要求
3.2.4 系统分配给硬件实现的功能	3.4.5 其他质量要求
3.2.5 系统分配给软件实现的功能	3.5 使用、保障要求
3.3 系统的物理特性	3.5.1 保障方案
3.3.1 物理技术要求	3.5.2 保障设施
——重量、重心	3.5.3 供应保障
——尺寸、体积……	3.5.4 人员与培训要求
	4. 验证与验收要求

图 2.6 系统规范的纲目示例

2.3.2 功能分析

通过任务要求分析可得到初步的系统技术要求,但如何将系统的技术要求分配给较低层次的功能区,以及如何通过较低层次的功能逻辑程序来实现系统上层的功能要求,则需要通过功能分析才能实现。

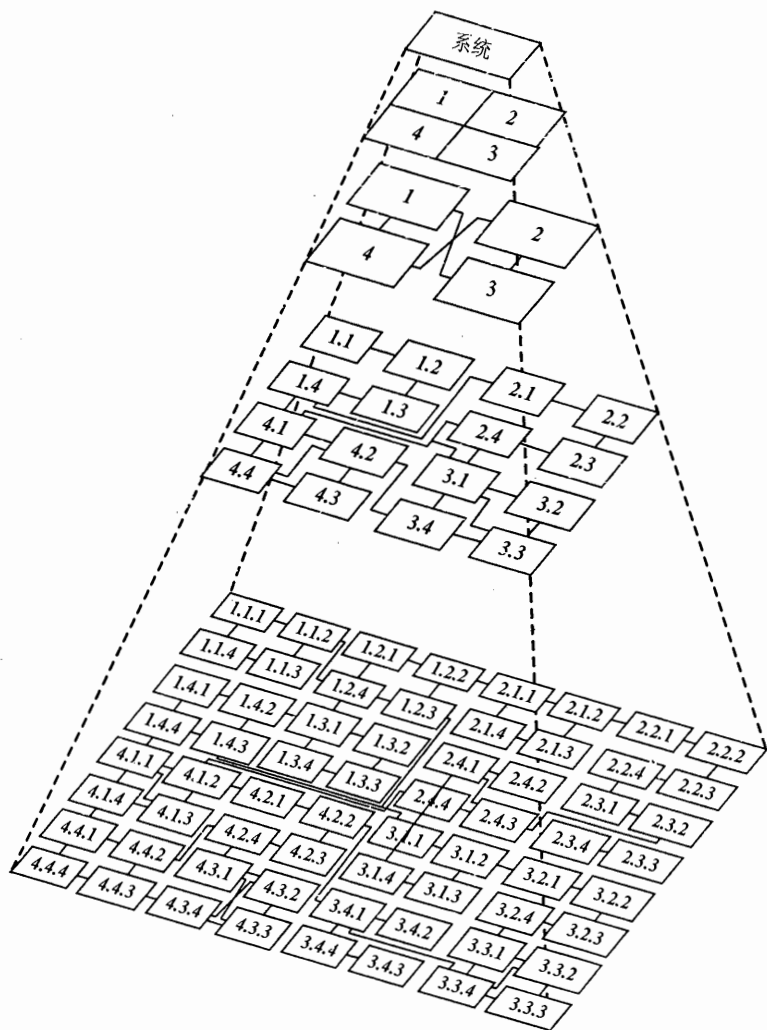
功能分析是确定、描述一个系统为完成其目标所必须具有的功能并建立功能之间关系的过程。功能分析将功能按逻辑程序进行自上而下的分解。在系统工程过程中,功能分析具有下述作用:

- 导出系统各个层次(分系统、部件、组件……)必须满足的要求;
- 确定系统所有层次的功能、性能和其他技术特性的度量;
- 为技术要求的向下分配提供一个可用的框架。

功能分析的框架是功能体系结构,功能体系结构是系统功能自上而下的一个层次分解结构。它不仅要示出系统与分系统必须执行的功能,而且还会示出上下层功能之间与同一层次诸功能之间的逻辑程序,如图 2.7 所示。由图可知,它也是一个层次体系结构。

功能分析的目的是要通过功能体系结构对功能进行分解,将高层次功能分解为低层次功能,并描述系统按一定的逻辑程序(流程)做些什么和有怎样的性能。

应当指出:功能体系结构是面向功能的,而不是面向解决方案的,即只确定实现系统的功能,而不涉及实现功能的具体硬件/软件配置。

图 2.7 功能体系结构^[1]

在功能分析时，主要的分析工具有功能流程框图、时间序列分析和 N^2 图。

1. 功能流程框图

功能流程框图是一个分层的逻辑序列图(见图 2.8)。它的主要作用是：

- 确定满足用户需求的总体功能(用顶层功能流程图表示)；
- 确定满足每一上层功能所需的下层功能序列(用第二、第三层及更下层的功能流程图表示)；
- 确定每个功能间的序列(顺序执行、并行执行、选择执行……)；
- 确定每个功能间的输入和输出。

图 2.8 是功能流程框图的一般示例。图 2.9 是一个空间飞行器功能流程框图的典型示例。

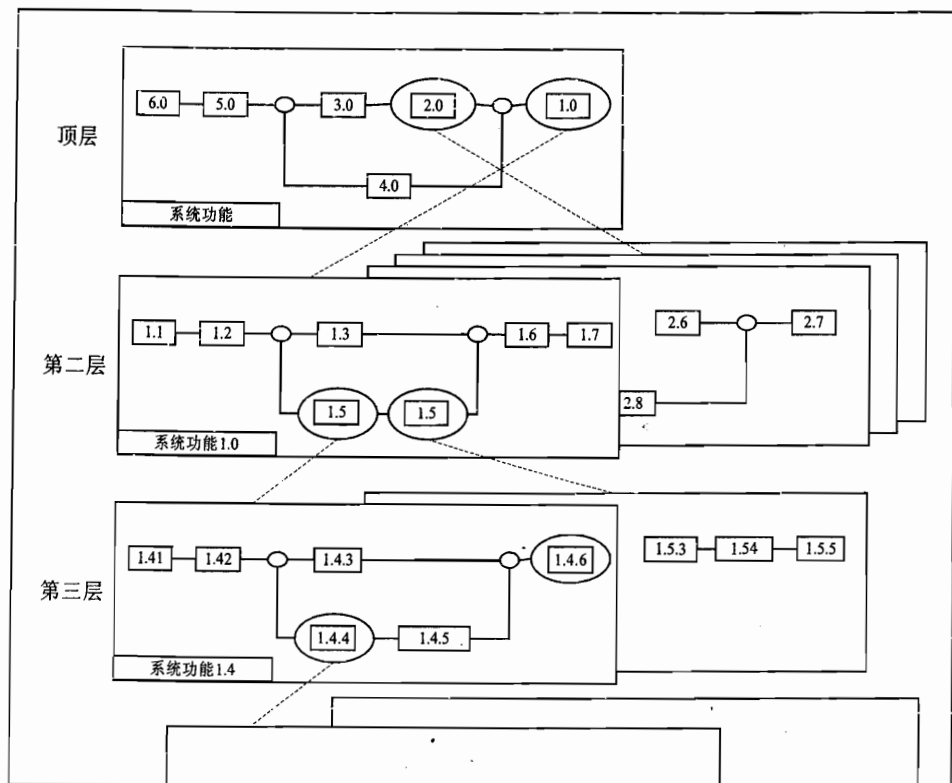


图 2.8 功能流程框图示例^[3]

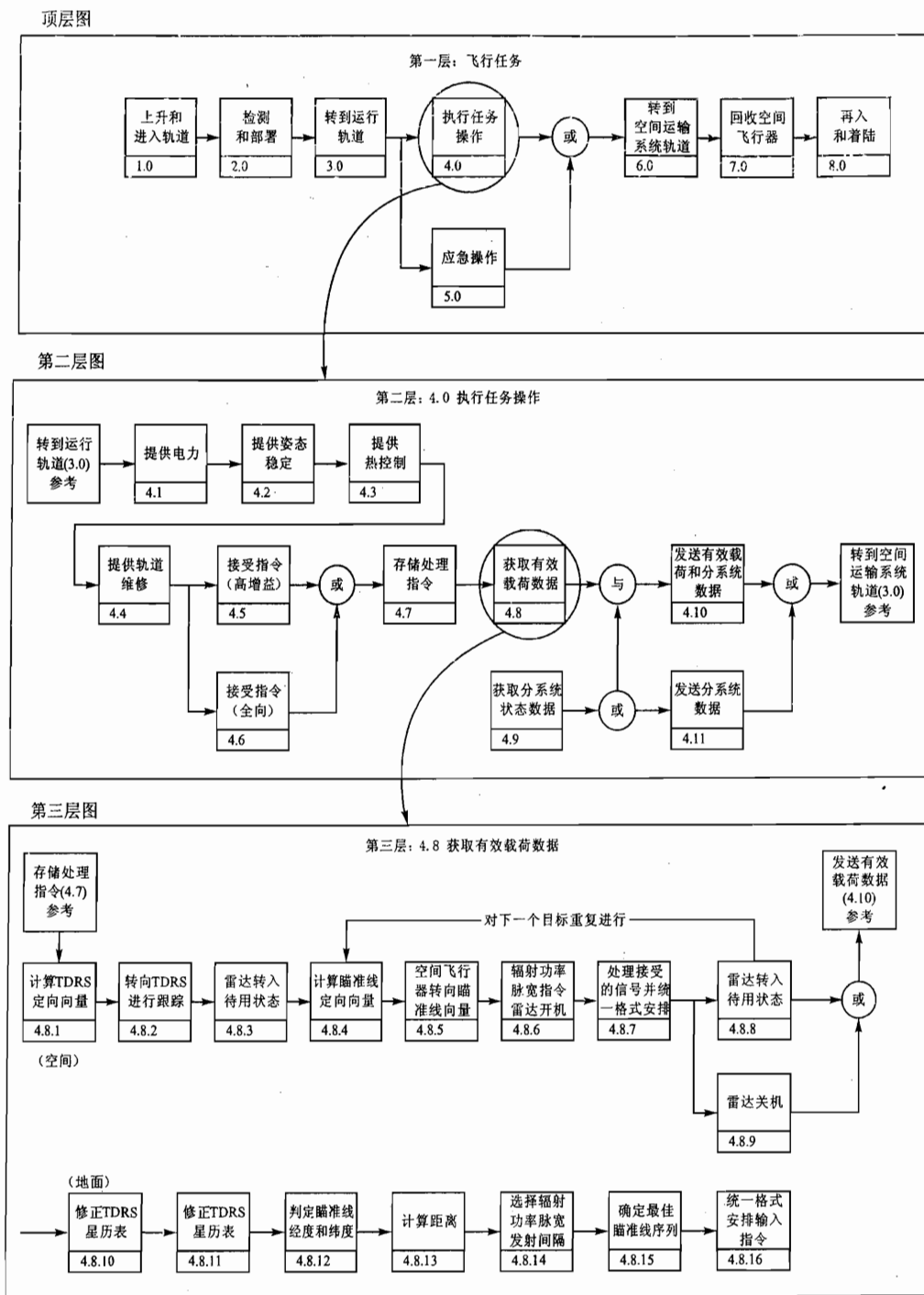
2. 时间序列分析

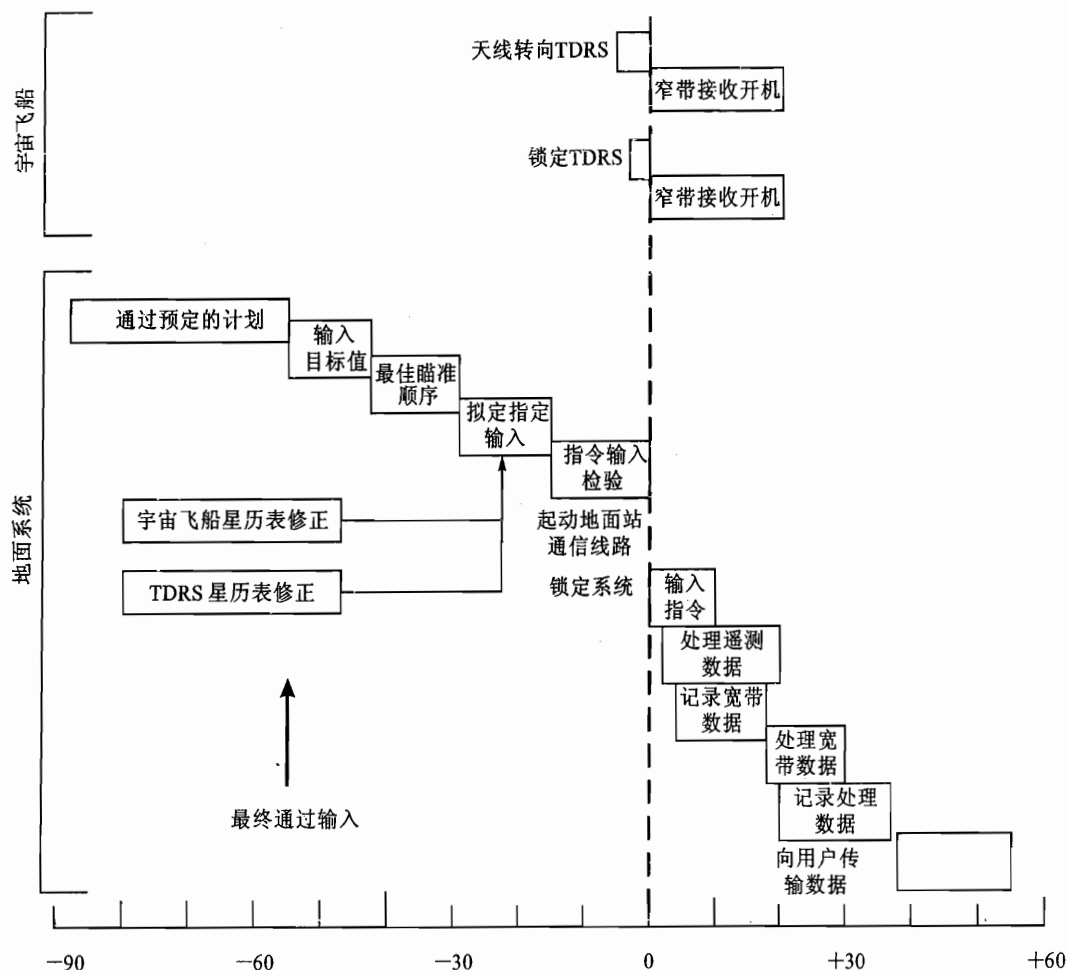
功能流程框图只表示了功能之间的逻辑程序和层次关系，而没有表示出各个功能的持续时间。而时间序列分析方法可以确定各功能执行的时间先后关系和持续时间的长短。

时间序列分析可定义功能的时间序列(如并发、重叠、顺序)关系，用来确定诸如飞行任务控制、飞机再次出动准备时间、维修停机时间等。

图 2.10 显示的是某宇宙飞船的飞行任务时间序列分析，0 s 是其发射时间，从倒计时 90 min 开始到发射及发射后 55 min 内的各个功能按时间顺序和持续时间进行了排列，可为发射前后的飞行任务控制提供清晰的时间顺序。

图 2.11 是维修停机时间序列分析。若要求一个 4 人维修组在 1 h 内维修完一个设备，从图中分析可知，A、B、C、D 四人维修时间为 55 min，可满足要求。若因需要将维修时间缩短为 40 min，则应分析其最长工作时间(08 序列任务：更换 V 型皮带，需 54 min)。图 2.11 的分析结果表明，在 4 人维修的情况下，该要求不可能满足。若一定要满足该要求，就应增加维护

图 2.9 某空间飞行器功能流程框图示例^[1]

图 2.10 飞行任务时间序列分析^[1]

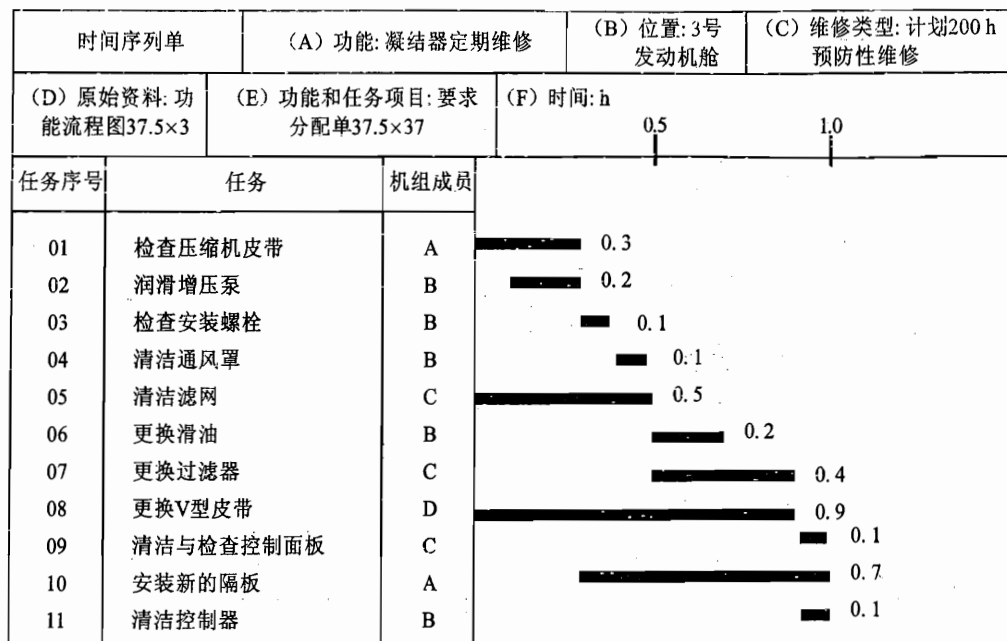
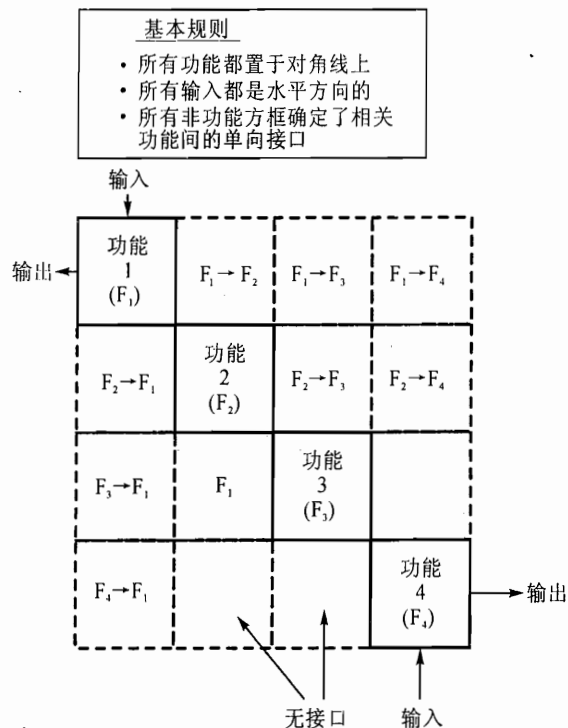
人员。

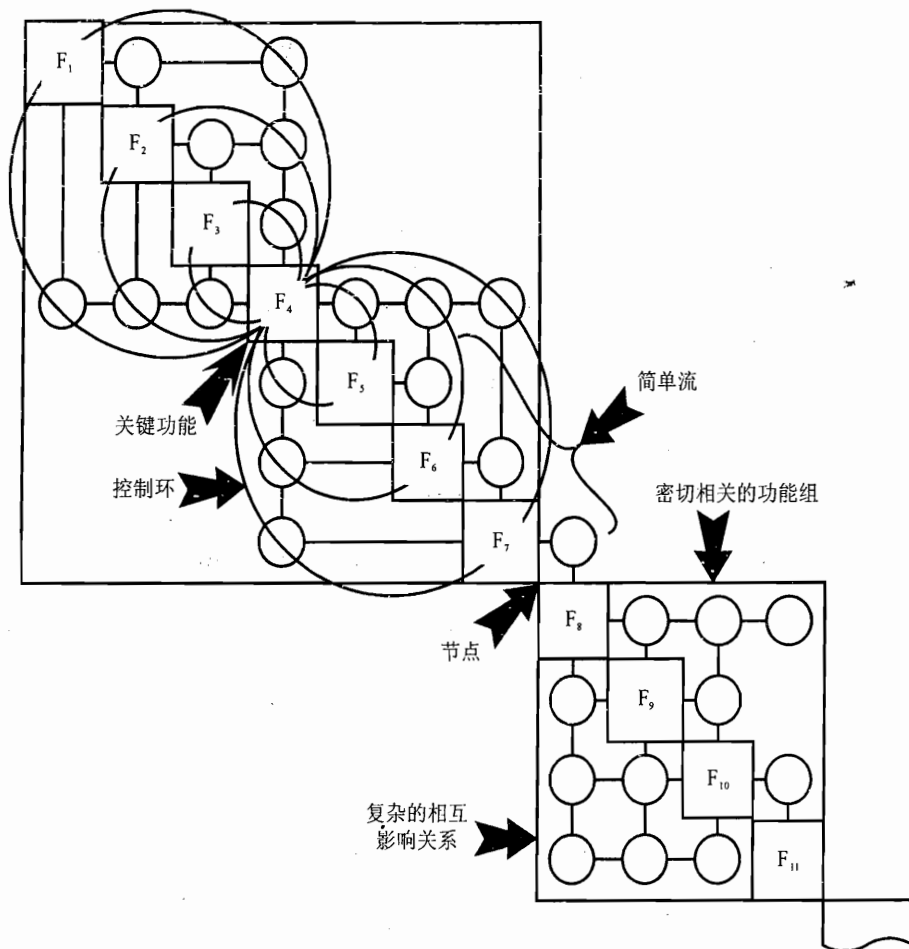
3. N^2 图

N^2 图可以判定功能间的输入-输出关系,并找出其中的关键功能。

N^2 图的基本规则及其示意图如图 2.12 所示。

由上述基本规则绘制的功能 $F_1 \sim F_{11}$ 的 N^2 图如图 2.13 所示。由图 2.13 可知:在所有功能中, F_4 是关键功能,因为它有从 F_1 、 F_2 、 F_3 、 F_5 、 F_6 、 F_7 来的 6 个输入,并且向这 6 个功能均有输出。图 2.13 还显示了两组功能间的关系: $F_1 \sim F_7$ 功能组与 $F_8 \sim F_{11}$ 功能组之间有交联,它们是通过 F_7 与 F_8 之间的节点相交联的。

图 2.11 维修停机时间序列分析^[1]图 2.12 N^2 图的基本规则及其示意图

图 2.13 N^2 图示例^[4]

4. 技术要求的向下分配

技术要求分配是系统功能分析过程的一个必然的逻辑延伸,也是与它不可分割的内容。进行技术要求的向下分配,是以系统的功能流程图和其他功能分析结果为依据进行的。技术要求的向下分配,应一直进行到能够根据分配的具体技术要求开展硬件或软件项目的设计为止。

技术要求的分配就是将描述系统功能和性能要求的参数及其指标向下分配。在技术要求分配中,有的参数和指标可以直接向下分配,如飞行器的总重,可直接根据功能层次体系分解为机翼、机身、发动机等分系统的重量,然后逐级下传,一直分配到部件、组件、零件为止。

但还有一些参数则不能直接进行分配,需要经过一定的分析和转换后才能向下分配。如对飞行器系统提出的使用可用性(A_0)和平均维修间隔时间(MTBM)是飞行器的总体技术参

数,无法直接向下分配到分系统和部件,需要根据具体情况转换为对应分系统或部件的可靠性、维修性参数才能进行分配。为此,要求进行可靠性参数指标和维修性参数指标间的权衡分析,还可能要利用计算机进行仿真研究,以判定分配的一组参数值能否满足飞行器系统的可用性(A_0)和维修性(MTBM)的指标要求。

下面介绍可靠性、维修性、可用性要求的下传与分配。

例如,对某飞机,给定的战备完好性指标是 $A_0=0.83$, MTBM=450 h, 每小时维修工时 MMH/OH=0.2。

当技术要求分配到3个分系统时,需将战备完好性指标 A_0 等转换为可靠性、维修性指标 MTTR 等(见图 2.14)。再下传到部件层次时,可靠性指标要转换成故障率 λ 。

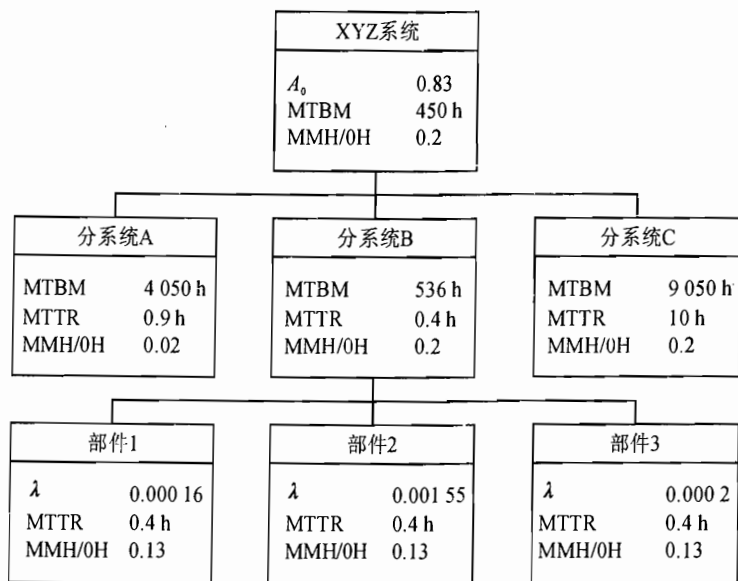


图 2.14 可靠性、维修性、可用性要求的下传与分配

技术要求分配到各个层次的各单元后,应以技术要求分配单的形式下达相关的设计部门。

技术要求分配单记录分配的功能、性能和飞行器系统之间的关系,它给出功能分析和技术要求分配的结果。为保持功能的可追溯性,技术要求分配单中的功能编号与功能流程图中的编号应一致。

2.3.3 系统综合

系统综合是将功能分析中形成的功能体系结构转化为物理体系结构的过程。只有通过系统综合方能给出满足飞行器所需功能的硬件、软件及它们之间的接口。因此,系统综合也是一



个定义组成系统的硬件和软件来实现系统功能的过程。

1. 系统物理体系结构的特征

系统物理体系结构具有下述特征：

- 物理体系结构是功能体系结构的映射和实现。因此,物理体系结构中的每一个硬件和软件至少应满足一项功能要求,或功能要求的一部分。当然,一个物理单元也可满足多项功能要求。物理体系结构和功能体系结构的映射,可以用图 2.15 所示的关系矩阵来表示。
- 物理体系结构可以用原理框图的形式来描述。
- 物理体系结构是否构建得合理、有效,应通过权衡研究进行分析。
- 物理体系结构必须描述物理单元(硬件、软件)之间的接口关系。
- 物理体系结构可作为制定飞行器系统的工作分解结构(WBS)和制定规范体系的基础。
- 现代飞行器的物理体系结构设计应遵循模块化设计和开放性的设计原则。

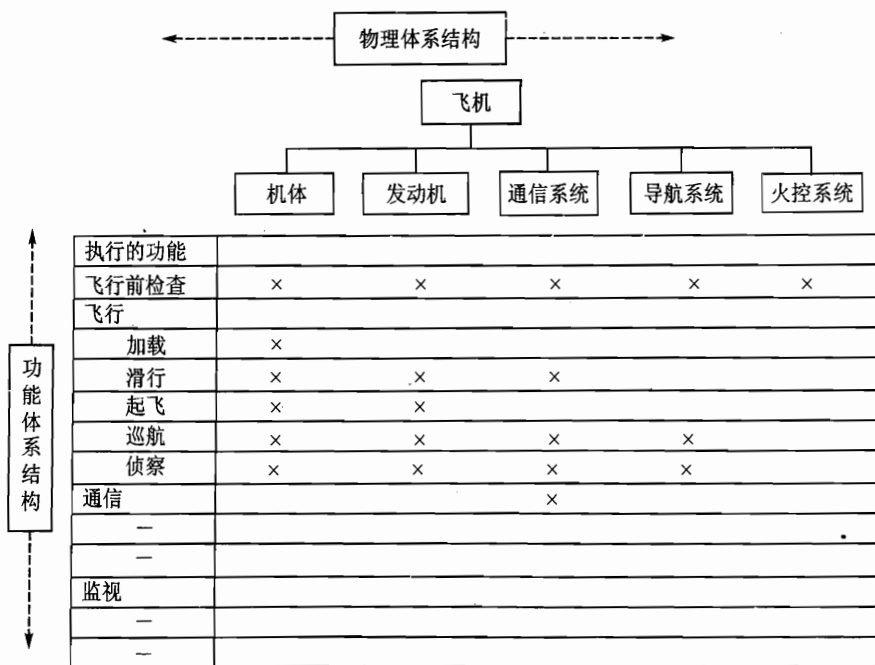


图 2.15 功能体系结构和物理体系结构的矩阵关系^[3]

2. 系统综合的工具和方法

系统综合的输入是功能体系结构;其输出则是物理体系结构,包括组成系统的部件与单元(含硬件、软件)以及各单元的接口和接口控制文件。

常用的系统综合工具包括:

(1) 原理框图：定义并描述组成系统的部件、单元及其相互关系。

(2) 仿真建模：对构建的物理结构进行观察、优化、预测和评价。

(3) 接口控制文件：定义和描述系统各部件的接口。

1) 原理框图(SBD)

SBD 描述硬件部件和软件部件的组成及其相互关系。它是由上至下逐级展开的。SBD 描述按功能体系结构确定的功能和性能要求相适应的物理构型,它可以表示构成该部件的硬件和软件的单元,以及系统各部件与其他系统或分系统之间的接口。

系统综合是一个创造性的过程。因为满足每一项规定功能要求的物理体系结构可以由多种硬件/软件体系结构来实现,许多功能既可以用硬件来满足,也可以用软件来实现,所以必须在系统综合过程中运用权衡分析的方法,从众多可行方案中权衡出一个最好的方案,亦即选择满足功能要求的最优的物理体系结构。例如,对于图 2.4 中给出的某导弹系统的功能框图,在将该功能框图转换成原理框图时,就需要进行仔细的权衡分析和多方案选择。例如,在图 2.4 中,我们给出了推进功能,而实现推进功能可采用固体火箭发动机、冲压喷气发动机、涡扇喷气发动机等。因此,必须通过权衡后,最终选择一种发动机(如固体冲压喷气发动机)来实现推进功能。

同理,在选择导引头时,可供选择的有红外导引头、激光导引头、雷达主动导引头、雷达半主动导引头等,也需经权衡分析后,选择出最恰当地满足导引功能的导引头(如红外导引头)。

权衡分析的具体方法可参见 6.1 节。

经过系统综合和权衡分析后,图 2.4 的功能框图就转换成该导弹的原理框图(见图 2.16)。

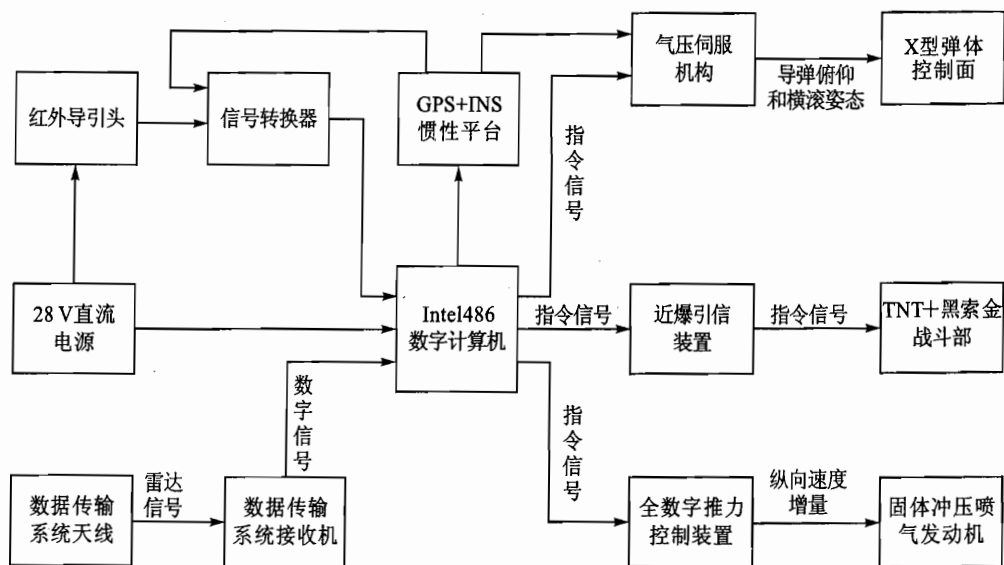


图 2.16 某导弹系统的原理框图



图 2.16 是该导弹系统的原理框图,图中显示的是组成部件的各物理单元,而不是功能了。利用原理框图最终可以确定构成系统的每个单元。即确定只产生单一、独立的功能,并在一定的逻辑程序内执行单一的任务且能单独加以测试的单元。

2) 仿真建模

原理框图虽是最基本的进行系统综合的工具,但由于飞行器系统的复杂性,仅依据原理框图还不足以对系统的全貌及其技术细节作出更为确定的描述,因而也很难仅仅根据原理框图使设计工作进行到有把握地投入实际生产的阶段。因此,利用其他建模手段,对于深入了解并解决工程问题同样是绝对不可缺少的。

模型是对真实世界的某种理想化的表述,是以物理的形式或数学的形式去代表一个系统或是它的一个组成单元,用来展示它们的工作状态。利用模型不但可以帮助设计人员去分析问题,进行预测和工程验证,控制设计工作的进展和实施培训,还有助于不同专业人员之间的相互沟通。

(1) 物理模型

物理模型能以各种物理手段提供关于系统的较为直观的描述,因而是飞行器研制中最常用到的一大类模型。物理模型可以是与真实产品一样大小的实体模型(如样机),也可以是缩比模型(如风洞模型);既可能是某种形式的像真实体,也可能是系统的某种模拟性的代表物。

(2) 数学模型

数学模型是真实系统的一种抽象的代表,它以一系列数学方程去描绘系统的状态。在这些数学方程中,用理想化的参数去代表系统各个组成单元的特性。利用数学模型可以定量地分析和确定各备选设计方案的性能特性。

根据不同的分析目的,可以采用不同类型的数学模型作为分析手段。静态模型只能表示稳态的情况,比如分析加载后处于平衡状态的机械结构受力情况;动态模型则可用来展示系统状态随时间变化的情况,如分析飞机自动驾驶仪在控制飞机飞行时的工作状况。

简单的数学模型可以用解析方法求解,但复杂的大型系统的数学模型,要用数字计算机进行运算。

进行数字仿真虽然有时候要消耗大量的机时,但它仍然以其具有的及时性、通用性和灵活性(易于改变参数值)而在各个工程领域得到了广泛的应用。比起建造一个真实的系统进行试验,进行数字仿真还是省钱、省时的。数字仿真又有连续数字仿真和离散数字仿真之分,在进行离散数字仿真时,还可以根据需要采用过程仿真模型和事件仿真模型。

建模和仿真能够在与系统真实环境逼真的条件下模拟产品及其过程,并给出设计方案(硬件和软件设计)的性能及其运行结果的预测。

通过模型与仿真可以优化硬件和软件的方案和参数,预测性能,并给各部件与单元分配最佳的性能要求,从而得出一个符合功能需求的物理体系结构。

3) 接口控制文件(ICD)

接口控制文件描述组成飞行器系统各单元间的输入、输出信息和物理接口间的链接。接口控制文件包括接口控制图、接口要求规范等文件。它描述相关部件或单元间的物理接口和功能接口,以建立并维护有接口关系的部件或系统(包括硬件、软件)之间的相容性。

在接口设计中,应尽可能地采用通用的商用接口,并降低专用接口的数量。应将开放式系统的方法作为接口设计的重要部分,即应选用在工业界中认可的规范或标准来定义系统的内部接口和外部接口。

在接口控制文件中,可以采用接口控制图的方式来描述系统各部分接口的关系。图 2.17 所示是一个典型的空间系统的接口控制图,而图 2.18 所示是一个软件配置项的外部接口图。

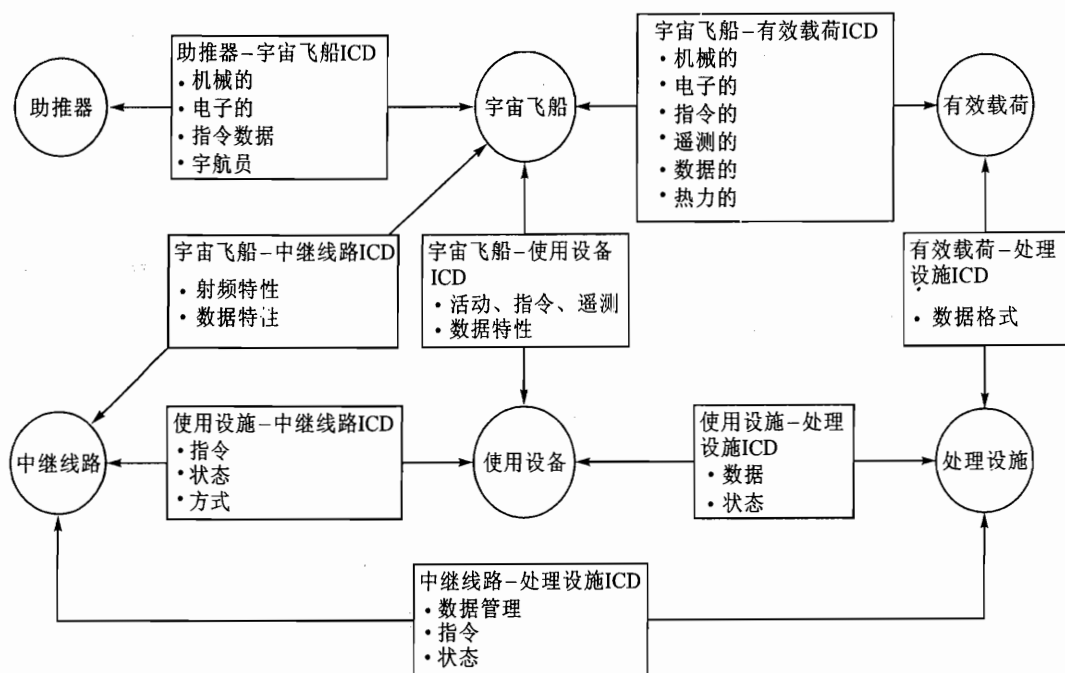


图 2.17 典型的空间系统接口控制图^[10]

接口控制文件的规范格式可参见图 2.19。

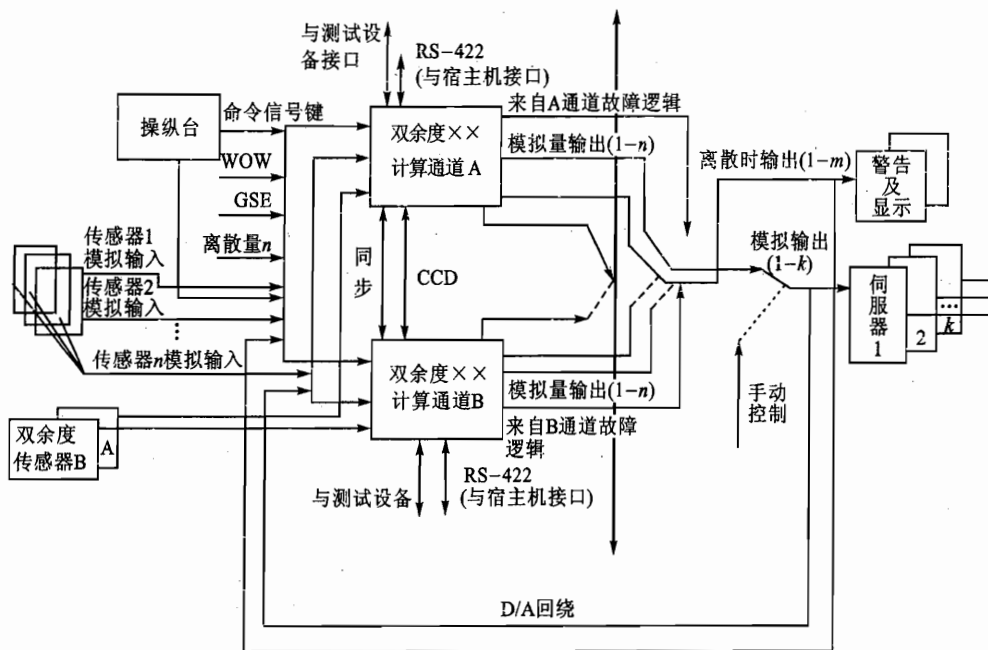


图 2.18 软件配置项的外部接口

1.0 范围: 目的与适用范围	3.5 软件
2.0 适用的文件	3.5.1 数据: 输入、输出、速率、精度
3.0 技术要求	3.5.2 信息: 格式、内容、存储
3.1 物理接口	3.5.3 规程: 启动、数据处理、证实、错误检测
3.1.1 机械技术要求: 外壳、附件、遮蔽、调准	3.6 硬件/软件
3.1.2 主要工装	3.6.1 接口: 图表、标准和惯例
3.1.3 质量参数: 重量、惯性矩、重心配置、轴线、模型转换 (数学的/物理的)	3.6.2 定时与时序: 控制与逻辑、相关关系、数据转换、输入数据读出
3.2 电子接口	3.7 环境
3.2.1 指令信号: 形式、速率、识别	3.7.1 结构: 振动、冲击、声振、加载、动态模式形态
3.2.2 数据信号: 射频特性、形式、速率	3.7.2 热: 温度范围、热速率、传热面
3.2.3 遥测信号: 形式、时钟、识别、记录	3.7.3 磁: 磁通量密度、变化率
3.3 电气接口	3.7.4 辐射: 类型、通量密度、总剂量
3.3.1 电功率: 类型、电压、功率范围、保护措施	3.7.5 工作环境: 压力、温度、密封度
3.3.2 接口引线分配	3.7.6 大气条件: 温度、气流速率
3.3.3 电磁兼容	3.8 安全性
3.4 液压/气压接口: 类型、流率、温度、压力	3.9 使用限制
	4.0 质量保证
	4.1 质量保证技术要求
	4.2 ICD 技术要求验证

图 2.19 接口控制文件的格式示例

2.3.4 系统工程过程的输出

系统工程过程的输出是确定系统要求和设计方案的各类文件,包括各子过程的输出、最终输出和规范。

1. 系统工程过程中各子过程的输出

- 任务要求分析中输出的系统技术要求文件;
- 功能分析中输出的功能体系结构和技术要求分配单;
- 系统综合中输出的物理体系结构、权衡分析报告和接口控制文件等。

系统工程过程中各子过程的输入、分析工具和方法及输出可参见图 2.20。

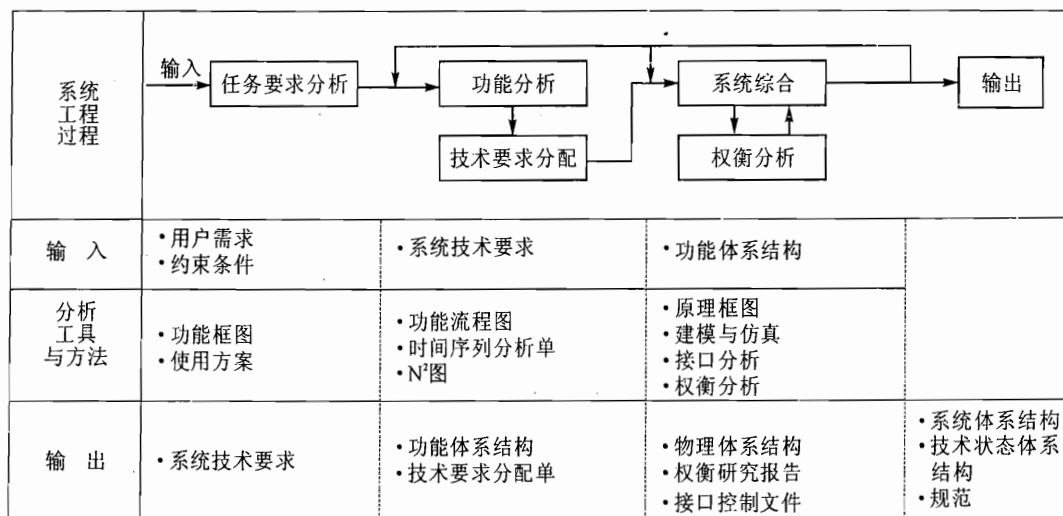


图 2.20 系统工程过程中的输入与输出结果

2. 系统工程过程的最终输出

系统工程过程的最终输出包括描述整个系统的系统体系结构和技术状态项目体系结构。

(1) 描述整个系统的系统体系结构

该系统体系结构应包括组成飞行器系统的硬件、软件和数据,以及维护保障该飞行器所需的设备、设施和人员,即构成飞行器系统的 5 大要素,如图 2.21 所示。

(2) 技术状态项目(configuration item)体系结构

该技术状态项目体系结构一般是以工作分解结构(WBS,详见第 6 章的描述)的形式来描述的。该体系结构以清晰的树状层次结构表明组成飞行器系统所有的单元,这些单元不仅包括飞行器产品(硬件、软件等),还包括研制该飞行器所必需的各项服务(如系统工程管理、项目管理、人员培训)和数据、资料等,如图 2.22 所示。

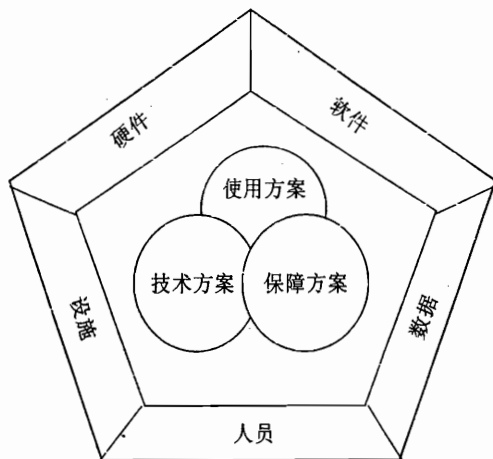


图 2.21 系统体系结构示意图

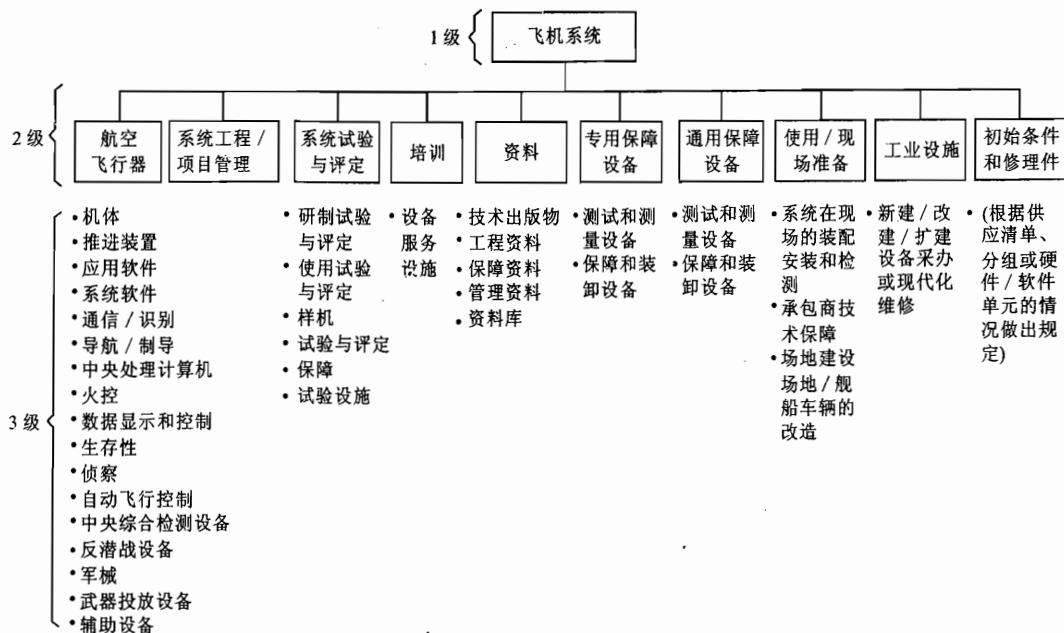


图 2.22 工作分解结构示例(摘自 MIL - HDBK - 881)

3. 规范

规范是系统工程过程中产生的最重要的技术文件,是飞行器研制、试验与鉴定和生产的依据。规范的作用是用来确定整个飞行器系统、分系统和部件的技术要求与质量要求。规范一般可分为系统规范、研制规范、产品规范、工艺规范和材料规范。

在系统工程过程中,输出的主要是系统规范和研制规范。

系统规范是阐述飞行器系统的技术要求,并将这些技术要求分配给各个分系统(或技术状态项目)。在系统规范中规定了飞行器系统的性能和功能技术要求,从而也就确定了系统的基本属性。系统规范为飞行器系统确立了功能基线。

研制规范是应用于系统第一、二层次(即系统、分系统)以下各较低层次单元的技术文件。它详细地阐明各个单元的性能、接口和其他技术要求,以便据此进行各部件的设计、研制、试验和鉴定。研制规范还规定了各部件的功能范围、质量保证和验收要求。研制规范一经批准,就成为技术状态管理中的分配基线。

第3章 工程专业的综合

3.1 概 述

飞行器系统的研制是一项高度综合性的工作,是由不同阶段的工作内容组成的系统工程过程,而进行系统综合则是贯穿系统工程过程始终的核心工作内容。此处所说的综合,从技术的角度看,主要是指以主体工程设计工作为主导的与各有关工程专业的综合。

飞行器系统的质量,除了功能性外,还包括可靠性、耐久性、环境适应性、维修性、测试性、保障性、安全性和经济性。

除了功能属性以外,其他属性均直接或间接与故障密切相关:

- 可靠性、耐久性和环境适应性从不同角度代表了产品无故障工作的能力;
- 维修性代表了便于预防和修复产品故障的能力;
- 测试性代表了诊断产品故障的能力;
- 保障性代表了针对故障的维修保障能力;
- 安全性代表了故障安全能力。

各有关的工程专业,是指应用某一特定范围的科学技术知识去支持并影响系统设计工作的那些专门的业务内容。在设计中融合进这些特定的工程专业工作,亦即通过这些工程专业的工作对飞行器主体设计施加必要的影响,从而保证研制出的系统在实际的使用环境中更为合理并具有更高效能。这些工程专业几乎覆盖了与研制各系统都相关的内容,如可靠性、维修性、安全性、可生产性、保障性、人素工程、零件/材料/加工、可运输性、电磁兼容性等。从产品全寿命周期一开始的立项论证阶段,就应该将有关的工程专业融入设计工作之中,从全系统、全过程、全特性的视角对工程专业综合进行规划与实施,如图 3.1 所示。

如图 3.2 所示,工程专业综合所涉及的可可靠性、安全性、维修性、测试性、保障性和战备完好与任务成功(持续能力)的兴起和发展是与军事需求密切相关的。

工程专业综合最重要的专业体现在可靠性、安全性、维修性、测试性和保障性等领域。如图 3.3 所示,通过抓住“故障”这一核心要素,将与故障相关的理论和技术发展成为一门相对独立的工程技术学科。

工程专业综合是研究产品全寿命过程中与故障作斗争的工程技术,它运用系统科学与系统工程的理论和方法,从系统的整体性及其同外界环境的辩证关系出发,研究产品发生故障的机理与规律、预防、预测、诊断与修复产品故障的理论与方法,并运用这些机理与规律、理论与方法开展一系列相关的技术与管理活动。

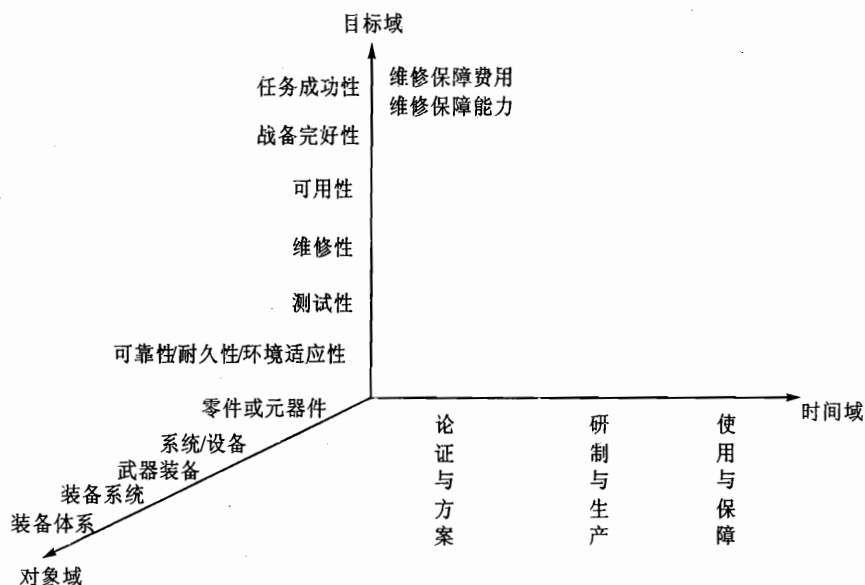


图 3.1 全系统、全过程、全特性工程专业综合

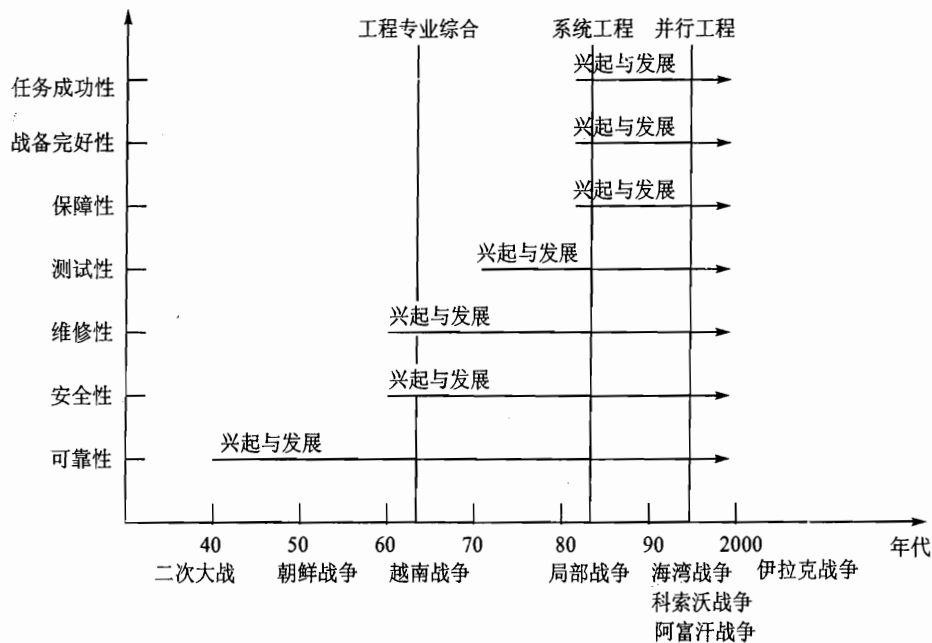


图 3.2 工程专业综合的兴起与发展

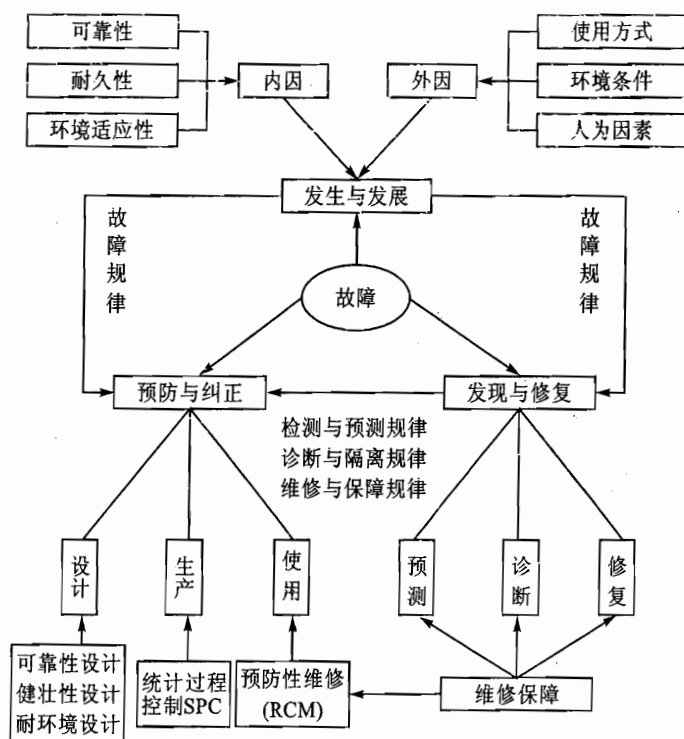


图 3.3 以故障为核心的工程专业综合

- 故障发生与发展：研究内因与外因相互作用下的产品故障发生规律(随机性规律与确定性规律)。
- 故障发现与维修：研究故障预测、诊断、修复的一般规律、方法与技术手段。
- 故障预防与纠正：研究预防与纠正故障的一般规律、方法与技术手段。

工程专业综合的总目标：提高装备的战备完好性，提高装备的任务成功性，减少维修保障资源和降低维修保障费用。

在系统研制的初期阶段，当备选的设计方案已具体到能对其进行分析的程度时，就可以开始逐步地将有关的工程专业融入设计工作之中。可靠性和维修性工程在所有相关的工程专业中对设计工作的影响最大。在这个阶段，各工程专业要在它们各自的业务范围内确定出相应的技术要求，并将相关的标准等转换为适用于具体项目的专用要求。所有这些从不同专业角度提出的技术要求，都要纳入到各级规范之中，从而保证这些规范都能体现出各专业的技术要求。各专业人员还要参与评审和分析进展中的设计成果，保证各方面所需要的特性能协调地综合到系统设计中。

随着设计工作进入到工程研制阶段，各专业都要提出工程研制阶段的详细工作计划，明确各专业的工作程序、内容、进度及所需的人力、物力。

在方案确定和验证阶段,还有一项重要的工作内容,就是提出专业方面的试验要求(如可靠性寿命试验、维修性演示验证试验、电磁兼容性试验等)和拟定相应的试验大纲计划,并将其纳入系统的试验与鉴定总计划中。

当进入工程研制阶段后,应集中于协调落实各专业的工作计划,包括各种试验与鉴定计划。

在整个系统工程过程中,参与各阶段的设计评审和各类试验的评估与鉴定工作,是各工程专业对系统的研制施加影响并保证系统具有所要求的全面性能的重要途径。各专业通过参与这一系列的工程技术活动,从各自的角度提出各阶段的决策性意见,这对于系统最终所能达到的费用-效能目标有着重大的影响。因此,随着研制工作的进展,恰当地组织好各工程专业的协同工作,充分地发挥他们各自的作用,是保证系统研制成功的关键环节。本章下面各节将分别叙述一些主要的工程专业在进行系统综合的过程中所起的作用。

3.2 可靠性工程

可靠性是指整个系统或它的组成在规定的时限内能按预期的要求进行工作的能力。可靠性高,就意味着产品在其给定的寿命周期内能持续地满足使用要求。作为正式的定义,应把可靠性理解为产品在规定的条件下和规定的时间内完成规定功能的概率。

随着人们日益重视系统的费用-效能的高低,又鉴于可靠性的好坏对系统的效能和费用有着重大的影响,因而在飞行器系统的研制过程中日益对可靠性提出更高的要求,也使可靠性工程专业对系统的设计发挥着更大的作用。在设计中考虑可靠性问题,或者说进行可靠性设计,必须既要遵循可靠性工程专业自身特有的规律,又要体现这些特有规律的专业纳入总的研制过程中,协调地加以推进。

3.2.1 可靠性要求

可靠性要求是进行可靠性设计、分析、制造、试验和验收的依据。正确地确定各项可靠性要求是一项重要而复杂的系统工作。设计、生产、管理人员只有在透彻地了解这些要求后,才能将可靠性要求正确地设计、生产到产品中去,方能有计划地实施有关的组织、监督和控制工作。

可靠性要求可以分为两大类:第一类是定性要求,即用一种非量化的形式来设计、评价,以保证产品的可靠性;第二类是定量要求,即规定产品的可靠性参数、指标和相应的验证方法,用定量方法进行设计分析,用增长或验证方法进行可靠性验证,从而保证产品的可靠性。

1. 可靠性定性要求

可靠性定性要求可以分为定性设计要求和定性分析要求两种。

1) 定性设计要求

主要的定性设计要求可参见表 3.1。

表 3.1 定性设计要求项目表

序 号	要求项目名称	目 的
1	制定和贯彻可靠性设计准则	将可靠性要求及使用中的约束条件转换为设计边界条件,给设计人员规定了专门的技术要求和设计原则,以提高产品可靠性
2	简化设计	减少产品的复杂性,提高产品可靠性
3	余度设计	用多于一种的途径来完成规定的功能,以提高产品的任务可靠性和安全性
4	降额设计	降低元器件、零部件的故障率,提高产品的基本、任务可靠性和安全性
5	制定和实施元器件大纲	对电子元器件、机械零部件进行控制与管理,提高产品可靠性,降低保障费用
6	确定关键件和重要件	把有限的资源用于提高关键产品的可靠性
7	耐环境设计	选择能抵消环境作用或影响的设计方案和材料,或提出一些能改变环境的方案,或把环境应力控制在可接受的极限范围内
8	热设计	通过元器件选择、电路设计、结构设计、布局来减少温度对产品可靠性的影响,使产品能在较宽的温度范围内可靠地工作
9	软件可靠性设计	通过采用 N 版本编程法、恢复块法及贯彻执行软件工程规范等来提高软件的可靠性
10	包装、装卸、运输与储存等设计	通过对产品在包装、装卸、运输与储存期间的性能变化情况的分析,确定应采取的保护措施,从而提高其可靠性
11	工程保证及生产质量保证	采用工程方法及生产质量保证方法来保证产品的可靠性

2) 定性分析要求

主要的定性分析要求可参见表 3.2。

表 3.2 定性分析要求项目表

序 号	分析项目名称	分析方法特点	目 的
1	功 能 危 险 分 析 (FHA)	综合的、系统的演绎方法	检查系统功能故障,确定设计方案的可行性,发现设计中潜在的问题,提出改进措施
2	故障模式和影响分析(FMEA)	系统的、自下而上的归纳分析法	评价每个零部件或设备的故障模式对装备或系统产生的影响,确定其严酷度,发现设计中的薄弱环节
3	故障树(或事件树)分析(FTA/ETA)	系统的、自上而下的演绎分析法	分析造成产品某种故障状态(事件)的各种原因和条件,以确定各种原因或原因的组。发现设计中的薄弱环节,提出改进措施

续表 3.2

序 号	分析项目名称	分析方法特点	目 的
4	区域安全性分析 (ZSA)	按装备的区域进行的分析、检查法	判断是否会由于系统、设备安装不当而产生不可接受的风险,或是否会由于该区域中某系统的设备故障而引起另一个系统的故障

2. 可靠性定量要求

作为设计中要考虑的因素,或者从总体上说作为一个(或一类)设计参数,必须能够对可靠性予以量化的表征和度量,以便能和其他设计参数一起进行设计权衡,并用这些定量的参数对设计结果进行验证和评估。

表征一架飞机的性能,不可能仅以单一的参数就把飞机的性能全面描述出来,而需要从不同的角度提出一系列参数,才能对飞机的性能给出综合性的描述。例如利用各种速度参数、爬升率、机动过载、载重和起飞重量等参数去描述飞机的总体性能。描绘系统的可靠性特性与之相类似,也不可能仅只使用单一的可靠性参数。可以从不同的侧面提出形形色色的度量参数,但总括起来可以把系统的可靠性参数归纳为下述 4 大类:

- (1) 与战备完好性相关的可靠性参数;
- (2) 与任务成功性相关的可靠性参数;
- (3) 与维修人力费用相关的可靠性参数;
- (4) 与后勤保障费用相关的可靠性参数。

表 3.3 列出了常用的 4 大类可靠性参数。

表 3.3 系统可靠性参数

类 别	可靠性参数
战备完好性	平均故障间隔时间 MTBF* 平均故障间隔飞行小时 MFHBF 固有可用度 A_i^* 使用可用度 A_o 满任务率 FMC 无维修待命时间
任务成功性	成功率* 完成任务的成功概率 MCSP 致命故障间的任务时间 MTBCF*

续表 3.3

类 别	可靠性参数
维修人力费用	平均故障间隔时间 MTBF* 平均故障间隔飞行小时 MFHBF 平均维修间隔时间 MTBM 平均非计划维修活动间隔时间 MTBUMA 首次翻修期限* 翻修间隔期限 T. B. O. 使用寿命* 储存期限
后期保障费用	平均故障间隔时间 MTBF* 平均需求间隔时间 MTBD 平均拆卸间隔时间 MTBR 每次拆换的平均零部件费用

* 合同中用的术语。

表 3.3 中所列各种参数的含义可参见专门的书籍,本书不再一一加以解释。不过应该注意到,并不是每个系统在进行可靠性设计时都要用到所有这些参数,而是要针对具体系统的实际情况选用其中的若干个参数。有时,根据需要,还可能采用表 3.3 中未列出的其他度量参数。

在选用合用的可靠性参数时,应充分考虑可靠性参数本身所具有的下述带有普遍性的特点。

1) 参数指标的综合性

综合性是指在很多情况下,需要将可靠性参数与下面即将讨论的维修性参数综合起来加以考虑,使二者在实现战备完好性、任务成功性或寿命周期费用目标时能匹配得更合理。以用来表示战备完好性的固有可用度 A_i 为例:

$$A_i = \frac{MTBF}{MTBF + MTTR}$$

它既与可靠性参数 MTBF 有关,也与维修性参数 MTTR 有关,所以 A_i 、MTBF、MTTR 三者间存在着需要相互协调的综合关系。

2) 参数的相关性

很多可靠性参数,往往在两个参数间存在着相互依存关系。忽略了这种关系,在提出不同参数的相应指标时就可能出现矛盾,而导致出现错误。例如 MTBF 和 MFHBF 间就通过称为运行比的 S/F 这一数值关联在一起,当某个具体系统的任务剖面和使用维修条件等确定之后, S/F 即是一个定植,从而也就限定了两个参数值间固定的相互制约关系。

3) 参数指标的阶段性

在不同的研制阶段,根据技术、资金等的现实条件及事物自身的发展规律,要对各可靠性参数提出从方案阶段的初步目标值分阶段地增长到成熟期的最终目标值的各阶段目标值。这种做法对于逐步提高系统的费用-效能指标非常有效。可靠性参数分阶段增长的情况如图 3.4 所示。

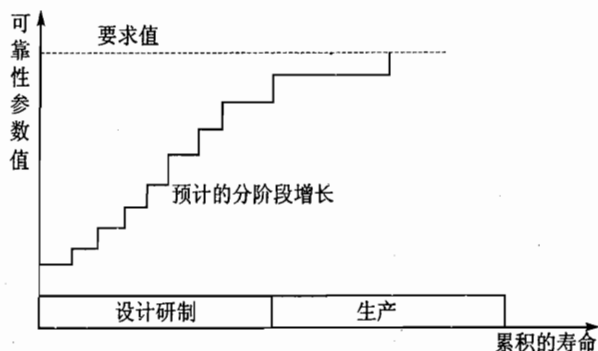


图 3.4 可靠性参数值的分阶段增长

4) 参数的使用术语和参数的合同术语间的可转换性

由于在实际使用场合与在承制方内厂环境下的参数验证条件不同,因而需要在适用于不同条件下的各参数间进行必要的正确转换。在表 3.3 所列的参数中,凡带“*”号的参数都属于合同中用的术语,一般用固定值表示;同一表中其余的参数都属于使用术语,一般用使用值表示。例如,MTBM 是使用术语,一般是由使用方提出的参数要求值。为了在合同中体现这一要求,必须把该参数转换为对应的合同术语,比如转换为对应的 MTBF 值。只有这样,承制方才能用内厂试验的方法对其所能达到的参数值进行验证。

3.2.2 可靠性工程实践

图 3.5 表示出了在系统寿命周期的各阶段有关可靠性工程专业的工作是怎样综合到研制过程中去的。在 GJB 450A—2004《装备可靠性工作通用要求》中,把可靠性工作项目按管理、设计分析、试验和验证评价进行了划分。

在方案阶段,应根据一个或几个已有的、类似系统的使用和保障经验,经过分析论证,针对每个被选系统方案选定可靠性的参数体系,并确定相应的门限值和使用目标值。对最终选定的系统方案,考虑到系统的可靠性有个增长过程,应制定出各阶段相应的最低可接受值(门限值)和目标值。

在工程研制阶段,对选定的系统方案在规定了稳定的参数指标及其容差值之后,即可进一步再分别转换为研制合同中的规定值(目标值)和最低可接受值(门限值)。在这个阶段,应进行可靠性的预计、分配、核验和再分配的工作;根据分析和试验结果,采取有效的纠正措施,及

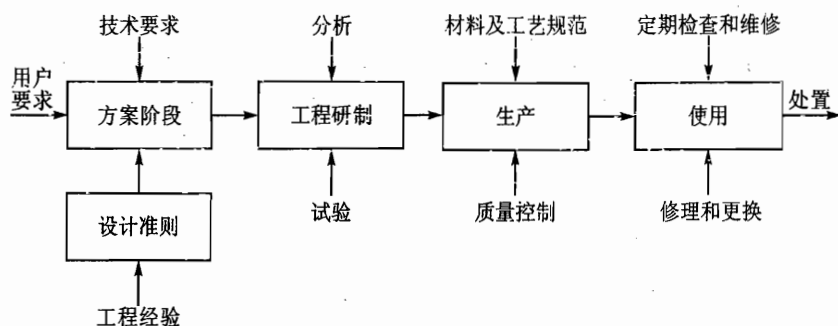


图 3.5 可靠性工程实践

早解决所发现的可靠性设计缺陷；还要实施可靠性增长管理，实现可靠性增长，以确保在制定生产决策前能充分地满足所规定的可靠性要求。

进入生产阶段时，以前类似系统的使用情况、验证试验的结果以及经过证实的有效设计纠正措施等都是进行投产决策的主要依据。由于生产中的薄弱环节和工艺缺陷而造成的重复出现的故障，应在这一阶段采取专门的质量控制措施加以清除，以保证在系统投入使用初期即可满足最低的使用可靠性要求。

在使用阶段，应对系统在实用条件下表现出的可靠性水平实施控制。要继续纠正因设计问题和质量问题引起的与可靠性有关的缺陷；还要纠正因使用方案和保障方案制定不恰当引起的可靠性缺陷；应不断地完善后期保障方案，以实现成熟期可靠性的使用目标值。

这样的综合过程相当典型，对于其他工程专业的综合，尤其是维修性工程的综合，在很大程度上与可靠性工程的综合过程相类似。

为确保取得可靠的系统或产品，应该从设计、生产制造、工程管理这 3 个主要方面采取提高可靠性的措施，以达到预期的目的。

1) 设计

正确而充分的可靠性设计，是达到可靠性要求的最基本的保证。为此，应做到：

- 对所用零件、元器件和材料的来源加以控制，并予以严格的鉴定和验收；
- 对零件和元器件要根据应力分析的结果采用必要的降额设计措施；
- 进行必要的、与可靠性相关的分析，如电子设备的潜在通路分析、故障模式及影响分析和维修分析等；
- 对设计要力求简化和标准化；
- 进行以暴露设计缺陷和验证纠正措施的效果为主要目的可靠性增长试验；
- 应根据对效能和费用的影响去考虑可靠性方面的设计特征。

2) 生产制造

能否完整地体现可靠性设计的意图,生产制造是个关键的环节。为此,应做到:

- 实施严格的工序控制,避免引入不合格的零件和元器件,避免产生不应有的工艺缺陷;
- 进行环境应力筛选,以便尽早、有效地暴露元器件或设备中存在的隐患;
- 实行无故障验收的原则。

3) 工程管理

- 实行设计和生产制造过程的监控报告制度;
- 及时进行故障分析,根据故障分析结果所采取的纠正措施,应贯彻使故障重现率和维修工作的难度尽可能减少的原则;
- 对纠正措施的效果进行跟踪和鉴定。

应当强调,系统或设备所具有的高可靠性是设计、生产制造、工程管理这3方面共同努力的结果。

必须认识到,设计上的缺陷、零件和原材料的缺陷及加工过程的缺陷都会导致故障。这些故障又都具有很大的变动特性,变动性又源于系统的生产条件和维修条件的变化。

传统上,以严格控制容差和增加检验次数的办法来提高产品的质量,同时也就是提高产品的可靠性。但这样做并不能从根本上解决问题,而且还相当费钱。要想减少从设计到加工的整个过程变动的的影响,唯一有效的途径还在于改进这一过程。

首先,应使系统的设计是“健壮”的,即要使系统设计成对外界生产条件等的变动不敏感。为此,就要在进行可靠性设计时特别注意选择最为有利的可靠性参数值,使之能在外界条件变动时仍能使系统自身的可靠性变动最少。

其次,要制定“健壮”的生产制造和修理过程,使得该过程对于环境条件、原材料状态、加工机械的状况及操作人员的素质情况不敏感。

与此同时,还要采取专门的以统计结果为基础的过程控制手段,对生产和修理过程随时进行监控,及时发现可能导致产生缺陷的变动和引起变动的关键因素,从而采取有效措施消除这类变动或尽量减少这类变动的发生。

要保证得到“健壮”的设计和采用“健壮”的过程,就要求从上到下的各管理层次都能对产品的可靠性和质量有正确的理解,并在此基础上直接参与设计和生产过程的改进工作。没有管理工作保证,就无法更全面地进行可靠性的工程实践。

表3.4从可靠性管理、设计分析、试验评价等方面列出了适合不同阶段、不同系统与设备相关的可靠性工作项目。

表 3.4 可靠性工作项目

类 别	项目 编号	工作项目	方案 阶段	工程研 制阶段	设计 定型	生产 阶段	系 统	分系统	设 备
可靠性管理	202	制定可靠性工作计划★	√	×	×	√	√	√	√
	203	对转承制方和供应方的监督和控制★	√	√	√	√	√	√	√
	204	可靠性评审★	√	√	√	√	√	√	√
	205	建立故障报告、分析与纠正措施系统★	△	√	√	√	√	√	√
	206	建立故障审查组织★	△	√	√	√	√	√	√
	207	可靠性增长管理	△	√	×	○	×	×	△
可靠性设计与分析	301	建立可靠性模型	△	√	×	○	√	√	√
	302	可靠性分配	△	√	×	○	√	√	√
	303	可靠性预计	△	√	×	○	√	√	√
	304	故障模式、影响及危害度分析★	√	√	×	△	√	√	√
	305	故障树分析	△	√	△	△	△	△	△
	308	制定可靠性设计准则及符合性检查★	√	√	×	○	√	√	√
	309	元器件、零部件和原材料的选择和控制★	△	√	√	√	△	△	√
	310	确定可靠性关键产品★	△	√	×	○	√	√	√
	311	确定功能测试、包装、储存、装卸、运输和维修对产品可靠性的影响	△	√	×	○	△	△	√
软件可靠性		软件可靠性控制★	√	√	○	×	√	√	√
可靠性 试验与 评价	401	环境应力筛选	×	√	×	√	×	×	√
	402	可靠性研制试验	×	√	×	○	×	×	△
	406	可靠性分析评价★	×	√	√	√	√	√	√

注：★——必须开展的工作项目；√——选用；△——根据需要选用；○——仅设计更改时选用；×——不选用。

从几十年的发展来看,常规的可靠性设计分析似乎与性能设计脱节,飞行器系统型号工作中存在“两张皮”的怪现象。可靠性设计分析结果没有对常规的性能设计产生影响。其主要表现在:

(1) 由于管理措施不严和监督的不力,因此在产品的每个研制阶段内,可靠性设计工作总是落后于性能设计工作,两者不能保持同步,使得可靠性设计和分析工作不能发挥应有的、尽早发现问题和缺陷的作用。

(2) 由于可靠性工程技术知识培训工作的滞后,因此使得大多数性能专业的设计人员对可靠性设计分析工作的重要性认识不足,对可靠性设计分析技术了解不深,在进行可靠性设计分析工作时应付了事,有的甚至出现概念错误,难以达到进行产品可靠性设计分析的真正目的。

(3) 除了上述的主观因素和管理因素外,从技术上看,现有的可靠性设计分析技术(包括相关的维修性、测试性等分析技术)内容广泛,涉及众多的标准、规范和方法,而每一项设计分析工作,如 FMEA、可靠性预计、容差分析等,都需要设计人员花费大量的时间去掌握和完成,因此性能设计人员缺乏和性能设计同步的可靠性自动化分析手段和分析工具,是导致“两张皮”现象的重要客观因素。如何将可靠性设计与性能设计相结合?如图 3.6 所示,从产品热设计、电路容差分析等应用是比较好的结合部位。这需要:一是有相关的硬件支持;二是有相关的 EDA 软件的支持。

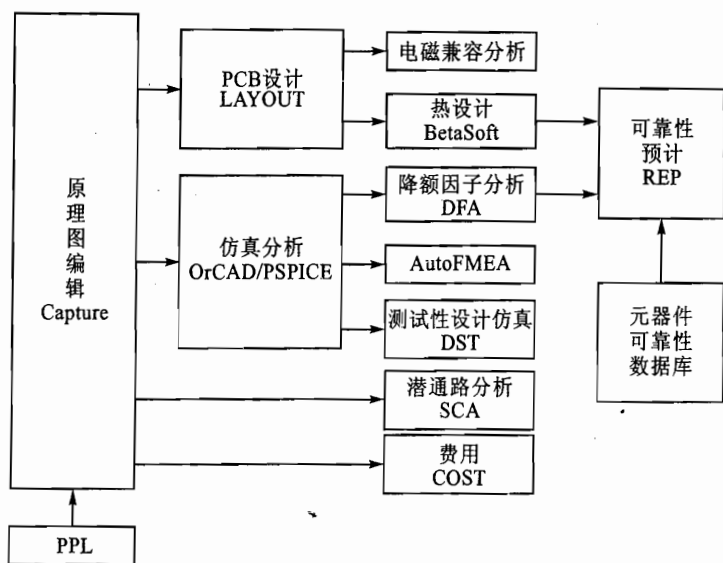


图 3.6 可靠性设计分析与电子产品性能设计的综合

软件工程与软件可靠性方面的要求等有关内容见第 5 章。

3.3 维修性工程

维修性是指当运用规定的程序和资源条件时,在每个规定的维修等级上,由具有规定技术水平的维修人员进行维修时,系统或产品保持或恢复到预定工作状态的能力。

实际上,可靠性和维修性是一件事物的两个方面,是彼此密切相关的两个系统特性。一方面希望所研制出的系统正常运行的时间(或使用寿命)能维持得越长越好;另一方面,由于客观上不存在绝对不出故障的系统,因而又希望在系统出现故障后,能在尽可能短的时间内恢复到正常运行状态。系统维持正常运行状态的能力自然首先取决于它所具有的可靠性的高低,但



同时也要看它所具有的维修性的好坏。可以说,系统具有好的维修性,是使它能充分发挥其固有可靠性潜力的基本保证。

3.3.1 维修性要求

1. 维修性定性要求

飞行器系统和设备应按照 GJB/Z91 开展维修性设计。维修性定性要求应遵循以下准则:

(1) 故障检测诊断迅速、简便、准确,主要设备应具有故障检测功能,对影响系统安全的重要电子系统及机械设备应具备机内测试功能。

(2) 具有良好的可达性,对舱内主要设备应设计维修通道、出舱方案,并预留维修空间。

(3) 符合人机工程要求,简化维修工作,减少维修差错,降低对维修人员的要求。

(4) 保证维修工作安全。

(5) 具有完善的防差错措施及识别标志。

(6) 提高标准化和互换性程度,减少维修设备、工具、备件和费用。

2. 维修性定量参数

维修性参数的提出,从大的方面与可靠性参数的提出是完全相同的,即也要从表征准备就绪程度、任务成功性、维修人力费用和后勤保障费用这 4 个方面去考虑系统维修性的度量问题。表 3.5 所列常用的维修性度量参数。

表 3.5 系统维修性参数

类 别	可靠性参数	类 别	可靠性参数
战备完好性	平均修复时间 MTTR* 平均系统恢复时间 MTTRS 平均维修时间 $\bar{M}$ 最大维修时间 $M_{\max ct}$ 平均不能工作时间 MDT 更换发动机时间* 再次出动机务准备时间	维修人力费用	平均修复时间 MTTR* 每飞行小时维修工时 MMH/FH 每工作小时维修工时 MMH/OH 故障检测率 故障隔离率 虚警率
任务成功性	恢复功能用的任务时间 MTTRF* 任务维修度 M_t	后期保障费用	平均修复时间 MTTR* 平均不能工作时间 MDT

* 合同中用的术语。

3.2.1 小节中所讨论过的关于可靠性参数的一些特点,完全适用于系统的维修性参数,这里不再重复。但在选定各种维修性参数时,除了要注意不同的维修性参数指标间要相互协调外,还应特别注意同一系统的维修性参数与可靠性参数指标值间的协调关系。应充分考虑到,系统的可靠性参数和维修性参数要形成统一的参数体系。

3.3.2 维修性工程实践

由于可靠性和维修性是紧密地联系在一起的,因此在工程实践方面二者也是协调开展工作的。在不同发展阶段所应进行的主要维修性工作与 3.2.2 小节中所述的关于可靠性的工程实践内容非常类似,所以在工程上常常把可靠性和维修性结合在一起考虑。表 3.6 是主要的维修性工作项目实施要求。

表 3.6 维修性工作项目实施表

序 号	工作项目	项目类型	研制阶段				
			方案阶段	工程研制阶段	设计定型	生产阶段	使用阶段
1	制定维修性工作计划	管理	→				
2	对转承制方和供应方的监督与控制	管理	→		→		
3	维修性工作评审	管理	→		→		
4	建立数据收集、分析和纠正措施系统	工程	→		→		→
5	维修性建模	工程	→	→			
6	维修性分配	计算	→	→			
7	维修性预计	计算	→	→	→		
8	故障模式及影响分析—维修性信息	工程	→	→	→		
9	维修性分析	工程	→	→	→		
10	维修性设计准则	工程	→	→	→		
11	维修性试验与评估	统计		→	→	→	→

不过,值得提出的是,系统在运行中所表现出来的维修性水平与维修技术人员和管理人员的业务水平有着极为密切的关系;而且,系统提供的后勤保障效能对于系统在运行中所能达到的维修性水平也有着举足轻重的影响。这些是维修工程实践与可靠性工程实践间最主要的不同之处,从而也决定了维修性工程实践在与可靠性工程实践以类似的程序同步向前推进的同时,还有着属于其专业范围的独特工作。

从维修性工程专业的考虑出发,首先应提出针对所研制系统的维修方案或维修方针。这些内容都应通过系统工程过程反映到系统的设计工作中去。维修方案或维修方针是根据系统在使用阶段如何得到后勤保障等情况确定的。它们明确了有关维修等级的划分;在不同的维修等级上进行故障的防止、检测、隔离和修复的要求;相应地对人员的技术水平、测试设备、修复与拆换时间的要求以及系统寿命周期内对维修费用的要求或限制等内容。维修方面的要求对于确定系统的可用度及系统寿命周期费用的高低,都是有重要影响的设计考虑因素。

要借助已有的工程经验和所建立的各种维修活动模型,进行维修工作分析和提出各项维修产品的详细维修要求。根据维修工作分析,即可拟定出定性和定量的维修设计准则,从而能在设计工作的初期就把维修性要求与系统的主体设计结合在一起。

当进入工程研制阶段时,随着设计工作的深入和设计细节的日趋明朗,即可预计出系统各组成单元或各项产品的维修性参数值,并将之与所分配的目标值相比较,以确定是否符合预定的技术要求。必要时,还要进行维修性参数的再分配,并相应地做出设计上的改动。在此过程中,为了得到有用的维修性参数估计值,有必要进行局部的或整体的维修性演示验证。

维修性工程涉及面很广,必须与可靠性工程、安全性工程、人素工程及后勤保障分析、可生产性分析、寿命周期费用分析等其他专业工作互相沟通,才能使系统在实际运行期间达到所要求的维修性水平。

3.4 安全性工程

系统具有安全性,是指在系统的运行过程中未发生人员的死亡、受伤及引起职业病的情况,也未发生设备或财产受到破坏或损失以及环境遭到破坏的事件。

由于设计缺陷、使用不当、环境变动或外界干扰等原因,在形形色色的工程系统中往往存在着各种潜在的不安全因素。不安全因素的存在可能会影响到制造、试验、装卸、运输、储存和操作等各个工作环节的正常运作,并对有关人员和设备造成危害。有毒物质的泄漏或核泄漏、电磁辐射、激光照射或爆炸等都是典型的不安全因素。

安全性工程专业在系统工程过程中所起的作用在于,要利用有关的专业知识、工程原理、技术、技能,以及以这些知识、原理和技能为基础制定的判断准则,在系统寿命周期的各个阶段去发现、排除或控制系统中存在的危险或不安全因素,并通过一系列有计划的活动对系统所达到的安全性水平实时监控和管理,求得以更好的费用-效能值来满足系统提出的安全性要求。

3.4.1 安全性评价

对系统所达到的安全性水平,通常是以其所具有的危险的严重程度及其出现的可能性大小来估计的。危险是指在系统中现存的或潜在的有可能导致意外事故的状况。比如,由于不应有的燃料事故的泄漏,并造成燃料出现在不允许出现的部位,即是一种危险。

危险的严重程度是由于危险的存在而可能导致的最坏潜在后果的一种定性估计。视危险可能对人员造成的伤害、引起的职业病以及对财产或设备造成的损坏程度,一般可将危险的严重程度定为4个等级:

- I级——致命级。可能造成人员死亡和整个系统损坏。
- II级——严重级。可能造成人员严重受伤,引起严重的职业病或主要系统的损坏。
- III级——有限级。可能造成人员轻度受伤,引起轻度的职业病或系统的局部损坏。
- IV级——极轻级。没有造成人员受伤,没有引起职业病或系统的损坏。

危险出现的可能性可以定量地表达或定性地表达。定量地表达危险出现的可能性,可以用系统在预定的工作寿命期间,每单位时间、每一意外事件、每一人群、每一个项目或每项活动可能的危险出现率来描述。当缺乏充分的统计数字依据,无法进行定量估计时,可以通过对同类系统在安全性方面的历史资料进行分析,得出对危险出现可能性的定性估计。表3.7给出了一种可供使用的危险出现可能性的定性序列实例。

表 3.7 危险出现可能性的定性序列

出现的可能性	等 级	单个项目	总体现象
频繁	A	可能频繁出现	有连续出现的经历
相当可能	B	在寿命期内会出现数次	会频繁出现
偶然	C	在寿命期内可能间或出现	将出现数次
很小可能	D	可认为不会遇到这种危险	未必出现,但不完全排除出现的可能性
极不可能	E	出现可能性几乎近于零	可认为不会遇到这种危险
不可能	F	实际上不可能出现	实际上不可能出现

对危险严重等级和可能出现危险的定性估计给定某种对应的数值,再按二者间一定形式的数学集合表达式,得出该数学集合的某种数学顺序安排,根据这种数字顺序即可作出危险的估计,并进而有针对性地采取必要的防范措施。

3.4.2 安全性工程实践

应用对危险进行分析的方法,即可提出关于安全性的设计准则,再通过设计评审判断设计结果是否符合安全性要求。当系统投入运行后,应通过对使用过程中的危险进行分析及对有关运行程序进行的相应评审,将安全性要求及安全性标准贯彻到使用过程中去。对那些影响安全性的关键项目、关键运行状态要予以监控,以便将潜在的危险减少到可以接受的程度。上述这些有关系统安全性的工作,应该在一个系统安全性大纲的指导下按规定要求进行。系统安全性大纲应按寿命周期各阶段的顺序加以制定。表3.8所列为安全性工作项目。

表 3.8 安全性工作项目

类 型	编 号	工作项目	方案 阶段	工程研制阶段			使用 阶段	实施结果
				初步 设计	详细 设计	生产 试验		
管 理	1	安全性工作计划	√	√	√	√	√	工作计划
	2	对系统和设备场所监督与控制	△	√	√	√	√	技术文件
	3	安全性评审	√	√	√	√	√	评审结论
	4	试验的安全性	√	√	√	△	×	技术文件
	5	事故报告与调查	△	√	√	√	√	运行报告
	6	安全性信息管理	△	√	√	√	√	技术文件
设计 与分析	7	初步危险表和初步危险分析	√	√	√	√	√	报告
	8	分系统危险分析	△	√	√	√	√	报告
	9	系统危险分析	△	√	√	√	√	报告
	10	使用和保障危险分析	△	√	√	√	√	报告
	11	安全性设计与控制	√	√	√	△	×	报告
	12	软件安全性	√	√	√	√	△	文件
验证	13	安全性验证	×	√	√	△	△	验证报告
培训	14	安全性培训	×	√	√	√	√	技术文件

注：√——适用；△——根据需要选用；×——不适用。

在方案探索阶段,安全性工程的工作重点是对各被选的系统方案进行安全性评估,主要的工作任务包括:对寿命周期内可能影响安全的所有材料、设计特点、生产制造过程、使用方案及使用环境进行评估;对每个备选方案进行初步危险分析;确定可能的安全性限度和提出系统安全性要求。

在方案确定阶段,主要的工作任务包括:完成和修正初步危险分析;判定最可行的主方案在设计、工艺、生产及使用和技术保障方面存在的可能影响安全的因素,并据此进行必要的改进;制定系统的安全性要求和应用于验证过程的判断准则;确保将系统安全性要求纳入系统的技术规范。

在工程研制阶段,主要的工作任务包括:对初步设计结果进行审查,保证有关的安全性要求在设计中得到体现,并对已发现的危险加以排除或控制;审查试验计划和程序,确保试验能安全进行;判定封存、包装、运输、装卸、试验、使用和维护等因素对安全的影响;审查各种工程技术文件,判定它们是否包含了有关安全性考虑的内容;检查是否具备足够的安全保障和防护设备。

在生产阶段,主要的工作任务包括:确定对关键零部件生产技术、装配程序、设备和实验



项目的检查要求,保证在生产过程中具备足够的安全措施,如建立必要的安全管理制度和在生产技术手册或加工工艺规程中编入必要的报警或警告等特别措施;对早期生产出的成品进行试验和鉴定,尽早发现和采取措施弥补安全上的缺陷;审查有关试验计划和程序,保证各项试验的安全进行;审查保证系统安全操作和维修所需的报警、警告和其他相关措施;审查有关封存、包装、装卸和运输等操作的相应规程能否保证安全。

在使用阶段,主要的工作任务包括:审查在外场进行定期检查或试验的程序,以确定安全性是否保持在应有的水平,尤其应注意哪些对安全性关系重大的零部件关键特性的变化趋势;根据实际使用情况,修正危险分析结果,并判定因技术上的变更可能引起的新危险,保证在调整系统的配置时应考虑到预计的变更对安全性产生的影响;评价外场故障分析的结果,提出关于采取修正措施的建议;对系统的运行情况进行监控,判断为应付使用和维修活动中可能遇到的意外情况而要采取的应急措施的充分性;对新提出的或修订过的使用和维修程序进行安全性审查,以保证满足原有的安全性要求;修订与安全性有关的设计手册、标准和规范等技术文件,以充分而及时地反映在安全性方面积累的经验和教训。

作为安全性过程活动的指导性文件,系统的安全性大纲应安排得能够保证定期地对系统的状态加以考查,能及时地分析危险和评定关于系统安全性工作的总效果。如果说安全性大纲是在系统寿命周期内用以指导安全性工作的纲要,那么系统的安全性要求则为系统的设计和使用确立了排除或控制危险的安全性准则,也确立了表征系统安全性水平的定量标准。

在一般情况下,系统安全性设计准则考虑了从标准、规范、细则、手册和指南等设计文件的适用性,到器材的选择、危险区或危险源的隔绝、环境的控制、人为失误的避免和防护措施的采用,直至警告和提醒标志的安排等各方面的细节。

定量的安全性水平通常是用概率数值表示的。在设计实践中,可采用故障树分析技术确定出操作危险的各事件的组合,并进而确定出它们的发生概率。再结合以往研制工作中的经验、教训和事故风险评估结果,确定出系统中各功能分系统的设计极限和使用极限。这样就可以根据实际情况确定出所采取的相应措施,以尽量减少或避免发生大事故的危险。

在系统安全性工程实践中,往往按如下的顺序去考虑满足系统的安全性要求或解决存在的危险问题:

- (1) 首先考虑从设计上采取措施排除危险,若不能排除,则从设计上采取措施加以控制。
- (2) 如果做不到以设计措施去排除或控制危险,则应使用固定的、自动的或其他类型的安全保护装置将危险控制到可以接受的程度。
- (3) 当用上述两种方法都无效时,应利用报警装置去检测出危险,同时发出报警信号,由操作人员在接收到报警信号后作出判断和采取必要的措施,以便制止危险的继续发展。
- (4) 当应用报警装置尚不能排除或充分控制危险时,就应既采用防护措施,又进行专门的人员培训,以达到控制危险的目的。

一个系统所能达到的安全性水平与在系统工程过程中安全性管理的程度成正比。使系统

能得到有效的安全保障的系统安全性大纲能否在工程实践中取得成功,取决于对安全性目标和安全性要求提得是否明确,也取决于能否正确地把这些目标和要求体现在系统的各组成单元上。

3.5 零件工程

零件是系统或设备的最基本的构成单元。它既是指不可再分的单一的工件,也是指不予拆开的由两个以上工件构成的组合体。如果将这个组合体拆开,则会破坏其所特有的功能。

一个大型的系统中包含了各种各样的大量零件。这些不同类型的零件,按照一定的功能要求,自下而上分层次地组成组件、部件和分系统,最后构成完整的系统。

通常,零件本身所具有的固有可靠性水平(或固有故障率)以及由设计所规定的零件的组合与应用方式,就基本上决定了系统的固有可靠性和维修性水平,并且也对系统后期保障费用的高低有着很大的影响。比如一个由 100 个零件构成的设备,当零件的固有故障率都为 10^{-3} 次/小时时,该设备的平均故障间隔时间为 10 h;当零部件的固有故障率都降至 10^{-4} 次/小时时,该设备的可靠性水平将提高一个量级,其平均故障间隔时间可增至 100 h。由此可见,正确地设计和选用所需的零件,是得到效能高而费用低的合用系统的最基本保证。

3.5.1 材料的选择

正确地选择材料,是零件设计过程中的关键一步。选材不当,不仅会导致零件失效,还会造成不必要的浪费。据统计,目前约有不下 4 万种牌号的金属材料和几乎同样数量的各种牌号的非金属材料可供选用。因此,可以说选择材料是个非常重要的决策过程。

正确地选择材料,不仅要使材料的特性符合零件的功能要求,还要考虑到与之相关的加工过程。在进行选材决策时,几乎面对的是无法计数的材料特性和加工处理方法的组合。在这种情形下,只依靠已有的经验或仅仅依据简单的材料性能手册去选择恰当的材料,已不能完全满足工程设计的要求。虽然材料本身的特性仍然是基本的选材依据,但同时还要根据系统的应用特点,考虑到诸如费用、加工技术、重量和供应来源等问题。

材料本身的性能一般可以用其固有的物理、机械、热、电或化学的特性加以表述。材料的这些特性是材料本身的基本构成与由它制成的零件所具有的技术性能间的联系纽带。用某种材料制成的零件,其性能或故障状态(疲劳、腐蚀、磨损等)与材料的特性(屈服限、拉伸与压缩强度、延展性等)间存在着紧密的联系。通过采用材料组分控制和热处理等方法,可以改变材料的构成,从而可以按需要改善材料的特性,进而使以该材料制造的零件满足一定的技术性能要求。

由于材料特性数据的不充分和不准确,以及操作者其他各种条件的限制,因此作出选材决策并不是一件轻而易举的事情。在一般情况下,如果仅仅简单地以某种特性更好的材料去代替原用的材料,而不相应地改变零件的设计方式,往往是不能充分发挥材料自身所具有的潜力

的。本质上,选材过程不仅是不同材料间的比较权衡过程,更是在不同的材料加工处理方法间的比较权衡过程。

选材一般要经历以下几个步骤:

① 分析对材料的要求。确定零件的工作条件和环境,并将之转换为对材料关键特性的要求。

② 筛选可供选用的材料。通过将大量的材料特性数据与所需求的材料特性进行对比,先筛选出少量几种符合要求的材料作为备选材料。

③ 在备选材料中择优。从零件的性能、费用、可生产性及货源等方面进行权衡分析,从各种备选材料中选取最为合用的材料。

④ 提出供设计用的资料。根据已有资料及试验结果确定出选定材料的关键特性,并从统计的角度得出在预计的特定使用条件下有关材料性能的可信指标。

典型的选材过程框图如图 3.7 所示。

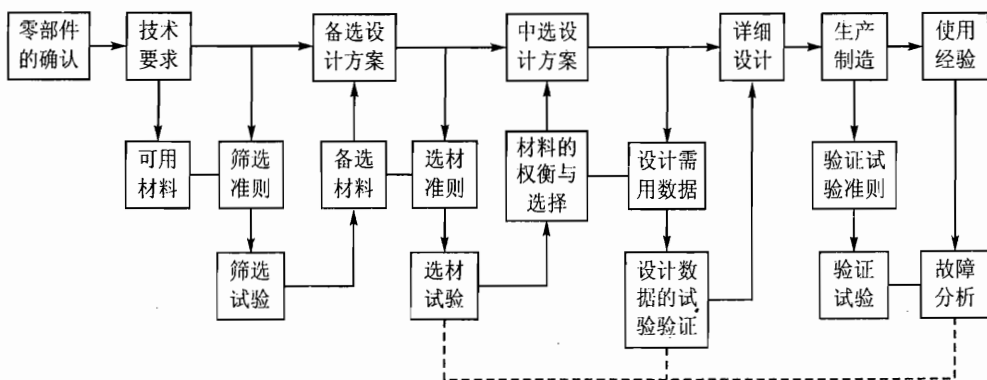


图 3.7 典型的选材过程框图

需要指出,设计过程关于所选定材料的特性数据必须是反馈该材料经过加工过程后所具有的对应数据,并应具有充分的可信度。这样,才能使所设计的零件在制造出来以后能达到给定的可靠性指标。

由于进行选材的权衡分析是件复杂的工作,所需比较的那些材料特性有时很难用可比的参数表示出来(如某些加工的难易程度),因此也就不容易做出明确的判断。常用的权衡分析方法有费用与性能指标对比法、加权特性指标法、价值分析法和故障分析法等。

3.5.2 加工过程的选择

在现代技术条件下,生产加工已不再是一种传统意义上的常规性活动了。设计、选材和加工过程三者已经形成了如图 3.8 所示的彼此密切相关的工程活动链。由于工艺技术日益先进而且变化很快,因此把设计和制造更紧密地联系起来是非常必要的。最为突出的一个例子就

是,随着半导体器件代替了真空管,使得电子器件的设计和加工不再是彼此相互独立的,而是不可分的,从半导体管的选材、器件的制造和电路的设计等功能都是紧密相关的;当发展到大规模集成电路时,从选择材料到系统设计,更是从头到尾都交织在一起,生产制造也越发变得不能与设计分割开来进行了考虑了。

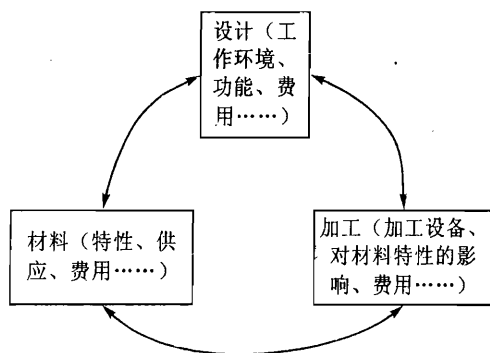


图 3.8 设计、材料与加工作用链

在生产制造过程中用到的加工过程,大体上可以划归为 8 大类:固化(铸造)、形变(锻造、轧制、拉伸和拔丝等)、切削(车、铣、刨、磨和抛光等)、聚合物加工(注塑、热成型等)、特种加工(压制、烧结、热压和塑变等)、接合(焊接、铜焊、钎焊、扩散连接、铆接、螺接和胶接等)、热处理和表面处理(碳化、氮化、喷涂、电镀和油漆等)以及装配。

由于同一件产品常常可以用数种不同的加工方法,因此要注意选择其中最合用的生产加工过程。在选择时,费用是个非常重要的因素。

考虑费用因素,不仅要考虑将原材料加工为最终产品所需的费用,还要考虑到材料的有效利用率以及加工方法对材料特性的影响和随后对零件技术性能的影响。

不同的加工方法、不同种类的原材料和不同的产品技术要求,都会影响到零件的加工费用。当以费用作为第一位原则时,应把材料的牌号(成分)、材料的形状(棒材、丝材、管材、板材、带材等)、所需尺寸(大小和容差)、热处理状况、表面状态、质量等级、所需数量和加工难易程度等因素综合起来加以考虑,把选材和选择加工方法联系起来做出决定。

3.5.3 零件控制

正确地设计新的零件或恰当地选择现成的零件,都需要进行仔细的权衡并对设计或选择过程慎重地加以控制。在开展系统研制工作的初期,就应该对零件的设计和选择过程实施严格的控制。这就是所谓的进行零件控制,其主要目的是以合理的费用求得合用而可靠的各类零件。

一个大型的复杂系统,通常总是包容了各类型的零件,有机械类的,也有电气和电子及其他类的。制造这些不同类型的零件,会用到种类繁多的零件和各种不同的加工方法;针对不同的材料和加工方法,在设计技术上也会做出不尽相同的考虑。这样一来,对由相当浩瀚的各色零件组成的群体实行有效的控制,必须要有真正起作用的有效的零件控制程序,即必须建立起能对新设计或选用的零件的种类、规格和质量进行严格控制的一整套管理制度和审定手续。

建立什么样的零件控制程序最合适,要视工程项目的具体性质而定。一般选择最有效的零件控制方法时要考虑的主要因素是:系统的可靠性要求,产品的结构复杂程度,零件总产量

的大小,所涉及的承制方的范围,对系统提出的使用性能的要求,全寿命期费用的约束以及在所需的这些零件中预计要购买的现成零件的范围等。

考虑零件控制的深度时,应区分工程项目的不同发展阶段,还要看未来达到什么样的研制目的才需要该种零件。比如,当仅是进行探索性研究时,实施零件控制效果不大,或可能无效;当进行工程研究或是进行试生产时,则应进行零件控制并将零件控制的重点集中在用于原型样机研制的零件上,此时应尽可能地选用成件,零件控制工作大量地集中于现成零件的选择过程;当已确定投入批生产时,则必须实行全面的零件控制。此外,为了在整个寿命周期内求得费用上的节约,还要结合使用过程中的后勤保障要求去考虑零件的控制工作。

在系统的研制过程中,实施有效的零件控制程序,必须有权威性的监督保证机构,比方说成立一个专门的零件控制部门。这个零件控制部门应完成如下工作:

- (1) 确定应遵循的零件控制程序,并保证该程序的有效实施。
- (2) 拟定一份适用于具体工程项目的零件选择清单。
- (3) 评审并核准未列入零件选择清单的拟选零件,确定它们的试验要求和合格验证准则。
- (4) 确定新研制零件的设计准则、试验准则和加工工序的监控准则等。
- (5) 及时地对实施零件控制过程中所产生的问题做出相应的处理、决断。

其中,拟定一份合用的工程项目零件选择清单,是实现零件控制的主要手段。鉴于一个新系统中所含的零件大部分都不会是新设计的,而且往往有很大的继承性,所以拟定该清单的主要意图是通过最大限度地压缩所用零件的品种、规格、级别或类别,达到最大限度地提高零件标准化程度的目的。当然,随着技术手段的持续进步和新的技术要求的不断提出,需要不断地对零件选择清单进行必要的调整。不过,无论如何,设计人员应尽可能按照零件选择清单去选择符合技术要求和可靠性要求的标准化零件。如果设计中要求用的零件是未列入零件清单的,同时也未曾经过合格验证的,或者是没有充分的使用记录的,则应进行必要的试验验证,并且还要经过审定和核准。当由许多不同的部门为同一系统的研制工作服务时,应该保证这些部门间都具有一致的零件标准。

3.6 人素工程

虽然近年来出现了许多高度自动化的设备,但就全局来说,一个系统要能正常、高效地运行,总是离不开“人”这个非常重要的环节。一架现代飞机需要以人作为控制其飞行的核心;即使是完全自动化的系统,也还需要由人去规定它的运行状态,并对其完成任务的情况进行监控和进行必要的调整。随着工程技术的发展,越来越需要考虑到人在技术发展中的作用,有关人的因素的问题也引起了越来越多的兴趣和关注。

在工程问题中讲人的因素,是指要去发现有关人的各种行为举止和能力等的信息,把这些信息应用于工具、机器、系统和它们所在环境的规划与设计,以期能使人在感觉舒适的状态下对它们进行有效而安全的利用。在这里,要把注意力集中于考虑人的特点以及在工作过程中

人与其接触到的产品、设备、设施、工作程序和环境之间的相互作用关系。考虑的重点放在人本身以及物质实体的设计如何影响到人,从中寻求能更好地与人的能力、限度和需求相匹配的系统构成和相适应的使用环境条件。

归纳起来,研究人的因素问题有两个主要的目的:一是提高工作效率和效能,比如操作时的方便程度,减少失误和提高生产率等;二是突出对人的重视程度,如改善安全性,减少人的疲劳和工作压力,增加人的舒适程度和提高工作时的满意程度等。

研究人的因素问题,并把研究结果应用于工程设计实践,所采取的主要途径就是把有关人类的能力、限度、特性以及行为举止等信息系统地应用到系统的设计、运行程序的设计和工作环境的设计中去。为此,要进行一系列的科学研究,去发现人对周围事物及环境等的反应规律,并以这些信息为依据提出相应的设计建议,利用这些信息去预计各个设计方案可能对人产生的影响,以便评估这些设计满足给定目标的程度。

在进行系统综合时,人素工程这一专业要遵循下述原则:

- (1) 物质系统、机器等是要服务于人类的,因此在设计时总要想到使用该系统或机器的人。
- (2) 要认识到不同的个人间在能力和各种限度间的差异,并使之在设计中有所反映。
- (3) 意识到对物质系统和运行程序等的设计要影响到人的行为状态和人的健康情况。
- (4) 在设计过程中强调试验数据的使用和评估。

(5) 要依仗科学的方法和利用客观的数据去验证假定,并由之得出有关人的行为状态的基本数据。

(6) 把物质系统、运行程序、环境和人员等视为一个大系统中的相互关联的组成因素,而不能把它们当作一个个彼此无关的孤立因素。

简言之,从人素工程的角度,在进行系统综合时要处理的主要是人-机间的接口问题,要应用有关人的各种信息和有关的规律去影响系统研制过程。为此,首先必须对人的行为状态、人的能力和极限等信息有全面而确切的了解,在此基础上提出有关的技术要求,拟定出相应的设计准则,用以指导设计,最后还要经过试验验证和评审,以保证所研制的系统从人素工程的观点看也是最合用的。

3.6.1 人素工程的研究方法

人素工程专业就其大部分内容来说,都属于带经验性的科学技术内容,有关人的能力等的各种信息大部分也都是以试验和观察为基础获得的。通过研究所得到的各种经验性数据,既可作为设计依据,又可用作对设计结果进行评估的参照基准。

人素工程的基本研究方法大体上可分为人的特性研究、实验性研究和评估性研究这3种类型。

1. 人的特性研究(描述性研究)

进行人的特性研究的目的在于收集有关描述人的行为状态和特制的信息,比如人体的尺

寸、听觉随年龄的变化情况、人的视觉感知能力及人的体力限度等信息。这些信息都是赖以作出设计决策的基础。收集操作者的意见,以发现设计缺陷和运行中产生的问题,是进行这类研究的一个实践性例子。

要做好人的特性研究工作,必须正确地选好进行研究的场合,定好表征相关特性的变量,取得有代表性的样本和做好所得数据的统计分析。

2. 实验性研究

进行实验性研究的目的在于测试某些特征量对人的某些行为状态的影响,比如估测系上座椅的安全带后,对人的手臂的功能动作会有什么样的影响。

为了研究得更深入,常常将现场的试验研究和实验室的试验研究综合起来,发挥它们各自的长处,同时还要结合起来进行仿真研究。

3. 评估性研究

进行评估性研究的目的在于从系统全局的角度出发,收集操作者使用某具体系统时所表现出的行为状态和行为特性的信息,借以评价设计的优劣和该系统在运行时对人所产生的影响。可以说,评估性研究是人的特性研究和实验性研究二者的综合。它是以整个系统或整个产品作为研究对象的。

评估性研究的内容比起实验性研究的内容来更要全面和广泛得多。进行评估性研究时,无论是对一个系统的预想得到的输出,还是未曾预想得到的输出,都要加以估测,并要把这些结果与预期的目标相对比。在评估性研究的内容中,往往还包括了关于费用-效能分析的相关内容。在系统的样机上进行的维修工作演示验证,从人素工程的角度来看,就是一种典型的评估性研究。

评估性研究应在真实的或接近真实的具有代表性的使用条件下进行,而且应该由具有代表性的实际操作者去参加研究。这意味着最好是在现场条件下进行评估性研究,从而也对研究工作的规划和组织实施提出了许多难题。

3.6.2 人素工程研究中的度量尺度

在进行人的特性研究时,要以各种方式描述人的特征和行为状态;在进行实验性研究时,要表述出人对系统运行状态变更后的反应;而在进行评估性研究时,则要以一定的方式对系统设计的优劣加以判断。所有这些都涉及如何正确地选择度量尺度的问题。虽然在进行不同内容的研究时所采用的度量尺度不可避免地会存在交叉和重叠的现象,但大体上可以把这些度量尺度区分为3种类型的度量尺度。

1. 系统特性方面的度量尺度

该度量尺度主要用于描述系统的功能作用,表征整个系统的工程状态。属于这类尺度的如设备的可靠性参数、维修性参数、使用保障费用以及常见于工程技术规范中的转速、重量、速

度等。这类度量尺度多用于评估性研究中。

2. 工作特性方面的度量尺度

该度量尺度主要用于描述工作任务是如何完成的,表征某项工作任务的输出结果。在完成该项任务时,可能涉及人的因素,也可能不涉及人的因素。属于这类尺度的,又可再划分为3个方面的内容,即有关任务输出情况的尺度,如飞机的出动架次率、执行一定飞行任务的耗油量等;有关任务输出质量的尺度,如失误的次数、飞行导航精度等;有关任务的时间尺度,如故障隔离时间、飞机起飞延误时间等。

3. 人的特性的度量尺度

该度量尺度主要用于描述人的反应情况,表征人在执行各种操作任务时的行为状态和反应特性。人的特性主要可以从以下3个方面度量。

1) 人的行为的度量尺度

该度量尺度从各种角度描述人所固有的能力,主要涉及工作频度的尺度(如在一定时间间隔内所能探测到的目标数)、工作强度的尺度(如驾驶杆杠力的大小)、潜在能力的尺度(如从一种活动转换到另一种活动的转换时间或反应时间)、持续时间的尺度(如执行跟踪任务时持续盯住目标的时间),以及关于人的可靠性尺度等。

2) 心理状态的度量尺度

该度量尺度用于表征在不同的环境条件影响下(热、振动、噪音、加速度等)进行体力或脑力活动时,对人造成的不同紧张程度。这类尺度涉及到人体生理系统、心血管系统、呼吸器官、神经系统、感官系统以及血液的化学组分等。

3) 主观反应度量尺度

该度量尺度用人的主观感受或主观判断去描述某些难于进行客观度量的事物。常见的如客机座椅的舒适程度、工具把手的合适长度等。显然,在应用主观反应度量尺度时,应特别注意尽量减少人的固有偏好的影响,甚至在所提问题的字句上或次序上的微妙变化都会对作出主观的评断有影响。

3.6.3 人素工程实践

为了达到人-机系统的优化,必须把操作人员的作用有效地结合到系统设计中去。在进行系统综合的过程中,人素工程专业的工作任务主要在于:根据人-机(包括软件)间的功能分配情况,判断人所要执行的工作任务的范围和关键程度,进而确定其是否超出了人的能力限度;利用实体模型或样机,对具有关键性的人的行为状态进行动态模拟,以便对设计方案加以评估;还要注意对处于危害性环境条件下工作的人员采取恰当的防护措施。

归纳起来,在研制工程系统时,人素工程涉及的工作应包括下述3个互相关联的内容:

1. 分 析

运用有关人素工程的原理,确定为完成既定的系统任务而在人员、硬件和软件间应达到的

最佳分工,即进行合理的功能分配。对所分配的功能要进一步加以分析,以便明确为实现各项功能所应完成的具体任务。在此基础上对每项具体任务再进行分析,特别注意研究人-机间的接口关系,保证对人员的操作要求不超过其能力限度。最后要把分析的结果具体地反映到系统和各分系统的初步设计中去。

2. 详细设计

在进行系统的详细设计工作时,应保证设计结果符合人素工程的考虑原则。凡关系到人的因素的设计内容,都应经过人素工程专业人员的评审。为了做出切合实际的评估,还必须进行必要的测试和实验验证。此外,在考虑系统本身设计的同时,还要考虑相关的工作环境和设施的设计,并根据人素工程分析结果制定出系统及其各组成单元的使用和维护规程。

3. 试验和评估

为了证实系统的设计符合人素工程的要求,并保证这些要求与系统的总体要求协调一致,也为了及时发现人-机系统中的缺陷,必须进行必要的试验验证和评估。

鉴于人的因素以及有关人的因素的各种信息和数据的特点,在设计过程中考虑人的因素的影响和应用相关的数据时还难以形成完整的工作程序。但如果能对人的因素予以较为系统化的考虑,那么至少可以使人们在进行设计时能充分注意到人的因素的特点。能促使人们系统化地去考虑人的因素的一种方法,就是列出一些在设计进行评审时起提醒作用的、包含一系列问题的清单,例如各种显示器安排得便于使用吗?是否充分地利用了人的决策能力和适应能力?各种操纵装置易于识别吗?等等。这样的问题清单虽不能概括所有方面,有时也不可能得到满意的回答,但对设计工作是有益的。

3.7 可生产性工程

可生产性工程专业的目的在于促使设计的结果能与进行生产制造的现实条件相对应。设计并制造出质量高而费用低的产品,需要许多专业的共同努力,其中包括要把关于进行生产制造的知识和设计知识结合起来,进行更有成效的设计权衡,并在生产制造过程中予以实现。

可生产性就是产品的设计特点和生产规划要素对于一件产品所产生的综合效果的反映。换言之,产品的可生产性指生产该产品的相对难易程度,而这种难易程度是受产品设计的特点和生产规划要素所制约的。可生产性好,意味着能够以现实可用的生产技术,按所需的批量,经济地按设计意图制造、装配、检验和测试所需的产品或系统。

上面说的设计特点,是指一项涉及在材料的选择、设计的简化、容差的要求、生产制造途径的灵活性以及配套技术资料的清晰和简洁性等方面所表现出的特色。生产规划要素则是指为完成产品的生产任务,在规划生产制造流程时应加以考虑的生产量、水处理、专用工具要求、人力、设施和原材料供应等有关内容。

强调可生产性的最终目的还在于以最少的费用和最短的时间获取所需的系统。在整个系统工程全过程中,可生产性工程专业都应与其他各工程专业相结合,对影响到系统可生产性的

各方面予以统筹的考虑,力图使系统具有尽可能高的可生产性。

3.7.1 影响可生产性的诸因素

可生产性水平高的产品,应该是能在较短的时间内,利用品种范围比较广的材料,以要求不过高的技能制造出来。能否做到这点,要受到众多因素的制约,对这些因素都应予以恰当的考虑。这些因素主要有以下几点。

1. 图纸和技术规范

工程图纸再加上相应的技术规范和标准等,应当能够完整地描述有关产品的全部特性。从要求具有绝对的确定性考虑,趋于要求图纸和技术规范对一切内容都规定很严格;但这样做,会与达到好的可生产性所要求的具有一定灵活性相矛盾。因此,应从设计的角度仔细地加以考虑,要在图纸和规范的确定性要求与具有更好的可生产性要求间求得合理的均衡。

2. 零部件的选择

由于零部件是组成完整产品的基本单元,它们的状态对产品所能达到的可生产性水平有很大的影响,所以要特别强调正确选用最有效的零部件的重要性。要拟定出控制零部件选择过程的导则、指南和要求等,并在研制过程中不断地对这些导则、指南和要求予以审查和落实。

3. 材料的选择

如 3.5 节所述,材料的选择会影响到加工过程,从而就会影响到生产的进度、成品的费用、成品的可靠性及所需准备的备件数量等,这些对于可生产性来说都是关键性的影响因素。虽然在开始的时候总是从满足性能要求出发去选材,但应注意到当材料的供应情况不佳时会反过来影响可生产性的提高。

4. 加工过程的选择

加工过程的选择与材料的选择有着密切的关系,要正确地确定加工过程,必须对设计与选材和加工过程间的相互关系有透彻的了解。此外,还应充分考虑到实现预定的加工过程的现实性,要为加工过程的选择留有必要的余地。

5. 预期的生产量

产量首先对设计有影响,而确定加工过程时,产量又是决定性的因素,因而它也会影响产品的可生产性。如果在确定产品的基本设计特点时,结合预期的生产量并考虑了易于生产的问题,那么产品的可生产性就有可能进一步提高。

6. 消耗性产品和非消耗性产品的影响

产品是否属于消耗性的,对于进行设计时所要考虑的内容有很大影响,因为这涉及要是否需要维修以及产品的预期寿命等问题。以寿命问题为例,产品的寿命要看预期的使用环境如何。对于战时和平时使用的产品,要按不同的原则去考虑它们的寿命。对战时使用的产品,倾向于不修,因而更多地按可消耗的原则进行设计;而对于平时使用的产品,则更多的



是从经济性的考虑出发,要求具有更长的使用寿命,因而按非消耗的原则进行设计。按消耗性原则设计,还是按非消耗性原则设计,又影响到对产品生产量的考虑。

从求得更好的可生产性水平出发,对于消耗性和非消耗性以及大批量和小批量的不同组合,在设计的处理上应是不同的。主要有以下4种不同的情况:

- (1) 消耗性、大批量,如弹药、防护服等。
- (2) 消耗性、小批量,如某些训练用器材、战略导弹等。
- (3) 非消耗性、大批量,如常规枪支、车辆等。
- (4) 非消耗性、小批量,如大型飞机、舰艇等。

7. 质量保证措施的影响

进行产品检验是质量保证的重要内容,而考虑到提高产品的可生产性,应该针对不同产品的特点,对抽检量、检验内容、检验方法及合格标准等因素慎重地加以权衡。

8. 设计简化程度的影响

在满足设计要求的前提下,使设计尽量简化无疑会大大地提高产品的可生产性水平,但设计简化到什么程度又与所能掌握的加工制造技术有着密切的关系,它受加工技术的制约。例如,采用精锻、精铸或粉末冶金方法,既可以进行精密成型,减少了后续的加工量,又可以把若干本来需要分开制造的几部分合成一体一次制出,从而也减少了装配工作量。可是如果不掌握精密成型手段,则这种从提高产品可生产性出发的设计简化意图就无法实现。

上面提及的各种影响因素,较多地侧重于进行零部件设计时应加以考虑的那些因素。但不应忘记,把这些零件组合在一起的装配过程设计得恰当与否,对组件、装配件以至整个系统的可生产性也都有相当的影响。

3.7.2 可生产性工程实践

与其他工程专业一样,在整个系统工程过程中,可生产性工程专业都要参与有关工程项目的研究、发展、试验和评估活动。这就需要考虑与可生产性有关的各个方面,从不同的侧面进行工作,并将这些考虑纳入系统的研制活动中。图3.9表示出应予以考虑的与可生产性有关的各因素。

在不同的工程发展阶段,可生产性工程的活动重点和工作深度要随着工程项目的进展有所变化。

1. 在方案探索阶段

在这个阶段,系统方案尚未很好地加以

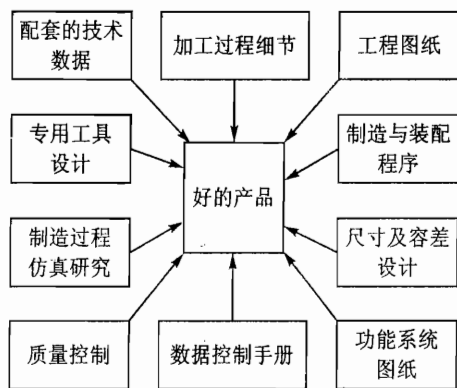


图 3.9 与可生产性相关的考虑因素

确定,此时对可生产性的考虑首先应从具有好的可生产性的要求出发,估测有可能被采用的技术手段及其相应的零部件设计过程与加工制造过程,还要尽早提出关于简单化和标准化方面的具体要求。但应注意可生产性要求与性能、可靠性、维修性和安全性等要求间的权衡,不能因过分强调可生产性而导致总体性能达不到基本要求。

为了与其他方面的考虑更好地协调,必须提出一个落实可生产性要求的计划。该计划规定了后续阶段进行可生产性评审及其有关工作的基准程序和目标要求,也规定了在工程研制阶段为达到预期的生产准备状态应进行的基础工作。

应通过设计要求与生产要求间的权衡,切实地进行初步的可生产性分析。此时,应考虑的问题可能包括:各种备选的制造和装配方法及实际具有的能力;备选的机械加工能力;关键材料的现状和展望;对安全、检验、测试、质量保证等方面的技术需求与实际可能间的差异;初步生产费用的估计及费用与进度间的平衡;技术风险,特别是与采用先进技术相关的风险因素等。

显然,为了周全地考虑到这些因素,需要设计、材料、质量保证和后勤供应方面的专业人员协同工作,其中生产制造工程专业的人员需要估计现有的和预计需要的工业生产能力和规模,并以此提出相关的建议,以保证能在预定的进度约束下,在后续阶段有充裕的时间落实这些建议。

毫无疑问,参与各种技术评审是把可生产性与其他工程技术要求恰当地结合起来的一种保证手段。比如,对于在发展初期可以预见到的重大的可生产性相关的因素(关键材料、工艺装备、加工方法和设施等),在进行系统要求评审时就应提出来加以考虑。

2. 在方案确定阶段

在这个阶段要核准设计方案,批准系统的研制费用和进度。此时,关于可生产性的考虑要更为具体,并且要进行更为大量的权衡分析。在设计仍处于进展的状态时,对可生产性(当然也包括其他工程专用)深入地加以考虑将对以后的设计工作带来很大好处。

该阶段进行可生产性的考虑重点在于提出详细的可生产性目标,并进一步仔细判断所存在的风险因素。比如,对材料,要考虑其供应来源、供货期及标准化等问题;对加工制造过程,要考虑预计的工艺技术能否实现,所有首创方法的可行性与存在的风险,以及进行验证性试验的准备条件等问题;对设计过程,要考虑零部件与所用材料是否已最大限度的标准化,设计中是否已反映了可生产性要求,以及现有的和新的工业资源的满足程度等问题。

通过评审,应明确前一阶段工作的成果,并根据样机试验和演示验证的结果判断仍然存在的或新出现的风险因素,同时要提出进一步的工程发展要求。比如通过系统设计评审,就应保证在设计决策中反映了可生产性分析的结果,同时又不致因过分照顾到可生产性要求而影响不能达到系统的总体性能要求。

3. 在工程研制阶段

应通过设计、试验、验证等工程活动,保证系统发展工作能顺利地转入实际生产阶段。为

此,应按规定的费用要求完成所有的生产设备和辅助性设备的设计,做好一切与向生产阶段过渡有关的准备。此时,做好涉及可生产性的工作,对于降低生产费用和优化生产进度有很大的作用。

在这个阶段,强调要对最终设计所能达到的可生产性水平进行验证,应继续执行或在必要时修订可生产性计划,一定要从材料的选择、制造技术和装配方法的落实等方面提前做好准备。

在从研制向生产过渡的过程中,凡解决有关生产计划、生产进度、生产流程、车间布置、材料的处置、加工方法、工艺装备和检验与测试设备等方面的问题时,都必须考虑到与可生产性有关的要求。

在工程研制阶段,要开展多种评审活动。它们都对可生产性有不同程度的影响。无论对于系统本身,还是系统的保障设备,对每个配置项目进行评审时,都要包含可生产性评审的内容。值得注意的是,随着设计被固定下来,进一步改善可生产性水平的机会越来越少了。

4. 在生产和使用阶段

进入生产阶段并不意味着与可生产性相关的工作业已终结。实际上,设计和生产往往是要并行开展,特别是对那些需要较早地提前进行生产准备工作的项目更是如此。例如工艺装备、原材料及外购件等都需要及早着手筹划。

虽然这时可生产性的考虑已不再像前面几个阶段那么强调,但如果此时仍十分注意尽量采用新的和成熟的生产工艺技术,并且注意保证在随后陆续作出的设计和加工工艺上的变动都能对提高可生产性水平有进一步的促进作用,那么仍然能够大大地降低生产费用。

在生产和使用阶段,与可生产性相关的过程工作更多地集中于对各种设计变更和生产过程变更进行评定方面,其目的是保证使这些变更至少不会降低原有的可生产性水平。

在工程设计实践中,可生产性工程发挥着积极作用。要使设计中预想的可生产性水平得以实现,还要在生产和使用保障过程中继续发挥可生产性工程专业的作用。

随着系统的不断更新和发展,系统的功能日趋综合化,使用环境也更加复杂多变,因而系统综合的范围也处在持续地变化和扩展的过程中。从系统综合的全局看,进行综合所涉及的工程专业除本章所述以外,还有其他许多工程专业的要求应予以考虑,例如保障性、可测试性、结构完整性、生存性与易损性、电磁兼容性、污染与腐蚀控制以及硬件与软件间的协调综合等。

第4章 综合保障工程

综合保障工程不仅关系到飞行器系统的战备完好性、作战效能和寿命周期费用或总拥有费用的高低,而且直接影响到飞行器的机动作战能力、高强度作战能力和持续作战能力的优劣。随着飞行器技术性能和复杂度的提高,导致飞行器的使用保障费用不断提高,目前,甚至达到全寿命周期费用的70%~80%。

传统的飞行器管理,重视设计制造,轻视使用保障,忽视综合保障,往往是在飞行器设计定型以后,甚至等到飞行器交付使用以后,才考虑各种保障问题,所以经常造成飞行器投入使用后迟迟不能形成战斗力的现象。对此,国外提出了“综合后勤保障(Integrated Logistics Support, ILS)”。在借鉴国外经验的基础上,我国先后制定并颁布了 GJB 3872—99(装备综合保障通用要求)、GJB 1371—92(装备保障性分析)、GJB 3837—99(装备保障性分析记录)等军用标准。综合保障工程与作战能力的关系如图 4.1 所示。

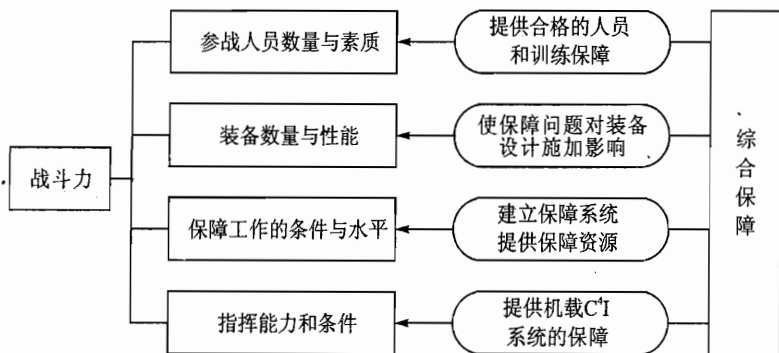


图 4.1 综合保障与战斗力的关系

对于综合保障, GJB 9001A—2001 中规定:“运用优化设计和可靠性、维修性、综合保障等专业工程技术进行产品设计和开发”;“按规定的要求确定并提出产品交付时需要配置的保障资源”;“规定产品使用所必需的保障方案和保障资源要求”;“交付的产品需经顾客验收合格,按规定提供有效技术文件、配套备附件、测量设备和其他保障资源”。

4.1 保障性

4.1.1 保障性定义

保障性(supportability)是指飞行器系统的设计特性和计划的保障资源能满足平时战备和

战时使用要求的能力。其中“系统的设计特性”是指与飞行器系统保障有关的设计特性,如使用(充填加挂)、维修、运输等。这些设计特性都需要设计人员充分考虑飞行器具有便于保障的特征,使飞行器设计得容易保障。“计划的保障资源”则是指飞行器使用与维修保障中所需要的人、物、信息等资源。这些资源所构成的保障系统与飞行器相互匹配,保证飞行器投入使用后,为达到规定的使用要求,能及时获得所需的设施、设备、备件、器材、工具、人力、技术资料等,并能够得到保障。飞行器系统具有便于保障特性和能够得到保障才是完整的保障性。

保障性与第3章中介绍的可靠性、维修性、测试性一样,是现代飞行器系统设计所考虑的一个综合的系统特性,成为与性能、费用和进度同等重要的参数。

4.1.2 保障性要求

保障性要求是指包括定性与定量描述用户对飞行器保障性的具体要求。保障性的目标是多样的,难以用单一的参数来评价,同时某些保障资源参数很难用简单的术语进行表述。通常的作法是:通过对飞行器的使用任务进行分析,考虑现有飞行器保障方面存在的缺陷以及保障人力费用等约束条件,综合归纳为一整套保障性参数,有些参数还要采用与现役飞行器对比的方式进行表述,而且保障性要求应尽量量化。

1. 定性要求

定性要求包括两个方面:一是与装备保障性设计有关的定性要求,主要是生存性、运输性、便于战场抢修等(便于装备实施保障的设计特性,例如便于操作,便于装卸运输和便于补充油、液、气、电、弹药等)的定性设计要求;二是保障系统及其资源的定性要求,主要是指在规划保障时要考虑、要遵循的各种原则和约束条件。归纳如下:

- (1) 装备应便于保障并减少保障工作量。
- (2) 提高装备自保障能力,减少对保障系统的依赖。
- (3) 要考虑装备的特殊保障要求。
- (4) 提高装备和保障资源的标准化、互换性程度。
- (5) 装备和保障资源有良好的运输性,便于包装、装卸、储存、运输。
- (6) 装备和保障资源有良好的环境防护能力,避免恶劣环境的伤害。
- (7) 装备与保障资源以及保障资源相互之间有良好的设计接口,技术协调、配套。
- (8) 保障资源有良好的继承性、通用性,减少品种规格要求。
- (9) 保障资源应满足齐全、配套、简便、适用、安全、快捷、经济、耐用等项要求,并与装备同步研制、同步交付。

初始保障方案是定性要求的重要组成部分,它是确定保障资源和建立保障系统的基础。

2. 定量要求

1) 保障性综合参数

保障性综合参数是描述保障性目标的参数,通常可用战备完好性目标值(readiness objec-

tive)来衡量。表 4.1 列出了与飞机和导弹相关的保障性综合参数。

表 4.1 常用的战备完好性参数

飞行器类型	典型战备完好性目标参数
飞机	使用可用度(A_0): 能工作时间与总时间之比
	能执行任务率(MC): 24 h 执行某规定任务所占时间之比
	出动架次率(SGR): 24 h 平均出动架次
导弹	使用可用度(A_0)
	储存可用度(SA)(导弹检测合格率)
	能执行任务率(MC)
	平均技术准备时间: 包括导弹检测时间、挂装飞机时间、拆卸工作时间等
	平均发射前准备时间

2) 保障性设计参数

保障性设计主要是根据外场装备使用、运输、维修等要求开展主装备与保障资源的接口设计。因此,保障性设计参数主要包括:

- ① 设备的尺寸、重量、功率等参数要求;
- ② 系统和设备的预防性维修周期。

3) 保障资源参数

保障资源参数分为保障系统参数和保障资源参数。

保障系统参数主要包括:

- ① 平均保障延误时间;
- ② 备件更换维修周转时间;
- ③ 各维修级别故障修复百分数;
- ④ 机动转移时保障资源的运输量。

这些要求对战备完好性有直接影响,是衡量保障系统效能的重要度量指标。

保障资源参数主要包括:

- ① 随装工具数量与重量;
- ② 各维修级别器材库存量;
- ③ 各维修级别保障设备利用率;
- ④ 各维修级别备件利用率;
- ⑤ 各维修级别保障设施利用率;
- ⑥ 使用人员数量与技能等级及培训率;
- ⑦ 各维修保障机构维修人员数量与技能等级及培训率;

⑧ 技术资料的种类、数量与差错率。

这些要求是经济性和战备完好性权衡的结果,将直接影响规划保障资源的结果,并影响保障系统的设计。

4.2 综合保障

4.2.1 基本概念

由于飞行器使用与保障的各个方面分散在各个部门管理,按照传统的做法,保障常常是在主系统设计完成之后,即将设计“冻结”之后,有的则是在系统交付用户之后,逐步在使用中开发。也就是在使用中遇到什么问题,解决什么问题。事实证明这种做法不仅造成保障费用高,而且系统长期形不成战斗力。综合保障这一概念,就是在这样的背景下,为了以可负担的寿命周期费用实现系统战备完好性目标而提出的。

综合保障是指在飞行器的寿命周期内,为满足系统战备完好性要求,降低寿命周期费用,综合考虑飞行器的保障问题,确定保障性要求,进行保障性设计,规划并研制保障资源,及时提供飞行器所需保障的一系列管理与技术活动。

综合保障作为系统工程过程的一个有机组成部分,强调:

(1) 保障系统与主装备同步论证、同步设计、同步试验、同步交付使用。

(2) 采用系统分析和综合协调的方法。其中包括:飞行器的使用要求与保障要求间的综合协调;飞行器设计与保障系统设计间的综合协调;飞行器所需的各类保障资源间的综合协调;新研飞行器与现役飞行器保障系统的综合协调;作战性能、保障性能、研制进度、费用间的综合协调。其主要目标是用可承受的寿命周期费用,在飞行器投入使用的时候,提交与飞行器相匹配的保障系统。

有两种规划装备所需保障的方法(见图4.2):传统的方法是先进行飞行器设计,待飞行器设计完成后,再设计保障系统;而综合保障则采用飞行器和保障系统的设计同步进行的方法。在设计初期,应统筹考虑飞行器与保障系统之间,以及保障系统内部各综合保障要素之间的相互关系。

只有综合权衡分析的方法才能达到满足任务要求的目的。这也是与先设计飞行器后规划保障的传统作法的显著区别。

综合保障的主要工作有:

- ① 制定与飞行器战备完好性目标和与飞行器设计间有最佳关系的保障性要求;
- ② 有效地将综合保障考虑纳入飞行器设计;
- ③ 研制与获取所需的保障资源;
- ④ 以可承受的寿命周期费用建立保障系统,实现飞行器早期投入使用,形成初始战斗能力。

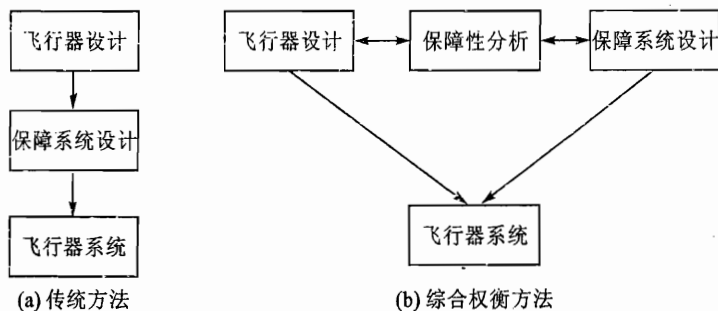


图 4.2 两种规划保障工作的方法

4.2.2 装备保障

装备保障是指为使飞行器处于战备完好状态并能持续完成作战任务所需的保障工作。装备保障主要包括飞行器的使用保障和维修保障，基本不涉及作战保障(气象、侦察、警戒、伪装等保障)和后勤生活保障(生活物资供应、医疗救护保障)。

1. 使用保障

飞行器的使用保障是指为保证飞行器正确操作使用，以便能充分发挥其作战性能所进行的一系列保障工作。在飞行器研制过程中，对使用保障的考虑主要包括：

- ① 使所设计的装备便于操作；
- ② 能迅速、有效地供应能源，加挂弹药和补充能力良好；
- ③ 有完善、适用的使用保障技术文件；
- ④ 使用中所需的检测设备及工具便于操作、携带和运送；
- ⑤ 适用的运输方式和运输工具；
- ⑥ 飞行器具有自保障能力(如机载辅助动力、机载制氧设备等)；
- ⑦ 有抢救与自救能力；
- ⑧ 飞行器能合理、方便地储存，并保障使用完好；
- ⑨ 有飞行器适用的场站、训练场所、仓库、码头设施等。

2. 维修保障

飞行器的维修保障是指为保持和恢复飞行器完好的技术状况所应进行的一系列维修工作。维修保障除各种预防性维修、修复性维修和战场抢修外，还包括保障维修所需的各种人员培训、器材供应、设备设施、技术资料等保障工作，是部队技术保障的主要内容。综合保障在飞行器研制过程中对维修保障的考虑主要包括：

- ① 制定合理的维修保障方案(预防性维修、修复性维修、战场抢修等)，规划维修所需的保障资源；

- ② 优化预防性维修工作,以减少维修停机时间和维修人员配备;
- ③ 便于进行修复性维修,尽量采用通用和简易的工具和设备;
- ④ 有简明而适用的维修技术文件,便于维修人员操作;
- ⑤ 易于实施维修人员的训练,维修人员不需要过高的文化水平;
- ⑥ 战场抢修所需工具及设备便于使用、携带和随同战斗部队转移;
- ⑦ 维修备件配套定额和供应方案力求标准化,减少供应品种和数量;
- ⑧ 有适应各维修级别的固定设施及相应的维修设备等。

4.2.3 综合保障要素

综合保障要素分为技术与管理 and 保障资源两大类,共 10 个要素,如图 4.3 所示。

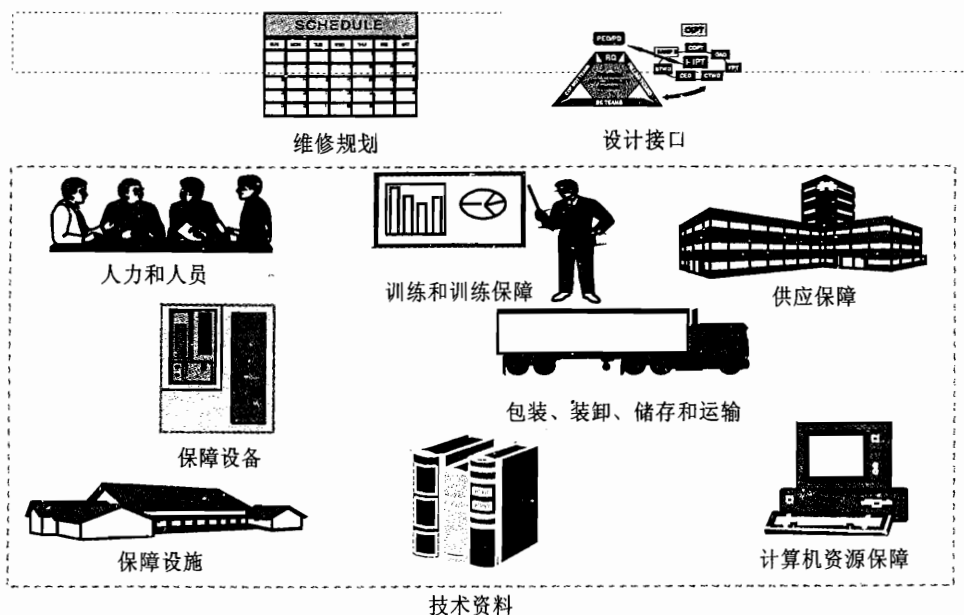


图 4.3 综合保障要素

1. 技术与管理要素

这一类要素综合考虑了保障资源各要素的配置及保障系统与系统保障性特性的关系。它共有两个要素,即维修规划和设计接口。

1) 维修规划

维修规划是指飞行器寿命周期中研究和制定维修保障方案,以及规划具体实施这种方案的各项工作的。

几乎综合保障的每一个要素都与维修规划有关,它们都需要维修规划提供信息,因此维修规划活动中的各种变化都必须及时提供给有关专业,以便在各专业活动中做出相应的变更。

2) 设计接口

设计接口主要指飞行器设计与保障系统设计之间的接口,它说明有关保障性的设计参数与战备完好性及保障资源要求之间的相互关系。例如战备完好性的可用性、战斗准备时间或再次出动时间等与所需的保障资源能否及时有效的提供有关,同时也与飞行器的某些使用可靠性和维修性参数有直接关系。可靠性和维修性的某些设计特性是与保障资源的确定有着密切的关系,例如根据故障模式影响分析确定维修工作内容,根据零部件拆装时间选用或设计拆装工具等。同时,保障工作也对飞行器设计提出相应的要求,如果在制定飞行器预防性维修计划对某些维修工作类型难以确定,则需要更改飞行器的设计,或为了使维修工作类型综合得合理,要求提高某些零部件的可靠性等。

2. 保障资源要素

保障资源要素共有 8 个,即人力和人员,供应保障,保障设备,技术资料,训练和训练保障,计算机资源保障,保障设施,包装、装卸、储存和运输。

1) 人力和人员

该要素是指平时与战时各维修级别上使用与保障飞行器所需的人员数量、专业及技术等级。以歼击机为例,简要说明如下:

- ① 机务大队设 1 个大队部、3 个外场中队、1 个定检中队、1 个导弹中队;
- ② 大队部设 1 个质量控制室、1 个飞参处理室;
- ③ 外场 3 个中队各设机械分队 2 个,军械、特设、飞控、火控、通导、质检分队各 1 个;
- ④ 定检中队设机械分队 2 个,军械、特设、飞控、火控、通导、电抗、质检、座椅、修理、附件分队各 1 个。

专业划分包括外场维修专业(机械、军械、特设、通导、火控、飞控共 6 个专业)和内场定检专业(机械、军械、特设、通导、火控、自控、座椅、电抗、附件、修理共 10 个专业)。

2) 供应保障

该要素是指规划、确定飞行器和设备补给品(备件、消耗品)的采购、分类、接收、储存、转运、配发以及报废处理等要求所需要的全部管理活动、规程、方法与技术。它包括初始供应保障、后续供应保障以及停产后的供应保障。

3) 保障设备

该要素是指保障飞行器和设备使用和维修所需的全部设备(移动的、固定的、临时的设备)。它包括搬运设备、拆装设备、维修设备、工具、计量与校准设备、试验设备、测试设备及监测故障诊断设备、修理工艺设备等,还包括保障设备自身的保障设备,如清洗油车的清洗设备等。

以歼击机为例,其地面保障设备分为 4 大类:专用地面维护设备、飞机技术维护工具、飞机维护地面检测设备和通用地面维护设备。

专用地面维护设备分为 10 类:牵引和系留设备、起重设备、梯架、拆装设备、系统维护设备、特种系统维护设备、停放维护设备、维护用具、系统维护用检测设备等。



飞机技术维护工具分为16类:1:1飞机技术维护工具、1:10机械专业飞机技术维护工具、1:10军械专业技术维护工具、1:10氧气专业维护工具、1:10仪表专业维护工具、1:10无线电专业维护工具、1:10电气专业维护工具、1:30飞/发定检工具、1:30军械定检工具、1:30航空设备定检工具、1:30无线电设备定检工具、1:30救生系统定检工具等。

飞机维护地面检测设备分为5类:移动式检测设备、飞机/发动机维护地面检测设备、航空军械维护地面检测设备、无线电电子设备维护地面检测设备、航空设备维护地面检测设备。

4) 技术资料

该要素是指将飞行器和设备要求转化为使用与保障所需的各种工程与技术信息的记录。它包括工程图样、技术规范、技术手册、技术报告、规程、细则、说明书、计算机软件文档等。这一切都适用于主装备、保障设备、训练器材、运输与搬运设备以及各种设施等。

以歼击机为例,第一部分为编号文件,是关于飞机、发动机、机载设备、地面设备和检测仪器的出厂履历本或合格证;第二部分为标准(校准)资料,是关于飞机各系统的技术说明与维修工艺方面的使用维护技术资料;第三部分为配套产品文件,是关于机载成品附件的技术使用手册,包括发动机、飞机各系统、航空军械、无线电设备、航空设备方面的机载成品附件的技术说明、使用手册、电路图册与航空弹药资料汇编等技术资料;第四部分为地面保障设备与检测仪器的技术说明与使用技术资料。

5) 训练和训练保障

该要素是指对飞行器和设备的使用维修与保障人员的训练过程、方法、规程、技术、训练器材与设备以及这些器材与设备的研制及保障规划。它包括单兵和集体训练、新飞行器训练、初始训练、正规训练和在职训练,以及训练设备的采办和安装。

6) 计算机资源保障

该要素是指使用与保障飞行器中的嵌入式计算机所需的设施、硬件、软件、文档、检测仪器、人力和人员。

7) 保障设施

该要素是指使用与保障飞行器和设备所需的永久性与半永久性的构筑物及其设备,如维修车间、训练场地及仓库、码头、船坞等。它包括对设施类型、设施设计与改进、选址、空间大小、环境要求及设备等方面的考虑。

8) 包装、装卸、储存和运输

该要素是指为保证飞行器和设备及其保障设备、备件得到良好的包装、装卸、储存和运输所需的资源、程序、设计考虑和方法。其中包括环境考虑、短期或长期的保管和运输性要求。

4.2.4 保障系统、保障方案、保障计划

保障系统是在寿命周期内用于使用和维修飞行器的所有保障资源及其管理的有机组合,是为达到既定目标(如使用可用度)使所需的保障资源相互关联和相互协调而形成的一个系

统,有时也称为保障分系统。虽然各类保障资源是根据飞行器战备完好性目标而研制和选用的,但只有保障资源还不能直接形成战斗力,需要将所有的使用与维修保障资源有机组合起来,形成保障系统,才能发挥每项资源的作用。

保障系统包括飞行器所需要人力、物力、信息等各种资源以及这些资源的管理。只有通过科学合理的管理,才能将分散的各种资源组成具有一定使用和维修功能的系统。保障系统主要由使用保障、维修保障、备件供应和人员保障等功能组成。因此保障系统又是为了实现上述保障功能的一整套资源与管理的系统。

保障方案是保障系统完整的总体说明,它满足飞行器的保障要求,并与设计方案及使用方案相互协调,一般包括使用保障方案与维修保障方案。

使用保障方案规定了装备规模、使用地域、任务剖面 and 装备使用方式、装备集中和分散使用的保障供应、装备储存和运输方式、油料与弹药的补充、使用人员的约束及主要保障资源的要求等。

维修保障方案包括维修类别、维修级别及其任务、维修间隔期、预计的主要维修资源和维修活动约束条件等。

保障方案构成如图 4.4 所示。

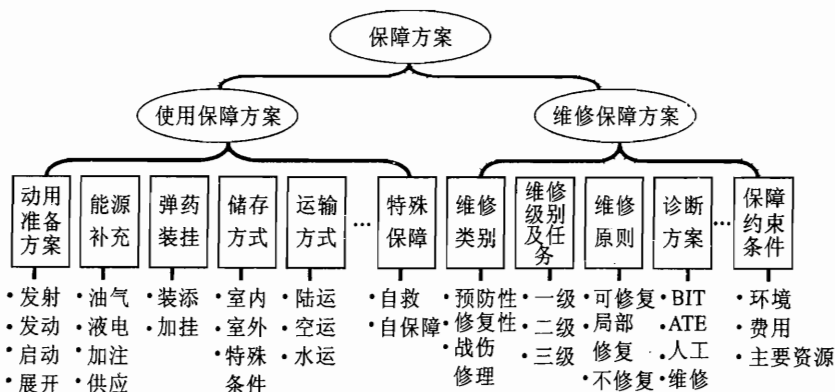


图 4.4 保障方案构成

从保障方案包括的内容可知,制定这种方案要考虑装备特点、保障和使用特性、资源要求和部队实际状况等诸多因素,因此应按功能分析、系统综合先拟定备选方案,再经权衡分析最后确定。

保障计划是保障性分析工作中的一个重要环节,它是一个详细的保障方案。保障方案通常只是装备保障系统的总体说明,而保障计划则进一步说明保障方案的具体实施内容与要求,它包括所有使用、维修工作和各类保障专业所需的硬件、软件项目的详细要求。保障计划可分为维修保障计划和使用保障计划。

保障方案、保障计划和保障系统之间在寿命周期中的基本关系如图 4.5 所示。保障方案

的制定是一个动态过程,自飞行器方案探索时提出初始保障方案,通过方案确定阶段和工程研制阶段对不同备选保障方案的权衡分析,得到优化的保障方案,并在工程研制阶段的后期进一步完善。保障方案和由它细化而成的保障计划等在研制过程中要通过评审、验证来评价。

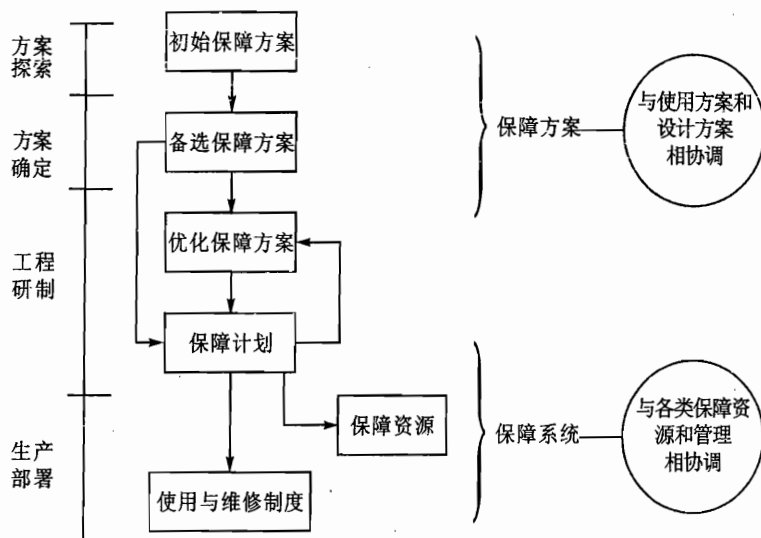


图 4.5 保障方案、保障计划和保障系统之间的基本关系

4.2.5 综合保障计划与综合保障工作计划

综合保障工作是飞行器系统工程过程的重要组成部分,贯穿于整个寿命周期。为了对寿命周期过程中的综合保障工作实施规划与管理,订购方必须从型号立项时起,即着手制订综合保障计划,全面规划并指导飞行器寿命周期的综合保障工作;承制方应根据订购方制订的综合保障计划,制订自己的综合保障工作计划,全面规划承制方(包括转承制方)的综合保障工作,以满足合同中规定的保障性定量、定性要求和综合保障工作要求。

1. 综合保障计划

综合保障计划是综合保障工作要求和职责方面的管理性文件。它包括了综合保障所有工作项目的细节及其与型号项目的关系,为决策人员提供了有关综合保障方面的必要信息。其中的保障计划是飞行器使用与维修的完整且详细的说明,对装备的保障具有重要的指导作用。

该计划应明确规定需要完成哪些综合保障工作,怎样完成,何时完成和由谁完成,亦即明确各种保障工作的目标、方案和实施方法。

综合保障计划是由订购方制订的,它是飞行器寿命周期各阶段开展综合保障工作的指导文件。在方案探索阶段应拟定初始的综合保障计划,并随着研制工作的进展补充完善。其中由承制方完成的工作应在合同中确定。

2. 综合保障工作计划

综合保障工作计划是有关承制方完成其承担的综合保障任务的工作程序、基本方法、进度要求和规划性文件。它由承制方在实施综合保障的整个工作过程中随着型号研制进展反复迭代而制订的。该计划有时也是招标书中要求提交的文件。在寿命周期不同阶段中,承制方提出的有关工作计划经订购方同意后,也可纳入综合保障计划作为附件,同时也可作为修订原有综合保障计划的参考。此计划有时可能成为招标书中的一部分。

这项计划大体上可分为4部分。

(1) 总体说明:承制方根据订购方提出的综合保障要求或综合保障计划的总则,按合同规定的范围和要求,编制综合保障工作计划。

(2) 综合保障工作的管理:包括管理的目标和原则、组织机构和职责范围、与订购方的协调关系、评审程序以及管理控制系统等。

(3) 综合保障工作的主要内容:制定保障方案、维修规划、保障性分析、试验与评价以及综合保障各要素的主要活动及其它它们之间的关系。

(4) 进度计划:综合保障工作计划的进度包括3个层次的进度,即型号研制进度、综合保障工作计划进度和各要素的进度。后两个进度必须相当详细并协调一致。

4.3 保障性分析

飞行器在使用过程中必须有与之相匹配的保障系统才能确保完成任务。要做到二者相匹配,使保障影响飞行器设计,又能提出正确的保障资源要求,必须同步研制飞行器及其保障系统。其关键在于获得与飞行器设计方案、使用方案协调一致的保障方案。4.2.5小节所述的综合保障计划的制订基本依据来自进行保障性分析所得的结果,而综合保障计划的适用性也要通过保障性分析加以判断,并根据分析的结果作适当修改。保障性分析是保证综合保障工程实现其目标的有效手段,也是实现综合保障工程与其他系统工程综合、协调的联系桥梁,是系统工程的一个组成部分。

一个完整的飞行器系统,应包括在规定的工作环境下使系统的工作和保障可以达到自给所需的一切设备、设施、器材、软件、数据、服务和人员。结合综合保障工程的概念可以把飞行器系统理解为飞行器与其保障系统的有机组合。因为保障系统正是将使用与维修保障所需的人员、备件、保障设备、软件和设施组合在一起,如果再加上飞行器(包括其上的各种设备)就是非常典型的飞行器系统。图4.6为飞行器系统的组成示意图。

从飞行器及其保障系统来看,二者不是简单的叠加关系。它们是从任务需求出发,通过对使用方案、设计方案和保障方案的综合分析和权衡,求得最佳的设计方案和保障方案,以满足任务要求,因此飞行器和保障系统是同步研制并综合协调的。

从飞行器的具体设计过程来看,传统的工程学科(机械学、热力学、电子学、流体力学……)需要和可靠性、维修性、人素工程、综合保障工程等专门工程学科综合,以进行飞行器具体设

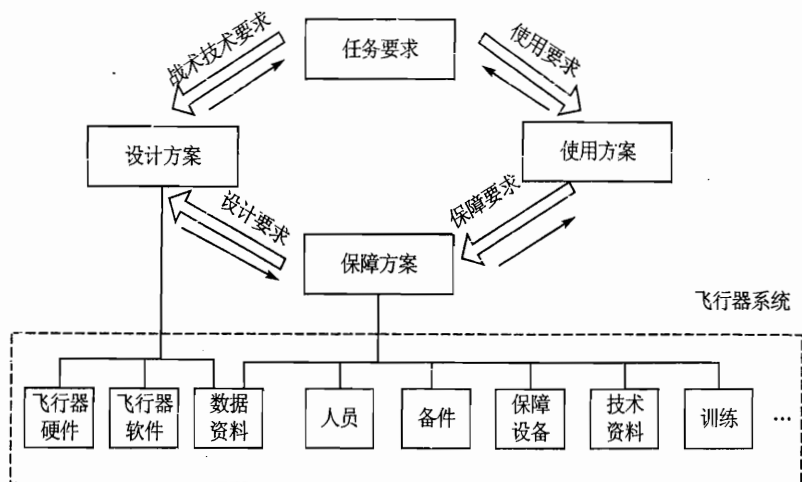


图 4.6 飞行器系统组成示意图

计。这也体现了飞行器及其保障系统的有机综合。

飞行器与其保障资源在设计上的协调和对任务的影响,可用飞机加油这一简单的保障问题为例加以说明。加注燃油时间,属于战斗准备时间,将直接影响飞机的战备完好性,要缩短加油时间,可采用高压和大流量的加油泵,以提高输油速度(这属于保障资源要求方面的问题),同时还要使加油能力与油箱布置、机内油道管路和输油接口设计所形成的受油能力(属于与保障有关的设计方面的问题)相匹配,才能达到目的。

4.3.1 保障性分析的概念

保障性分析是综合应用一些分析技术(如故障模式影响及危险性分析、以可靠性为中心的维修设计分析、寿命周期费用分析、费用效能分析等)对系统和设备的综合保障所进行的分析。国外称其为“后勤保障分析(Logistics Support Analysis, LSA)”,也可称为综合保障分析。

保障性分析的作用可概括为以下4个方面:

- (1) 提出和确定与保障有关的设计因素,用于影响飞行器设计,使飞行器设计既满足任务要求,又便于实施保障。
- (2) 在研制飞行器过程中,应尽早地确定影响保障和费用的主导因素,以便确定分析工作的重点,并及时制定改进的目标和解决的方法。
- (3) 提出飞行器使用与维修所需的各类保障资源的要求,以便进行保障资源的研制与采购。
- (4) 利用分析所得到的数据资料形成保障性分析记录,建立统一的保障数据库,便于飞行器设计和综合保障工作的使用。

保障性分析在寿命周期内与飞行器设计和保障系统设计之间的基本关系如图 4.7 所示。

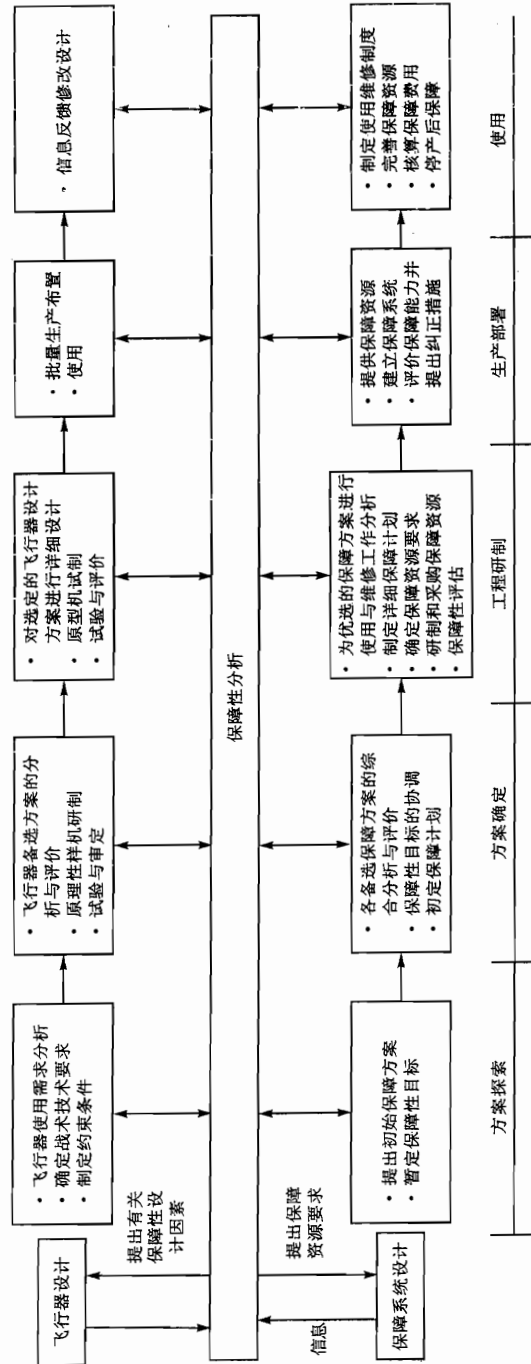


图4.7 飞行器及其保障系统设计与保障性分析的关系

由图可知,保障性分析对飞行器的输出是提出与保障有关的设计因素,而保障系统的输出则是保障资源要求。

4.3.2 常用的保障性分析方法

如图 4.8 所示,常用的保障性分析方法主要包括故障模式影响及危害性分析(FMECA)、以可靠性为中心的维修分析(RCMA)、修理级别分析(LORA)、使用与维修任务分析(O & MTA)、费用分析等。

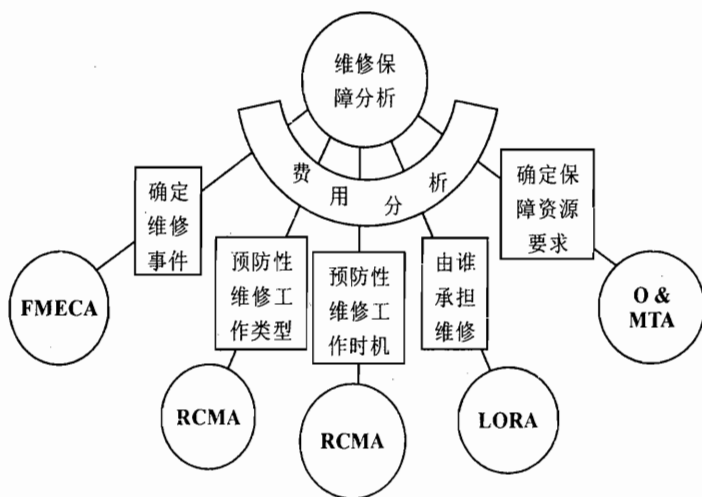


图 4.8 保障性分析方法

1. 故障模式影响及危害性分析(FMECA)

FMECA 用于确定修复性维修项目、预防性维修分析的输入、制定故障查找程序。

在规划维修的分析过程中,FMECA 的主要作用是确定修复性维修项目和要求。首先分析出产品可能存在的各种故障模式,然后分析这些故障能否通过改进设计来消除或用预防性维修来减缓。如果不能,则应确定进行修复性维修。在此基础上,就可以考虑故障的诊断、隔离与定位,拆卸与分解,更换故障件(或原件修复),组装和安装、调试等作业,从而可以提出检测手段、拆装工具、所需备件、人员、技术资料等与规划保障资源相关的信息。

FMECA 的具体做法参见 GJB 1391—92《故障模式影响及危害性分析程序》。

2. 以可靠性为中心的维修分析(RCMA)

RCMA 用于确定预防性维修的工作类型,探索维修工作间隔期。

RCMA 有 7 种工作类型:保养、操作人员监控、使用检查、功能检查、定期拆修、定期报废、综合工作。

RCMA 是按照以最少的维修资源消耗保持装备固有可靠性和安全性的原则,应用逻辑决

断法确定装备预防性维修要求的分析方法。其目的：一方面是为设计更改提供必要的输入；而另一方面(更重要的一个方面)是确定需要进行预防性维修的工作项目,包括其维修工作类型及预防性维修工作的间隔期,从而可以通过分析提出进行预防性维修工作所需的保障资源。

1) 系统和设备以可靠性为中心的维修分析方法

此方法适用于各种类型的设备预防性维修大纲的制订,以确定系统和设备的预防性维修要求,包括维修的产品和项目、维修方式或维修工作类型、间隔期以及维修级别。其步骤如下:

- ① 确定重要功能产品;
- ② 进行故障模式影响分析(FMEA);
- ③ 应用逻辑决断图确定预防性维修方式或维修工作类型;
- ④ 确定预防性维修工作的间隔期;
- ⑤ 确定维修级别;
- ⑥ 进行维修间隔期探索。

2) 结构以可靠性为中心的维修分析方法

此方法用于确定结构项目的预防性维修要求,主要是检查维修等级、方法和间隔期以及维修级别。该方法适用于飞机的结构,由于目前结构的故障仍属隐蔽功能故障,对操作人员是不明显的,所以必须进行预防性维修。适用于结构项目预防性维修的工作类型只有两种:视情检查和对安全寿命结构项目的定时报废。视情检查按一定间隔期进行,分为一般目视检查、详细目视检查和无损检测3种。

3) 区域检查分析方法

此方法适用于需要划区进行检查的大型飞行器装备。除了对重要功能产品、重要结构项目应计划一定的维修工作以外,对于非重要功能产品、非重要结构项目以及电缆、导管、接头等管线的一般状况,也应定期检查。因为它们除了使用耗损以外,还会受到因维修邻近装置而引起的偶然损伤。此方法并不针对某一产品(项目)或其某一故障模式,而是检查产品(项目)的明显问题,如松动、过热、腐蚀、劣化、裂纹、磨擦、脱开、丢失、泄漏、偶然损伤等。

RCMA的具体做法参见GJB 1378—92《装备预防性维修大纲的制定要求与方法》。

3. 修理级别分析(LORA)

LORA用于确定维修工作在哪一级进行,包括经济性和非经济性分析两类。

LORA的目的是确定维修工作在哪一个修理级别上执行最为经济有效。在维修规划的分析过程中,应用LORA进一步确定各修理级别的维修工作内容,从而为确定每个修理级别上的保障资源提供输入。

LORA的具体做法参见GJB 2961—97《修理级别分析》。进行LORA工作时,可按该标准附录E“应用指南”对工作项目加以剪裁。

4. 使用与维修任务分析(O & MTA)

O & MTA用于确定每项使用保障任务和每一修理级别维修任务所需的保障资源要求。

使用与维修工作任务包括使用工作任务、预防性维修工作任务、修复性维修工作任务和战伤修理任务。

应对每项任务进行详细分析,将任务划分为作业和工序,然后对执行每个工序所需的人员及技术水平、工具、设备、备件和消耗品、所需的工时和时间、采用的标准与技术文件等逐个加以分析,并列出清单,进而汇总提出所需的保障资源要求。

5. 其他方法

1) 费用分析

费用是综合保障追求的目标之一,也是飞行器研制的约束条件,因此需要在综合保障工作实施过程中做好费用分析与控制。

2) 比较分析

选定(或构建)一个能代表新飞行器特性的基准比较系统,与新飞行器对比,从而提出新飞行器的保障方案,并进一步修改完善。

4.3.3 保障性分析信息

在飞行器研制过程中,开展综合保障工作需要大量的信息,这些信息主要通过保障性分析产生,所以称为保障性分析信息。对这些信息的记录、处理和应用是保障性分析工作的重要组成部分。由于保障性分析信息的应用广泛、作用重大,记录和处理工作又十分繁杂,因此日益受到重视。当前,我国已参照美军标 1388-2B 制定了国军标 GJB 3837,提出了有关保障性分析数据记录要求,以规范保障性分析记录。

保障性分析信息主要包括:

(1) 飞行器系统的使用与维修信息。这些信息明确了对产品使用与维修的要求,包括任务和使用要求数据、平均故障间隔时间、平均修复时间、所保障的产品的总数、年计划的维修工时等。

(2) 飞行器系统的可用性、可靠性、维修性、故障模式影响分析信息。

(3) 飞行器系统使用与维修工作清单、使用与维修工作任务分析、人员与保障要求信息。还包括每一项可修项目的使用与维修工作任务,以及完成每项使用与维修工作任务所需要的训练、人员、保障设备和供应保障要求等。

(4) 保障设备与训练器材要求信息。

(5) 受试件的要求与说明信息。

(6) 设施要求信息。

(7) 人员数量和专业技能要求信息。

(8) 包装和供应要求信息。

(9) 运输性工程分析信息。

- 将保障性分析所得的大量数据和信息按统一规定的格式填写,形成保障性分析记录,实质上是建立一个独立的数据库,作好数据管理工作。它规定了数据元定义和编码、数据字段长度和格式以及数据关系表等,以便实现自动数据传输和处理。保障性分析数据流和系统工程的接口关系如图 4.9 所示。图中,A、B、C…分别表示对下一层次的数据关系式。

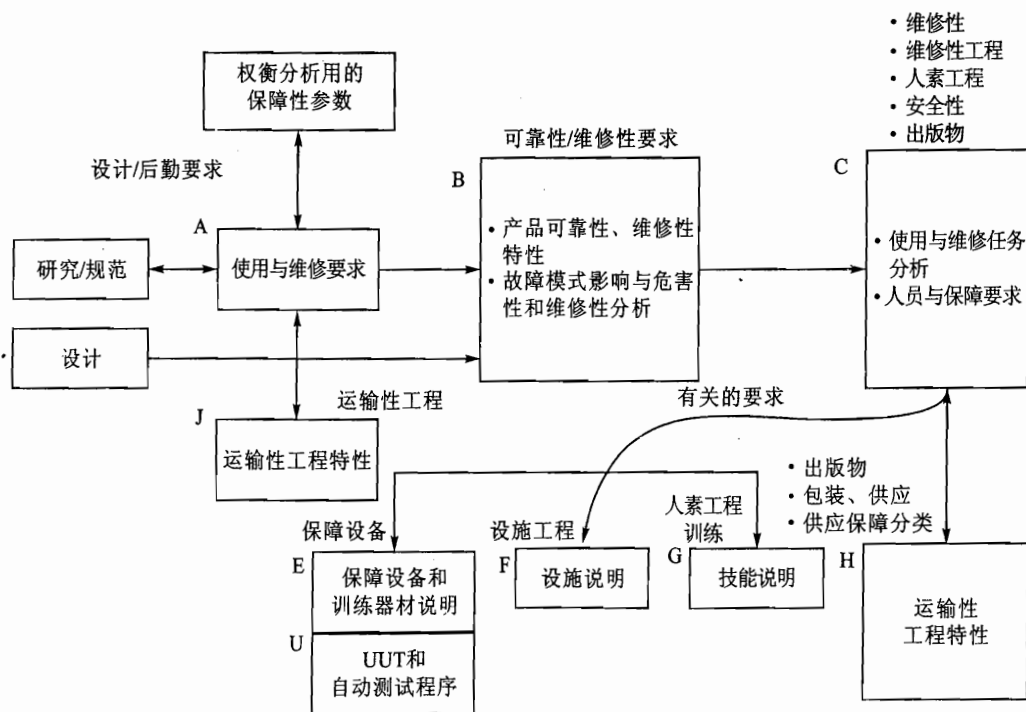


图 4.9 保障性分析数据流和系统工程的接口

4.4 保障资源研制

综合保障工作的最终输出是提供飞行器系统保障所需的保障资源,并将这些资源有机地综合成保障系统。通过保障性分析,飞行器系统所需的保障资源要求在保障计划(包括维修保

障计划)中已有明确的结论。现在需要将保障资源要求转化为直接可利用的资源,这就是根据保障资源要求进行研制、采购或利用用户已有的资源。

飞行器系统使用保障与维修保障所需的资源是不同的。使用保障资源主要有充、填、加、挂、储、运所需的保障设备及测试设备,使用操作人员及其训练与训练设施以及使用所需的技术资料等。维修保障资源主要有维修保障设备、备件、维修人员及其训练、维修设施及维修技术资料。这两方面的资源在大多数情况下是不能通用的,但研制的基本方法是相似的。现仅就备件和器材供应、保障设备这两方面的资源加以探讨。

4.4.1 备件和器材供应

备件和器材供应包括使用 and 维修所需更换的备件和消耗品,主要确定备件需要量和库存量。在提供用户使用的层次上,有初始供应和正常供应两类。

1. 确定备件需要量

使用与维修工作中所需的备件量很大,耗费资金很多。维修所需备件的品种、数量主要是根据飞行器系统硬件设计、保障性分析、有关试验所作的消耗统计等资料而定。通常利用的数据有平均故障间隔时间、更换率、备件修复可利用率和废品率等。

初始备件供应的理论需求量应根据实际情况加以修正,如使用强度和环境、使用和维修人员技术水平、零部件质量以及管理水平等。战时还应研究战伤损坏率等。

当数据不足时,可利用类似飞行器系统的类似零部件估计,也可以从用户年实际消耗分析而得。

所需备件及器材应有专门的各类供应清单,说明备件名称、规格、种类、数量及特殊要求等。这是保障工作的一项重要文件。

2. 确定备件库存量

备件库存是指在飞行器使用与维修过程中,为保证其工作正常进行,备件已经取得,而尚未正式投入使用,存储在仓库的过程。存储的数量即库存量。存储过程包括:

- (1) 从现存备件中提交使用,库存量下降;
- (2) 从生产厂或备件源订货补充库存;
- (3) 备件周转保障一定的库存量。

这个过程理论上十分简单,但它受诸多因素影响,如装备使用与维修备件需求的波动;备件供货的时间间隔;备件生产周期;仓储环境、地理位置和运输条件以及备件储存寿命和备件价格等。因此,合理库存便成为极其复杂的问题。

备件储存分为平时周转储存和战备储存。平时储存可利用各种模型辅以必要的修正系数或经验系数加以计算,并在使用过程中加以完善。战备储存则根据任务统一规划分级储存,其中包括战略、战役和战术储存。

平时周转储存可利用各种库存模型确定其库存量,目标是满足使用费用最低,即既不积压

资金,又要保证需求。模型中要根据供需情况作必要的假设和简化。

常用的库存模型有:

(1) 按备件供货时间的库存模型。这种模型可分为边进货边消耗,瞬时进货和允许暂时缺货等几种。

(2) 随机型库存模型。这种模型将备件需求量以不同概率分布(如二项分布、正态分布或泊松分布)来描述,以确定库存量。

(3) 供应期库存模型。这种模型是根据备件生产周期、供货周期及订货发货的制约和运输限制而确定出的平均日需要量与供应期天数作为基础来制定库存量的。

制定库存量还应考虑安全储备要求,通常根据不同零部件的实际情况增加一个系数。另外,对各级维修机构应储备某种零件的天数要作分别的规定,如基层级储备天数应少于中继级,而基地级的储备天数最长,安全储备可以用附加天数计算。

3. 供应保障工作的内容

供应保障的备件和消耗品供应包括初始供应和后续供应,有时也包括使用试验和评价时的供应(按合同和规定执行)。供应保障工作的内容主要有:

(1) 确定各维修级别所需备件和消耗品的清单,如零件供应清单、散装品供应清单及修复件供应清单等。

(2) 拟定新研飞行器系统及其保障设备、搬运设备、训练器材所需备件的订购要求,包括检验、生产管理、质量保证措施及交付要求。

(3) 使用和维修备件有关的仓库储存管理,包括备件的采购、验收、分发、储运及剩余物资处理等。

(4) 进行数据收集、分析并及时修订备件需求,以便调整库存,改进供应,减少浪费。

(5) 拟定飞行器系统停产后的备件供应计划。

图 4.10 为备件供应研制过程示意图,图中虚线部分为保障性分析工作。

4.4.2 保障设备

1. 确定保障设备

详细的确定保障设备是保障性分析中使用与维修工作分析的结果,而在研制早期就应对保障设备做出初步规划。在方案阶段应利用基准比较系统对新研制的飞行器系统所需的主要保障设备,特别是专用昂贵设备做出预测。有时保障设备还会成为初始保障方案的一项主要内容,昂贵的保障设备会成为费用主导因素,对备选保障方案的评价有重要作用。

在维修方案明确后,从每一维修级别的维修任务可以确定保障设备的具体要求,并据此评定各维修级别维修能力与维修设备是否配套。通常,基层级所需保障设备显然应少于中继级,否则需要重新分配维修任务。在做费用权衡时,如果价格十分昂贵的保障设备要在基层级配备,就要慎重,甚至要考虑修改维修方案,直至修改飞行器系统设计。

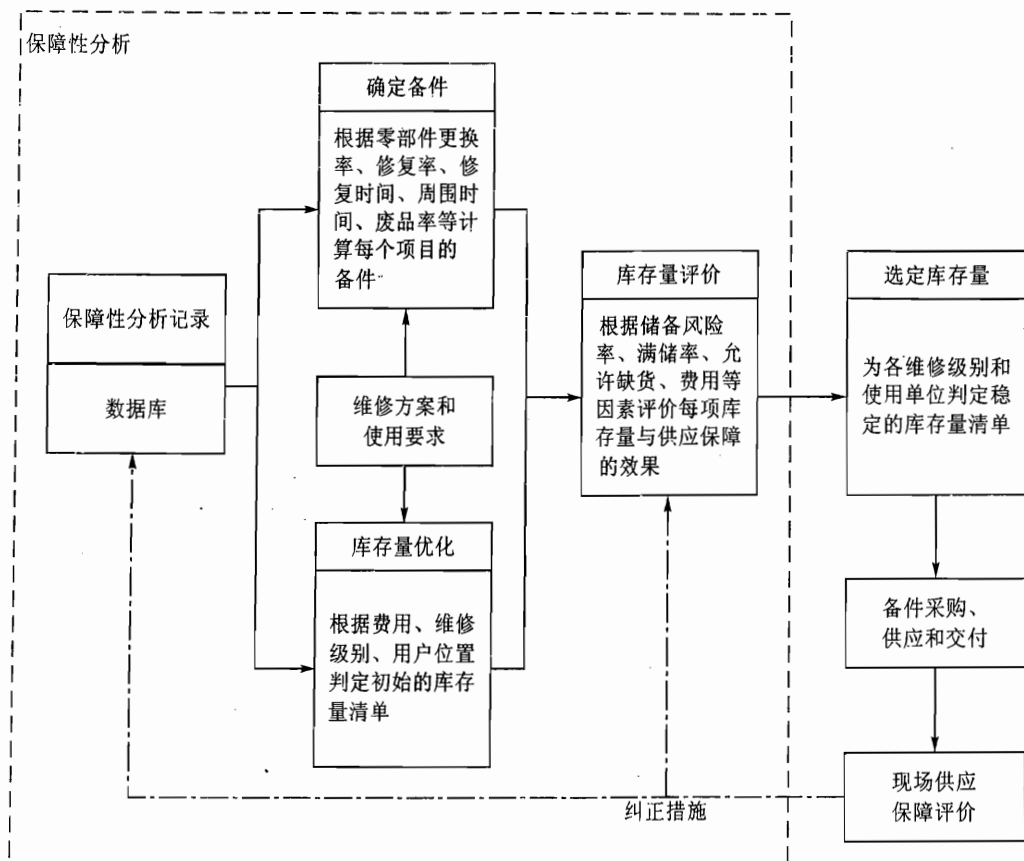


图 4.10 备件供应研制过程示意图

图 4.11 是保障设备的研制过程,其中相当大的一部分工作量是在保障性分析后做的,因此需要及早安排,以便保障设备能与飞行器同步提供给用户。

2. 研制保障设备的主要工作

在研制(包括购买)保障设备前,要进行完整的规划,并做好费用和进度的安排,其主要工作有:

(1) 拟定保障设备的范围与主要内容,如随机(车)工具、中修工具、自动测试设备、起重运输设备及加工工艺装置等。从保障计划中可以获得所需内容,也可由类似飞行器的实际保障设备中预计。

(2) 拟定初始保障设备清单,其中包括标准的和专用的。这份清单还可作为使用维修工作分析时选用保障设备的有价值的资料。

(3) 通过保障设备的综合权衡形成选定的保障设备清单,其中要从维修级别的任务、保障设备利用率、保障设备本身的保障要求及费用等方面做出分析与评价。

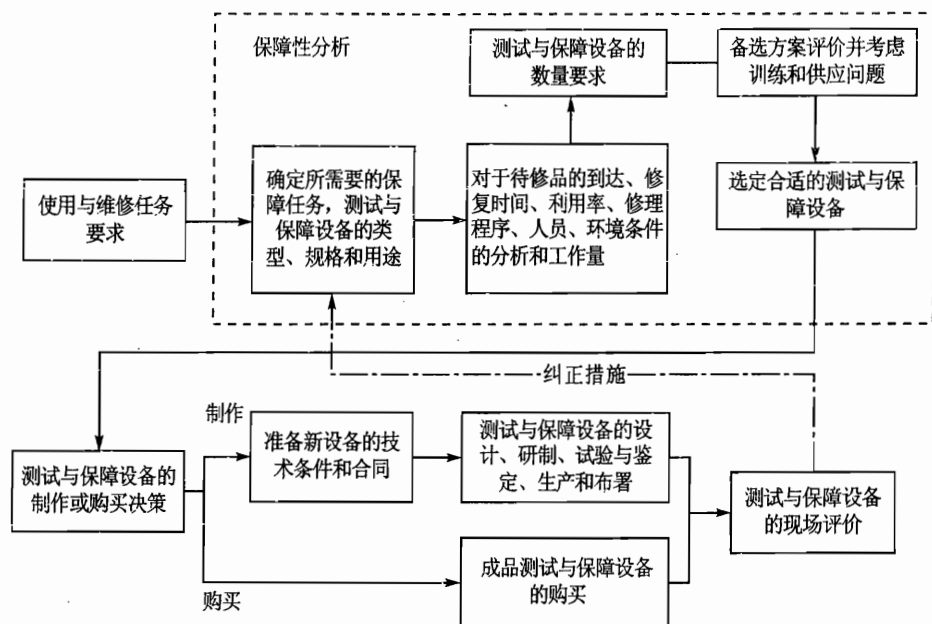


图 4.11 保障设备研制过程

- (4) 确定保障设备制作或购买,拟定制作的要求或对承担此项任务的转承制方提出要求。
- (5) 编制保障设备的技术手册,其中应说明设备的工作原理、流程图、使用维修方法、测试技术条件以及保障要求等。
- (6) 拟定保障设备操作人员的培训要求并编制教材。
- (7) 提出保障设备的设施要求,如动力、空间、环境和专门的基础建设。
- (8) 保障设备与飞行器使用维修匹配性验证和现场评价。
- (9) 保障设备(硬件和软件)交付部队的计划。

4.3 综合保障试验与评定

飞行器系统的综合保障试验与评定是为了验证并考核所建立的保障系统在使用期间达到规定战备完好性目标的充分程度。通常保障系统必须与被保障的飞行器系统一起运行才能进行分析与评价,如果单独去考核综合保障各专业和保障系统的工作,往往不能完全反映保障的质量,也不能达到预期的效果。由于考核综合保障工作所建立的保障系统,同时也考核了飞行器的有关保障的设计特性,因此,从这个意义上说综合保障的试验与评定也可以称为保障性的评估。

实施保障性评估要达到以下目的:

- 为验证战备完好性提供实测的数据;

- 暴露综合保障工作中存在的问题,以便及时纠正;
- 验证满足合同要求的程度,包括使用和保障资源与飞行器的匹配程度;
- 考核所建立保障系统在使用环境中的效能。

根据执行的时机,可将保障性评估作如图 4.12 所示的分类。

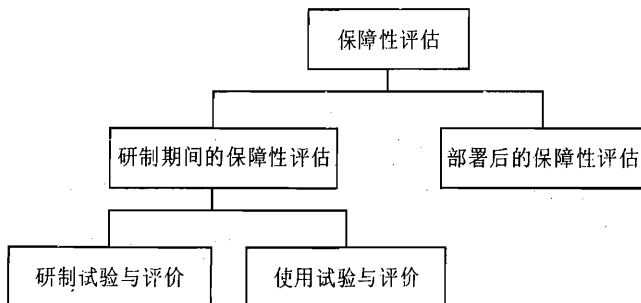


图 4.12 保障性评估分类

1. 研制期间的保障性评估

飞行器在研制期间要进行研制试验与评价(DT & E)和使用试验与评价(OT & E)。这两种试验与评价都是型号试验与评价总计划(TEMP)的组成部分,有明确的要求,综合保障工作要参与其中。

2. 部署后的保障性评估

部署后的保障性评估是在飞行器正常使用后一定时期对其实际保障能力的评定。综合保障专业负责规划、实施这项评估。

按综合保障评价的工作内容,也可将其分为以下 4 个方面的评估工作:

(1) 设计接口评价:对承制方的综合保障有关设计与纠正措施反馈的接口的评价。此评价从方案阶段到工程研制阶段进行连续评估,用以保证维修保障计划与硬件特性相互兼容,并通过保障设备功能试验评价这些设备与飞行器使用检查程序的兼容性。

(2) 硬件检查:用以检查飞行器硬件的可接受程度及其满足保障性要求的能力,如检查所选择的维修工作和维修保障计划,验证故障隔离及故障件拆卸和更换的要求等的满足程度,并进行必要的设计更改。

(3) 统计数据的核实:这是在飞行器初始部署期间为验证其保障特性而核实所统计的综合保障有关数据,以评定统计的有效程度,作为更新保障资源规划和预算工作的基础,并核实合同与定量的保障系统要求之间的符合程度。

(4) 部署后整套保障系统的评估:主要评估所规划的保障资源,用于保障所部署飞行器的适用性和充分程度。

第 5 章 飞行器研制中的软件工程

5.1 概 述

在当代采用高新技术的飞行器研制中,系统工程与软件工程是密不可分的。这不仅是因为现代飞行器中装备了大量的软件,而且由于在飞行器研制中纯粹的软件开发和开发组织是少见的。软件多是与系统一起开发,并集成到系统中。因此迫切需要将软件工程,乃至软件采办与系统工程集成在一起。

5.1.1 软件在现代飞行器中的重要作用

现代高新技术在飞行器研制中的应用日趋广泛,其标志之一是数字化的设备大量应用于飞行器。因而计算机软件在飞行器中得到大量的应用,不论在规模上,还是在重要性上均呈急剧上升的趋势。

表 5.1 展示了美国第二、三、四代歼击机上航电系统中由硬件和软件承担其功能的百分比。由表 5.1 可知,美国的歼击机每更新一代,其由软件实现的功能就翻一番。

表 5.1 软件在美国航电系统中的应用

第 X 代飞机	型 号	航电系统功能	
		硬件实现/(%)	软件实现/(%)
第二代	F-111	80	20
第三代	F-16	60	40
第四代	F-22	20	80

在国内,飞行器的状况也相类似,在我国新研制的军用飞机中,其飞控系统、火控系统、发动机燃油调节系统,甚至弹射救生系统等均采用了由软件实现其控制功能,逐步取代了原有的机械或光学设备。在某些机种上,机载软件已超过 100 万行源代码。

毫无疑问,软件在飞行器中起着越来越重要的作用。但不幸的是,软件的质量和可靠性却远不如人意。特别是与硬件可靠性相比,软件的可靠性一般要比硬件低一个数量级。下面是美国一个权威机构在 2000 年 3 月发布的关于美国当前软件开发质量的统计数据。

- 软件项目中途终止的占 25%;
- 软件产品在交付时通常在产品中还残留 15% 的缺陷;

➤ 软件公司花在软件返工(修改)上的资源为 30%~44%;

➤ 软件失效往往比硬件失效高一个数量级。

与美国的软件开发水平相比,我国的飞行器软件开发水平还相差很远,因而软件的质量和可靠性水平更令人担忧。软件故障已成为引发飞行器故障的主要原因。不少事实证明,软件故障是导致型号失败,甚至灾难性事故的重要因素。

5.1.2 软件的质量特性

软件的质量特性是一组描述和评价软件产品质量的属性。

据 ISO/IEC 9126—91,软件质量可定义为 6 个特性和 27 个子特性。

软件的 6 个质量特性是:

① 功能性(functionality): 当软件在指定条件下使用时,软件产品满足规定需求和隐含需求功能的属性。

② 可靠性(reliability): 在规定的条件下和规定的时间区间内,软件实现其规定功能的能力。

③ 易用性(usability): 在指定条件下使用时,与用户使用软件所需努力程度有关的属性。

④ 效率(efficiency): 在规定条件下,软件产品可提供的性能水平与其所用资源的数量相关的属性。

⑤ 可维护性(maintainability): 软件产品可被修改、维护的能力。

⑥ 可移植性(portability): 软件产品从一种环境转移到另一种环境的能力。

以上 6 个特性及其派生的 27 个子特性的关系如图 5.1 所示。

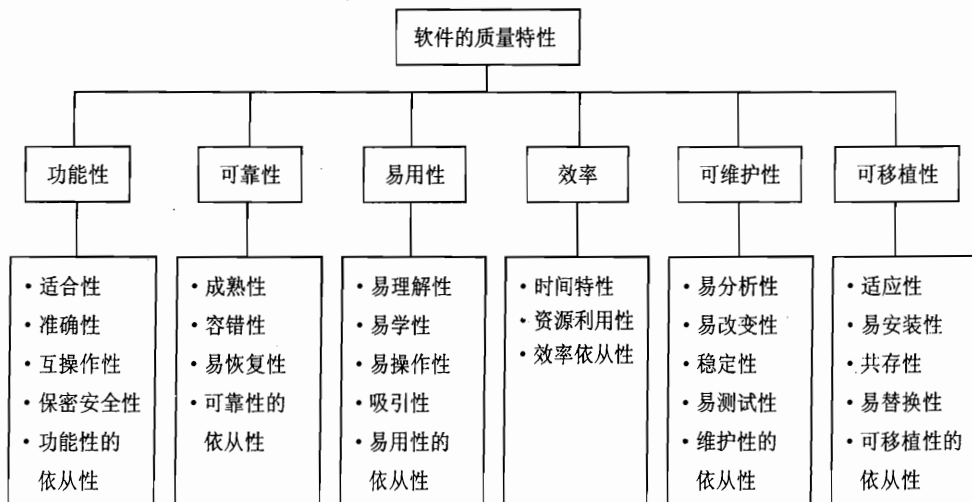


图 5.1 软件的质量特性及其子特性

据 John Masa 在“软件可靠性工程”中的论点和 IEEE 于 2004 年 AR&MS Tutorial Notes “Software Reliability and Maintainability”中的论述,均认为在软件质量属性中“可靠性是最重要的软件质量属性”。

5.1.3 软件的质量模型

对于软件的质量特性,必须予以量化与测量。因为对任何事物:如果不能测量它,也就不能控制它。对软件的上述质量特性建立量化的模型是为了更好地控制软件质量。

1. 功能性模型

软件的功能性是指软件产品的功能满足用户需求的程度。软件功能性模型 F 为:

$$F = \frac{\text{软件产品能提供的功能数 } n}{\text{用户在需求规格说明中要求的功能数 } N}$$

这里的“功能”是指广义的功能,包括性能(功能实现的准确程度)、互操作性、完整性等特性在内。

2. 可靠性模型

软件的可靠性是指与规定的时间区间内软件无失效运行的状况相关的特性。软件可靠性模型 R 一般可为:

$$R = \frac{\text{规定的软件运行时间区间 } T_R(h)}{\text{在规定时间内软件的失效数(个)}}$$

3. 易用性模型

软件的易用性是指与使用软件的努力程度(如培训、操作难易程度)相关的特性。软件易用性模型 U 为:

$$U = \frac{\text{为使用软件的学习和培训工作量(人时)}}{\text{软件开发工作量(人时)}}$$

U 越小,表示软件越容易使用。

4. 效率模型

软件的效率是指与计算机软件资源利用率(如存储空间、处理时间、通信时间等)相关的特性。软件效率模型 E 为:

$$E = \frac{\text{软件实际利用的资源}}{\text{分配给软件的可利用资源}}$$

E 太小,就表示实际利用的资源太小,存在资源的浪费;但 E 也不能过大,一般要求 $E \geq 0.8$,即对存储空间/处理时间等要求有至少 20% 的余量。

5. 可维护性模型

软件的可维护性是指当软件失效后,与软件故障定位、排除的难易程度等有关的特性。软件可维护性模型 M 为:

$$M = \frac{\text{需要维护与修正软件所有故障花费的总时间 } T_M(h)}{\text{需要维护与修正的软件故障总数(个)}}$$

M 越小,表示软件的可维护性越好。

6. 可移植性模型

软件的可移植性是指软件在另一个环境(硬件配置与/或软件系统环境)使用时,与其转换工作的努力相关的特性。软件可移植性模型 P 为:

$$P = \frac{\text{将软件移植到另一环境中所花的工作量(人时)}}{\text{软件开发工作量(人时)}}$$

P 越小,表示软件的可移植性越好。

5.1.4 软件质量的 Pareto 原理

Pareto 是意大利的经济学家,他在分析社会财富分配状况时得出了“20%的富人掌握了 80%的社会财富,其他 80%的人只占有 20%的社会财富”。这一原理随后被广泛地应用于其他领域,如系统工程中的“八·二定律”及硬件研制中对硬件系统关键件的分析等。

Pareto 原理在软件开发中也被证明是正确的。Barry Boehm 给出了软件质量和软件现象所遵循的 Pareto 原理:

- ① 20%的软件模块,包含了软件中 80%的缺陷;
- ② 20%的软件缺陷,消耗了 80%的更改维护费用;
- ③ 20%的软件改进,花费了 80%的适应性维护费用;
- ④ 20%的软件模块,占用了 80%的执行时间;
- ⑤ 20%的软件模块,消耗了 80%的资源(人力、经费等);
- ⑥ 20%的软件工具,占用了 80%的全部工具使用时间。

本书作者再补充两条重要的、软件质量的 Pareto 原理:

- ① 20%的软件缺陷,造成了 80%的软件故障;
- ② 20%的软件开发和管理人员(骨干),决定了 80%的软件开发质量。

根据 Pareto 原理,质量管理之父 Juran 提出了在质量管理中存在着“少数的重要事情(vital few)”和“多数的琐碎事情(trivial many)”的观念。这就给软件质量管理指明了方向:在有限的资源条件(有限的人员、时间、经费)下,抓软件质量要抓“少数的重要事情”,要抓 20%的关键模块,对这些关键模块的开发过程必须从文档、评审、配置管理、测试等方面严加控制,就可有效地提高软件质量。

5.1.5 软件质量与软件工程化

如何保证软件产品质量,一直是软件工程界十分关注、致力解决的问题。20 世纪 60 年代末发生的世界软件危机的主要表现之一就是软件质量差。当时,软件界共同探讨解决软件危

机的方法,得出的唯一结论就是吸取硬件的工程经验用于软件开发,即实施软件工程。30多年来,国际软件工程界一直在探索和发展软件工程的实施方法,如今软件工程已经成为一个专门学科;虽然“软件危机”问题尚未完全解决,但实践已经表明,实施软件工程的确是保证软件质量、解决软件危机的唯一有效方法。

实施软件工程,确保软件质量的核心就是用软件工程方法组织软件开发。

众所周知,产品质量主要取决于产品研制过程的质量。软件产品也是如此,飞行器软件质量主要取决于软件的开发过程。由于软件产品是“通过承载媒体表达的信息所组成的一种知识产物”,软件产品一旦形成,在没有人为改动的情况下,就有“千篇一律”和“一成不变”的特点,因此,软件产品的质量主要由软件开发过程决定。

用软件工程化方法组织软件开发包含两方面的内容:

- 用软件工程方法开发软件,即软件的工程化开发;
- 用软件工程方法管理软件开发,即软件的工程化管理。

要确保飞行器软件的质量,必须一方面重视软件的工程化开发,另一方面实施软件的工程化管理,两者不可或缺。这也是人们通常所说的软件的质量是设计(开发)出来的,也是管理出来的。

1. 软件的工程化开发

软件的工程化开发,就当前的飞行器软件而言,主要与软件开发的方法学有关。

当前的软件开发方法学主要有结构化、面向对象、净室和形式化等方法。

1) 结构化方法

结构化方法包括结构化的分析、结构化的设计、结构化的编程和结构化的测试。

结构化方法认为软件系统是以一定的结构形式存在的,由若干子系统构成。软件系统可以以一定的准则,自顶向下进行层次分解,直至分解到低层次的模块。

对飞行器中应用的大量嵌入式软件,现在大多仍采用结构化的开发方法。

2) 面向对象方法

面向对象方法是以对象为中心构造模型和组织软件系统。这种方法认为客观世界由对象组成,不同对象间的相互作用和联系构成了不同的系统。应用计算机解决问题的方法空间应当与客观世界的问题空间相一致。

面向对象方法中的对象是由数据及其上的操作组成的封装体。对象是类的实例,而类则是具有相同属性和服务的对象的集合。

3) 净室(clean room)方法

净室方法是在结构化分析和设计方法的基础上,增加了需求分析和设计的形式化方法。这种方法认为软件程序设计组应努力开发出在进入测试之前就几乎无错的系统。



4) 形式化方法

形式化方法是以严格的数学证明为基础的。这种方法要求软件需求规格说明要用形式化的语言描述,以保证其正确无误,然后经过一系列变换直到产生出可执行程序。

由于这种方法需要较多的数学验证,目前尚缺少可支持的工具,因此在飞行器的工程研制中尚未得到应用。

在飞行器的软件开发中,选择何种开发方法取决于软件项目的特点,能得到的支持环境和技术支持,以及开发人员的技术水平和经验等因素。

鉴于上述软件开发方法学在一般软件工程书籍中多有阐明,本书不再赘述。

2. 软件的工程化管理

软件的工程化管理极其重要。总结国内外飞行器软件开发的经验和教训,可以认为管理仍是决定软件质量的关键所在。

软件工程化管理是指对于一个软件工程项目为了确定和满足需求所必须进行的一系列组织、计划、协调和监督等工作。多年来,经过大量调查研究发现:管理仍然是开发软件项目成败的关键。

早在20世纪70年代中期,美国国防部就组织力量研究软件项目失败的原因,发现在失败的软件项目中,70%是由于管理不善所造成的,因而认为管理影响全局,并掀起了研究软件管理技术的热潮。20年后,根据美国3份经典研究报告,这一状况并未得到转变:软件开发仍然很难预测,大约只有10%的项目能够在预定的费用和进度下交付符合需求的软件;管理仍然是软件项目成败的主要因素;并指出开发过程中的返工是软件过程不成熟的标志。

软件工程管理具有以下几个特征:

- 没有适当的管理,软件开发不可能完成好,也就谈不上软件工程化;
- 软件工程项目越大、越复杂,管理工作量占整个软件研制工作量的比例也越大;
- 管理的基本目标是以最小代价达到对工程项目预定的要求,基本任务是保证恰当地确定指标要求和圆满地实现指标要求。

软件工程化管理的关键是:

- 对软件开发过程的全过程控制;
- 对软件质量的全方位管理;
- 建立多层次的软件开发、管理体系。

5.2 现代软件开发与管理的三维模型

本节提出的现代软件工程化管理的理念,是为软件开发与管理构建一个全方位、全过程、多层次的三维体系。这三维是:

- 时间维:对软件生存期的全过程进行控制。
- 空间维:对软件质量有关的关键因素实施全方位管理。



► 组织维：构建从软件开发个人、软件开发小组到软件开发单位的多层次的管理模式。

图 5.2 展示了现代软件工程化管理的三维框架。

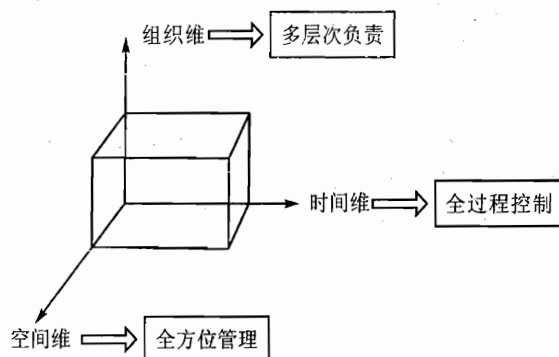


图 5.2 现代软件工程化管理的三维框架

5.2.1 软件生存期的全过程控制

对软件生存期的全过程控制是指软件开发应按软件生存周期分阶段制定并实施计划，并逐个阶段予以验证。对软件生存期的每个阶段均应明确其进入条件、主要工作、方法和工具、基本要求、主要质量控制手段、主要质量度量、阶段产品和完成标志。

表 5.2 概括了上述的基本内容。

表 5.2 软件生存期的全过程控制

研制阶段	本阶段的进入条件	主要工作	管理任务	主要质量控制手段	完成标志	阶段产品
系统分析与软件定义	型号总体已确定对计算机系统的功能分配	- 分析系统要求和 使用环境 - 论证软件项目可 行性 - 确定软件关键性 等级 - 拟订软件开发任 务书 - 明确软件验收方 法	- 分析风险 - 确定指标 - 拟订质量大纲 要求 - 确定资源保证 - 组织供需双方 协作	- 评审 - 规范	软件开发任 务书通过评 审,并形成功 能基线	- 系统和段 设计文件 - 软件开发 任务书

续表 5.2

研制阶段	本阶段的进入条件	主要工作	管理任务	主要质量控制手段	完成标志	阶段产品
软件需求分析	- 已有交办和承制双方签署的软件开发任务书 - 功能基线已建立 - 软件项目组已成立	- 确定运行环境 - 确定功能、性能和接口要求,编写软件需求规格说明 - 确定关键软件功能 - 制定软件开发计划 - 制定软件系统测试计划	- 制定质量大纲实施计划 - 组织验证与评审 - 进行配置管理 - 实施计划管理 - 组织供需双方协作	- 评审 - 规范 - 结构化分析及其工具(如数据流图 DFD、数据字典 DD 等) - 配置管理	软件需求规格说明通过正式评审,并形成分配基线	- 软件需求规格说明 - 软件开发计划(含质量保证计划和配置管理计划) - 软件系统测试计划
软件设计	概要设计	- 软件需求规格说明已通过评审,分配基线已建立 - 软件开发计划已批准	- 组织验证与评审 - 记录并报告问题 - 进行配置管理 - 实施计划管理	- 评审 - 规范 - 结构化设计及其工具(如结构图 SC、流程图等) - 配置管理	软件概要设计通过评审,并纳入受控库	- 概要设计说明 - 软件集成测试计划
	详细设计	软件概要设计已通过评审			软件详细设计通过评审,并纳入受控库	- 详细设计说明 - 软件单元测试计划

续表 5.2

研制阶段	本阶段的进入条件	主要工作	管理任务	主要质量控制手段	完成标志	阶段产品	
软件实现	详细设计已通过评审	- 编写程序 - 进行调试 - 进行静态分析和单元测试 - 编写软件使用说明 - 设计测试用例, 编写测试程序 - 编写单元测试报告	- 组织评审 - 记录并报告问题 - 进行配置管理 - 实施计划管理	- 评审 - 规范 - 结构化编程及其工具 (如编程语言、调试工具、测试工具等) - 配置管理	- 软件无错误地通过编译或汇编 - 完成软件静态、动态单元测试并通过评审 - 软件单元纳入受控库	- 源程序 - 测试用例和测试程序 - 软件单元测试报告 - 软件使用说明 (初稿)	
软件测试	软件集成测试	- 软件单元通过了单元测试 - 已具备集成测试环境和工具	- 执行软件集成测试计划 - 编写软件集成测试报告 - 完成软件使用说明的编写	- 测试管理 - 分析风险 - 组织评审、记录并报告问题 - 确定可否供系统测试	- 评审 - 规范 - 测试技术、工具 - 配置管理	- 通过软件集成测试评审 - 阶段产品纳入受控库	- 可运行的程序及数据 - 软件集成测试说明 - 软件集成测试报告 - 软件使用说明
	软件配置项测试	- 已完成了软件集成测试和评审 - 已具备 CSCI 测试环境	- 测试整个程序 - 试用软件使用说明 - 编写配置项测试报告	- 进行配置管理 - 实施计划管理		- 通过 CSCI 测试评审 - 阶段产品纳入受控库	- 配置项测试说明 - 配置项测试报告
	系统测试	- 已通过 CSCI 测试 - 已具备系统测试环境	- 按系统测试要求开展测试工作 - 编写系统测试报告	- 分析并报告问题 - 组织问题追踪 - 进行配置管理 - 组织转阶段评审	- 评审 - 规范 - 配置管理 - 系统测试技术与环境	- 通过系统测试正式评审 - 阶段产品纳入受控库	- 系统测试报告 - 软件问题报告单和修改报告单

续表 5.2

研制阶段	本阶段的进入条件	主要工作	管理任务	主要质量控制手段	完成标志	阶段产品
软件验收交付	- 达到任务书中规定的验收条件 - 完成任务书中规定的各类文档和测试	- 验收测试与审核(可利用已有测试与审核结果) - 验收评审 - 进行软件产品移交	- 组织验收 - 记录并报告问题 - 问题归零	- 评审和审计 - 验收测试技术、工具	- 软件产品纳入产品库 - 软件产品向交办方移交	- 产品移交文件 - 软件验收报告 - 软件验收测试报告
软件使用和维护	软件产品正式交付给使用方	- 使用 - 记录运行状态与问题 - 维护 - 回归测试	- 分析并报告问题 - 组织问题追踪 - 进行配置管理	- SFRACAS 管理 - 回归测试 - 评审		- 问题报告单 - 修改申请报告单 - 新版本全套文档

5.2.2 软件质量的全方位管理

软件质量的全方位管理是对影响软件质量的各个关键要素进行严格管理,使软件开发按照软件质量要求规范化地实施。对软件质量的全方位管理包括软件需求管理、软件分级管理、软件文档管理、软件配置管理、软件评审管理、软件测试管理、软件 FRACAS 和软件分承包单位管理。

软件质量全方位管理的示意图如图 5.3 所示。

5.2.3 构建多层次的软件开发管理模式

构建多层次的软件开发管理模式是将软件开发者个人、软件开发小组、软件开发单位/公司组成一个 3 层次的开发、管理体系,建立起从个人到小组到单位/公司的有机联系的层次递阶的管理模式。这里,我们不妨借鉴近年来由美国 SEI 提

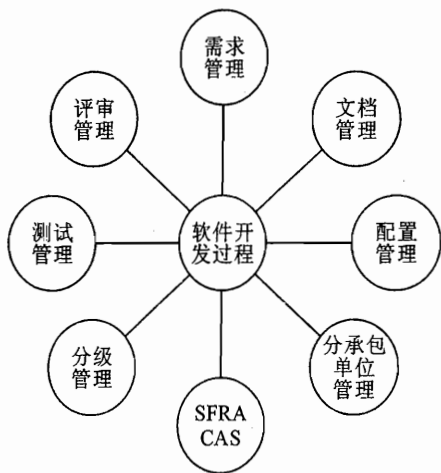


图 5.3 软件质量全方位管理的示意图

出的:

- 针对个人软件开发过程的 PSP(Personal Software Process);
- 针对开发小组软件开发过程的 TSP(Team Software Process);
- 针对单位/公司的软件能力成熟度 CMM(Capability Maturity Model for Software)。

PSP、TSP 和 CMM 都不仅涉及软件开发过程,还涉及对软件开发的管理。它们组成了一套先进的软件开发管理思想,不仅可以用来衡量软件开发单位、项目组和个人的能力,而且能有效地帮助他们改进和优化管理,从而提高其软件开发能力和产品质量。

我们提出的个人、项目组、单位三位一体的多层次软件开发、管理模式(参见图 5.4),是为了改变传统的软件开发管理模式。在传统的软件开发管理模式中,管理只是软件开发单位管理部门的事,不能调动和激发软件开发个人和小组参与管理的积极性。而在 PSP 中,不仅要求软件开发个人进行软件开发工作,还要求个人参与管理工作:要求每个软件开发个人自己进行项目管理和质量管理,并制定出管理计划和具体的、定量的质量要求。TSP 也类似(详见 5.5.2 小节)。

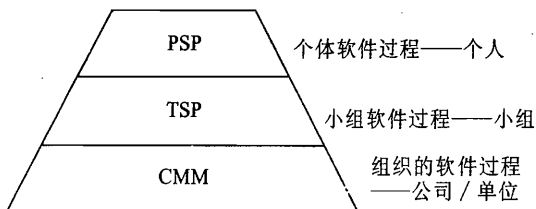


图 5.4 三位一体的多层次软件管理模式

构建这种三位一体的多层次软件管理模式,可以使软件开发组织从混乱、无序、不可控的软件开发状态下转变为有序、有纪律、可控的并不断改进的良性状态。

本章提出的现代软件开发和管理的三维模型,从时间维、空间维、组织维各方面确保了现代飞行器软件开发过程和软件产品的质量。下面将详细阐明全过程、全方位、多层次开发与管理的实施方法。

5.3 时间维——对软件生存期的全过程控制

5.3.1 过程的定义

1. 过程(process)

过程是指通过人员、设备、器材和规程(procedures)的交互作用,以期提供一个规定的服务,或生产一个规定的产品。过程示意图如图 5.5 所示。

每个过程是一组(或多组)被定义的应在某个给定时间内完成的活动(activities)或任务(tasks)。这些活动的特点是能够被可度量、可控的输入和输出所描述的。

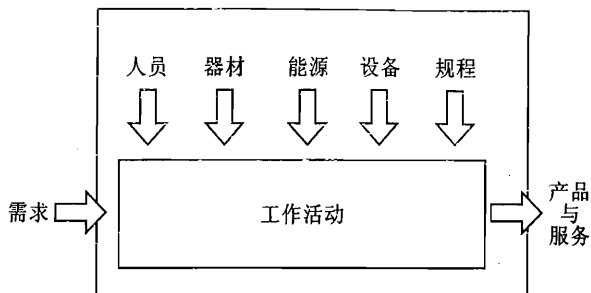


图 5.5 过程示意图

2. 软件开发过程

软件开发过程是指将用户要求/需求转化为软件产品的过程。

为了确保有效的质量控制和质量保证,管理者应当将开发过程分解为子过程,并满足以下的质量准则:

- ① 降低开发过程的复杂性,以确保可见性(visibility)和可控性;
- ② 确定并按时间顺序指定各控制点,在这些点上能验证过程的正确趋势和结果的正确交接;
- ③ 标识出对质量和项目成功至关重要的任务;
- ④ 在项目特定的组织结构中对所有质量功能提供清晰的标识和分配;
- ⑤ 使用已证实和正确推导及解析的质量度量;
- ⑥ 提供开发过程有效管理的质量数据。

5.3.2 软件过程管理的关键活动

软件过程管理由定义过程、度量过程、控制过程和改进过程 4 个关键活动组成。其示意图如图 5.6 所示。

1. 定义过程

与定义、实施和保持过程相关的关键目标是:

- ① 设计能满足或支持需求目标的过程;
- ② 标识和定义与过程性能有关的问题;
- ③ 提供支持软件活动所需的整套方法、人员和实践经验;
- ④ 确保软件组织具有实施和保持过程的能力(技术、培训、工具、设施和资金)。

2. 度量过程

度量是发现从可接受的性能偏离的基础,它也是辨识过程改进时机的基础。过程度量的关键目标是:

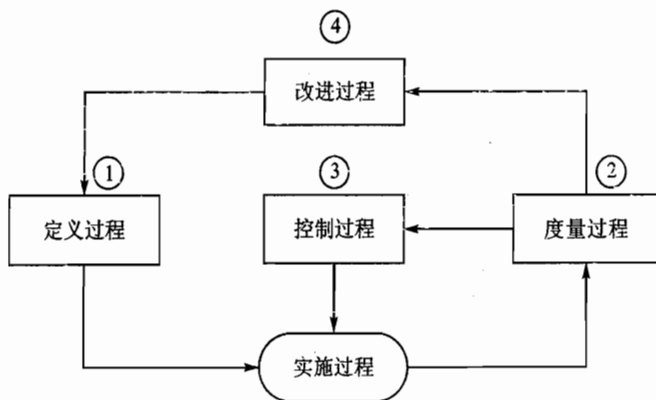


图 5.6 软件管理过程的 4 个关键活动

① 收集能对每个过程进行性能度量的数据。

② 分析每个过程的性能。

③ 保存并使用数据,以便:

- 评估过程的稳定性和能力;
- 解释观察到的结果并予以分析;
- 预计未来的费用和性能;
- 提供基线和标准;
- 绘制趋势图;
- 确认改进过程的时机。

3. 控制过程

控制过程就是将过程保持在正常的(固有的)性能边界之内,亦即使过程的运行始终如一,这涉及:

① 测量:获得关于过程性能的信息。

② 探测:分析得到的信息,以辨识出由于特定的原因引起的过程偏离。

③ 纠正措施:采取步骤来排除由于过程中特定的原因引起的过程偏离,以及消除使过程偏离预定目标的因素。

需要确立和保持以上 3 个关键活动,以控制软件过程:

- 确定过程是否受控(亦即测定的性能的偏差是否稳定);
- 辨识因特定原因引起过程变化而造成的性能偏离;
- 消除造成性能偏离的根源,以稳定过程。

4. 改进过程

① 了解现存过程的特征及影响过程能力的因素。

- ② 规划、实施和验证改进过程的活动,以使其更好地满足业务需要。
- ③ 评估改进后的效果和获得的效益,并将其与改进过程所花的费用相比较。

5.3.3 过程控制要点

- ① 软件开发过程应当是可度量、可控、可改进的。
- ② 软件开发过程应按时间顺序划分各子过程,对各子过程应确定各控制点,即提供每个子过程明确的开始与终止判据。
- ③ 清楚地描述过程中的各项活动、任务及其结果,特别应标识出对软件项目成功至关重要的任务。
- ④ 使用正确的、解析的质量参数,对软件开发过程提供有效的质量度量。
- ⑤ 在软件开发过程中,应将管理活动与工程活动结合起来。
- ⑥ 在软件开发过程中,必须认真进行软件工程的实践。
- ⑦ 标识软件开发过程中的风险领域,确定描述与跟踪各风险因素的方法和降低风险的途径。

5.3.4 软件生存期各阶段的过程控制

1. 系统分析与软件定义阶段

本阶段主要关心的是飞行器系统中与软件开发相关的部分。在飞行器系统将任务分配给了其中的计算机系统后,计算机系统首先对其上级系统对它的需求进行分析和确定,然后进行系统体系结构设计,将系统要求合理分配给计算机软件、硬件以及可能存在的人工操作,定义各软件开发项目,并将系统对软件的需求编写成软件开发任务书。

1) 进入条件

飞行器系统已经确定对计算机系统的功能分配。

2) 主要工作

- ① 系统分析;
- ② 系统设计;
- ③ 定义各软件开发项目;
- ④ 确定软件的关键级别和是否需要进行第三方独立测试;
- ⑤ 编制软件开发任务书(其中明确各种技术和管理要求,应特别注意规定验收方法);
- ⑥ 评审;
- ⑦ 签署软件开发任务书。

3) 方法和工具

进行系统分析和设计应采用系统工程的方法。在进行任务分解时,可采用结构化的思想,自顶向下,逐步细化。

在进行系统分析和设计时,可以采用一些图形工具来开展工作。通常使用以下图形工具:

- ① 系统流程图:表示系统的操作控制 and 数据流。
- ② 系统资源图:表示适合于一个问题或一组问题求解的数据单元和处理单元的配置。
- ③ 数据流程图:表示求解某一问题的数据通路。
- ④ 数据流图(DFD)。
- ⑤ 功能图:表示一个系统的模块层次关系,以及进一步描述模块信息的输入—处理—输出图(IPO)。
- ⑥ 结构图(SC)。

4) 基本要求

① 系统要求应满足:可追溯性;与飞行器系统所分配功能的一致性;可测试性;设计、操作和维护的可行性。

② 系统设计应满足:技术途径可行;软件、硬件、人工操作功能分配的权衡考虑恰当;可追溯性;与系统要求的一致性;设计和所用标准恰当;操作和维护的可行性。

③ 软件开发任务书应对软件的所有主要功能、性能技术指标进行定义,并与系统要求和系统设计一致。

④ 软件的接口说明完整、一致,满足可测试性。

⑤ 软件安全性分析和关键性分级正确。

⑥ 软件独立测试工作有安排。

⑦ 软件验收和交付要求全面、明确、可检查。

⑧ 进度和经费安排合理。

5) 阶段产品

与软件开发相关的阶段产品有:

① 系统要求规格说明;

② 系统设计说明;

③ 软件开发任务书。

软件开发任务书经评审通过后,纳入软件配置管理受控库,并以此形成软件配置管理的初始基线——功能基线。

6) 完成标志

① 完成了所有的阶段产品。

② 系统要求和系统设计通过了有关评审,并经批准生效。

③ 软件开发任务书通过评审,双方签署生效,软件功能基线建立。

2. 软件需求分析阶段

软件需求分析主要解决的问题是系统(或用户)要软件“做什么”。

软件需求分析阶段的任务是全面确定被研制软件的运行环境、功能、性能等全部需求;对

软件开发、质量保证、配置管理工作进行策划;编写软件需求规格说明和软件开发计划等重要文档。

软件需求规格说明是软件开发中最重要的技术文档,它是整个软件开发工作的基础。

软件开发计划是贯穿整个软件开发过程的综合性管理计划,是整个项目研制管理的依据。

1) 进入条件

① 已产生获得评审通过的、交办方和承制方签署的软件开发任务书。

② 软件开发项目组成立。

③ 以软件开发任务书为代表的配置管理功能基线已建立。

2) 主要工作

软件需求分析阶段的主要工作:首先是建立软件的逻辑模型,然后通过研究模型来获得软件的全部需求。

① 建立逻辑模型,描述待开发的软件将要“做什么”,可以自顶向下地把系统对软件各主要功能要求分解成功能层次体系。在一个完整、协调、一致的描述产生之前,模型的各部分可能需被多次说明。

② 定义软件需求,通过研究软件的逻辑模型,获得以下各类软件需求:

- 功能需求:规定软件应该“做什么”,也可以在其中包括性能属性。
 - 性能需求:规定可测量变量的数值(如频率、精度、速率),可以归并在各种功能的定量指标中。不允许用定性的说明来描述性能。性能属性可以作为一个值的范围提出,如:可接受的最坏情况、正常情况、最好情况值。
 - 接口需求:规定必须相互作用或通信的硬件、软件和数据库元素。接口需求应当划分为与软件、与硬件和通信接口。
 - 操作需求:规定系统将如何运行以及如何与操作人员通信,包括全部用户接口、人机接口需求。
 - 资源需求:规定诸如处理能力、主存储器、磁盘空间等物理资源的上限。
 - 验证需求:规定如何验证软件的一些限制。
 - 质量需求:规定保证适合于质量目的的软件属性。
 - 可靠性需求:规定在平均有效周期内可接受的软件平均失效前时间(MTTF),或可接受的软件失效最小间隔时间。
 - 维护性需求:规定如何易于修复缺陷和使软件适应于新的需求。应当用诸如修复缺陷的平均时间(MTTR)等定量的术语来说明。
 - 安全性需求:规定减少由于软件失效而导致的危险的需求。
- ③ 制定软件开发计划,包括:
- 软件规模估算。
 - 资源计划:根据任务和规模估算确定软件开发需要的人力资源,以及需要的硬件、软件

支持,包括软件开发环境、系统软件、计算机辅助软件工程(CASE)环境、测试环境等。

- ▶ 成本估算: 根据软件规模和所需资源估算软件开发成本。
- ▶ 进度安排: 剪裁和确定本软件开发过程的各个研制阶段。根据 40-20-40 规则分配工作量,即前期的分析和设计阶段占约 40%的工作量,中期的编程占约 20%工作量,后期的测试占约 40%的工作量。另外,还应确定各研制阶段的工作周期。

3) 方法和工具

(1) 采用结构化分析方法进行软件需求分析工作时,可采用数据流图来描述系统的逻辑模型,采用控制流图来描述系统的控制流,采用状态转换图来进行控制说明,采用数据字典来描述数据流图中的数据流,采用结构化的语言、决策表、决策树来描述数据流图中的处理。

(2) 采用面向对象分析方法时的功能分解和需求描述:

- ① 建立问题域的对象模型: 描述问题域类和对象,以及它们之间的关系。
- ② 建立问题域的动态模型: 描述对象模型中类和对象的动态行为。
- ③ 建立问题域的功能模型: 用一组数据流图描述问题域数据之间的依赖关系,以及有关的数据处理功能。

④ 描述软件配置项的外部接口: 对每个负责外部接口的类和对象进行描述。

⑤ 描述 CSCI 的内部接口: 以交互作用图标识各类和对象之间的接口。

4) 基本要求

软件需求分析阶段各项工作应满足以下要求:

(1) 软件开发计划

- ① 应满足软件任务书要求;
- ② 应对各研制阶段的工作设置明确的检查标志;
- ③ 各项工作及责任应明确落实到具体研制人员和管理人员。

(2) 环境约束

软件需求分析工作必须以软件的运行环境为基础。

(3) 质量和可靠性要求

软件需求应满足系统的质量和可靠性要求。

(4) 安全性分析

应进行软件的安全性分析,确定软件的危险状态及不允许发生的事件,确定软件的关键性功能。

(5) 软件需求规格说明的编制要求

必须用确定的方法正确而恰当地定义软件的功能、性能等所有软件需求。

(6) 软件需求规格说明的质量要求

- ① 无歧义性: 仅当对每一个需求只有一种解释时,软件需求规格说明才是无歧义的。
- ② 完整性: 软件需求规格说明应满足规定的各类需求。

③ 可验证性：仅当软件需求规格说明中每一个需求都是可以验证的，这个软件需求规格说明才是可验证的。

④ 一致性：仅当软件需求规格说明中各项需求的描述是不矛盾时，它才是一致的。

⑤ 可修改性：如果一个软件需求规格说明的结构和风格使得其在需求有必要改变时，修改是易于实现、完整、一致的，那么这个软件需求规格说明被称作是可修改的。

⑥ 可追溯性：当软件需求规格说明中每一个需求的来源是清晰的，在以后增补或修改软件需求规格说明的需求时可以容易地引证其来源，那么这个软件需求规格说明就是可追踪的。

(7) 评审要求

软件需求规格说明、软件开发计划、软件系统测试计划应进行评审。软件需求规格说明必须通过有任务交办方参加的评审。

5) 度量

① 需求规格说明书的页数/字数；

② 需求规格说明书的需求数；

③ 进入分配基线后的需求更改数；

④ 进行需求分析的人-月数；

⑤ 需求完整性度量：

$$R_I = 1 - \frac{\text{补充的需求数}}{\text{需求总数}}$$

⑥ 需求稳定性度量：

$$R_S = 1 - \frac{\text{更改的需求数}}{\text{需求总数}}$$

⑦ 需求可验证性度量：

$$R_V = \frac{\text{可验证的需求数}}{\text{需求总数}}$$

⑧ 需求测试覆盖率：

$$R_{TC} = \frac{\text{已测试的需求数}}{\text{需求总数}}$$

⑨ 需求测试成功率：

$$R_{TS} = \frac{\text{已通过测试的需求数}}{\text{需求总数}}$$

6) 阶段产品

① 软件开发计划；

② 软件需求规格说明；

③ 软件接口说明(必要时编制,当相关内容不很多时,可与软件需求规格说明合编)；

④ 数据要求说明(必要时编制,当相关内容不很多时,可与软件需求规格说明合编)。

软件需求规格说明及其配套文档经评审通过后,纳入软件配置管理的受控库,并以此形成软件配置管理的一条重要基线——分配基线。

7) 完成标志

- ① 完成了所有的阶段产品。
- ② 软件开发计划经批准后生效。
- ③ 软件需求规格说明通过评审。
- ④ 以软件需求规格说明为核心的配置管理分配基线已建立。

3. 软件设计阶段(概要设计和详细设计)

1) 软件概要设计阶段

本阶段任务是根据软件需求规格说明,建立软件总体结构与功能模块(软件部件)间的关系,定义各功能模块的接口、控制接口,设计全局数据库/数据结构,规定设计限制,编写软件概要设计说明。

(1) 进入条件

- ① 软件开发计划经批准生效。
- ② 软件需求规格说明正式获得评审通过。
- ③ 以软件需求规格说明为核心的配置管理分配基线已建立。

(2) 主要工作

① 建立物理模型

在从逻辑模型到物理模型的转换过程中,要建立设计准则,按设计准则将各功能分配到各软件部件(功能模块),并定义各软件部件的输入和输出。设计准则还应当满足一些非功能要求 and 设计质量准则,并被记载在有关文档中。

建模是一个迭代过程,直到每一部分都有一个清晰的描述为止。应使用计算机辅助软件工程(CASE)工具来辅助建立物理模型。

② 编制软件概要设计说明

使用文字和图示技术全面、具体地说明物理模型,形成软件概要设计文档。

a. 各软件部件功能定义:每一个软件部件的功能都可以上溯到软件需求规格说明,软件概要设计说明的详细程度仅限于指出每个软件部件应满足的功能需求,而不必涉及如何去满足这些需求。同样,各软件部件之间的接口关系也仅限于要交换哪些信息,不必说明如何交换这些信息(除非它对选择设计的成败有直接影响,从而作为后续设计的限制和约束来加以描述)。每个软件部件都必须确定输入数据(I)、要完成的功能(P)和输出数据(O)信息。输入和输出数据应当以数据结构加以定义。

b. 数据结构定义:定义与各软件部件接口相关的数据结构。外部接口可以考虑单独在接口控制文档中集中定义,作为软件概要设计说明的补充和附录。

c. 控制流定义:可以作顺序和并发运行及同步和异步特性描述。

d. 计算机资源利用: 开发环境和运行环境所需的计算机资源(例如 CPU 的速度、内存容量、系统软件等)必须进行估算,并在软件概要设计说明中确定。软件需求规格说明中的任何资源需求都对软件的设计具有约束。

③ 编程语言选择

一般可能在软件概要设计之前已指定使用的编程语言,在进行体系结构设计中应给予充分的考虑。

④ 可靠性设计

应根据系统的可靠性要求,按照可靠性设计准则开展设计工作。要特别注意对软件出错处理的设计工作,不仅要设计对硬件等外部系统出错时的处理,而且还要在假定最终软件产品必定还会有错的情况下,进行软件出错的处理设计。

⑤ 安全性分析和设计

在概要设计中应充分考虑系统对软件的安全性需求,进行安全性分析,确定安全关键软件部件,采取相应的安全性设计措施。

(3) 方法和工具

有许多方法可用于进行软件的体系结构设计,比较常用的是面向数据流的结构化设计方法。此外还有面向数据结构的方法、以信息隐藏为原则的方法等。

(4) 基本要求

① 设计完整: 覆盖软件需求规格说明中描述的所有软件需求。

② 接口清晰: 必须定义软件的主要部件和它们之间的接口,必须定义和引用所有外部接口。

③ 详细一致: 说明必须一致并足够详细,以便制定详细的实现计划。

④ 弱耦合: 总的原则是尽可能采用非直接耦合、数据耦合、特征耦合等弱耦合形式;减少控制耦合;必要时才使用外部耦合和公共耦合,但一定要最少,而且要特别注意绝对不能采用内容耦合。

⑤ 高内聚: 划分模块时应避免使用低内聚的模块,依据具体情况多用中高内聚模块,特别是功能内聚的模块,尽可能遵守“一个模块,一个功能”这一原则,它是衡量模块独立性的最高标准。

⑥ 作用范围在控制范围之内: 受某个判定影响的所有模块都应该在该判定所在模块的管理之下。

⑦ 控制规模和扇入、扇出数。

⑧ 追溯性: 软件概要设计说明中应有软件需求和软件结构中各软件部件的交叉对照表。

⑨ 资源和余量设计: 整个系统在存储量、输入和输出通道以及处理时间方面至少应留有20%的余量。

(5) 阶段产品

① 概要设计说明。

② 外部接口设计说明(本文档在必要时单独编制,用于描述软件的所有外部接口的数据结构,作为概要设计说明的补充和附录。在相关内容明确和不很多时,本文档可与概要设计说明合编)。

(6) 完成标志

① 完成了所有的阶段产品。

② 概要设计说明通过评审。

③ 概要设计说明和外部接口设计说明文档纳入配置管理受控库。

2) 软件详细设计阶段

应对所有的软件单元进行详细设计,设计包括算法和内部数据结构在内的内部细节,为编写源程序提供必要的说明;编写软件详细设计说明。

对于规模不大、层次较少的软件,可将软件详细设计阶段和概要设计阶段合并成统一的软件设计阶段,但也应分两步开展工作,先进行结构设计,再细化进行详细设计,只不过这两步之间不再需要经过阶段转移的批准和控制了。

(1) 进入条件

① 软件概要设计说明通过评审。

② 软件概要设计说明和外部接口设计说明已纳入配置管理受控库。

(2) 主要工作

① 逐级细化软件部件:从顶层开始,逐级将概要设计产生的软件体系结构中的每一个软件部件分解细化,将设计进行到低层,直至形成若干个软件单元(可编程模块)。

② 软件单元过程描述:对各个软件单元进行过程描述,其中包括确定模块内的算法和数据结构。

③ 确定接口信息:设计和确定所有软件单元之间的详细接口信息。

④ 编写软件详细设计说明。

⑤ 对软件详细设计说明进行评审。

(3) 方法和工具

为了在详细设计阶段把处理描述转换成为准确的、结构化的过程描述,需要使用描述过程的详细设计工具。详细设计工具可以分为以下几类:

① 图形工具:逻辑构造用具体的图形来表示。

② 列表工具:用一个表来表示过程的细节,这个表描述了输入、处理和输出信息。

③ 语言工具:用类语言来表示过程的细节,这种类语言很接近编程语言。

不管属于哪一类,每一种设计工具都应当能表现出控制的流程、处理功能、数据的组织以及实现的细节。

(4) 基本要求

软件的详细设计工作和软件详细设计说明文档应满足以下技术要求：

- ① 按结构化程序设计原则进行设计；
- ② 采用规定的工具来描述软件单元；
- ③ 符合可靠性设计准则；
- ④ 详细地规定各单元间的接口；
- ⑤ 对每个单元确定所有的输入、处理、输出；
- ⑥ 对符号命名确定统一的规则并按规定使用；
- ⑦ 只使用基本控制结构来描述各软件单元的过程；
- ⑧ 详细设计与软件需求可追溯；
- ⑨ 详细设计与概要设计一致；
- ⑩ 在各部件和单元的需求之间内部一致；
- ⑪ 单元测试的要求和方法正确。

(5) 设计阶段的度量

- ① 软件设计文档的页数/字数；
- ② 设计阶段的人-月数；
- ③ CSC 的数量；
- ④ CSU 的数量；
- ⑤ 设计基线确立后，设计文档更改数；
- ⑥ 在设计阶段发现的需求错误数。

(6) 阶段产品

软件详细设计说明(采用某种详细设计工具对软件单元进行的描述应作为软件详细设计说明的一部分纳入文档)。

(7) 完成标志

- ① 完成了软件详细设计说明。
- ② 软件详细设计说明通过评审。
- ③ 软件详细设计说明纳入配置管理受控库。

4. 软件实现(编码与单元测试)阶段

软件实现阶段要进行编码和单元测试两项工作。

首先，采用规定的编程语言，对所有软件单元进行代码编制，并对编制的代码进行编译、调试，直至编译无错误通过。

之后，对完成编码的模块进行单元测试。其任务是检验所有软件单元的设计和实现。内容包括测试模块的接口、局部数据结构、重要的执行路径、出错处理、影响前面各点的边界条件等 5 个基本特性。

1) 进入条件

- ① 软件详细设计说明已通过评审。
- ② 软件详细设计说明已纳入配置管理受控库。

2) 主要工作

- ① 编码：对每个软件单元用指定的程序设计语言进行编码，包括对源代码的注释。
- ② 编译调试：保证编码完成的软件单元无错误地通过编译。
- ③ 静态分析：采用静态分析工具对源代码进行静态分析，找出其中的错误和可疑之处。记录所发现的程序错误，并对源代码进行调试和修改。

④ 代码审查：对编译通过的软件单元源代码，在进行动态测试之前，根据代码审查单，对源代码逐条阅读和审查，以发现错误和可疑之处。记录所发现的程序错误，并对源代码进行修改。审查人员通常应包括该单元开发者之外的软件开发人员。

⑤ 单元动态测试准备：补充和完善软件单元测试用例，准备必要的软件单元动态测试工具，设计和编制单元测试所需的桩模块和驱动模块，准备必要的测试数据。

⑥ 单元动态测试实施：根据计划进行测试，并与预期结果进行比较和分析，以发现错误和可疑之处。应有记录地对源代码进行调试和修改，并针对修改进行回归测试和必要的补充测试。

- ⑦ 评审：对编制的源代码和进行的单元测试工作进行审查。

3) 方法和工具

- (1) 采用结构化编程方法。
- (2) 进行单元测试，包括对软件单元进行静态分析和代码审查以及统计逻辑覆盖率。
- (3) 采用静态分析技术。静态分析是一种对代码的机械性的、程式化的特性分析方法。静态分析一般常用软件工具进行，包括控制流分析、数据流分析、接口分析和表达式分析。

① 控制流分析：使用控制流程图系统地检查程序的控制结构的工作。控制流按照结构化程序规则和程序结构的基本要求进行程序结构检查。

② 数据流分析：用控制流程图来分析数据处理的异常现象(数据异常)，这些异常包括初始化、赋值或引用数据的序列的异常。

③ 接口分析：在接口分析中应进行接口一致性的设计分析，它涉及模块之间接口的一致性以及模块与外部数据库之间的一致性。程序的接口分析涉及子程序以及函数之间的接口一致性，包括检查形参与实参类型、个数、维数、顺序的一致性。当子程序之间的数据或控制传递使用公共变量块或全局变量时，也应检查它们的一致性。

④ 表达式分析：某些错误的出现与程序中表达式的计算相关，如果对表达式进行分析，就可以避免这些错误。

表达式中的错误常常由于不正确地使用括号造成。如果程序设计标准要求全部表达式都必须用括号括起来，则很多这样的错误就可以避免。



另一类错误出现在数组引用时,它包括数组下标越界。

对包含有除式的代数式,必须检查其除数是否为零。

(4) 代码审查:由代码审查员根据软件详细设计说明,按照规定的代码审查单,对被审查的程序代码逐条进行审查,以确保编码与设计的一致并确保编码的正确性。应着重审查错误代码而不是多余代码或代码的遗漏。审议结果不用来评价个人的工作能力。

代码审查是发现软件错误的有效手段。其检查内容包括:

- ① 检查代码和设计的一致性;
- ② 检查代码是否遵循编码标准及可读性;
- ③ 检查代码的逻辑表达的正确性;
- ④ 检查代码结构的合理性。

为提高软件代码审查效率并保证审查质量,对会议时间和参加人数作如下规定:

- 每次审查会的参加人数至少3人,最多不超过7人;
 - 每次审查会议的时间至少45 min,最长不超过2 h;
 - 每小时阅读的代码不超过150~200代码行。
- ⑤ 单元测试:对软件单元进行静态和动态测试,并统计逻辑覆盖率。

4) 基本要求

(1) 模块结构:应采用结构化编码方法进行代码编制。

(2) 单元测试用例应满足以下要求:

- ① 用指定值、异常值和极限值验证全部计算;
- ② 验证全部输入数据的各种选择;
- ③ 验证全部输出数据的各种选择和格式;
- ④ 每个单元的全部可执行语句至少执行一次;
- ⑤ 在每个分支点进行选择的测试。

(3) 测试覆盖率:采用白盒法进行测试时,应考虑测试用例对程序内部逻辑的覆盖程度。

对单元测试,语句覆盖和分支覆盖是最基本的要求。此外,由于程序中错误(异常)处理工作的重要性以及其结构相对简单,要求错误处理要做到路径覆盖。对质量要求高的软件单元,可根据情况提出条件覆盖、分支/条件覆盖、条件组合覆盖以及其他更高的覆盖要求。

5) 实现阶段的度量

- ① 可执行代码行数;
- ② 通过代码审查发现的软件缺陷数;
- ③ 每个模块 LOC 的规模;
- ④ 通过代码审查发现的需求错误和设计错误;
- ⑤ 在实现阶段中需更改的需求数和设计缺陷数;
- ⑥ 注释行占代码总行数的百分比。



6) 阶段产品

① 软件源代码。

② 所有的测试用例、测试数据设计和详细的测试结果记录。另外,单元静态分析结果和代码审查结果也应一并保留。

③ 软件单元测试辅助程序、用于进行单元测试的驱动程序以及桩模块等辅助程序,属于内部产品,也应仔细保存,以便于评审和回归测试。

7) 完成标志

① 软件单元无错误地通过编译或汇编。

② 完成代码静态分析和代码审查。

③ 完成单元的动态测试并达到测试要求。

④ 通过软件实现阶段评审。

⑤ 所有软件单元纳入软件开发单位的配置管理受控库。

5. 软件测试阶段(软件集成测试、系统测试)

1) 软件集成测试

得到经过单元测试的程序后,下一步就要依据概要设计确定的软件结构,按照软件集成测试说明制订的软件集成“工序”,把软件单元逐步组装成完整的软件配置项,并不断发现和排除错误,以保证链接、组装的正确性。在软件最后集成为配置项后,应按软件需求规格说明的要求对软件进行配置项测试,亦即确认测试。

(1) 进入条件

① 被集成的软件单元无错误地通过编译或汇编。

② 被集成的软件单元通过单元测试达到了测试要求。

③ 被集成的软件单元已置于软件开发单位的软件配置管理之下。

④ 已具备了在软件开发计划和软件部件集成测试说明中要求的软件集成测试环境和测试工具。

(2) 主要工作

① 集成和测试:通过一个逐步进行的过程,在每一个集成测试的步骤中都要进行以下工作:

➤ 验证软件单元可在一起执行。

➤ 验证软件的控制路径,包括通过人-机界面和外部、内部控制的正确性。

➤ 测试软件内部数据的完整性,这些测试可能要求测试工具监督数据值以及由于数据值的变化而引起的系统变化。

➤ 验证软件部件的外部接口、功能、性能、精确度、错误识别和恢复。当新的软件部件加到已有的软件部件中时,还要验证旧的软件部件仍保持原有的特性和功能。

② 编写软件集成测试报告:对软件集成测试工作结果进行综合分析和评价,形成软件集成测试报告。



③ 评审：审查软件组装过程的恰当性和软件集成测试的充分性，评价测试结果的有效性和软件结构与接口间的协调性，评价对所发现的问题进行软件设计修改和程序代码修改的正确性。

(3) 方法和工具

① 软件集成测试策略

软件集成测试策略有两种：非增量方式和增量方式。

- 非增量方式：先测试好每一个软件单元，然后一次组装在一起再测试整个程序。这种方式快捷，占用机器时间少，有利于并行开发，但容易引起混乱。
- 增量方式：逐步把下一个要组装的软件单元或部件，同已测好的软件部件结合起来测试。

② 测试用例设计

a. 基于需求的测试用例

根据软件需求说明设计合理的测试用例。测试用例应包括正常范围测试用例和异常范围测试用例。异常范围测试用例又称为鲁棒测试用例。

正常范围测试用例的目的是用来证明软件响应正常输入和条件的能力。正常范围测试用例包括：

- 应该用有效等价类和边界值方法来测试实型和整型输入变量。
- 对与时间有关的功能，例如滤波器、积分器和延时器，应该用代码的多次迭代来检查该功能在上下文环境中的特性。
- 对于状态转换，应开发能在正常运行中遍历各种可能的转换的测试用例。
- 对于用逻辑方程表达的软件需求，正常范围测试用例应能验证变量的使用和布尔算符。

异常范围测试用例的目的是要验证软件响应异常输入和条件的能力。异常范围测试用例包括：

- 应该用无效值的等价类来测试实型和整型量；
- 应在异常状态下测试系统的初始化；
- 应确定输入数据的可能的失效模式，特别是来自外部系统的复杂数字数据串；
- 对于循环计数是一个计算值的循环，应开发企图计算超出范围的循环计数值的测试用例，然后演示与循环有关程序的鲁棒性；
- 应检查用于超出帧时间的保护机制是否能正确响应；
- 对与时间有关的功能，例如滤波器、积分器和延时器，应开发算术溢出保护机制的测试用例；
- 对于状态转换，应开发能引发软件需求不允许的转换的测试用例。

b. 基于结构的测试用例

根据项目要求的不同结构覆盖准则，设计相应的测试用例：

- 关键软件要达到语句覆盖、判定覆盖和判定/条件覆盖；
- 重要软件要达到语句覆盖和判定覆盖；
- 一般软件要达到语句覆盖。

(4) 基本要求

① 软件集成测试的要求

- 必须对有调用关系的软件单元之间的所有调用进行测试,验证每个调用接口的完整性和一致性;
- 应对软件经受错误影响的能力进行测试;
- 应测试在各种外部输入下,从外部接口采集和(或)发送数据的能力,包括对正确数据及状态的处理,对接口错误、数据错误、协议错误的识别及处理。

② 软件集成测试的通过准则

- 软件单元无错误地连接;
- 满足各项功能、性能要求;
- 对错误的输入有正确处理的能力;
- 对测试中的异常有合理解释;
- 人机界面、对外接口正确无误。

(5) 测试阶段的度量

- ① 每种测试发现的软件缺陷数;
- ② 每种测试期间发现的软件需求错误和设计错误数量;
- ③ 每种测试期间需修改的需求说明和设计文档数量;
- ④ 白盒测试中的结构覆盖率;
- ⑤ 黑盒测试中的需求覆盖率;
- ⑥ 测试成功率:

$$T_s = \frac{\text{通过测试的用例数}}{\text{总测试用例数}}$$

- ⑦ 因修正缺陷而更改的源程序语句行数(或机器指令字数)/源程序总行数(总字数)(%)。

(6) 阶段产品

- ① 软件集成测试说明。
- ② 软件集成测试报告,含所有的软件集成测试结果。
- ③ 所有的软件问题报告单及软件修改报告单。
- ④ 与软件修改报告单相一致的、经过修改的全部软件源代码。

(7) 完成标志

- ① 完成了所有的阶段产品。
- ② 进行了在软件集成测试说明中计划的所有测试工作,并达到了通过准则。
- ③ 所有软件问题按要求进行了处理。
- ④ 经过回归测试的软件程序重新纳入配置管理受控库,并通过了配置审核。
- ⑤ 软件集成测试阶段评审通过。

⑥ 软件集成测试说明、软件集成测试报告、软件问题报告单、软件修改报告单纳入配置管理受控库。

2) 系统测试

飞行器系统中包含的计算机系统(含软件和硬件)往往作为大系统中的一个分系统、子系统或设备。在这种情况下,被研制的软件是否能与系统中其他的软、硬件正确地对接,是否正确地完成了上级系统分配给它执行的功能、性能,必须通过系统测试来给予确认。

软件系统测试的目的,是在真实的系统工作环境下或仿真测试环境中检验软件是否能与系统正确连接,并确认软件是否与软件开发任务书中提出的软件功能、性能和接口要求一致。

对实时嵌入式软件而言,系统测试是最重要的测试,因为前述的所有测试均不能测试出实时软件中潜藏的时序错误和软、硬件接口错误。为此,进行系统测试的软件必须在目标机上运行,并采用真实的 I/O 接口。

(1) 进入条件

- ① 软件通过了集成测试/配置项测试。
- ② 所有软件的文档和源代码纳入配置管理受控库。

(2) 主要工作

① 建立真实的系统测试环境或仿真测试环境:所谓真实的系统工作环境实际上就是由尽可能采用的真实系统加上一些必要的仿真器组成。但软件的运行硬环境和接口,除特殊情况外都应该采用真实的部件。

② 设计软件测试用例:根据系统测试的要求详细设计出每个测试的测试用例。

③ 执行系统测试:执行系统测试的测试用例,按操作规程操作,详细记录试验情况。

④ 分析:比较和分析每个测试用例的预期结果与实际测得结果,判断测试是否通过。若发现问题,应及时填写软件问题报告单。

⑤ 编制系统测试报告。

⑥ 评审:对系统测试工作、试验结果进行评审。

⑦ 软件修改和回归测试。

(3) 方法和工具

应根据情况选择进行下列软件系统测试:

① 安装性测试。

② 功能测试(含边界测试)。

③ 性能测试:应注意对数据处理精度、时间控制精度、时间测量精度、响应时间、控制信号波形等性能指标和要求进行测试。

④ 强度测试(含时间强度测试和性能强度测试)。

⑤ 降级能力强度测试:对在计算机部分硬件失效(或瞬间失效)时各种可能的降级方式进行测试,并检查其自恢复能力。

⑥ 外部接口测试。

⑦ 安全性测试：应注意进行硬件和软件在各种故障模式下的测试；最坏配置情况下的测试；错误操作情况下的测试；多机系统出现故障切换时，系统的功能、性能连续平稳性测试。

(4) 基本要求

① 测试的重点放在实时系统的时序是否正确，软件与相关系统的接口上不允许有不协调之处；

② 注意所有输入/输出信息的边界测试；

③ 要将软件及其所属系统组合在一起进行强度测试；

④ 应对系统进行闭环的、非侵入性的实时测试；

⑤ 所有软件问题报告单和软件修改报告单纳入配置管理受控库；

⑥ 所有发现的问题都得到处理。

(5) 阶段产品

① 系统测试说明(含软件测试用例汇总)。

② 系统测试报告(含所有的系统测试结果)。

③ 所有的软件问题报告单和软件修改报告单。

④ 系统测试通过的软件源代码。

(6) 完成标志

① 完成了所有的阶段产品。

② 所有已发现的异常影响均被排除；或异常的影响虽未消除，但已弄清带着异常运行的风险，并经认可。

③ 通过了系统测试评审。

④ 本阶段所有的阶段产品均在配置管理的控制之下。

6. 软件验收与交付工作

软件产品在研制完成后，要进行软件的验收和交付。

1) 软件验收前提

提交验收的软件项目必须具备以下条件：

① 软件通过配置项测试(对嵌入式软件，必须通过系统测试)；

② 完成任务书(合同，下同)中规定的各类文档；

③ 软件产品已置于配置管理之下；

④ 达到任务书中规定的验收条件。

2) 软件验收依据

进行软件验收的依据是软件开发任务书、任务书引用或附录的有关技术文件、任务书或型号系统规定执行的标准和规范。

3) 软件验收过程

软件验收交付必须按下列步骤进行:

- ① 提出软件验收申请;
- ② 制定软件验收计划;
- ③ 成立软件验收委员会;
- ④ 进行软件验收测试;
- ⑤ 进行软件验收评审;
- ⑥ 形成软件验收报告;
- ⑦ 移交软件产品。

软件验收测试和配置审核是软件验收评审前应完成的两项重点工作。软件验收测试和配置审核工作,在验收委员会的主持下,由验收测试组、配置审计组具体完成。

4) 软件验收评审

在完成软件验收测试和配置审核的基础上,验收委员会对被验收软件进行验收评审。

7. 软件使用与维护阶段

软件维护是在软件产品交付之后,为纠正故障及改进性能和其他属性,或使产品适应改变了的环境所进行的修改活动。

软件的维护与硬件的维修不同,对软件的维护就意味着修改,不存在如硬件那样更换备件的维修工作。

1) 维护工作内容

- ① 纠错性维护:包括纠正设计错误、程序错误、数据错误和文档错误。
- ② 适应性维护:包括适应影响系统的规则或规律的变化;硬件配置的变化,如机型、终端、外部设备的改变等;数据格式或文件结构的改变;以及软件支持环境的改变,如操作系统、编译器或实用程序的变化等。
- ③ 完善性维护:包括扩充和增强功能,如扩充解题范围和算法优化等;改善性能,如提高运行速度、节省存储空间等;以及为便于维护,如为了改进易读性而增加一些注释等。

2) 软件维护过程

软件维护过程可按下列步骤实施:

- ① 收集软件维护信息;
- ② 确定软件维护类型;
- ③ 软件维护的申请与审批;
- ④ 软件维护的计划与实施;
- ⑤ 软件维护评审与验收。

3) 软件维护记录

每次软件维护完成后,必须做好软件维护记录。软件维护记录的内容包括:

- ① 被维护软件的名称；
- ② 软件维护类型；
- ③ 软件维护开始和结束的日期；
- ④ 软件修改前后的源语句总数；
- ⑤ 修改的源语句总数；
- ⑥ 增加的源语句总数；
- ⑦ 删除的源语句总数；
- ⑧ 软件修改花费的总人时数和总机时数；
- ⑨ 软件维护前的运行次数；
- ⑩ 软件维护前发生故障的执行次数。

5.4 空间维——软件关键质量因素的全方位管理

对软件质量的全方位管理是把与软件质量有关的各方面的工作均予以管理。这主要包括软件分级管理、软件需求管理、软件文档管理、软件配置管理、软件评审管理、软件测试管理、软件 FRACAS 和软件分承包商管理。

5.4.1 软件的分级管理

应根据软件失效后对系统安全性和功能的不同影响程度,将软件划分为若干个关键度级别。在软件研制过程中,应视软件的关键度级别采取不同的管理措施。

1. 软件分级方法

按软件失效后对系统安全性的影响,可将软件划分为 A、B、C、D 四个等级(见表 5.3)。

表 5.3 软件安全关键等级分级方法

级 别	定 义	危害事例说明
A	失效可能导致灾难性危害的软件	人员死亡、系统报废、环境灾难
B	失效可能导致严重危害的软件	人员严重受伤、严重职业病、严重系统损坏、严重环境危害、任务失败
C	失效可能导致轻度危害的软件	人员轻度受伤、轻度职业病、轻度系统损坏、轻度环境危害、影响完成任务
D	失效可能导致轻微危害的软件	使用中增加麻烦、不方便,但不影响完成任务

2. 软件分级管理

鉴于飞行器软件种类繁多,管理工作涉及的人力、物力资源消耗甚大,所以不得不分类,别一律对待。软件分级管理可依据软件的级别,在文档、评审、测试等方面有不同的管理要求。

5.4.2 软件文档管理

软件文档是软件产品的组成部分,编写软件文档是软件开发的基本任务之一。

软件文档起着如下重要作用:

- (1) 软件开发工作在一定阶段的成果和结束标志。
- (2) 向管理人员提供软件开发过程中的进展和情况,使软件开发过程的可见性增加,便于管理。
- (3) 记录开发过程中的技术信息,便于协调以后的软件开发、使用和修改。
- (4) 提供与软件运行、维护和培训有关的信息,便于管理人员、开发人员、操作人员和用户之间相互了解。
- (5) 为软件测试提供编制测试用例与预期结果的依据。

在软件开发项目策划期间,应根据具体任务要求和已确定的软件开发程序,确定具体的计算机软件配置项或软件系统应产生的文档种类。

软件文档种类和文档的产生时间如表 5.4 所列。

表 5.4 软件文档产生时间

文档名称	开发活动						
	系统 要求 分析 和 设计	软件 需求 分析	概要 设计	详细 设计	编码和 计算机 软件单 元测试	计算机 软件集 成和测 试	计算机 软件系 统测试
系统和段设计说明	——						
软件开发计划		——					
软件质量保证计划		——					
软件配置管理计划		——					
软件需求规格说明		——					
接口需求规格说明		——					
软件设计说明			——	——			
接口设计说明			——	——			
版本说明文档							——
系统测试计划		——	——	——			
系统测试说明			——	——	——	——	
系统测试报告					——	——	——
软件产品规格说明							——
计算机系统操作员手册							——
软件用户手册							——
软件程序员手册							——
固件保障手册							——
计算机资源综合保障文件							——

5.4.3 软件需求管理

需求管理是软件工程化管理的基础。其目的是在用户与软件项目组织之间达成对用户需求的一致理解。

需求管理的要点是：

- 分配给软件的系统需求是受控的,并建立了相应的配置管理基线——分配基线;
- 软件的需求与软件开发计划和活动应保持一致。

需求管理的关键是：

- 掌握与跟踪需求的变更情况;
- 掌握需求的状态和评审结果。

(1) 控制需求的变更情况

由于在进行需求分析时,需求分析人员与用户对需求的理解不一致,或用户通常会在软件开发过程中补充一些新的需求,因此需求变更是软件开发过程中常见的现象。对需求的变更必须严格加以控制,这主要体现在:

- 定义需求基线(即分配基线);
- 评审提出的需求变更,评估每项变更的可能影响,并决定是否批准变更;
- 以一种可控的方式将需求变更融入到项目中,并使当前的项目计划与需求一致;
- 使每项需求都能与其对应的设计、源代码和测试用例、测试结果联系起来,以实现跟踪;
- 在整个软件项目开发过程中跟踪需求状态及其变更情况。

对需求的变更,应进行以下的度量:

- (用户与开发项目组)提交的软件需求变更个数;
- 已批准变更的需求个数;
- 更改后已进入分配基线的需求个数。

还应统计需求变更在各阶段的分布,即:

- 需求分析阶段的需求变更次数;
- 设计阶段的需求变更次数;
- 实现阶段的需求变更次数;
- 测试阶段的需求变更次数;
- 外场试飞和使用阶段的需求变更次数。

(2) 跟踪需求的状态变化

来自于用户的需求在系统分析与软件定义阶段处于“已定义”状态,这是需求的初始状态。已定义的需求经过评审后认为合理、可行,并被写入“软件开发任务书”中,则该需求就转入了“已批准”状态;然后经过软件开发项目组的需求分析后转入“已分析”状态;并经过设计、实现、测试后逐一转入“已设计”、“已实现”、和“已验证”状态。对软件的状态转换过程必须进行跟

踪,这就是软件的可追溯性。这种可追溯性应当是双向的,既可以从“已定义”状态一直跟踪到已验证状态;也可以从已验证状态反向追溯到“已分析”、“已批准”状态。软件需求的状态转换过程如图 5.7 所述。

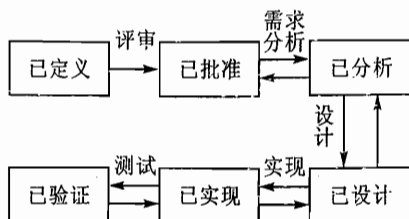


图 5.7 软件需求状态的转换

5.4.4 软件评审管理

软件评审的作用是：

- (1) 对各阶段的产品或过程进行审查和评估,以评价其是否符合规定的要求;
- (2) 发现存在的缺陷;
- (3) 采取补救措施;
- (4) 找出在性能、安全性和经济方面可能的改进。

在软件生存周期过程中一般需要进行多次评审,每次评审各有不同的具体要求。一般,在软件开发过程各阶段末设置评审点。

1. 系统分析和软件定义阶段评审

1) 评审目的

在系统分析和软件定义阶段,要形成系统对软件的需求说明或软件任务书,即建立功能基线。其目的是确保所提出的任务书能正确反映使用需要,并且所提要求是可以实现的,所确定的研制经费和周期是与要求相匹配的。

2) 评审对象

软件任务书或系统/段设计文档(以下简称任务书)。

3) 评审要求

① 审查任务书中软件需求的定义内容是否明确、无二义、完整、合理、可行;一时难以明确定义的需求是否说明了处理时限和负责人。

② 审查软件需求定义内容是否遵守任务书规范的要求。

③ 注意安全关键软件的可靠性和安全性要求,特别是不希望发生的事件,以及有关的验收方法是否明确。

④ 审查中发现的重要问题应得到解决,必要时应修改软件需求定义。

⑤ 评审通过的软件需求定义即软件承制方接受的软件需求定义,应作为软件开发的功能基线,纳入软件配置管理。

2. 软件需求分析阶段评审

1) 评审目的

① 确定软件开发计划和软件需求规格说明是否符合任务书要求;

② 确定该阶段文档能否作为概要设计的依据,能否转入概要设计阶段。

2) 评审对象

软件需求分析阶段的全部阶段产品。

3) 评审要求

(1) 审查软件需求规格说明

- ① 软件需求分析方法、工具使用是否合适。
- ② 软件需求内容是否符合系统的软件需求定义的要求。
- ③ 软件需求规格说明的质量要求是否满足系统定义的要求。
- ④ 是否明确了对软件的可靠性和安全性要求。
- ⑤ 是否明确了对软件的维护性要求。
- ⑥ 是否明确了对软件的数据保密性、完整性要求。

(2) 审查软件开发计划

- ① 软件开发阶段划分是否明确。
- ② 计划是否明确、合理、可行。
- ③ 软件开发计划中软件配置管理要求是否明确。
- ④ 软件开发计划中软件质量保证计划是否明确。

3. 软件概要设计阶段评审

1) 评审目的

- ① 确定软件结构设计是否实现了软件需求规格说明的要求；
- ② 评价软件设计方案与数学模型的可行性和先进性；
- ③ 评价接口设计方案和运行环境的适应性；
- ④ 审查数据库设计的完备性和一致性；
- ⑤ 确定该阶段文档能否作为详细设计的依据和能否转入详细设计阶段。

2) 评审对象

软件概要设计阶段的全部工作。被评审的文档包括软件概要设计说明、数据库设计说明和计算机软件集成测试计划。

3) 评审要求

审查软件概要设计说明：

- ① 审查软件设计方法是否合理。
- ② 审查软件设计工具的使用情况。
- ③ 审查软件的结构和模块划分,采用结构化设计方法时应审查:
 - 是否满足层次化结构设计要求；
 - 是否满足模块化结构设计的要求；
 - 是否满足模块间的调用要求；
 - 是否满足模块独立性要求；

- 是否满足模块扇入、扇出要求;
- 对系统运行过程的描述是否清楚;
- 审查软件设计是否满足正确性、完整性、一致性要求。

模块的功能和接口应满足正确性、完整性要求,在功能和性能上应完全覆盖已定义的软件需求,并与软件需求保持一致,且具有良好的可追溯性。

- ④ 审查软件的性能和资源使用要求是否明确、合理。

应确定进程及关键模块对存储量、处理时间、输入/输出通道、通用、专用外部设备环境等的要求。整体指标应满足软件需求规格说明的指标和余量要求。

- ⑤ 审查是否满足接口设计要求。
- ⑥ 审查控制流、数据流设计是否明确。
- ⑦ 审查是否满足可靠性、安全性设计要求,并进行了安全性分析。
- ⑧ 审查数学模型是否满足要求。

4. 软件详细设计阶段评审

1) 评审目的

- ① 确定软件部件(单元)设计是否实现了软件概要设计说明的要求;
- ② 评价软件单元设计描述(包括其中的算法和数据结构)的正确性;
- ③ 评价接口信息详细设计描述的正确性;
- ④ 若有数据库应用,则审查数据库设计的正确性和一致性;
- ⑤ 审查软件单元测试计划的合理性和完备性;
- ⑥ 确定该阶段文档能否作为软件编码的依据和能否转入软件实现阶段。

2) 评审对象

软件详细设计阶段的全部工作。被评审的文档包括软件详细设计说明、数据库设计说明和软件单元测试计划。

3) 评审要求

(1) 审查软件详细设计说明

- ① 审查设计方法是否合理。
- ② 审查设计工具的使用情况。
- ③ 审查功能设计是否符合概要设计的规定,且具有良好的可追溯性。
- ④ 审查软件的算法和数据结构。
- ⑤ 审查软件单元是否满足模块化结构设计的要求及其模块复杂性是否合理。
- ⑥ 审查模块描述是否满足如下要求:
 - 给出各模块的功能描述、接口描述和局部数据定义;
 - 描述应明确、清晰,无二义性;
 - 明确说明必要的精度和处理时间的考虑;

➤ 说明关于余量(存储、处理时间)的安排;

➤ 描述安全关键的特征。

⑦ 审查是否满足接口设计要求。

(2) 审查数据库设计说明

① 审查数据库逻辑设计。

② 审查数据库物理设计。

③ 对数据库系统中的软件单元要重点审查:

➤ 对软件单元中嵌入的过程命令描述是否满足要求;

➤ 是否逐个描述软件单元中使用的局部数据;

➤ 是否逐个描述软件单元中引用的数据元素及数据元素组合;

➤ 是否清楚地描述软件单元与数据库之间的接口信息(包括接口类型、数据特征、接口协议、通信特征等);

➤ 是否清楚地描述软件单元运行状态下的对外关系(与数据库和其他软件单元之间的控制关系);

➤ 是否清楚地描述软件单元运行的约束条件、限制,异常和出错处理;

➤ 数据库详细设计中是否满足概要设计中提出的质量要求。

5. 软件编码和单元测试阶段评审

软件编码和单元测试工作一般只进行内部评审,评审细则由软件项目负责人制定。对软件编码,通过静态分析进行仔细审查;对单元测试,按照规定的单元测试要求和计划评审单元测试工作。

6. 软件集成测试评审

1) 评审目的

审查软件集成测试过程和测试结果的有效性,看其是否达到了集成测试目的。

2) 评审对象

软件集成测试阶段的工作及测试记录。被评审的文档包括软件集成测试报告等。

3) 评审要求

(1) 审查文档是否齐全:

➤ 是否提交了软件集成测试计划、软件集成测试说明和软件集成测试报告;

➤ 是否有完备的软件集成测试记录。

(2) 审查测试活动的独立性:测试前是否成立了测试小组,是否能客观地进行测试。

(3) 审查测试方法的正确性:是否采用了增量测试法;集成是否通过自底向上、自顶向下或两者结合的方法进行的。

(4) 审查测试结果、测试说明、测试报告。

(5) 检查在以下项目的测试中是否满足了测试要求:

① 软件单元间的接口关系:

- 是否在每增加一个新的软件单元时,均验证了新加入单元与已存在单元间接口的正确性(主要验证单元的引用关系及数据关系);
- 是否对全部的调用对都进行了测试。

② 功能测试:

检查测试过程在每增加一个软件单元时是否都进行了不少于一次的测试,同时:

- 对每项功能的测试是否至少被一个测试用例或认可的异常所覆盖;
- 对每项功能都用正常值、异常值和边界值进行了测试;
- 对每项功能都用假设的数据类型及数据值作为输入进行了测试,并对软件排斥不规则输入的能力也进行了检查。

③ 性能测试:

- 对软件部件的性能是否都至少用一个测试用例进行了测试;
- 对有时间控制的软件部件是否都测试了软件的运行时间;
- 对有处理精度要求的软件部件是否都检查了其运算精度。

④ 各软件部件输入/输出数据处理的检查:

- 各软件部件输入/输出信息格式和内容的测试结果是否满足要求;
- 是否对各软件部件输入/输出的处理能力进行了检查。

⑤ 每个软件部件的错误处理能力的检查:

- 是否对每个软件部件的错误处理能力进行了检查;
- 对关键等级的软件是否按软件需求的要求作了容错处理能力的检查。

7. 系统测试评审

1) 评审目的

- ① 审查系统测试过程和测试结果的有效性,是否达到了系统测试的目的;
- ② 审查在真实应用环境下运行时,软件是否满足任务书和软件需求规格说明中的全部要求;

- ③ 审查在真实计算机环境下系统集成后运行的稳定性和安全、可靠性。

2) 评审对象

系统测试的工作及测试记录。被评审的文档包括系统测试计划、系统测试说明和系统测试报告等。

3) 评审要求

(1) 审查文档

① 测试文档是否齐全:

- 是否提交了系统测试计划、系统测试说明和系统测试报告文档;
- 是否有完备的系统测试记录。

② 文档是否符合规范。

(2) 审查测试环境

① 测试环境应是真实应用的软、硬件环境或对真实环境逼真的仿真环境；

② 检查测试环境的配置是否满足要求。

(3) 审查测试结果、测试说明、测试报告

① 审查测试项目的完备性：

- 按系统测试实施方法(或相应规定)中的要求逐项审查,看其是否进行了全部测试；
- 对所有功能测试、性能测试、强度测试、安全性测试、可靠性测试、余量测试均已取得测试结果；
- 所有接口及操作过程都得到了验证；
- 所有测试用例都已执行；
- 对未执行的测试用例,测试组应作出合理解释。

② 审查测试用例和测试过程的充分性：

- 测试用例和测试过程是否确切地规定了测试的输入、测试步骤、预期的输出和评价测试结果的评估方法；
- 系统测试是否至少有一次覆盖了任务的全过程。

③ 审查测试结果的合理性和正确性：比较在预定环境下测试结果与预期结果的符合程度,差别是否在合理的范围内。

(4) 检查项目测试

① 功能测试：

- 对每项功能是否至少用一个测试用例进行了测试；
- 是否用每个功能的合法边界值和非法边界值进行了测试；
- 是否用一系列真实值或仿真值对系统的功能进行了测试；
- 是否至少用一个被认可的异常(如飞行异常)对系统的功能进行了测试；
- 是否用任意的随机数据作为软件的非正常输入数据对软件的适应能力进行了检查。

② 性能测试：

- 是否对需求分析提出的各项性能指标都进行了测试；
- 是否可以说明实时数据处理精度能满足需求中的要求或在可接受的范围之内；
- 是否可以说明实时(即时间特性)性能满足要求；
- 是否在实时条件下,对系统处理最大信息量的能力进行了检查,结果是否满意；
- 是否对系统中各配置项之间的协调工作能力进行了测试。

③ 强度测试：

在预先商定的时间段内,程序在设计的饱和状态下运行时,考查软件的所有性能。因此,评审时要审查时间强度测试和性能强度测试。

时间强度测试：是指测试时间的长度是否满足对强度测试所作的时间规定。一方面要特别审查时间长度是否大于任务执行时间，另一方面要审查测试时间的连续性。

性能强度测试：

- 测试结果中是否包含处理的信息量在超过设计最大值时的测试数据；
 - 是否进行了数据传输能力的饱和试验(要求系统比设计能力传输更多的数据，以考查网络传输，内存的写入、读出，外部设备及软件处理等的超负荷能力)；
 - 是否进行了降级处理的强度测试；
 - 是否考察过在部分硬件发生故障时，系统在降级处理时的正确性和恢复正常运行的能力。
- ④ 安全性和可靠性测试：是否进行了下列各项测试，并达到了预期的结果。
- 用输入异常信息的方法测试软件系统的异常处理能力；
 - 用产生一系列强制性故障的方法检查软件系统的异常处理能力；
 - 制造条件对双工切换、多机替换的正确性和连续性进行测试；
 - 根据需求分析的要求产生一系列应急事件，考验系统的应急处理能力；
 - 考查关键部件的容错处理能力；
 - 检查系统防止非法进入的处理能力；
 - 根据软件需求(如果有)的要求，用制造系统崩溃的方法检查系统快速恢复的能力；
 - 检查出现误操作时系统的适应能力；
 - 必要且可能时，进行软件可靠性测试并评估软件可靠性。

(5) 对外部接口及人机界面的检查

① 系统输出信息的信息格式及内容是否达到了接口控制文件的要求；

② 从测试说明和测试结果的输出信息、提示信息全面检查人机界面的操作是否简便易行、安全可靠。

(6) 审查系统对发现错误后的处理

审查该软、硬件系统在真实环境或仿真环境下运行时，对发现错误后进行的处理是否满足要求。

5.4.5 软件配置管理

1. 软件配置管理的任务

软件配置管理的任务包括：指明配置管理项的功能特性和物理特性，并建立配置管理项的标识体制；控制对这些特性的更改；记录、报告更改处理以及执行状态；对配置进行审计和评审等。

2. 软件配置管理的工作内容

软件配置管理的工作内容主要包括：

- (1) 设立软件配置管理机构，明确机构职责与分工，明确该机构与其他组织的关系。

(2) 制订软件配置管理计划,以规划整个软件生存周期中软件配置管理的活动,对软件配置管理计划进行必要的维护。

(3) 确定并执行软件开发过程中要用到的、与软件配置管理有关的标准、规定或约定。

(4) 标识基线,对于每个基线,必须描述基线的项(包括应交付的文档和程序),与每个基线有关的评审、批准事项及验收标准,在建立基线的过程中用户和开发者的参与情况等。

(5) 制定并执行更改申请规程、文档更改规程,实施配置控制。

(6) 制定并执行配置状态记录和报告规程,实施配置状态的记录和报告。

(7) 实施配置的审计。

3. 软件配置管理计划

软件配置管理计划的目的在于对所开发的软件规定各种必要的配置管理条款,从而使所交付的软件能够满足项目委托书中规定的各种需求,以及经批准的软件需求规格说明书中规定的各项具体要求。

软件配置管理计划的作用在于使软件开发单位在开发项目所属的各子系统(其中包括为该项目研制或选用的各种支持软件)时,都应该执行配置管理计划中的有关规定。认真执行一个完善的软件配置管理计划,必定能提高软件的质量,最终满足用户的需求。

4. 软件配置管理活动

1) 配置标识

(1) 确立基线。在软件生存周期中,主要有 3 种基线,即功能基线、分配基线和产品基线,如图 5.8 所示。

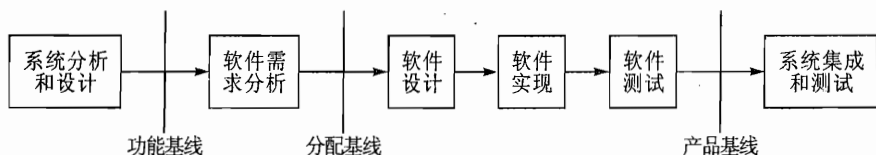


图 5.8 配置标识的 3 种基线

对于每个基线,应该指明:

- 应进入配置受控的每个配置管理项;
- 与每个基线有关的评审、批准事项及验收标准;
- 在建立基线的过程中用户和开发者的参与情况。

(2) 标识软件配置管理项。

2) 配置控制

(1) 入库控制

入库控制包括:

- ① 作为受控的软件配置管理项必须存入受控库;

② 只有满足要求的、作为产品交付给用户使用的软件配置管理项,才能从受控库转入产品库;

③ 各基线所产生的阶段产品,入库前均需经软件配置管理员审计,确认合格后方可入库。

(2) 更改控制

在软件生存周期中,对已进入受控库或产品库的任一软件配置管理项的任何更改都必须履行正规的审批手续。

3) 软件配置管理状态的记录与报告

软件配置管理状态报告的目的是提供开发过程的历史记录,因此在报告中应指明各软件配置管理项的现行状态,何时因何故发生了何事(入库、更改等)。

4) 软件配置审计

应对软件进行功能配置审计(以验证软件的功能和接口与软件需求规格说明的一致性)和物理配置审计(以检查程序与文档的一致性、文档与文档的一致性、交付的产品与任务书及标准规范的一致性)。

5.4.6 软件测试管理

国内外的众多事例说明,当代飞行器中的大量灾难性事故是由软件故障引起的。因此,如何减少软件缺陷已成为软件界面临的严峻挑战。软件测试是当前保证软件质量的关键手段之一,特别是软件独立测试已引起了国内外软件业的高度重视。

软件测试的重要性,是因为:

(1) 在软件开发的各阶段中,出现错误的机会相当多。例如软件需求的描述、软件的设计、软件的编码都可能出错;就连简单的程序输入也可能引入错误。随着软件的规模和复杂度的增加,使得错误发生率增高,软件错误造成的灾难也越来越严重。

(2) 软件测试费用在软件开发总成本中占有很大的比例。在国外,软件测试人员与开发人员比例为4:6,测试费用与开发费用的比例为1:1,而在安全关键软件的研制中,如飞行控制软件,测试费用可能相当于软件工程所有其他费用总和的3~5倍。

(3) 在目前软件工程开发方法学指导下开发软件,软件测试乃是保证软件质量和可靠性的重要手段。

(4) 软件测试能为软件质量的定量评定提供依据。

1. 软件测试的目的

软件测试的目的是检测和报告在软件开发过程中形成的错误。

2. 软件测试的正确认识

(1) 确定预期输出(或结果)是测试用例必不可少的一部分。

(2) 彻底检查每个测试结果。本来检查每个测试结果是软件测试的必须步骤,但却经常被忽视。事实上,在最终所发现的错误中,有相当大一部分在前面的测试情况中已显露出来,

由于人们未能细心检查先前的测试结果而遗漏。

(3) 对非法的和非预期的输入情况,也要像对合法的、预期的输入一样,编写测试用例。在测试时,有一种自然的倾向,就是用合法的、预期的输入设计测试用例。实际上,用非预期的、不合法的输入用例进行测试,查出错误的可能性更大。

(4) 检查程序是否做了要做的事仅是成功的一半,另一半是看程序是否做了不要它做的事。

(5) 在规划测试时,不要设想程序中不会查出错误。要明确软件测试的目的是发现错误,而不是证明软件的正确性。

(6) 一段程序中存在错误的概率与在这段程序中已发现的错误数成比例。

(7) 测试是一项非常复杂的、创造性的、需要高度智慧的挑战性任务。设计合理的测试用例,高效地发现软件错误,需要很大的创造性,因此软件测试是一项极具挑战性的工作。

3. 软件测试工作流程

软件测试是软件工程中的一个子过程,为使软件测试工作系统化、工程化,必须合理组织测试过程管理,包括编制测试计划、组织项目人员、建立测试环境、监控测试进展等。

软件测试工作流程是:

(1) 制定测试计划:包括测试活动的范围和测试的方法;制定测试要素(如覆盖程度);明确测试终止的要求和异常终止的情况;确定资源要求;制定总的进度安排。

(2) 制定测试说明:确定需要测试的特性;细化测试计划的有关内容(测试方法、所需资源、详细进度);设计测试用例集;制定测试规程并给出相关的说明。

(3) 执行测试计划:实现测试设计(获得测试数据),执行测试规程(运行测试,判定结果,对每次失效加以记录、分析和解决);核对终止情况(是否正常终止,必要时补充测试用例)。

(4) 评价测试效果和被测试软件:评价测试工作和被测软件;当发现测试工作不足时,应修订测试计划,重复步骤(1)~(3)的工作,直到测试完备时止。

(5) 产生完整的测试文档,包括测试计划、测试说明、测试报告。

1) 编制测试计划

软件测试计划是软件测试的总方针,是指导测试过程的决定性部分。测试计划应有以下内容:

(1) 目的:必须明确每一测试阶段的目的。

(2) 完成标准:必须给出判断每个测试阶段是否完成的标准。

(3) 时间安排:对于每个测试阶段,都要有一个时间表,指出设计、写出和执行测试用例的时间。

(4) 明确责任:对于每一个测试阶段,要具体指出谁来设计、写出、执行并验证测试用例,谁来修改出现的错误。

(5) 测试用例库的规范格式。

(6) 工具:必须确定所需要的测试工具,制订出计划,即谁来准备这些工具,这些工具怎

样用,什么时候用。

(7) 硬件配置: 如果需要特殊的硬件配置或设备,那么应描述出它们的要求,以及怎样才能符合要求和什么时候要用到这些硬件。

(8) 组装方式: 测试计划要确定程序怎样拼装在一起的(如按自顶向下的增式测试)。包含若干主要程序或主要分系统的系统可能是逐次地拼装在一起的。

(9) 记录手段: 必须明确一种方法,用来记录测试中取得的进展。这些进展包括找出易错模块,估计相对于时间表的测试进展,可供使用的资源和完成的标准。

(10) 纠错手段: 要确定某种方法,用以报告查出的错误,记录改正错误的进度以及对系统所做的修改。纠错计划还要包括时间安排、职责、所用工具及机器时间。

2) 组织项目人员

软件测试与软件开发一样,自动化程度很低,在测试过程中,人是决定性的因素,测试项目的管理在很大程度上是对人的管理。

测试人员应具有软件测试经验和软件专业知识,在测试各应用领域软件时,应具有相应背景知识。测试人员还应具有较强的分析问题和解决问题的能力。

软件测试项目人员分为: 项目负责人、测试人员、质量保证人员、配置管理人员。根据项目的规模大小,有可能一人身兼数职,但职责必须明确。

(1) 项目负责人的职责:

- 组织协调测试工作,确定项目人员,并对所承担项目的质量负责;
- 负责组织测试过程,包括制定项目计划,报告项目进度等;
- 负责与委托方负责人的沟通和交流。

(2) 软件测试人员的职责:

- 负责编制测试计划、测试用例、测试程序;
- 负责实施测试,记录并报告发现的被测软件的问题,同时对测试的质量负责;
- 按照软件测试计划规定的过程实施同行评审。

(3) 质量保证人员的职责: 负责监督所有项目的质量保证活动,确保这些活动是在受控状态下进行的。

3) 建立测试环境

按照项目软件开发计划和软件测试计划要求,配置相应的软硬件环境。良好的测试环境,包括已加载到目标计算机的并在目标计算机环境的高度逼真仿真环境中运行的软件。为了提高测试的可信度,应使用目标计算机仿真器或宿主计算机模拟器来完成测试。选择的测试应在综合的目标计算机环境中完成(因为某些错误仅在这种环境中才能检测到)。

图 5.9 展示了某嵌入式软件的系统测试环境,该测试环境的主要功能包括:

- 按照实际的使用(包括时序)生成测试用例并驱动目标系统运行;
- 实时监控,分析软件的外部测试运行结果;

- 自动收集测试结果；
- 收集失效数据,进行软件可靠性评估；
- 支持回归测试。

该系统由主控机、相关系统仿真器(简称仿真器)、激励信号生成器(简称激励器)3部分及网络连线等组成。

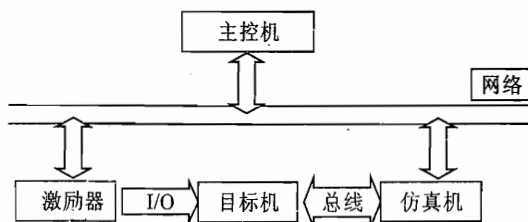


图 5.9 嵌入式软件的系统测试环境

4) 测试用例评审

测试用例应该经严格的评审,以证实其满足设计要求,主要应该检查:

- 每一项用例都应该包括目的、输入、条件、达到需求覆盖的期望结果及通过/失效标准；
- 基于需求覆盖分析显示每一软件需求都有相应的测试用例；
- 基于需求覆盖显示测试用例满足标准和鲁棒测试标准；
- 对实型和整型输入变量使用了有效等价类和边界值；
- 对多重迭代检查其在上下文环境中的功能特性；
- 对于状态转换,测试用例包含所有正常操作下可能的转换；
- 如果软件需求被表示为逻辑等式,那么正常情况下测试用例是否验证变量用法和逻辑操作符。

5) 监控测试进展

为了保证软件测试项目按预定的计划进行,对测试过程进展要实施控制。控制是以计划为基础的。由于软件产品的特殊性和软件工程的不成熟,制订软件计划比较困难,因此通常把一个大的任务分为若干小任务,再针对每个小任务分别制订计划。

项目控制包括进度控制、人员控制、经费控制和质量控制。

6) 测试结果审查

测试结束后,应保存运行结果,并对完成的测试程序和测试结果进行审查。一般,要进行覆盖分析。覆盖分析分两步进行,即需求覆盖分析和结构覆盖分析。第一步分析与软件需求有关的测试用例,以证实所选的测试用例满足指定的准则。第二步证实基于需求的测试程序测试了代码结构。

(1) 需求覆盖分析

这种分析的目标是确定基于需求的测试对软件需求的验证情况。这种分析可能要求增加

基于需求的测试用例。基于需求的测试覆盖分析应表明:

- 每一项需求都有测试用例;
- 测试用例满足正常和鲁棒测试准则。

(2) 结构覆盖分析

结构覆盖分析分两类:一类是单纯结构覆盖分析,是指项目不要求进行需求测试,这时的结构覆盖分析就是按照结构覆盖准则的要求,检查是否有未覆盖的部分;另一类是需求测试后的结构覆盖分析。

5.4.7 建立软件的失效报告、分析和纠正措施系统

失效报告、分析和纠正措施系统是可靠性工程中的一项重要技术。其基本思想也适用于软件产品,应用于软件时称之为“软件失效报告、分析和纠正措施系统(Software Failure Report Analysis and Corrective Action System, SFRACAS)”。SFRACAS 是一个闭环控制系统,要求将软件的失效加以记录、报告,找出失效原因,采取纠正措施。

1. 软件问题报告

软件问题报告概述在配置控制下的程序或文档中发现的各种问题,它是纠正措施过程的输入。在软件运行期间发现的问题一般应按下列类型分类:

- 软件问题:软件不按照文档运行,但文档是正确的。
- 文档问题:软件不按照文档运行,但软件运行是正确的。
- 设计问题:软件按照文档运行,但存在着设计缺陷。设计缺陷可能并不总是引起一个直接可见的运行失效,但具有产生深层次问题的可能。

2. 软件问题分析及纠正措施

软件问题的分析及纠正措施应遵守下述 5 条原则:

- (1) 对每个问题按类型和优先次序分类。
- (2) 对问题进行分析,找出其根源(根源分析),分析其影响和趋势。
- (3) 问题报告是分析和纠正措施过程的输入(使用方的追加要求也是一种问题报告)。
- (4) 分析和纠正错误的过程应闭环,以确保发现的问题均能及时报告并进入分析和纠正措施过程,解决问题,跟踪和报告状态。
- (5) 对纠正措施应予评价,核对所发现的问题是否已解决。

3. 软件失效报告、分析和纠正措施系统报告

从发现问题起,经过分析及采取纠正措施,到核对其实施效果止,应整理成 SFRACAS 报告。可采用软件配置管理中变更控制所用的软件问题报告单和软件修改报告单的格式,将它们和问题影响分析信息一起纳入软件配置管理。

5.4.8 对分承制单位的管理

软件承制单位必须对分承制单位的软件质量加以监控,其监控措施主要是:

- (1) 应与分承制单位协商后,确定分承制单位开发软件的级别,并检查其相应的管理措施。
- (2) 对分承制单位的软件需求规格说明予以正式评审,确认分承制单位的软件需求规格说明符合承制单位下达(或认可)的软件任务书的要求。
- (3) 对分承制单位软件的开发过程进行检查,特别对关键软件应参与(或主持)分承制单位的软件详细设计和软件测试报告的评审。
- (4) 对分承制单位交付的软件产品按国军标 GJB 1268A《军用软件验收要求》的要求进行验收。

5.5 三位一体的软件开发管理模式

1991年,美国卡耐基-梅隆大学软件工程研究所(SEI)推出了CMM,即软件能力成熟度模型。它是用来评价软件开发单位的软件能力成熟程度,能有效地帮助软件开发单位改进与优化管理。但CMM是针对软件开发单位的,而且CMM只告诉软件开发单位应当做什么,而不指出怎样去做。

众所周知,软件是人们思维逻辑的产物,人是软件开发的最基本要素,同时也是最主要的风险源。因为软件中的缺陷都是由于软件开发者的失误,在需求分析、设计、编程等软件开发过程中引入的,所以软件开发者自觉地进行管理对保证软件质量是至关重要的。

为此,SEI又在1995年提出了PSP,即个体软件过程。它对软件开发个人如何规划、控制和改善其软件开发过程指明了方向。只有每一个软件开发者自觉地关注软件过程的质量,软件质量才能得到保证。

对于一般的飞行器软件,由于其规模日趋巨大,只靠少数几个人来开发是不可能的,因而要组织软件开发项目组来承担开发任务。如何组建一个具有高度凝聚力的软件开发团队(team),如何确定团队中每个成员的角色,以及为这些角色确定管理目标及量化的质量评价准则等,都是必须回答的问题。SEI的W. S. Humphrey于1999年推出了TSP,即小组软件过程,就回答了上述诸问题。

如是,在这些20世纪90年代推出的过程控制方法的基础上,可以构建出三位一体的软件开发管理模式,即个人—项目组—软件开发单位的多层次软件开发、管理模式(参见图5.4)。这种开发、管理模式,能使软件开发组织从混乱、无序、不可控的软件开发状态转变为有序、有纪律、可控的并不断改进与优化的良性状态。

5.5.1 软件开发者的自我管理——个体软件过程

个体软件过程(Personal Software Process, PSP)是由美国SEI的W. S. Humphrey于

1995年提出的。PSP是一种可用于控制和改进个人软件开发工作的自我改善过程。使用PSP,可使软件开发人员减少软件缺陷(据统计,在开发过程中总缺陷减少58%,在测试阶段发现的缺陷减少71.9%;并增加生产率,使生产率提高20.8%)。在CMM1.1版中,CMM的18个关键过程域中有12个与PSP有关(参见图5.16中带*号的关键过程域)。

1. 个体软件过程的框架

PSP为软件工程师提供了发展个人技能的结构化框架。图5.10展示了共包括4个层次PSP的结构化框架。

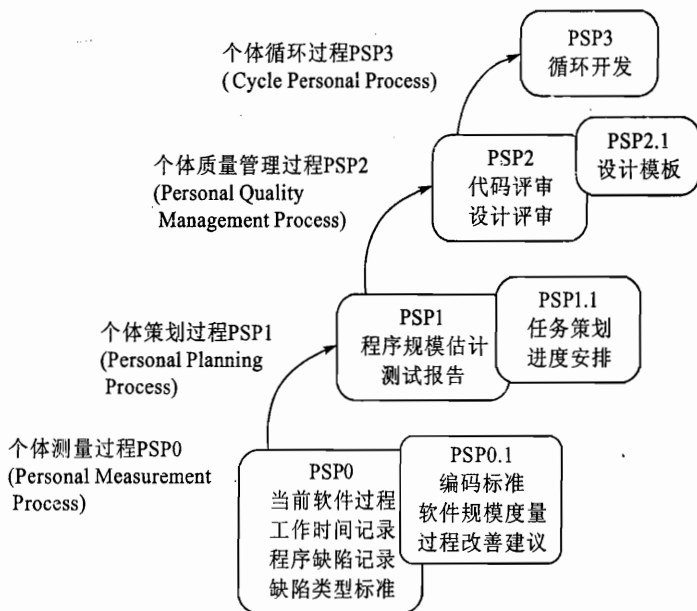


图 5.10 PSP 的框架^[5]

1) 个体度量过程 PSP0 和 PSP0.1

PSP0的目的是建立个体过程的基线,包括计划、开发和后置处理,并作一些必要的度量,如测定软件开发时间、度量引入的缺陷个数、度量排除缺陷的个数等,以作为测量PSP过程改进的基准。

PSP0.1增加了编码标准、程序规模度量和过程改善建议等3个关键过程域,其中过程改善建议表格用于随时记录过程中存在的问题、解决问题的措施以及改进过程的方法,以提高软件开发人员的质量意识和过程意识。

2) 个体规划过程 PSP1 和 PSP1.1

PSP1的重点是个体计划,引入了基于估计的计划方法,用自己的历史数据来预测新程序的大小和需要的开发时间,并使用线性回归方法计算参数,确定置信区间,以评价预测的可信程度。PSP1.1增加了对任务和进度的规划。

在 PSP1 阶段应该学会编制项目开发计划,这不仅对承担大型软件的开发十分重要,即使是开发小型软件也必不可少。因为,只有对自己的能力有客观的评价,才能作出更加准确的计划,完成客户委托的任务。

3) 个体质量管理过程 PSP2 和 PSP2.1

PSP2 的重点是个体质量管理,根据程序的缺陷状况建立检查表,按照检查表进行设计复查和代码审查,以便及早发现缺陷,使修复缺陷的代价最小,而且随着个人经验和技术的积累,还应学会怎样改进检查表以适应自己的要求。PSP2.1 则论述设计过程和设计模板技术,介绍设计方法,并提供了设计模板,但 PSP 并不强调选用什么设计方法,而是强调设计完备性准则和设计验证技术。

实施 PSP 的一个重要目标就是学会在开发软件的早期实际地、客观地处理由于人们的疏忽所造成的程序缺陷问题。人们都期盼获得高质量的软件,但是只有高素质的软件开发人员并遵循合适的软件过程,才能开发出高质量的软件,因此,PSP2 引入并着重强调设计复查和代码审查技术,一个合格的软件开发人员必须掌握这两项基本技术。

4) 个体循环过程 PSP3

PSP3 的目标是把个体开发小程序所能达到的生产效率和生产质量,延伸到大型程序。其方法是采用螺旋式上升过程,即迭代增量式开发方法,首先把大型程序分解成小的模块,然后对每个模块按照 PSP 过程进行开发,最后把这些模块逐步集成为完整的软件产品。

在本章中,将重点阐述:

- 如何估计待开发软件规模的大小;
- 如何估计在开发过程中可能引入的缺陷数;
- 如何估计在 PSP 过程中被排除的缺陷数。

2. 程序规模估计

程序规模一般可以程序代码行数来表示,因此程序规模估计主要是估计待开发的代码行多少。估计待开发程序代码的方法很多,本章仅介绍一种软件功能点估算方法^[6]。

用功能点法估算软件规模的步骤是:

- ① 确定软件的特性和功能范围;
- ② 将软件的功能进行分解;
- ③ 为每个独立功能确定其乐观值(S_o)、可能值(S_m)和悲观值(S_p);
- ④ 计算期望值 $EV = (S_o + S_m + S_p) / 6$;
- ⑤ 计算出软件的功能点 FP;
- ⑥ 根据不同的编程语言,由 FP 估算出软件代码行数 LOC。

软件功能点是基于软件信息域的可计算的度量,并考虑到对软件复杂性的评估而得出的。

$$\text{功能点 FP} = X[0.65 + 0.01 \times \sum F_i] \quad (5.1)$$

式中: X 为信息域值; F_i 为复杂度调整值。

1) 信息域值 X 计算

信息域值的计算如图 5.11 所示。

测量参数	计数	加权因子				
		简单	平均	复杂		
外部输入数	? ×	3	4	6	=	?
外部输出数	? ×	4	5	7	=	?
外部查询数	? ×	3	4	6	=	?
外部文件数	? ×	7	10	15	=	?
内部文件数	? ×	5	7	10	=	?
信息域值						?

图 5.11 信息域值计算

2) F_i 的取值

F_i 的取值可参见图 5.12。

评定每个因素的尺度是 0~5:

		0	1	2	3	4	5
		没有影响	偶有影响	轻微影响	中等影响	较大影响	严重影响
序 号	复杂性因素						
1	系统是否需要可靠的备份和恢复?						
2	是否需要数据通信?						
3	是否有分布处理的功能?						
4	性能是否关键?						
5	系统是否运行在既存的高度实用化的操作环境中?						
6	系统是否需要联机数据项?						
7	联机数据项是否需要输入处理以建立多重窗口显示或操作?						
8	主文件是否联机更新?						
9	输入、输出、文件、查询是否复杂?						
10	内部处理过程是否复杂?						
11	程序代码是否被设计成可复用的?						
12	设计中是否包括转换和安装?						
13	系统是否被设计成可重复安装在不同机构中?						
14	应用是否被设计成便于修改和易于用户使用?						

图 5.12 F_i 的取值

3) 功能点估算

分别统计出待开发软件的输入、输出数和文件数等,并分出简单、复杂等类别,据不同类别

乘上图 5.11 中相应的加权因子,即可得出如表 5.5 所列的计算结果。

表 5.5 功能点估算举例

测量参数	功能点			
	简 单	平 均	复 杂	合 计
外部输入数	$6 \times 3 = 18$	$2 \times 4 = 8$	$3 \times 6 = 18$	44
外部输出数	$7 \times 4 = 28$	$7 \times 5 = 35$	$0 \times 7 = 0$	63
外部查询数	$0 \times 3 = 0$	$2 \times 4 = 8$	$4 \times 6 = 24$	32
外部文件数	$5 \times 7 = 35$	$2 \times 10 = 20$	$3 \times 15 = 45$	100
内部文件数	$9 \times 5 = 45$	$0 \times 7 = 0$	$2 \times 10 = 20$	65
信息域值	304			
影响系数	1.15			
功能点总数	350			

4) 由功能点估算代码行规模 LOC

对不同的编程语言,对应一个功能点 FP 的代码行数平均值如表 5.6 所列。

表 5.6 LOC 与 FP

程序设计语言	LOC/FP(平均值)	程序设计语言	LOC/FP(平均值)
汇编语言	320	C++	90
C	128	Visual Basic	32
Cobol	105	Smalltalk	22
Fortran	105	Powerbuilder	16
Pascal	90	SQL	12

若某待开发软件的编程语言为 C++,则由表 5.7 和表 5.8 可得其预计的软件代码行 LOC 为

$$\text{LOC} = \text{FP} \times (\text{LOC/FP})_{\text{C++}} = 350 \times 90 = 31\,500$$

3. 度量引入缺陷个数

在软件开发过程中,引入软件缺陷是在所难免的。几乎所有的软件开发人员,在需求分析、设计、编码中都会引入缺陷。据统计,在设计阶段,设计人员一般会在 1 h 内引入 1~3 个缺陷;在编码阶段,一般会在 1 h 内引入 5~8 个缺陷;如果以代码行数中的缺陷数统计,则一般编程人员的平均缺陷引入率为 1 000 行代码中引入 50~100 个缺陷^[7]。

缺陷的度量单位是缺陷密度:

$$D_K = D/\text{KLOC}$$

式中: D 为缺陷数目; KLOC 为千行源代码。

软件开发人员可根据自己的经验和统计数据来估计缺陷密度。例如软件开发人员 A, 去年共开发了 5 个软件, 其程序规模和缺陷数如表 5.7 所列。

表 5.7 软件规模与缺陷数统计

程 序	规模(LOC)	缺陷数(D)	程 序	规模(LOC)	缺陷数(D)
1	12 000	90	4	25 000	126
2	5 000	124	5	20 000	207
3	1 200	16	Σ	63 200	563

开发人员 A 的缺陷引入率或缺陷密度为

$$D_K = D/\text{KLOC} = 563/(63\,200 \div 1\,000) = 8.91$$

由此估计该开发人员 A 的待开发项目的缺陷引入率。

由本小节“2. 程序规模估计”估计该待开发项目的规模为 31 500 LOC, 分配给开发员 A 完成的软部件约有 8530 LOC, 则可能的缺陷引入个数为 $D = 8.53 \times 8.91 = 76$ 个。

4. 度量排除的缺陷个数

据国外的统计^[7], 代码审查每小时约可排除 6~10 个缺陷(取均值 8 个), 缺陷排除率 η_w 为 40%~70%。现取其均值 $\eta_{wcp} = 0.55$, 则可得审查排除缺陷数为

$$D_w = D \times \eta_{wcp} = 76 \times 0.55 = 42$$

审查时间为

$$T_w = D_w / 8 = 42 \div 8 = 5.25 \text{ h}$$

据统计^[7], 单元测试每小时约可排除 2~4 个缺陷(取均值 3 个), 其缺陷排除效率 η_s 为 25%~35%。现取其均值 $\eta_{scp} = 0.30$, 则可得单元测试预计可排除缺陷数为

$$D_s = (D - D_w) \times \eta_{scp} = (76 - 42) \times 0.30 = 10$$

单元测试所需时间 T_s 为

$$T_s = D_s / 3 = 10 \div 3 = 3.33 \text{ h}$$

故可得开发人员 A 的测试计划如表 5.8 所列。

表 5.8 测试人员 A 的测试计划表

测试类别	预计的排除缺陷数	测试时间/h
代码审查	42	5.25
单元测试	10	3.33
小 计	52	8.58

由表 5.8 可得:

(1) 估计的残留软件缺陷数:

$$D_R = D - D_W - D_S = 76 - 42 - 10 = 24 \text{ 个}$$

(2) 缺陷排除效益:

$$D_B = \frac{\text{排除缺陷数}}{\text{排除缺陷耗费的工时}} = \frac{D_W + D_S}{T_W + T_S} = (42 + 10) / (5.25 + 3.33) = 6.06 \text{ 个/小时}$$

(3) 个人的缺陷排除效率 D_η

$$D_\eta = \frac{\text{排除缺陷数}}{\text{总缺陷数}} = \frac{D_W + D_S}{D} = \frac{(42 + 10)}{76} = 68\%$$

一般要求个人的缺陷排除效率应达到 70%。

上述软件的残留缺陷数完全是根据开发人员 A 的经验数据估计出来的,没有一般意义。

以下给出估计软件中残留缺陷数的方法。

假设: 每个缺陷被发现的概率相同; 缺陷数样本应足够大。

方法: 对同一个软件, 由测试工程师 A 和 B 分别进行独立的测试。其中工程师 A 发现的软件缺陷数为 N_A , 工程师 B 发现的软件缺陷数为 N_B 。在他们二人发现的缺陷中, 相同的缺陷数为 N_C , 则其发现的缺陷数为

$$N = (N_A + N_B) - N_C \quad (5.2)$$

据“捕获-再捕获”法则^[8]可得:

软件中的总缺陷数为

$$N_O = \frac{N_A \times N_B}{N_C} \quad (5.3)$$

软件缺陷中残留的缺陷数则为

$$N_R = \left(\frac{N_A \times N_B}{N_C} \right) - (N_A + N_B - N_C) \quad (5.4)$$

于是, 可得发现软件缺陷的效率为

$$\eta = (N_A + N_B - N_C) \times N_C / (N_A + N_B) \quad (5.5)$$

对上述估计的改进:

总缺陷数 N_O 的无偏估计为

$$N'_O = \frac{(N_A + 1) \times (N_B + 1)}{(N_C + 1) - 1} \quad (5.6)$$

方差为

$$\sigma^2 = \frac{(N_A + 1) \times (N_B + 1) \times (N_A - N_C) \times (N_B - 1)}{(N_C + 1)^2 + (N_C + 2)^2} \quad (5.7)$$

5.5.2 软件开发者的团队管理——小组软件过程

TSP 的目的就是要建立一个具有高度凝聚力的软件开发团队(team)。这些团队应当团

结、协作、高效地工作,从而能使其整体的实力大于各个个人实力之和。

实施 TSP 的基本条件是:

- ① 软件开发小组要完成的任务必须有明确的定义,即小组知道他们应完成什么任务。
- ② 小组的规模及其成员必须明确确定,每个人都应知道他自己和其他小组成员在组内的角色。
- ③ 小组对其承担的软件开发项目能自始至终地进行控制。

1. TSP 的简单框架

TSP 是通过多个开发周期,亦即多个循环来完成最终产品开发的。正如 PSP 也是通过循环开发的螺旋式上升过程逐步完成大型程序开发的。

TSP 的框架是由决定策略、制定计划、分析需求、进行设计、编码实现、实施测试和后期维护 7 个步骤组成的。

在每一个周期(循环)中应考虑:

- 每个周期内都必须完成一个最终产品的前期可测试产品;
- 每个周期应划分到可以保证在该周期内完成开发和测试任务;
- 各个周期的阶段产品可以组合成所需的最终产品。

用这种策略开发软件,先在第一个周期内开发出最小功能组合的合格产品,然后,在以后的每个循环中加进新的功能,并不断地进行测试和验证,这样可以确保最终能得到符合需求的、可运行的软件。

2. 小组及其角色的管理目标及其度量评价

1) TSP 小组的构成

一个飞行器系统或分系统的软件开发组织一般由 3 个层次组成:

第一层次:软件开发经理或软件总师,负责系统或分系统的软件开发管理。

第二层次:软件项目经理或软件项目组长,负责由该项目组承担的软件开发任务。在这同一层次上的还有软件质量保证经理、软件配置管理经理等。

第三层次:软件开发人员,包括软件设计与编程人员、测试人员、配置管理员、质量保证人员(可兼职)等。

这 3 个层次的体系结构可参见图 5.13。

对一个 TSP 小组来说,其组成就是图 5.13 中以粗黑框标示的部分,即由软件项目经理(小组组长)、开发人员、测试人员、配置管理员、质量保证人员和技术支持人员组成,其中部分人员可兼职。

2) 管理目标

对任何组织来说,首要的事是确定目标,不同的管理层次其管理目标不同。应当通过使所有成员参与目标的定义过程,来帮助小组成员接受全组的目标,并进而确定自己的目标。

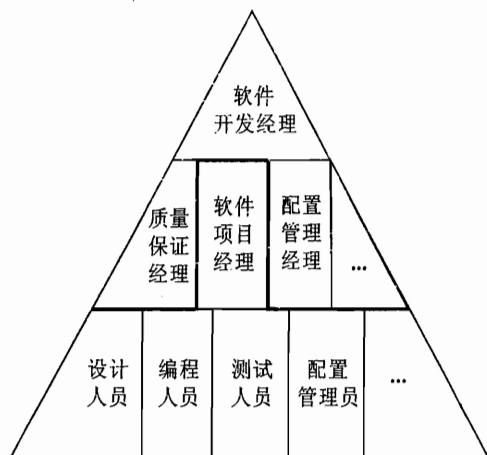


图 5.13 软件开发角色层次结构

(1) 软件开发经理(软件总师)的管理目标

- 满足用户(使用部门)的要求；
- 保证软件的开发质量；
- 减少软件开发周期与成本；
- 组织培训软件开发人员；
- 引进新技术与新工具。

(2) 软件项目经理的管理目标

- 组建和领导一个有效的项目组；
- 使软件项目计划的执行状况符合要求；
- 使软件项目的资源使用状况符合要求；
- 使软件的开发质量符合要求；
- 确保软件过程和活动的可视性；
- 关注开发人员的工作经历和人员稳定性。

(3) 软件开发人员的管理目标

- 个人工作进度与计划相比的执行情况；
- 个人的软件开发质量(如引入的缺陷数、排除的缺陷数等)；
- 个人的计算机资源使用情况。

(4) 小组的目标

- 开发出一个高质量的软件产品；
- 开发任务能在给定的周期内完成；
- 对软件开发过程进行有效的控制、良好的管理。

对小组的这些非量化的目标,应给予度量,这样才能知道是否达到了预定的基本目标,并对小组软件过程加以控制。

3) 度量目标

度量目标是整个软件过程度量中最关键的一步。它包括项目组(TSP 小组)的度量目标和项目组成员层中各个角色的度量目标。项目组及组内各个角色的度量目标关系图可参见图 5.14。

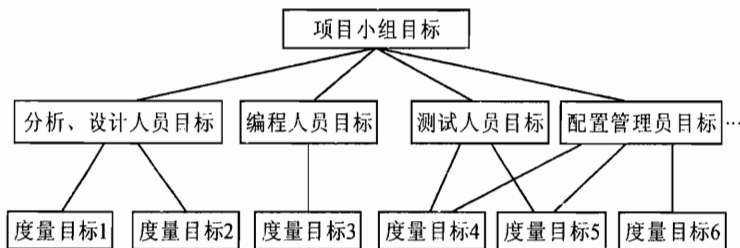


图 5.14 项目组及组内各个角色的度量目标关系

(1) 项目小组的度量目标

① 开发出一个高质量的软件产品

度量目标 1.1: 最终的软件产品实现规定需求的 100%。

度量目标 1.2: 在编译和静态测试后,发现软件的缺陷率应达到 80%。

度量目标 1.3: 在系统测试中发现的缺陷数为 0。

② 开发任务能在给定的周期内完成

度量目标 2.1: 对给定的周期提前或延后的天数 $\Delta T \leq 3$ 天。

③ 对软件开发过程进行有效的控制和良好的管理

度量目标 3.1: 对产品规模大小的估计误差 $\leq 20\%$ 。

度量目标 3.2: 对产品开发费用的估计误差 $\leq 20\%$ 。

度量目标 3.3: 对软件各基线中每个配置项变更的控制和纪录达 100%。

(2) 小组中各个角色的度量目标

小组的度量目标必须分解到小组中的每个角色,只有每个角色实现了自己的度量目标,小组的度量目标才会实现。

① 软件分析、设计人员的度量目标

度量目标 1: 软件需求的可验证性应达到 100%。

度量目标 2: 软件需求的二义性应达到 0。

② 软件编程人员的度量目标

度量目标 3: 在软件测试前,发现的软件缺陷率 $\geq 70\%$ 。



③ 软件测试人员的度量目标

度量目标 4: 在静态测试中,发现的软件缺陷密度 $\leq 10/\text{KLOC}$;而在单元测试中,发现的软件缺陷密度 $\leq 5/\text{KLOC}$ 。

度量目标 5: 在系统测试中,发现的软件缺陷密度 $=0$ 。

④ 软件配置管理员的度量目标

度量目标 6: 对软件各基线中每个配置项变更的受控率和记录达 100%。

以上仅是度量目标的示例,对一个给定的软件开发项目,项目小组应根据小组的基本目标,将每个目标分解到组内的每个角色,并为每个角色确定各自的度量目标。

3. 建立 TSP 标准的质量评价准则

为了衡量 TSP 开发过程的质量及最终软件产品的质量,必须建立一套标准的过程与产品的质量评价准则,这是 TSP 开发高质量软件的重要组成部分。本节给出了 TSP 在开发软件过程中及最终软件产品的质量评价准则。准则中的各个数据取自于国外 TSP 的质量标准。对我国的 TSP 小组,可参照这些数据和自己的统计数据来制定 TSP 的质量评价准则。

1) 软件开发过程中的缺陷引入率

缺陷引入率一般以每小时的开发工作(包括需求分析、设计、编码等)中引入的软件缺陷计算,参见表 5.9。

2) 软件开发过程中的缺陷排除率

对软件中已经引入的缺陷,通常采用查错方法排除。一般的查错方法在编码之前是评审(包括内部同行评审和正式评审),在编码完成后则采用测试(静态、动态)方法查错。

缺陷排除率一般以查错(评审、测试)工时中发现的缺陷数估计,参见表 5.10。

表 5.9 软件各开发阶段的缺陷引入率

开发阶段	缺陷引入率(个/小时)
需求分析	0.25
概要设计	0.25
详细设计	2.0
编 码	4.0

表 5.10 各种排错手段要求的排除率

排错阶段	缺陷排除率(个/小时)
需求评审	0.5
概要设计评审	0.5
详细设计评审	2.0
代码审查	6.0

3) 控制软件开发过程的质量准则

控制软件开发过程的质量准则有:

- ① 检查需求分析的时间应为进行需求分析时间的 25%;
- ② 检查需求规格说明的速率应小于 2 页/小时;
- ③ 用于设计评审(含同行评审和正式评审,下同)的时间应为设计时间的 50%;

- ④ 检查概要设计说明的速率应小于 5 页/小时；
- ⑤ 检查详细设计说明的速率应小于 100 伪代码行/小时；
- ⑥ 用于需求分析和设计的时间应比用于编码的时间长 2 倍；
- ⑦ 代码复核(含代码审查)的时间应不小于编码时间的 50%，最好是 75%；
- ⑧ 代码复核的速率应小于 200LOC/小时；
- ⑨ 每复核 1 h 代码，应发现 3 个以上的缺陷；
- ⑩ 复核代码时发现的缺陷应是编译时发现缺陷的 2 倍。

4) 软件产品的质量评价准则

这里提出了两个关于软件产品的质量评价准则：一个是软件无缺陷百分比；另一个是软件总需求的实现率。

(1) 软件无缺陷百分比(PDF)

软件无缺陷百分比是指某一开发阶段结束时，软件残留的缺陷数与总缺陷数之比：

$$PDF = 1 - \frac{D_i}{D_o}$$

式中： D_i 为某开发阶段结束时的残留缺陷数； D_o 为软件产品总缺陷数。

表 5.11 要求各开发阶段的 PDF

开发阶段	PDF
单元测试	0.50
集成测试	0.70
系统测试	0.85~0.95

要求各开发阶段的 PDF 能达到表 5.11 的要求。

由表 5.11 可知：软件在系统测试后交付时，其残留的软件缺陷率为 5%~15%，亦即其无缺陷率可达 85%~95%。一般认为，软件的无缺陷率达到 85% 用户即可接受，达到 95% 则用户有最高的满意度。而无缺陷率达到 100% 的软件产品几乎是不存在的，或者是代价极为昂贵的。

(2) 软件总需求的实现率

$$\text{软件总需求的实现率} = \text{已实现的需求总数} / \text{总需求数}$$

若软件总需求的实现率 > 99%，则用户较为满意；若 < 95%，则应予以警告。

5.5.3 软件能力成熟度模型(CMM)

1. CMM 的概念

CMM(Capability Maturity Model for Software)即软件能力成熟度模型，是美国卡耐基-梅隆大学软件工程研究所(SEI)于 1991 年提出的一套先进的软件开发过程管理思想和方法。它可以用来评价软件开发单位的能力成熟程度，并有效地帮助软件开发单位改进与优化管理，提高软件开发能力和产品质量。

CMM 的基本理念是持续的过程改进。它构建了一个由 5 个成熟度级别组成的过程改进框架，以渐进的方式将 5 个成熟度级别定义为一个顺序的等级，用以判断组织软件过程的成熟



程度,并评估其软件过程能力。一个成熟度等级是迈向一个成熟软件过程的、有明确定义的台阶,每一个成熟度等级提供了持续过程改进的一个层次,如图 5.15 所示。

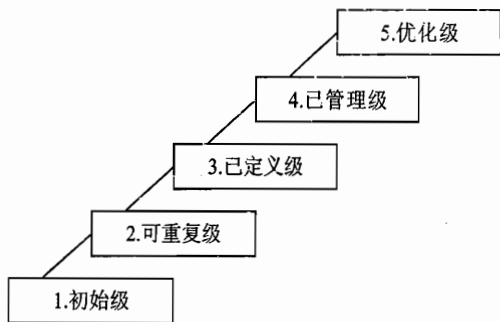


图 5.15 CMM 的结构图示

CMM 的目标是:

- 提供对软件开发组织的软件过程能力进行评估的一个基本框架。
- 指导软件开发组织一个逐步改进其软件能力成熟度的分层框架,即为软件开发组织设计一条从混乱、无序的软件过程达到有纪律、有序的成熟过程的进化途径,包括指出进化的步骤、活动、措施及必要的资源和条件等。
- 选择软件过程改进策略的一个指南:
 - 确定当前的软件过程能力;
 - 识别对软件质量和过程改进最关键的问题,了解软件开发组织的强项和弱项。

CMM 只回答“做什么”,而“如何做”则由开发组织自己决定。

2. CMM 的特征及其关键过程域

CMM 分成 5 个成熟度等级,其特点是:

- 每个成熟度等级用一个关键词来描述该等级的特征,并用一段文字来解释该等级的特征。
- 每个成熟度等级包含一组关键过程域,指明处于该等级的软件开发组织最应关注的几个软件过程(包括工程过程和管理过程)领域。
- 当该等级中所有关键过程所描述的目标全部达到时,就认为已经达到该成熟度等级。

CMM 的关键过程域如图 5.16 所示。

CMM 的 5 个级别各自的过程特点、行为特征和关键过程域可参见表 5.12。

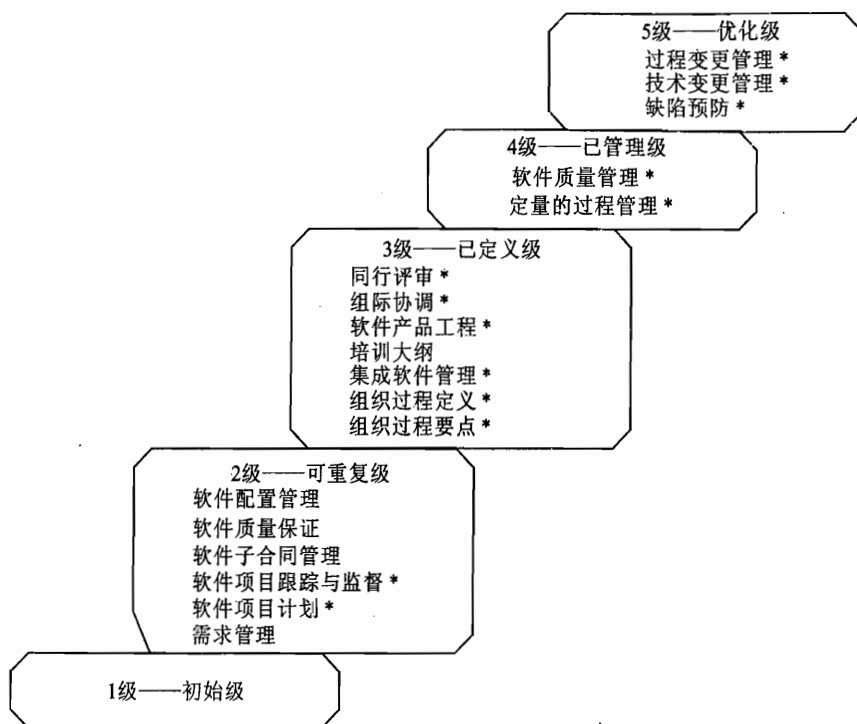


图 5.16 软件过程成熟度模型的关键过程域

表 5.12 CMM 的过程特点、行为特征和关键过程域

等 级	过程特点	行为特征	关键过程域
1 级	- 过程没有定义 - 过程有时混乱 - 成功靠个人努力	- 项目计划无效 - 管理是反应式的 - 开发环境不稳定	无
2 级	- 项目过程有定义 - 过程有纪律 - 能重复类似的成功	- 基于项目经验,规划新项目管理 - 项目管理制度化、有效 - 组织上有项目管理的方针	- 需求管理 - 软件项目计划 - 软件项目跟踪和监督 - 软件子合同管理 - 软件质量保证 - 软件配置管理



续表 5.12

等 级	过程特点	行为特征	关键过程域
3 级	• 标准化 • 全组织统一 • 管理和工程活动协调	• 项目裁剪执行组织的标准过程 • 组织定义统一的标准过程和裁剪方法	• 组织过程要点 • 组织过程定义 • 培训大纲 • 集成软件管理 • 软件产品工程 • 组织协调 • 同行评审
4 级	• 已全面测量 • 有定量理解与控制 • 可预测	• 项目测量生产率和质量,并定量控制过程的变化 • 组织设定定量质量目标和测量大纲	• 定量的过程管理 • 软件质量管理
5 级	• 过程不断改进	• 项目分析缺陷原因,评价过程 • 组织识别最佳技术,持续改进过程	• 缺陷预防 • 技术变更管理 • 过程变更管理

3. 各级 CMM 的渐进过程

CMM 是一个渐进的过程。各级 CMM 的过程从无序到有序,从定性管理到定量管理,并发展为不断改进与优化。其过程、人员、技术和测量都在不断地改进、提高和优化。对这些渐进的概要描述可参见表 5.13。

表 5.13 沿 CMM 等级渐进的过程描述

项 目	等级 1	等级 2	等级 3	等级 4	等级 5
过 程	几乎没有稳定的过程,“仅仅是去做”	在项目层有文档化且稳定的估计、策划和承诺过程,问题一旦发生就加以识别和纠正	集成的管理和工程过程在全组织均使用;对问题进行预想和预防或使其影响最小	过程得到定量的理解和稳定;个别问题的根源得到了理解和清除	过程不断得到系统化的改进;问题的公共根源得到了理解和清除
人 员	成功取决于个人;救火是一种实际工作方式;不同业务部门间不协调,还可能有矛盾	成功依赖于个人管理系统支持;承诺得到理解和管理;人员得到培训	若干项目组一起工作,一般作为 1 个一体化的产品组;按照角色规划并提供培训	每个项目内部有强烈的团队意识	全组织都有强烈的团队意识;每个人都参与过程改进

续表 5.13

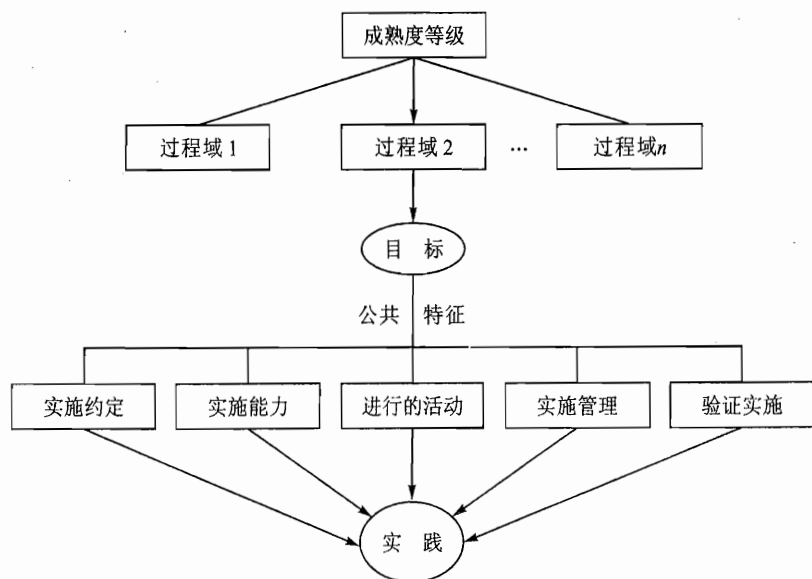
项 目	等级 1	等级 2	等级 3	等级 4	等级 5
技 术	引进新技术是危险的	技术支持已建立了稳定的活动	新技术都在定性基础上进行评价	新技术都在定量基础上进行评价	新技术的追踪和推广应用都是经过周密考虑的
测 量	数据采集和分析是无序的	策划和管理个别项目所用的数据	在已定义过程中采集和使用数据;各项目系统地共享数据	数据定义和采集在全组织是标准化的;数据被用来定量地理解过程和使过程稳定	数据被用来评价和选择过程改进方法

5.5.4 从 CMM 到 CMMI

CMMI(Capability Maturity Model Integration)是集成的能力成熟度模型。该模型于 20 世纪 90 年代末,由美国 SEI 提出。使用 CMMI 的目的是通过采用集成模型,例如系统工程和软件工程的集成模型来提高工程系统的效能、投资回报率和效率。

CMMI 最初综合的模型有 3 种,即软件工程的模型 SW-CMM V2.0、系统工程的模型 EIA/IS731 和综合产品研制(IPD)的模型 IPD-CMM V9.0 a。由此可见,该模型是为了改进系统工程和软件工程这两者的过程而开发的。

CMMI 模型有两种表示方法:连续表示法和分级表示法。为了与本章前述的 CMM 更好地衔接,我们采用分级表示法。分级表示法的 CMMI 模型又有成熟度等级、过程域、公共特征、目标和实践等部件,如图 5.17 所示。

图 5.17 CMMI 模型的部件^[5]

CMMI 成熟度等级也分为 5 级,每一级的过程特点、过程域和公共特征可参见表 5.14。

表 5.14 CMMI 的过程特点、过程域与公共特征

等 级	过程特点	过程域	公共特征
1 级 (已实施级)	过程性能不稳定、混乱,并且可能不满足质量、成本和进度目标	无	
2 级 (已管理级)	过程已管理,对项目过程均有计划、实施、监督和控制,以达到既定目标	(1) 需求管理 (2) 项目策划 (3) 项目监督和控制 (4) 供应方协议管理 (5) 测量和分析 (6) 产品与过程质量保证 (7) 数据管理 (8) 配置管理	(1) 实施约定: 包括确保过程能建立且将持久的实践。实施约定一般涉及建立组织的方针政策和领导责任。 (2) 实施能力: 包括为完全地实施过程建立必要条件的实践。实施能力涉及资源、组织机构和培训。 (3) 实施活动: 包括直接实现一个过程域的实践。实施活动一般涉及制定计划和规程,进行工作,跟踪工作以及必要时采取纠正措施。 (4) 实施管理: 包括收集和分析与该过程有关数据所必需的测量实践。这些测量数据用来提供对过程性能的深入了解。 (5) 验证: 包括确保符合已建立过程的实践。验证一般涉及评审和审计。
3 级 (已定义级)	过程已定义,这种定义过程是从组织的标准过程集剪裁得到的已管理过程	(1) 顾客和产品需求 (2) 技术解决方案 (3) 产品集成 (4) 产品验证 (5) 确认 (6) 组织过程焦点 (7) 组织过程定义 (8) 组织培训 (9) 集成项目管理 (10) 风险管理 (11) 决策分析和决定	
4 级 (已量化级)	过程是一种定量管理过程,是利用统计和其他定量技术加以控制的已定义过程	(1) 定量管理质量和过程 (2) 组织过程性能	
5 级 (已优化级)	过程是优化过程,是基于对过程固有的过程变异原因的理解,并加以改进的已量化过程	(1) 原因分析与解决 (2) 组织过程技术创新 (3) 过程创新推广应用	

5.6 下一代飞行器研制中的软件可靠性问题

5.6.1 美国 F-22 的教训

F-22 是美国,也是当今世界上唯一的、已服役的第四代战斗机。F-22 软件控制着飞机上 80% 的功能,其费用占飞机工程与制造研制费用的 30%。F-22 在验证与确认阶段制定了其风险范围,其软件开发的风险列为最高风险区。

F-22 是目前美国空军飞机中最复杂的软件密集系统。那么,F-22 的软件是怎么开发的呢?美国人认为:近 30 年的实践表明,软件工程的确是摆脱软件危机,确保软件质量的有效途径。IBM 公司在总结其成功开发航天飞机飞行软件的经验时说明,其主要的经验有两条:

一是认真实施软件工程;二是特别加强软件测试。F-22 正是严格地遵循这两条经验来开发其软件的。

1. 认真实施软件工程

(1) F-22 软件开发强制性要求遵循 MIL-STD-2167A 标准。该标准提供软件开发、软件测试和文档编制的结构化过程。

(2) F-22 软件开发强制性要求采用 Ada 军用标准语言,

(3) 为降低风险,F-22 航空电子软件是分块批次开发的,开始时开发基本功能和处理部分,随着每个批次软件的成功发布,航空电子系统不断成熟,最后实现更为复杂的航空电子系统功能。

(4) 采用综合产品组(IPT)。

(5) 软件开发团队的成员采用相同的软件开发环境。F-22 的软件工程环境(SEE)是美国国防部采用的最完善的工程环境,用于软件设计、编码和测试,

(6) 重视软件工程师的培训。共有 5 个培训中心,到 1995 年已完成了 50 000 h 的培训。

F-22 航空电子软件开发采用的是多 V 模型的开发方法。

F-22 上的航空电子硬件和软件共开发了 4 批次:

- 1.0 批次(1999 年上半年开发)提供了基本飞行的航空电子功能和主要的雷达能力,装有 50% 航空电子系统的源代码行,并提供了传感器与驾驶员数据流。
- 2.0 批次(1999 年下半年开发)增加了电子战(EW)传感器(用于自我防护)功能,开始传感器融合,无线电频率协调和重新布局。
- 3.0 批次(2000 年 4 月)增加了其他雷达、EW 传感器模块和先进通信导航识别(CNI)功能。3.1 批又增加了通信数据链,包括联合战术信息分布系统(JTIDS)接收能力、小编队内数据传输线,以及 GBU-32 联合直接攻击弹药 JDAM 发射能力。
- 4.0 批次(2005 年)增加了头盔瞄准具(HMS)、AIM-9X 空对空导弹综合和联合战术



信息分布系统发射能力,用于工程与制造研制后的初始作战能力评价。

2. 特别加强软件测试

F-22 的航空电子系统软件测试分 4 个阶段:

- (1) 在航电系统软件各组成部件(CSCI)开发出来后,在各软件开发试验室内测试。
 - (2) 将整个任务软件包与安装到航空电子综合试验室(AIL)中的真实传感器硬件综合,测试其使用情况。
 - (3) 将软件包再装载到由波音 757 改装的 F-22 航空电子系统飞行试验台(FTB)试飞。
 - (4) 航电系统软件在 F-22 试验机上完成所有的试验。
- F-22 的航空电子系统软件系统测试如图 5.18 所示。

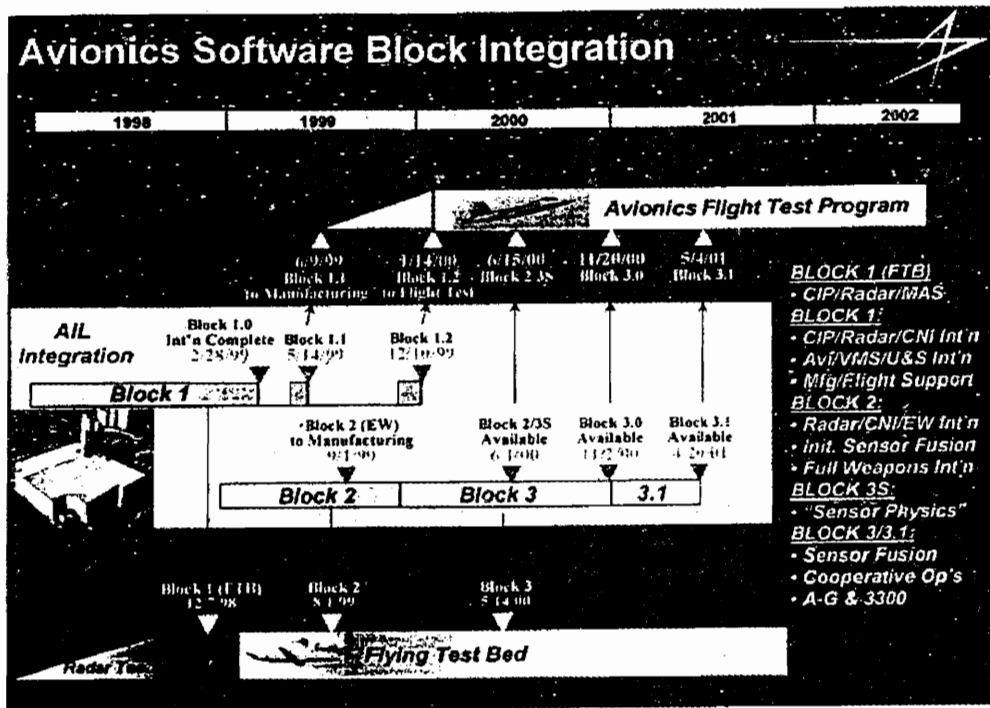


图 5.18 F-22 的航空电子系统软件系统测试

1) 航空电子综合试验

航空电子综合试验是开发新一代综合航空电子系统所必需的基础试验。为此,各国的航空电子开发商都采用了航空电子综合试验室(AIL),用于试验与评价航空电子系统(包括硬件和软件)的性能与可靠性。

F-22 航空电子系统包括软件与硬件的 AIL 试验,到 2002 年 10 月共进行了 21 000 h。

F-22 的 AN/APG77 雷达,从 1991 年到 2002 年 11 台雷达共计进行了 12 000 h 的试验。

航空电子综合实验室的建造耗资 2 500 万美元,在 AIL 五年的综合测试花费共 10 亿美元。

2) 飞行试验台(FTB)试验

F-22 航空电子系统在波音-757 飞行试验台(FTB)试飞。1998 年 3 月,第一架装载航空电子系统的波音 757 飞行试验台首次飞行;到 2001 年 6 月,航空电子系统在 FTB 上进行了 650 h 的飞行试验;到 2002 年初,累计飞行 1 000 h。

3) F-22 研制试验飞机飞行试验

F-22 研制试验飞机飞行试验安排:第一批 9 架试飞飞机中有 6 架专门用于航空电子系统的飞行试验,试验的时间达 3 000 多小时,大约占飞机飞行试验时间(4 583 h)的 2/3,主要用于各种传感器的综合或传感器信息的融合。F-22 从 2000 年到 2002 年共飞行了 35 个月。

3. F-22 在飞行试验中出现的软件可靠性问题

按理说,经过这么长时间,如此充分的地面测试和空中试验,应当能确保 F-22 的研制飞行试验顺利进行,但事实并非如此。

据 2003 年 4 月美国空军负责采办的助理部长说,F/A-22 项目已攻克了大部分技术难关,解决了尾翼抖振和驾驶舱颤噪问题。最近的飞行试验显示,该项目最后的遗留问题就是软件的不稳定性。美国空军要求 F/A-22 整个软件包(具有 2 百多万行 Ada 程序)工作 20 h 不会发生使任何 1 个航电组件失效的问题。在 2003 年 2 月中旬的试验中,3 h 的飞行试验就出现 1 次软件错误,而实验室试验结果是 8 个多小时出现一次软件错误。

美国空军称,自 2001 年 1 月以来的 F-22 飞行试验过程中,出现了许多航空电子软件可靠性的问题,造成了许多次飞行试验任务中断,受到了各方的批评。据称,航空电子设备的组成部件如雷达、电子战系统、通信系统、导航系统等都不存在问题,问题在于各个组成部件之间的综合。特别是软件的综合问题,航空电子软件在技术上遇到的最大挑战是成功实现多传感器的数据融合。其中的一些问题在波音 757 试验台上就被发现,而另外的一些问题则等到装备到了 F-22 上试飞之后才被发现。由于软件的问题,F-22 的座舱系统每运行 2 h 就要关闭一次。航空电子软件可靠性的问题能造成部分航空电子系统如雷达处于异常状态,甚至完全不能运行。在这种条件下,试飞员必须重新启动 F-22 航空电子系统。

这一系列严重的软件问题,使美国空军和承包商承受了巨大压力。美国空军称,软件可靠性问题是能够解决的。为此,美国空军加大了投入,通过提高技术人员的素质,提供更多的专用测试工具,并加强与工业界的合作,以尽快解决 F/A-22 的软件可靠性问题。空军组织工业界专家成立了独立专家组,对航空电子软件可靠性的问题进行研究和评估。由于软件开发的延迟迫使空军延长了项目发展阶段,造成超过 8.76 亿美元的费用超支。

在研制进度延后了一年半,费用超支 8.76 亿美元后,美国空军称,F-22 已解决了航空电子软件可靠性问题,并称,软件可靠性已不再是问题。

但就在军方于 2004 年 5 月宣称 F-22 的软件可靠性问题已得到“解决”后,却又出现了一



系列事故。

2004年12月20日,一架美军F/A-22“猛禽”战斗机坠毁。这架崭新的生产型战斗机在起飞的过程中失控,坠地爆炸,飞行员弹射逃生。空军参谋长Jumper指出:本次事故是由与飞控有关的软件问题造成的。这是F/A-22战斗机生产型的首次坠毁事故。此前F/A-22战斗机的原型机YF-22由于软件问题在降落时发生过一次坠毁事故。

2007年2月,美国空军12架F-22A“猛禽”战斗机计划飞抵日本冲绳的空军基地,开始首轮海外部署。12架“猛禽”从夏威夷飞往日本,途经国际日期变更线时,飞机上的全球定位系统纷纷失灵,多个电脑系统发生崩溃,多次重启均告失败。飞行员们无法正确辨识战机的位置、飞行高度和速度,他们不得不掉头返航,折回到夏威夷的空军基地。

鉴于这次行动属于美国国防部对F-22A进行初始作战试验与鉴定阶段(IOT & E)的内容,洛·马公司表示将对所有已出厂的87架F-22A进行全面检查,对有问题软件系统实施修改升级。

对于这种在软件密集型飞行器研制中忽视软件可靠性而引发软件故障频发的问题,美国的Koss博士曾指出:在现代军用飞机研制中,“软件可靠性受到很大的忽视……,一些比较大的命令控制系统对软件可靠性不做任何评估”。他还指出:“在现代军用飞机上,软件已经超过100万行源代码,交付可靠的计算机硬件的能力是确定的,提交的系统软件将决定整个系统是否满足使用的可用性。但是软件可靠性仍常常被忽视,即使今天,实施软件可靠性的组织仍然少得可怜”。

我们在前面已指出,美国在航天飞机和第三代飞行器研制中得出软件开发必须“认真实施软件工程,特别加强软件测试”的宝贵经验,并取得了很大成效,但这种经验用到第四代软件密集型的先进飞行器中显然是不够的。这是因为,从第三代装备到第四代装备的跨越中,软件功能倍增,软件规模倍增,但软件问题也同时倍增,特别是软件可靠性问题成了严重制约第四代装备研制的瓶颈。

因此,对于下一代先进飞行器的软件开发必须是认真实施软件工程和软件可靠性工程。

5.6.2 F-35 软件可靠性的实施

1. F-35 软件特点

F-35 软件具有如下特点:

- 软件庞大、复杂:软件规模大于680万行的代码。
- 软件开发组织:庞大、复杂:多公司配合,国际化团队,多地点软件开发。
- 产品开发管理:全面的软件计划、标准及度量,保证有组织的软件开发工作。

2. F-35 软件可靠性的实施

F-35 的软件开发吸取了F-22的教训,十分重视软件可靠性。

首先,F-35建立了软件管理团队,从组织上保证了软件可靠性工作的开展。该管理团队负责引入、开发和应用软件可靠性工程,以获取软件可靠性的统计度量,并利用这些度量来预计软件质量。

F-35强调软件可靠性是软件产品质量和软件产品完整性(integrity)的一个指标(indicator),并要求在软件开发过程和系统综合与测试中均应开展软件可靠性工作。

F-35提出的软件可靠性的关键是:

- 确定可靠性指标;
- 制定软件操作剖面(operational profiles);
- 可靠性测试(按用户实际使用产品的方法测试软件系统);
- 可靠性测量。

软件可靠性技术用来:

- 预计一个新软件期望的缺陷数;
- 预计软件何时可交付;
- 预计为满足软件质量目标需要进行多少测试;
- 定量地预计软件交付时的质量;
- 预计外场的软件失效率。

鉴于F-35尚在研制中,目前我们了解到的F-35在软件可靠性方面开展的工作有:

(1) 管理并报告软件可靠性。在管理方面强调软件是系统中的一个重要组成部分,并借鉴硬件研制中的管理方法,包括报告软件的质量度量、概率度量等。

(2) 收集软件可靠性数据,并采用度量数据收集的工具。F-35要求持续跟踪软件缺陷,以估计和预计软件可靠性增长,从软件设计、编码、集成到系统综合各阶段,采用缺陷注入与缺陷探测方法持续跟踪软件缺陷。

(3) 软件可靠性预计和度量。软件管理部门使用历史数据或测试数据来预测软件可靠性增长,并跟踪软件可靠性指标。例如,利用测试过程中统计的累计缺陷数、缺陷率(defect rate)与软件质量指标相比较。

(4) 当软部件产品集成时,通过数据整合(如各软部件的缺陷数、累计缺陷强度(defect intensity)等数据来跟踪和预测软件可靠性MTTF。

(5) 通过测试计划和测试资源的使用来确认测试费用和约束条件。F-35的测试类型与F-22类似,即包括软件CSCI_s的测试、航电综合试验室(AIL)测试、飞行试验平台(波音737改装)和F-35原型机(共12架)试飞等组成。

(6) 在飞行器部署后,软件可靠性能用来估计外场的软件失效率,因为软件可靠性测试所发现的是运行软件失效(operational software failure)。可利用外场的使用强度(field usage intensity)来估计期望的软件失效数。

软件一旦交付后,就具有常值的失效/缺陷率(constant failure /defect rate)*。

5.6.3 下一代飞行器研制中的软件可靠性工程

1. 软件可靠性工程模型

1) 软件可靠性工程 Musa 模型

图 5.19 是 John . Musa 给出的软件可靠性工程的过程框架,姑且称之为软件可靠性工程的 Musa 模型。该模型将软件可靠性工程过程定义如下:

- 定义“必要的”可靠性;
- 开发运行剖面并准备测试;
- 执行测试及失效数据处理;
- 得到软件的可靠性。

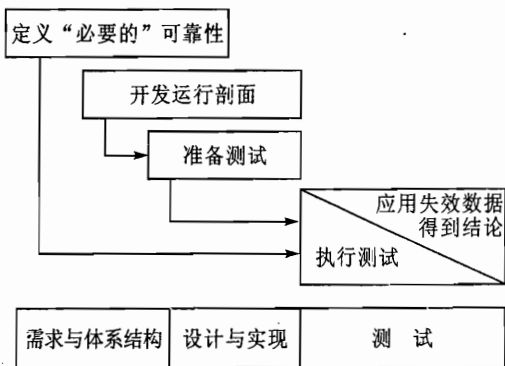


图 5.19 软件可靠性工程的过程框架 (Musa)^[13]

我们认为此模型的不足之处在于:

- 软件可靠性工程遍历的阶段只有需求、设计、实现和测试阶段,软件可靠性工程应当覆盖软件生存期全过程,软件可靠性的增长一直要延续到使用维护阶段。
- 缺少软件可靠性分析与设计,软件可靠性是设计出来的,在定义“必要的”可靠性参数和指标后,接下来应进行软件可靠性分析和设计。
- 仅仅定义“必要的”可靠性是不够的,还应进行软件可靠性估计,包括软件可靠性早期预计和增长预计。

为此,下面给出我们的软件可靠性工程框架。

2) 软件可靠性工程 Ruan 模型

图 5.20 展示的软件可靠性工程模型是作者对图 5.19 所示的 Musa 模型的改进:

* 这是假设软件失效前工作时间符合指数分布,则软件失效率为常数。

- 它包括了软件的整个生存周期,这意味着软件可靠性工程活动必须涵盖软件生存周期,而不是到测试阶段就结束。
- 它包含了软件可靠性分析和软件可靠性设计的重要内容,亦即软件可靠性是设计出来的。
- 它不仅包括了软件可靠性参数与指标的确定(即定义“必要的”可靠性);而且还涉及对这些指标的预计,这对可靠性设计方案的权衡、选择十分重要;还包括在测试阶段和使用、维护阶段进行的软件可靠性预计和估计。
- 它包含了软件可靠性管理,软件可靠性是设计出来的,是管理出来的。

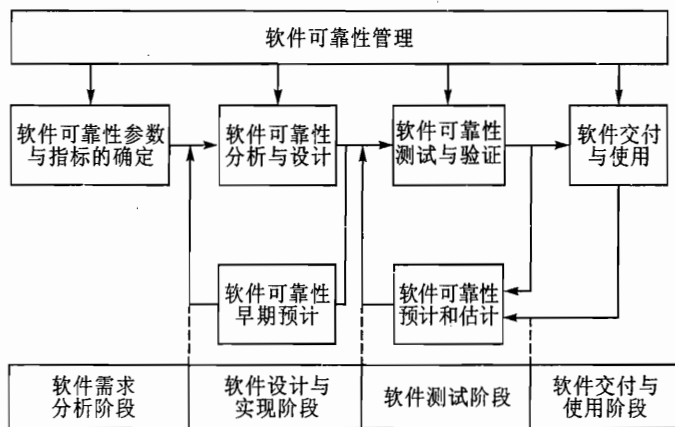


图 5.20 软件可靠性工程框架(Ruan)

2. 软件可靠性参数选取与指标的确定

1) 一般的软件可靠性参数

(1) 可靠度 R

软件可靠度 R 是指软件在规定的条件下、规定的时间段内完成预定功能的概率。或者说，是软件在规定时间内无失效发生的概率。

(2) 失效率

失效率是指在 t 时刻尚未发生失效的条件下,在 t 时刻后单位时间内发生失效的概率。

(3) 平均失效前时间 MTTF 和平均失效间隔时间 MTBF

① 平均失效前时间 MTTF 定义

MTTF 是指当前时间到下一次失效时间的均值。假设当前时间到下一次失效的时间为 ξ ,且具有累计概率密度函数 $F(t) = P(\xi \leq t)$,即可靠度函数 $R(t) = 1 - F(t) = P(\xi > t)$,则

$$MTTF = \int_0^{\infty} R(t) dt$$

② 平均失效间隔时间 MTBF 定义

MTBF 是指两次相邻失效时间间隔的均值。假设两次相邻失效时间间隔为 ξ ,具有累计

概率密度函数 $F(t)=P(\xi \leq t)$, 即可靠度函数 $R(t)=1-F(t)=P(\xi > t)$, 则

$$MTBF = \int_0^{\infty} R(t) dt$$

2) 结合飞行器特点的软件可靠性参数

(1) 成功率

软件的成功率是指在规定的条件下软件完成规定功能的概率。

某些一次性使用的系统或设备, 如弹射救生、导弹等系统中的软件, 其可靠性参数即可选用成功率。

(2) 任务成功概率 MCSP

任务成功概率是指在规定的条件下和规定的任务剖面内, 软件能完成规定任务的概率。

某些情况下, 存在一些自然的任务时间, 如军事飞行任务等, 人们有理由关心无失效地完成任务的概率, 此时, 即可选择任务成功概率作为软件可靠性参数。

(3) 由平均失效前时间派生的参数

对于不同的飞行器可派生出不同的软件可靠性参数, 如对于飞机、宇宙飞船, 可以使用平均失效前飞行小时。

(4) 平均致命性失效前时间

平均致命性失效前时间是指仅考虑致命失效的平均失效前时间。所谓致命性失效是指使系统不能完成规定任务或可能导致人或物重大损失的软件失效或失效组合。

对于不同的飞行器系统, 同样能派生出不同的参数, 如对于飞机、宇宙飞船, 可以使用平均致命失效前飞行小时。

3) 软件可靠性参数的选取

(1) 一般的软件可靠性参数选取原则

在选取软件可靠性参数时, 一般可考虑以下几个原则:

- 对失效发生频率要求较低的系统, 可靠性参数可选失效率或失效强度, 如操作系统、电话交换系统软件等。
- 对在规定时间内对无失效工作要求比较高的系统, 可选可靠度作为软件可靠性参数, 如工业过程控制系统软件。
- 对使用比较稳定的软件, 可选平均失效前时间/平均失效间隔时间作为软件可靠性参数, 如通用软件包等。

(2) 飞行器软件可靠性参数选取原则

在选取飞行器软件可靠性参数时, 除了考虑一般的软件可靠性参数选取原则外, 还应考虑以下的原则:

- 考虑飞行器特点。例如, 对于飞机可选用平均失效间隔小时。
- 考虑软件所在系统的可靠性要求。飞行器软件多用于嵌入式计算机系统, 选取软件可

靠性参数时应考虑选取与嵌入式计算机系统相同的可靠性参数,如某航空惯导系统的可靠性参数为 MTBF,则软件可靠性参数也应选用 MTBF。

- 考虑软件可靠性验证方法。如果是试验室内验证,则一般选用合同参数;如果是外场使用验证,则选用使用参数。
- 考虑软件的使用要求。例如一次性使用的系统软件可靠性参数可选用成功率。

4) 软件可靠性指标确定依据

关于可靠性指标的确定方法,国外有一些定量的方法,例如生存周期最优化法、发行日期法等。但是这些方法是以充分的数据为前提的。在我国,目前难以得到必要的数据,因此,在此仅提出确定软件可靠性指标的基本原则,即“需要”与“可能”相结合,经综合权衡后加以确定。具体要考虑的方面包括:

- 使用方的要求;
- 系统的重要程度;
- 系统的可靠性要求(从国外的经验看,计算机系统软件的可靠性水平比硬件的可靠性水平至少要低一个数量级);
- 国内外软件可靠性的水平;
- 相似软件的可靠性水平;
- 软件管理、开发人员、软件开发技术水平及开发工具的使用等;
- 进度要求;
- 经费保障;
- 指标要求的可验证性。

3. 软件可靠性设计与软件可靠性分析

1) 软件可靠性设计

(1) 软件可靠性设计原则

软件可靠性设计的实质是在常规的软件设计中,应用各种必要的方法和技术,使程序的设计在兼顾用户的各种需求时,全面满足软件的可靠性要求。软件的可靠设计和软件的常规设计紧密地结合,贯穿在常规设计过程的始终。因此,要实现软件可靠性设计首先必须贯彻软件开发工程化。

(2) 软件可靠性设计的内容

软件可靠性设计的内容可归结为 3 个方面:避错设计、查错和改错设计、容错设计。

避错设计是使软件产品在设计过程中不发生错误或少发生错误的一种设计方法。总的设计原则是控制和减少程序的复杂性。避错设计适用于一切类型的软件,它体现了以预防为主的思想,是软件可靠性设计的首要方法,应当贯彻于设计的全部过程中。遗憾的是,由于客观事物的复杂性和设计人员认识的局限性,设计错误是没法完全杜绝的。

查错和改错设计是指在设计中赋予程序某些特殊的功能。查错设计是程序在运行中自动

查找存在错误的一种设计方法。改错设计是指在设计中赋予程序自我改正错误、减少错误危害程度的能力的一种设计方法。

容错设计是指在设计中赋予程序某种特殊的功能,使程序在错误已被触发的情况下,系统仍然具有正常运行能力的一种设计方法。

限于篇幅,关于软件可靠性设计较为详细的内容可参见文献[13]。

2) 软件可靠性分析

(1) 软件失效模式和影响分析(SFMEA)

飞行器软件在使用中发生失效(不可靠)会导致飞行器任务的失败,甚至导致灾难性的后果。因此,应在软件开发过程中,对可能发生的失效进行分析,采取必要的措施避免引起失效的错误引入软件,同时为避免类似问题的发生提供借鉴。这些工作将会大大提高使用软件的可靠性,减少由于软件失效带来的各种损失,提高装备的战斗力。

SFMEA 是在软件开发阶段的早期,通过识别软件失效模式,分析造成的后果,研究分析各种失效模式产生的原因,寻找消除和减少其有害后果的方法,以便尽早发现潜在的问题,并采取相应的措施,从而提高软件的可靠性和安全性。

(2) 系统 SFMEA 的分析步骤

系统 SFMEA 分析一般包括以下几个步骤:

① 系统定义。在系统定义中应说明系统的主要功能和次要功能、用途、系统的约束条件和失效判据等。系统定义还应包括系统工作的各种模式的说明、系统的环境条件,以及软、硬件配置。

② 危险分析。识别对系统功能和安全性影响较大的危险事件,确定软件安全关键单元,即对上述危险事件的出现有直接或间接关系的单元,作为 SFMEA 分析的重点。

③ 结构分解。结构分解就是将软件按照一定的分解原则划分为若干逻辑部分,以确定系统的分析级别,以及最小分析单元。结构分解的原则可基于软件系统的功能、结构特征或工作模式等,应根据软件的具体情况确定。

④ 失效模式分析。针对每个最小分析单元,确定其潜在的失效模式。在 IEEE 软件异常分类标准中给出了软件失效模式的分类,在进行 SFMEA 分析时可作为参考。

IEEE 软件失效模式分类如下:

- 操作系统失败;
- 程序挂起;
- 程序失败;
- 输入问题;
- 输出问题;
- 未达到性能要求;
- 整个产品失效;

► 系统错误信息等。

针对每个失效模式,分析所有可能的失效原因(参见表 5.15),并分析其对本单元、上一层次,直至整个软件系统的影响,以及失效影响的严重性,从而确定需要采取的改进措施。

表 5.15 潜在的软件失效原因表

一般失效原因	具体失效原因	一般失效原因	具体失效原因
逻辑遗漏或执行错误	• 遗忘细节或步骤 • 逻辑重复 • 忽略极限条件 • 不必要的函数 • 需求的错误表述 • 未进行条件测试 • 检查错误变量 • 循环错误	数据操作错误	• 数据初始化错误 • 数据存取错误 - 标志或索引设置不当 - 数据打包解包错误 - 变量参考错误数据 - 数据越界 • 变量缩放比率或单位不正确 • 变量维度不正确 - 变量类型错误 - 变量下标错误 • 数据范围不对
算法的编码错误	• 等式不完整或不正确 - 丢失运算结果 - 操作数错误 - 操作错误 - 括号使用错误 • 精度损失 - 舍入和舍去错误 - 混合类型 • 标记习惯不正确		
软、硬件接口故障	• 中断句柄错误 • I/O 时序错误 - 时序错误导致数据丢失 • 子函数或模块 - 子函数调用不当 - 子函数调用位置错误 - 调用不存在的子函数 - 子函数不一致	数据错误或丢失	• 传感器数据错误或丢失 • 操作数据错误或丢失 • 嵌入到表中的数据错误或丢失 • 外部数据错误或丢失 • 输出数据错误或丢失 • 输入数据错误或丢失

失效影响严重性分类如表 5.16 所列。

表 5.16 失效影响严重性分类

严重性类别	严重性定义
灾难性	阻碍航空器继续安全飞行和着陆
严重	降低航空器的性能或飞行员克服不利操作状态的能力,包括大大降低安全裕量或功能能力,身体疲劳或高负荷使飞行员不能准确或完整地完成任务,对人员造成严重的或潜在的致命伤害
较重	可能降低航空器的性能或飞行员克服不利操作状态的能力,包括较大地降低安全裕量或功能能力,较大地增加飞行员的工作量或降低工作效率,可能对人员造成伤害
较轻	不会严重降低航空器的性能或飞行员克服不利操作状态的能力,包括稍微降低安全裕量或功能能力,稍微增加飞行员的工作量或降低工作效率

在进行失效模式分析时,应填写 SFMEA 工作表格。SFMEA 工作表格应能完整地体现分析的目的和取得的成果。表 5.17 是一张 SFMEA 工作表格的示例,表中记录了 SFMEA 分析的结果。

表 5.17 SFMEA 工作表格示例

索引号	单 元	功 能	失效模式	可能的 失效原因	失效影响			严重性	改进措施
					单 元	子系统	系 统		
1.1	输出	输出数据提交用户显示	数值高于正常范围	逻辑问题、计算问题、数据操作问题	N/A	无	任务降级	较轻	
1.2			数值低于正常范围	逻辑问题、计算问题、数据操作问题	N/A	无	任务降级	较轻	
1.3			输出数据没有显示	逻辑问题、接口或时序问题	N/A	无	任务中止	严重	
1.x			x	x	x	x	x	x	

表 5.17 中各项含义如下:

- 索引号: 用于组织各项记录。
- 单元: 软件单元的名称。
- 功能: 单元具有的一般功能。
- 失效模式: 与功能相关的失效模式。
- 失效原因: 失效模式的可能原因。

➤ 失效影响: 对单元、高一层子程序和系统(在处理、输出等方面)的影响。

➤ 严重性: 按失效影响的严重性划分。

4. 软件可靠性测试

1) 软件可靠性测试的目的

软件可靠性测试是指为了满足软件可靠性要求,验证软件是否满足可靠性要求,评估软件的可靠性水平而对软件进行的测试。其采用的是基于软件操作剖面(对软件实际使用情况的统计规律的定量描述)对软件进行随机测试的测试方法。通过软件可靠性测试可以达到以下目的:

(1) 有效地发现程序中影响软件可靠性的缺陷,缺陷排除后,即可实现可靠性增长:软件可靠性是指“在规定的时间内、规定的条件下,软件不引起系统失效的能力,其概率度量称为软件可靠度”^[13]。软件的“规定的条件”主要包括相对不变的条件和相对变化的条件,相对不变的条件如计算机配置及其操作系统;相对变化的条件是指输入的分布,用软件的操作剖面来描述。按照软件的操作剖面对软件进行测试,一般先暴露在使用中发生概率高的缺陷,然后是发生概率低的缺陷。而高发生概率的缺陷是影响产品可靠性的主要缺陷,通过排除这些缺陷可以有效地实现软件可靠性增长。

(2) 验证软件的可靠性是否满足一定的要求:可以根据用户的可靠性要求确定可靠性验证方案,进行可靠性验证测试,从而验证软件可靠性的定量要求是否得到满足。

(3) 预计软件的可靠性:通过对软件可靠性增长测试中观测到的失效数据进行分析,可以了解当前软件可靠性的水平,预测未来可能达到的水平,从而为开发管理提供决策依据。

(4) 估计软件的可靠性水平:通过对软件进行可靠性摸底测试,估计软件已经达到的可靠性水平。

软件可靠性测试中暴露的缺陷,既可以是影响功能需求的缺陷,也可以是影响性能需求的缺陷。软件可靠性测试方法从概念上讲是一种黑盒测试方法,因为它是面向需求、面向使用的测试,所以它不需要了解程序的结构以及如何实现等问题。软件可靠性测试通常是在系统测试、验收、交付阶段进行,并且主要是在实验室内仿真环境下进行。当然,也可以根据需求和可能在用户现场进行。

2) 软件可靠性测试过程

软件可靠性测试的一般过程如图 5.21 所示,主要活动包括构造操作剖面、生成测试用例、测试数据及测试环境的准备、可靠性测试运行、可靠性数据收集、可靠性数据分析和失效纠正。

(1) 构造操作剖面:软件的操作剖面“是指对系统使用条件的定义。即系统的输入值用其按时间的分布或按它们在可能输入范围内的出现概率的分布来定义”^[13]。粗略地说,操作剖面用来定量地描述软件的实际使用情况。操作剖面是否能代表、刻画软件的实际使用,取决于可靠性工程人员对软件的系统模式、功能、任务需求及相应的输入激励的分析,取决于他们对用户使用这些系统模式、功能、任务的概率的了解。操作剖面构造的质量将对可靠性测试、

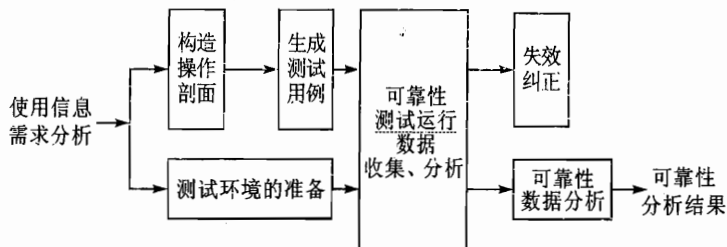


图 5.21 软件可靠性测试过程

分析的结果是否可信产生最直接的影响。

(2) 生成测试用例：软件可靠性测试采用的是基于操作剖面对软件进行测试的方法。因此，可靠性测试所用的测试数据是根据操作剖面随机选取得到的。

(3) 测试环境的准备：为了得到尽可能真实的可靠性测试结果，可靠性测试应尽量在真实的环境下进行，但是在许多情况下，在真实的环境下进行软件的可靠性测试很不实际，因此需要开发软件可靠性仿真测试环境。比如，对于大多数嵌入式软件，由于与之交联的环境的开发常常与软件的开发是同步甚至是滞后的，因此无法及时进行软件可靠性测试；而在有些系统中，由于交联的设备非常昂贵，因此无法用于需要进行大量运行的可靠性测试。

(4) 可靠性测试运行：是指在真实的测试环境或可靠性仿真测试环境中，用按照操作剖面生成的测试用例对软件进行测试。

(5) 数据收集：收集的数据包括软件的输入数据、输出结果，以便进行失效分析和回归测试；软件运行时间数据，可以是 CPU 执行时间、日历时间、时钟时间等；可靠性失效数据，包括每次失效发生的时间或一段时间内发生的失效数。失效数据可以实时分析得到，也可以事后分析得到。数据收集的质量对于最终的可靠性分析结果有着很大的影响，应尽可能采用自动化手段进行数据的收集，以提高效率及准确性和完整性。

(6) 数据分析：主要包括失效分析和可靠性分析。失效分析是根据运行结果和预期结果判断软件是否失效，以及失效的后果、原因等；而可靠性分析主要是指根据失效数据，估计软件的可靠性水平，预计可能达到的水平，评价产品是否已经达到要求的可靠性水平，为管理决策提供依据。

(7) 失效纠正：如果软件的运行结果与需求不一致，则称软件发生失效。通过失效分析，找到并纠正引起失效的程序中的缺陷，从而实现软件可靠性的增长。这一任务一般不是由软件可靠性测试人员，而是由软件开发人员完成。

3) 软件可靠性增长测试

软件可靠性增长测试是为了满足用户对软件的可靠性要求，提高软件可靠性水平而对软件进行的测试，是为了满足软件的可靠性指标要求，对软件进行测试—可靠性分析—修改—再测试—再分析—再修改的循环过程。软件可靠性增长测试过程如图 5.22 所示。

软件可靠性增长测试的特点如下:

- 测试目的: 通过测试-可靠性分析-修改-再测试-再分析-再修改的循环过程,使软件达到可靠性要求。
- 测试人员: 通常由软件研制方面而非使用方进行测试。
- 测试阶段: 通常在软件系统测试阶段。
- 测试场所: 一般在实验室中进行。
- 测试方法: 基于操作剖面的随机测试方法。
- 测试特征: 当测试过程中软件出现失效后,修改软件,排除引起失效的缺陷,从而实现软件可靠性的增长。
- 失效数据: 收集每次失效发生的时间或一段时间内发生的失效数。
- 数据分析: 根据软件可靠性增长预计模型预计软件的可靠性。

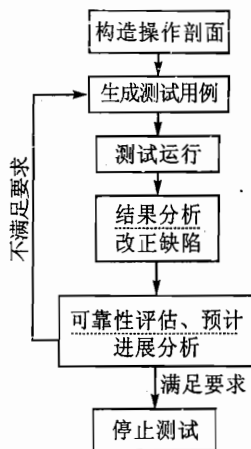


图 5.22 软件可靠性增长测试过程

4) 软件可靠性验证测试

(1) 基本概念

软件可靠性验证测试是为了验证在给定的置信度下,软件当前的可靠性水平是否满足用户的要求而进行的测试。即用户在接收软件时,确定它是否满足软件规格说明书中规定的可靠性指标要求。软件可靠性验证测试过程如图 5.23 所示。

软件可靠性验证测试的特点如下:

- 测试目的: 定量估计软件产品的可靠性,并作出接受/拒收回答。
- 测试人员: 通常由使用方参与测试。
- 测试阶段: 软件确认(验收)阶段。
- 测试场所: 既可在实验室测试,又可现场安装测试。
- 测试对象: 软件产品的最终形式,而不是中间形式。
- 测试方法: 基于软件操作剖面的随机测试方法。
- 测试特征: 不进行软件缺陷剔除。
- 测试方案: 根据与用户事先商定的生产方风险 α 、使用方风险 β 、MTBF 的检验上限 θ_0 、MTBF 最低可接受值 θ_1 等要求来确定。

软件可靠性验证测试同时又属于一种特殊产品的可靠性验证试验,它与已经比较成熟的硬件可靠性验证试验相比有如下特点:

- 实施环境: 是指测试时的软硬件环境,而不是温度、湿度、振动等物理环境。
- 实施成本: 软件产品的测试不存在硬件试验中样本的物理损耗及失效后零部件更换等



物理费用(当然软件产品所赖以运行的硬件产品在测试中是有损耗的),因而测试主要受与时间有关的资源限制。

- 实施机理: 软件测试是通过测试数据来激发软件中已存在的缺陷,因而测试数据的选取非常关键。
- 实施风险: 在双风险测试方案中,对软件所取的生产方和使用方风险可以比硬件低 1~2 个数量级。
- 实施结果: 软件验证符合要求后,不能直接交付,还要对验证测试中发现的缺陷更正,为避免更正中出现新的缺陷,一般还要通过无失效考核测试才能交付。

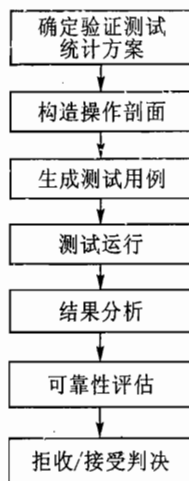


图 5.23 软件可靠性验证测试过程

(2) 软件可靠性验证测试工作内容

在整个软件工程的过程中,软件可靠性验证测试在每一阶段要做的工作如图 5.24 所示。

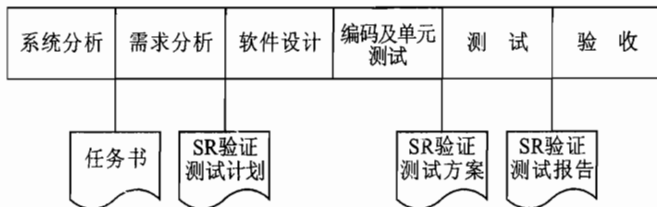


图 5.24 软件可靠性验证测试工作内容

软件可靠性验证测试统计方案主要有 3 种: 定时截尾测试方案, 序贯测试方案和无失效运行测试方案。

5. 软件可靠性管理

软件可靠性管理是对软件生存期各个阶段中的软件可靠性工程活动进行规划和控制。1992 年, AT & T Bell 实验室定义了一个最佳的 SRE 大纲中应包括的软件可靠性工程活动, 共 4 项。

1) 需求阶段的软件可靠性工程活动

此阶段的工作重点是确定产品的可靠性需求。首先定义失效类型, 并分析失效对用户的影响。失效的可能性取决于用户如何使用产品。在开发的早期阶段, 用功能剖面体现用户的使用, 它描述用户执行的高层次功能以及这些功能执行的频数。在确定整个产品功能时要进行权衡, 虽然产品特性越多, 用户越满意, 但是会增加开发费用, 特别是会增加与管理产品可靠性相关的费用。

从可靠性角度出发,此阶段的输出是软件可靠性目标值。正确地定义系统和软件需求是最重要的,这是因为需求缺陷是现场维护的主要问题。这些需求不仅要定义不同失效等级的目标值,还要定义满足目标值的条件。

综上所述,需求阶段的软件可靠性工程活动内容包括:

- 确定功能剖面;
- 定义并划分失效等级;
- 确定用户的可靠性需求;
- 进行权衡研究;
- 确定可靠性目标值。

2) 设计和实现阶段的软件可靠性工程活动

这一阶段的工作重点是开发满足可靠性要求的产品。首先将总的可靠性要求分配至产品的各个组件。可以采取若干步骤管理产品使其满足可靠性要求。有一些技术从硬件/系统可靠性方法借鉴而来,如故障树分析(FTA)、失效模式和影响分析(FMEA)。由于软件失效是由设计/开发过程中残留的缺陷引起的,因此可靠性方法必须针对的是引入缺陷并在后续开发阶段传播缺陷的软件过程。

当前软件产品中既有新的组件,也有重用的组件,对这些重用软件的可靠性也必须加以管理。

综上所述,设计和实现阶段的软件可靠性工程活动内容包括:

- 分配组件的可靠性;
- 管理产品,使其满足可靠性目标;
- 基于功能剖面分配资源;
- 管理缺陷的引入和传播;
- 测量重用软件的可靠性。

3) 测试阶段的软件可靠性工程活动

此阶段活动的目的是测试并确认软件,使其满足软件可靠性目标值。软件产品的测试必须仿真用户是如何使用的,这样才能准确地反映产品的可靠性。

测试阶段的软件可靠性工程活动内容包括:

- 确定操作剖面;
- 进行可靠性增长测试;
- 跟踪测试进程;
- 预计还要进行的测试量;
- 验证可靠性目标值是否满足要求。

在需求阶段,开发的功能剖面已经开始描述用户是怎样使用软件产品的,随着设计开发的进行,按照系统提供的完成具体功能的运行重新定义运行剖面。该运行剖面定义了运行的频



率和顺序,可用于开发测试用例。在测试期间,收集失效数据用于计算可靠性增长模型的参数,然后根据模型可以给出软件产品当前的可靠性水平,估计为了达到可靠性目标所需要的测试时间等。

4) 交付后和使用维护阶段的软件可靠性工程活动

在此阶段应继续进行可靠性管理。其中的一项工作是估计用于售后服务人员和修正软件产品缺陷人员的数量。要跟踪软件产品的现场使用可靠性,确保用户满意产品的可靠性。当发现的软件缺陷修改后,可靠性将继续增长。应该合理安排有较大改进的软件的发布时间,使得新的产品版本其可靠性也是用户可接受的。利用此阶段收集到的信息可以改进影响可靠性的开发过程,改进后续的产品版本或新的产品的可靠性。

综上所述,交付后和使用维护阶段的软件可靠性工程活动内容包括:

- 预计售后人员的需求量;
- 监控现场使用可靠性,并与目标值进行比较;
- 跟踪用户的可靠性满意程度;
- 安排引入新的软件特性;
- 指导软件产品和过程的改进。

5.7 下一代飞行器研制中的软件系统测试问题

5.7.1 关于嵌入式软件测试环境的思考

本节主要探讨嵌入式软件的系统测试问题,因为对单元、集成、配置项测试的方法在5.3.4小节中已有阐述。

对飞行器嵌入式软件的系统测试历来有两种不同的观点。

一种观点是特别强调测试环境的“真实性”,认为软件的系统测试应在“实况试验”环境中进行,系统测试人员坐在“真实”的飞行员座舱内,所测试的全部航电系统以及发动机控制系统、非电系统(如电源等)全部是真实的产品,并以真实的 I/O 交联在一起。其优点是可以逼真地反映出真实飞行器可能发生的各种事件,但缺点也是明显的:

- 构建这样一个全实物测试环境费用昂贵,而且当一个被测软件在进行测试时,它的所有交联设备都要与它一起工作,试验件的实物损耗很大,测试代价高昂。
- 由于所有的交联系统全是实物,所以在进行异常情况测试、强度测试等需要施加高负载和超负载的测试时,会对系统产生破坏,甚至发生危险。若为了避免这种情况发生,则不少测试用例无法执行。据我国的经验,在这种环境下的系统测试,将有 20%~30% 左右甚至更高比例的测试用例不能执行,亦即 $T_s \leq 0.80$ 。

另一种观点认为对嵌入式软件的系统测试最好在半实物仿真测试环境下进行。这种半实物仿真是指嵌入式软件运行采用真实硬件环境,而对于被测软件与外界环境及其他设备之间

的输入/输出则通过软件进行仿真。半实物仿真测试具有较高的逼真度。这是因为嵌入式软件的运行环境和接口都采用了真实的硬件设备,所以对于测试中系统的输入/输出信号保持了其原有的物理特性,这一点比全数字仿真更接近真实情况;同时,采用软件方式模拟系统输入信号的逻辑特性,使得半实物仿真比全实物仿真具有更加灵活的测试可控性,而且减少了采用硬件仿真造成的浪费。

我国不少软件测评中心,应用中航一集团公司计算机软件可靠性管理与测评中心自主开发的 GESTE 测试环境进行航空、航天等实时嵌入式软件的系统测试,测试用例的执行率 T_e 达到 95%~100%。

我们也许无法从理论上证明上述两种方法何种为佳,但“实践是检验真理的唯一标准”,下面以两个实际的飞行器为例,进行比照:

(1) 美国的 F-22

如所周知,F-22 的软件系统测试费时长(1998—2002 年),投入大(采用了航电综合实验室、飞行试验平台和 6 架专门进行航电系统试验的试验机),且是在“实况试验”的环境中进行的系统测试,但是测试效果并不佳。除了 5.6.1 小节已提到的在所有试验完成后,其生产型的 F-22 发生了由于飞控软件故障引起坠机外,还在 2007 年 2 月 12 架 F-22 从夏威夷飞往日本途经国际日期变更线(东经 180°)时,飞机上的多个电脑系统发生崩溃,多次重启均告失败,飞行员不得不掉头返航,折回到夏威夷的空军基地。

其原因就是 F-22 强调的是“实况试验”,而在长达 5 年的试验中,真实飞机并未飞越过东经 180° ,所以软件中潜藏的错误一直没有被发现。

(2) 我国的某型号飞机

无独有偶,我国的某型号飞机在进行的软件系统测试中也发现了这一问题。该型飞机在自西向东“飞越”东经 180° 时,飞机不能按照装订的航线正常飞行,而是立即掉头向西返航。但是,我们是先于美国 F-22 发生事故的 2 个月前,即 2006 年 12 月在地面半实物仿真测试环境中发现该软件错误的,当时的测试记录如表 5.18 所列。

表 5.18 某型号飞机的软件系统测试记录

问题序号	用例编号	缺陷描述	问题类型	问题等级	阶段	修改情况
1	T17-02-GN05	航线飞行中,由西向东经过东经 180° 时,遥测中显示的待飞距错误,飞机不能按照装订的航线正常飞行,而是掉头向西飞行	需求问题	关键	第一次测试	已经修改
2	T17-02-GN06	在计算大地坐标系到平面坐标系的转换时,没有考虑东西经 180° 是同一条经线的情况,坐标系转换错误导致待飞距计算错误				

在半实物仿真测试环境下,可以让软件在测试环境中经历各种情景及各种超负荷的非正常情况,从而尽可能多地让软件经历“所有可能的失效”,并暴露出软件中潜藏的缺陷。

因此,笔者建议的方法是:上述两种方法的结合,即首先进行地面的半实物仿真测试,让软件在仿真测试环境中经历“所有可能的失败”,然后在分析、排除软件的缺陷后,进入定型试飞阶段,让飞行器在真实的“实况环境”中进行试验(因为地面的半实物仿真不可能100%模拟真实飞行器的全部情景)。两者相结合的测试方法应当会显著提高试验的效果和效率。

5.7.2 模型驱动的软件仿真测试技术

模型驱动的软件仿真测试方法(Model-Driven Simulating Testing, MDST),是一种在下一代先进飞行器的软件测试中应用的方法。

模型驱动的仿真测试是将模型驱动体系结构(Model-Driven Architecture, MDA)方法与软件仿真测试结合,整个测试活动围绕建模开展,从测试需求分析,到测试用例生成,以及测试用例执行,最后是测试结果分析,处处都以模型为指导。MDST测试兼有传统软件仿真测试和模型驱动的软件测试(Model Driven Testing, MDT)的优点,在实时嵌入式软件测试领域有着良好的应用前景。

将MDA方法应用到软件测试过程中,将带来测试质量和测试效率的提高。模型驱动的软件测试的目标是通过支持测试资源的重用来缩短测试时间。另外,模型可以为被测系统及测试本身提供一个清晰、明确、统一的视图。更重要的是,模型驱动测试将测试逻辑和测试实现分开。因此,测试开发者就可以专心于详细地设计测试用例,将测试执行留给测试工具来自动完成,从而使开发者从繁重的测试工作上,如创建测试脚本,回收测试结果,编写测试报告中解放出来,避免了人工测试劳动所导致的人为错误。

模型驱动测试将测试逻辑与具体的测试实现相分离,从而使得测试人员更加关注于测试逻辑的设计,而具体的测试实现以及测试执行完全由模型驱动测试工具来完成。借助于模型驱动的测试方法和技术,测试模型可自动转化为最终可执行的测试脚本,这样就将测试脚本的维护转化为对测试模型的维护,当程序发生变化时,测试人员只需要对测试模型进行修改,然后通过模型自动生成测试脚本即可,从而降低了测试脚本的维护费用。同时,模型驱动测试可以有效地支持回归测试。由于测试模型易于维护和修改,当开发人员提交了一个新的版本之后,测试人员可以利用原有的测试模型对其稍加修改,对新系统进行快速的回归测试。

模型驱动测试为测试人员提供了简便、有效的方式,以实现高效、自动化的测试方法和技术。模型驱动测试通过测试模型描述测试的结构、行为和测试数据等众多测试因素,从而简化测试的开发,提高测试的效率。整个模型驱动的测试过程以测试建模活动为核心,测试人员创建测试所需要的静态架构、动态行为、测试数据等测试模型,然后利用这些测试模型实现测试行为的自动控制和测试脚本的自动生成。

1. 仿真测试原理

采用仿真策略进行测试是仿真测试框架的基本特征之一,也是最重要的基础。在框架原则层的原理解释中,首先对仿真测试原理进行分析。对仿真测试原理解释,主要包括仿真测试的目的、仿真测试的本质、仿真的对象和仿真的方式4个方面。

1) 软件测试中仿真的目的

仿真是真实过程或系统在整个时间内运行的模仿。在研究、分析系统时,对随时间变化的系统特性通常是通过一个模型来进行研究。凡是利用计算机在模型上而不是在真实系统上进行实验、运行的研究方法,都可认为是仿真。仿真的开展离不开模型,事实上,仿真就是采用模型来再现真实情况,而模型是系统、过程或现象的物理、数学或其他逻辑的表达。

仿真方法在软件开发,尤其是嵌入式软件开发和验证中已经得到了广泛的应用,所以将仿真引入到嵌入式软件的测试领域也就成为顺理成章的事情。而采用仿真的方法进行嵌入式软件测试,也的确具有真实使用环境中运行测试所不具有的许多优势。

- ▶ 减少测试的费用: 仿真测试中利用模型支持测试,大大减少了因开发硬件环境和交联设备等给测试带来的巨大经济开销。
- ▶ 缩短测试时间: 利用模型支持测试,可以方便地对模型进行修改和控制,减少了真实环境中因为对硬件环境的改动或对交联设备的运行控制而导致的过长的测试时间。
- ▶ 提前测试的进行: 仿真测试中的模型可以与被测软件的开发同时进行,这样在被测软件的硬件运行环境或者其交联设备没有完成前也同样可以进行测试,提前了测试开展的时机。
- ▶ 降低测试的风险: 仿真测试中模型的使用大大降低了在真实环境中测试可能造成设备损坏甚至人员伤亡的概率,将软件测试的风险降到最低。
- ▶ 提高测试的可重复性: 仿真测试中模型的使用解决了回归测试中的重复测试问题,使得完成的测试可以通过模型记录下来,避免了真实环境中运行测试的随机性。

由此可见,在嵌入式软件测试中,仿真方法是一个必不可少的测试手段。

2) 仿真测试方法的本质

在嵌入式软件测试中使用仿真方法由来已久,但是还没有对仿真测试的本质进行过系统分析。仿真测试就是通过使用软件和硬件的方法,模拟被测软件的交联系统及其物理信号的输入、输出,仿真一个被测软件运行的真实环境,并在该环境下进行各种类型测试。作为一种方法,仿真测试具有完整的体系。

(1) 基本概念

仿真测试中涉及以下主要概念:

- ▶ 运行环境: 被测软件在实际使用中所处的软件和硬件环境。
- ▶ 测试环境: 被测软件在测试过程中所处的软件和硬件环境。在仿真测试框架中,因为通常采用仿真的手段构建测试环境,所以本文可称之为仿真测试框架。



- 仿真模型：测试中利用软件算法实现对被测系统输入信号逻辑特性进行模拟的软件实体。
- 仿真原型：仿真模型所模拟的运行环境中的软件实体、硬件设备或是外界物理环境。
- 仿真测试：通过仿真模型建立嵌入式软件的测试环境，并在此环境下进行的测试。

(2) 基本公理假设

基本公理假设是仿真测试方法的前提和基础，这些假设并不能对其进行严格的证明，但它是经过实践证明的正确命题。

仿真测试的基本公理假设：

- ① 独立性假设：对软件的仿真测试不改变被测软件的固有属性。
- ② 环境同构假设：进行仿真测试所使用的测试环境，必须满足被测软件的固有属性在测试环境中的信息表现与在真实使用环境中是等价的。
- ③ 缺陷本质假设：软件缺陷是软件的固有属性之一。
- ④ 有效性假设：测试环境中发现的软件失效，在真实使用时，并且在一定条件触发下必定会出现。

第①条假设是软件测试的基本前提，它说明仿真测试将软件整体作为一个系统看待，而软件的测试过程是对软件系统的认识过程，是以从系统的外部环境的角度进行测试，是一种独立于被测软件客体的测试手段。这条假设也是其他假设的基础。

第②条假设说明测试环境与真实运行环境之间是同构的，即采用测试环境模拟软件真实运行环境这种做法的合理性是因为它们对于从被测软件获得信息的角度来看是等价的。

第③条假设说明了软件缺陷的性质。软件是复杂的人脑产物，软件的产生要经过分析、设计和实现等若干过程，同时也造成了软件的缺陷。软件缺陷同结构、功能、性能等一样是软件的固有属性之一。

第④条假设说明了测试结果的合理性，在前两条假设的基础上，可以通过三段论推理得到。因为软件缺陷是被测软件的固有属性，而根据第②条假设即可得到软件缺陷的真实性。

(3) 仿真测试的基本特性

上面的基本假设奠定了仿真测试的基础，仿真测试利用仿真方法构建测试环境对嵌入式软件进行测试。这种方法具有以下主要特性：

- 可仿真性：由仿真测试的定义可以看出，仿真测试中“仿真”的概念主要体现在被测软件的测试环境方面。从仿真测试的基本假设中知道，测试环境虽然是一种仿真环境，但是它应该在得到被测软件测试信息方面与运行环境是同构的。
- 测试性：由于测试环境是对运行环境的仿真，因此一方面测试环境与其原型之间保持着一定的相似性；另一方面测试环境要具有一定的测试性，能够灵活地实现嵌入式软件测试所要求的对被测软件输入的可控制性。
- 可替换性：由于测试环境具有仿真性和测试性，因而使得仿真测试方法在对被测软件

的测试环境方面具备了代替真实运行环境的能力。在测试过程中,仿真测试得到的测试结果与运行测试或者被测软件实际使用中的结果应该是一致的。

3) 对系统环境的仿真

(1) 相关概念

从仿真测试方法的假设中可以看出,被测试的软件是一个独立的系统,无论是仿真测试中的测试环境,还是实际的运行环境,都是被测软件这个系统的外部环境。因此在分析“仿真的对象”这个问题之前,首先应明确几个与“系统”和“环境”有关的概念。

① 系统的环境

一个系统之外的一切与它相关联的事物构成的集合,称之为系统的环境。更确切地说,系统 S 的环境 E 是指 S 之外一切与 S 具有不可忽略的、联系的事物集合,即

$$E_s = \{x \mid x \in S \text{ 且与 } S \text{ 具有不可忽略的联系}\}$$

定义中的“不可忽略”是一个模糊用语,不能作非此即彼的理解。上面的式子表明,系统的环境只能在相对的意义上确定。在不同研究目的下,或对于不同的研究者,同一系统的环境划分也有所不同。

环境复杂性是造成系统复杂性的重要根源。因此对系统的研究必须研究它的环境以及同环境的相互作用。环境意识是系统思想的一个基本点。

② 系统的行为

系统相对于它的环境所表现出来的任何变化,或者说,系统可以从外部探知的一切变化,称为系统的行为。行为是系统自身的变化,是系统自身特性的表现,但又与环境有关,反映环境对系统的作用或影响。

③ 系统的功能

功能是刻画系统行为,特别是系统与环境关系的重要概念。系统的任何行为都会对环境产生影响。系统行为所引起的有利于环境中某些事物乃至整个环境存续与发展的作用,称为系统的功能。

④ 系统的性能

系统的性能是指系统在内部相干和外部联系中表现出来的特性和能力。

⑤ 系统功能与环境的关系

系统环境对于目标系统的研究起到了至关重要的作用。虽然系统的功能与系统结构关系密切,但系统的功能是由结构和环境共同决定而非单独由结构决定的。系统功能的发挥需要环境提供各种适当的条件、氛围。为充分发挥系统功能,需适当选择、营造、改善环境。只有当环境给定后,才可说结构决定功能。

如果仿真测试中的待测试软件被看作一个系统,那么它所依赖的硬件运行环境、交联设备以及外界物理环境就构成了被测软件系统的环境。对真实运行环境的仿真实际上就是对软件系统环境的仿真,测试中被测软件的所有行为和功能体现都是通过测试环境驱动的。



由上面系统环境的相对性可知,不同的测试方法可能会有不同的测试环境的构造。下面对测试环境的分类就体现了这一点。

(2) 测试环境的分类

根据对软件运行环境仿真程度的不同,仿真测试可以分为全数字仿真、半实物仿真和全实物仿真。通过比较研究发现,半实物仿真是一种最优方案。

如图 5.25 所示,嵌入式测试的半实物仿真是指嵌入式软件的运行采用真实硬件环境,而对于系统与外界环境及其他设备之间的输入/输出则通过软件的方式进行模拟。半实物仿真测试具有较高的逼真度。这是因为嵌入式软件的运行环境和接口都采用真实的硬件设备,所以对于测试中系统的输入/输出信号保持了其原有的物理特性,这一点比全数字仿真更接近真实情况;同时,采用软件方式模拟系统输入信号的逻辑特性,使得半实物仿真比全实物仿真具有更加灵活的测试可控性,而且减少了采用硬件仿真造成的浪费。

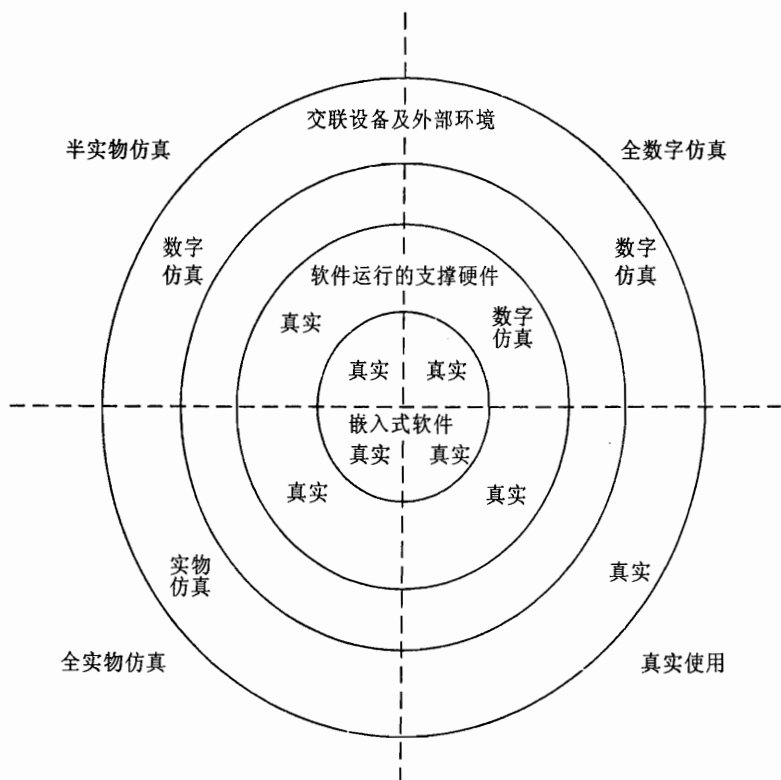


图 5.25 仿真测试的类型

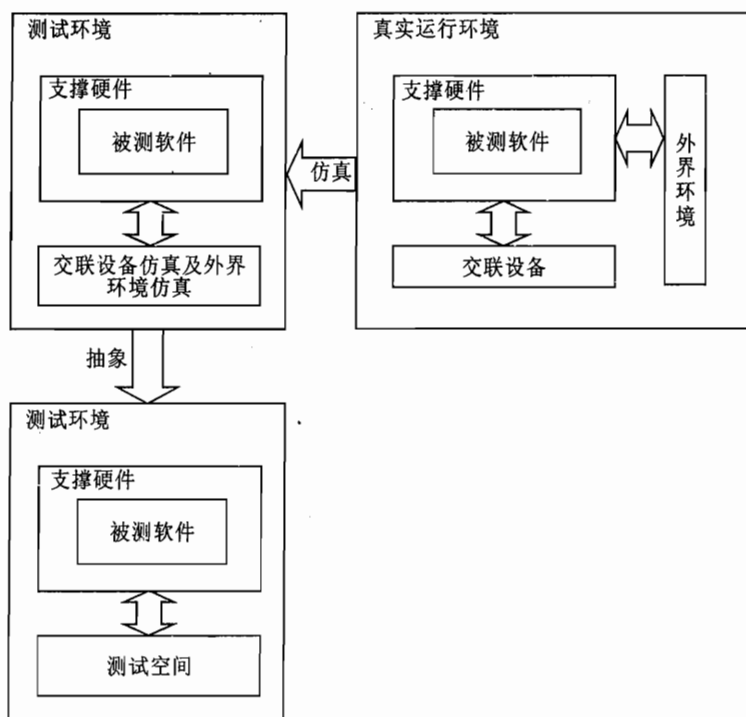
下面的讨论中,“仿真测试”一词将特指半实物仿真测试。

4) 对环境仿真的抽象

对于嵌入式软件运行环境的半实物仿真,具体说来就是对被测软件的交联设备和外界物理环境的仿真。而从环境对软件系统的作用来看,本文可以将半实物仿真抽象为对系统输入信息的仿真。计算机系统是一个离散系统,所以软件系统的输入具体是指输入变量,输入信息就是变量的相关信息。

(1) 相关概念

如图 5.26 所示,对真实运行环境的仿真最终抽象为软件的测试空间概念。下面将首先明确几个概念。

图 5.26 对环境仿真的抽象^[24]

① 输入变量

输入变量是任何存在于系统外部并影响此系统的数据元素。由程序计算出来的,同时并不存在于程序外部的中间数据不是输入变量。输入变量的概念是逻辑上的,并非物理上的。

系统所有的输入变量构成的集合是系统变量集合 V 的一个子集,记为:

$$V_1 = \{v_1, v_2, \dots, v_n\}, n \in N, \text{ 并且 } V_1 \subseteq V$$

输入变量集合是一个测试对象范畴的概念,它随着测试对象的指定而确定。



② 输入状态

输入状态是系统输入变量取值的完整集合。每个输入变量的实际取值按照其输入顺序/时序排列起来,构成输入空间中的一个输入状态。

③ 输入空间

系统所有可能的输入状态集合称为输入空间。每个软件客观上都对应着一个输入空间(又称输入域)。输入空间通常被认为等于软件输入值的取值范围(包含合法的和非法的取值)的笛卡尔积。笛卡尔积可能会产生非法的输入状态。软件的输入空间包含合法的输入空间和非法的输入空间。一般来讲,软件对于输入空间中的任何一个输入状态都应该做出正确响应,不管该输入状态是合法的还是非法的。对于非法的输入,软件应该正确地排除其干扰。

④ 测试空间

与输入变量和输入空间对应还定义了输出变量和输出空间的概念。为了方便讨论,将输入空间和输出空间统称为软件的测试空间。

(2) 利用测试空间表达测试环境

将交联设备和外界环境的仿真用测试空间的概念进行表达,目的是可以借用数学知识对仿真测试中的诸多方面进行讨论,增强仿真测试的精确性和形式化程度。例如通过数学形式表述测试空间可以方便地利用概率知识进行空间的分析,为测试用例的设计和讨论测试的揭错概率和充分性等问题奠定了基础。

另外,测试空间概念的引入为半实物仿真中的数学仿真部分提供了依据。测试空间中描述了软件系统输入/输出变量的各种信息,这些变量是连接测试环境与被测软件之间的桥梁,它们具有物理和逻辑两种特性。物理特性是指变量传输所经过的硬件信息,测试中这种特性是通过真实的 I/O 接口以实物仿真的形式来实现的。变量的逻辑特性是指变量的产生机理、传输的时间要求以及变化规律等信息,这些特性是在测试中通过数学仿真来完成的。

2. 模型驱动的软件仿真测试

1) 测试模型的定义

模型是指一个系统、实体、现象或过程的物理、数学或其他逻辑的表示形式,是人们认识事物的一种概念框架。也就是用某种形式近似地描述或模拟所研究的对象或过程。在仿真领域构造模型是为了研究原型。客观性、有效性是对建模的首要要求,反映原型本质特性的一切信息必须在模型中表现出来,通过模型研究能够把握原型的主要特征。另外,模型的主要功能是提供一个框架,能够适当的整理和组织观察数据、资料、信息,对原型系统的行为特征和运行演化规律作出解释。

在仿真测试框架中,我们引入测试模型的概念:以某种形式描述的测试解决方案。从定义可以看出,测试模型具有以下基本特征:

- 测试模型与测试解决方案有关。测试模型是对测试解决方案进行模块化和组件化表示的一种组织形式,它是提高测试框架可扩展性和可维护性的重要手段之一。

- 测试模型是对相关测试知识的描述。测试模型的内容是对解决方案的某一方面知识的描述,与模型定义一样,这种知识可以是一个实体,也可以是一个过程,甚至是一种思想。
- 测试模型的形式不固定。任何一种模型都具有某种形式,一般称之为模型语言。在仿真测试框架中,测试模型没有一种固定的标准语言,不同类型的测试模型可以有不同的描述形式。

2) 测试模型的分类

对测试模型的分类可以有多种依据,既可以按照功能分类,也可以按照建立的阶段分类。本节按照功能分类对各种测试模型进行介绍。

按照测试模型的功能可以将其分为测试用例模型、测试环境模型和软件缺陷模型。

(1) 测试用例模型的主要作用是辅助生成测试用例。测试用例模型是一个模型包,包括的内容有被测软件模型、测试空间模型和用于测试用例选取的策略模型以及具体的测试用例。被测软件模型是根据测试的目标不同,对被测软件的不同方面进行表示的一类模型,例如功能模型、性能模型、使用模型等。测试空间模型是根据被测软件模型生成的一种带有被测客体信息的、用于测试的模型,它是产生测试用例的一个基础模型。测试策略模型是测试主体选用的选取策略可以根据实际情况进行修改和更换,是测试主体测试意图的体现。利用测试策略模型和测试空间模型可以进行测试用例的设计与选取。测试用例模型是仿真测试框架中的一类重要模型。

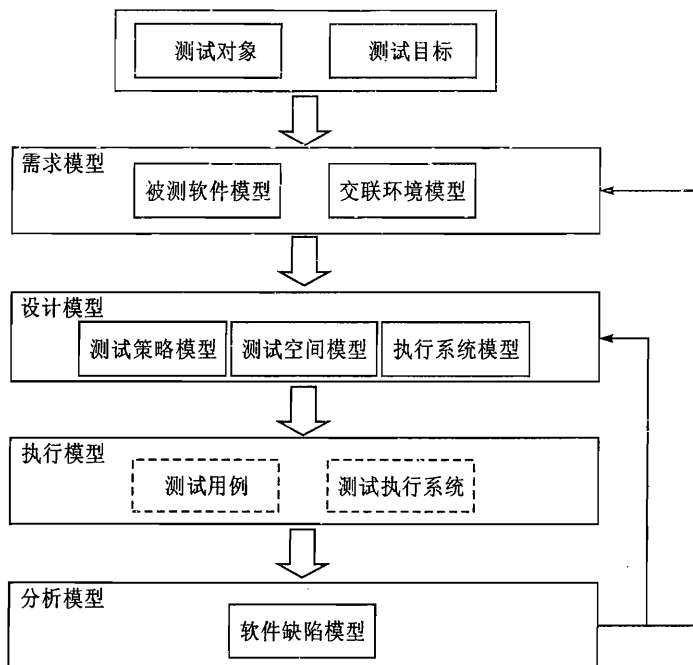
(2) 测试环境模型的主要作用是辅助构建仿真测试的执行环境。测试环境模型也是一个模型包,包括被测软件的交联环境模型和测试执行系统模型以及根据模型实现的测试执行系统。交联环境模型是根据被测软件的实际使用环境,利用仿真的方法构建的一个模拟真实环境。交联设备模型中包括被测软件与外界环境和交联设备之间的输入/输出关系以及输入/输出数据的逻辑特性和物理特性。测试执行系统模型是构建的仿真测试环境的运行时系统模型,测试执行系统中提供了仿真模型运行所需要的任务调度、数据处理和数据通信等重要功能组件,最终的测试是通过仿真模型在测试执行系统中的运行来完成的。

(3) 软件缺陷模型的主要作用是记录测试发现的软件失效,同时用来指导测试用例的选取与生成。软件缺陷模型的第二个作用尤其重要。根据对软件测试的认识论解释,对软件的测试过程就是对被测软件的缺陷属性的认知过程,可以说,软件缺陷模型是对被测软件本身属性的建模,只是这个模型所反映的信息是通过测试过程不断发现和补充的。软件缺陷模型中应该记录软件失效(或缺陷)与测试用例之间的某种关系,使得它可以用来对测试用例的策略进行信息反馈,提高发现错误的能力。软件缺陷模型是仿真测试过程中的一个反馈点。

3) 模型驱动的涵义

模型驱动的涵义可以通过按测试阶段对测试模型的分类图进行说明,如图 5.27 所示。

模型驱动的涵义简单说来就是将测试过程理解为创建模型及模型演化的过程。模型驱动

图 5.27 模型驱动测试的涵义^[24]

的原理正是软件测试认识论的体现,模型的创建过程就是测试主体对被测软件逐渐深化认识的过程,所以软件测试的认识论解释是模型驱动测试的理论基础,同时,模型驱动测试也是软件测试认识论在技术层次的必然反映。

从图 5.27 中可看出,仿真测试框架中将测试过程分为需求、设计、执行和分析 4 个主要阶段,每个阶段都由构建某类测试模型的活动构成,测试过程的推进是通过测试模型之间的演化来完成的。

(1) 测试需求阶段首先要根据测试对象和测试目标建立被测软件模型和交联环境模型。这两类模型是对被测软件的相关属性建立的信息模型,属于测试客体软件信息场的一部分。需求阶段的测试模型是整个测试过程的初始模型,是驱动后续测试模型建立的原动力。

(2) 测试设计阶段建立的模型包括测试空间模型、执行系统模型和测试策略模型。这 3 类模型是在需求模型的基础上演化而来,它们属于测试方法和测试工具的范畴。测试空间模型和测试策略模型是根据被测软件模型中表述的测试属性以及交联环境模型中的输入/输出信息而建立的为生成测试用例服务的模型。其中测试策略模型集中体现了测试者的测试知识和测试经验,是测试主体认知结构的主要体现;执行系统模型则主要依据交联环境模型而建立,是构建测试环境的主要依据。

(3) 测试执行阶段似乎没有测试模型的直接参与,但是从某种程度上讲,测试用例和测试执行系统都可以看成是一种实体模型。

(4) 测试分析阶段要根据测试结果建立软件缺陷模型。缺陷模型是测试结果的信息模型,它反映测试认识过程的迭代性,是深化认识过程的增益环节。仿真测试框架中缺陷模型的建立集中体现了测试认识论中重视测试客体特殊性的思想。

4) MDST 过程

模型驱动的软件仿真测试(MDST)是通过各类测试模型的建立和演化过程来驱动完成的。

MDST 简单说来是这样的一个过程:

- 观察被测系统的行为;
- 将系统行为经形式化描述后,提供一个测试生成程序;
- 让测试生成程序来创建和运行测试用例;
- 由测试结果分析工具自动对比测试执行结果和系统的预期输出,对软件质量进行评估。

模型驱动的软件仿真测试过程是一个测试模型的设计和创建的过程,如图 5.28 所示。

MDST 主要包括如下活动:

(1) 交联环境建模

采用基于接口的仿真建模技术,用接口来确定模型的结构,解决模型的框架设计问题和接口的通信问题。

(2) 测试需求建模

测试人员首先需要确定被测系统和测试目标,即测试需求描述。在进行测试模型的创建之前,必须确定软件系统的哪些组件或子系统是需要被测试的,即被测系统(SUT)功能建模;哪些组件或子系统是对被测系统测试所必需的,即 SUT 的测试环境建模。

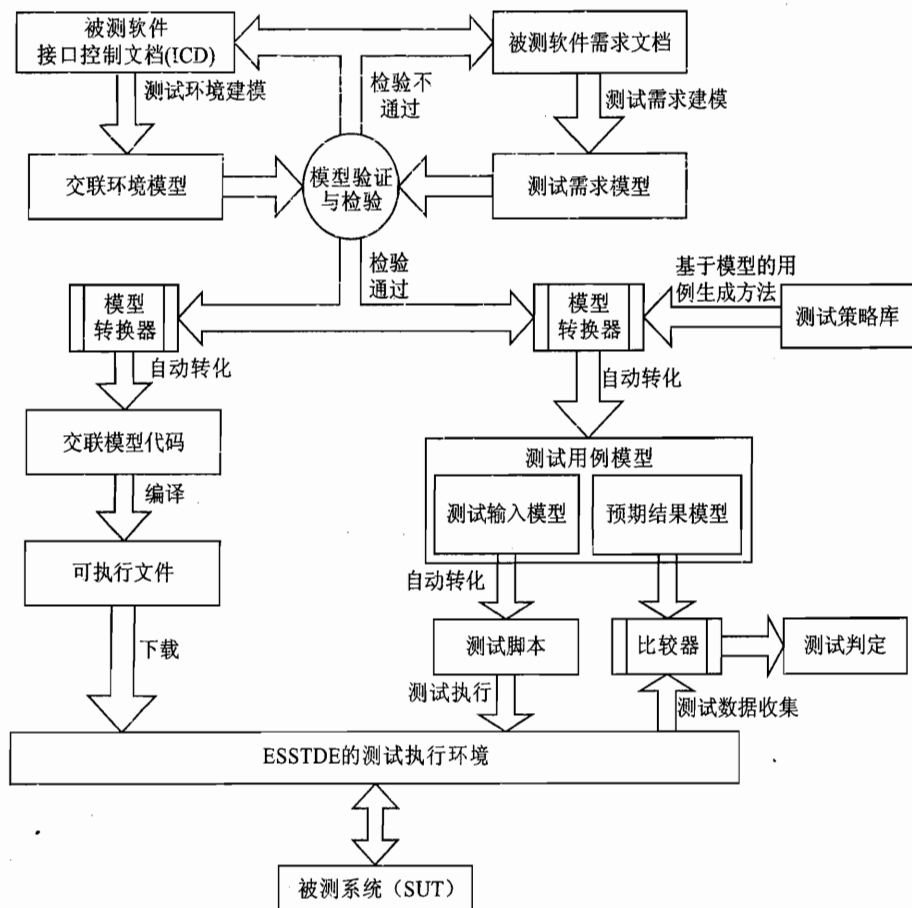
在确定了测试需求之后,MDST 的下一步就是进行测试模型的设计,确定测试架构的描述和测试行为描述。在测试架构模型的描述中,需要确定软件测试中涉及的所有测试组件以及它们之间的通信关系;在测试行为模型描述中,通过测试架构模型中的测试组件描述被测系统行为,进而实现测试用例模型的描述。测试模型可以手工设计,对于具有详细设计模型的软件系统来说,还可以通过重构、扩展开发阶段中的设计模型而获得。

(3) 对测试模型的校验和验证

模型必须能正确地反映系统的静态特性和动态行为,因此必须对模型进行验证和确认。现有的模型验证方法通常有观察法、专家面试法、灵敏度分析法、参数验证法、统计方法、模型比较法、模型分离法、静态一致性检验法和动态一致性检验法等。

(4) 根据测试需求模型自动生成测试用例

自动生成测试用例是 MDST 的关键功能。可引入测试策略库,采用基于模型的测试方法自动生成测试用例。

图 5.28 模型驱动的软件仿真测试流程^[24]

(5) 用例的自动执行

MDST 支持测试的自动执行。MDST 方法能够支持测试用例模型到测试脚本的自动转化,从而避免了测试脚本的开发,提高了测试效率并减少了测试人员的工作量;同时,MDST 方法将对测试脚本的维护转化成对测试模型的维护,从而减少了测试维护的费用。

(6) 测试结果的收集与分析

测试执行结束后,从各个测试组件或测试机器上收集测试执行结果,并为测试执行的判定与跟踪提供测试判定模型和测试跟踪模型,以便于对测试的结果进行比较和分析。

利用模型驱动的仿真测试原理和方法,我们构建了一个分布式实时仿真测试系统,可以对下一代飞行器装备的大规模复杂软件的被测系统(如数据高度融合的通用综合处理器 CIP)进行软件系统测试(见图 5.29)。

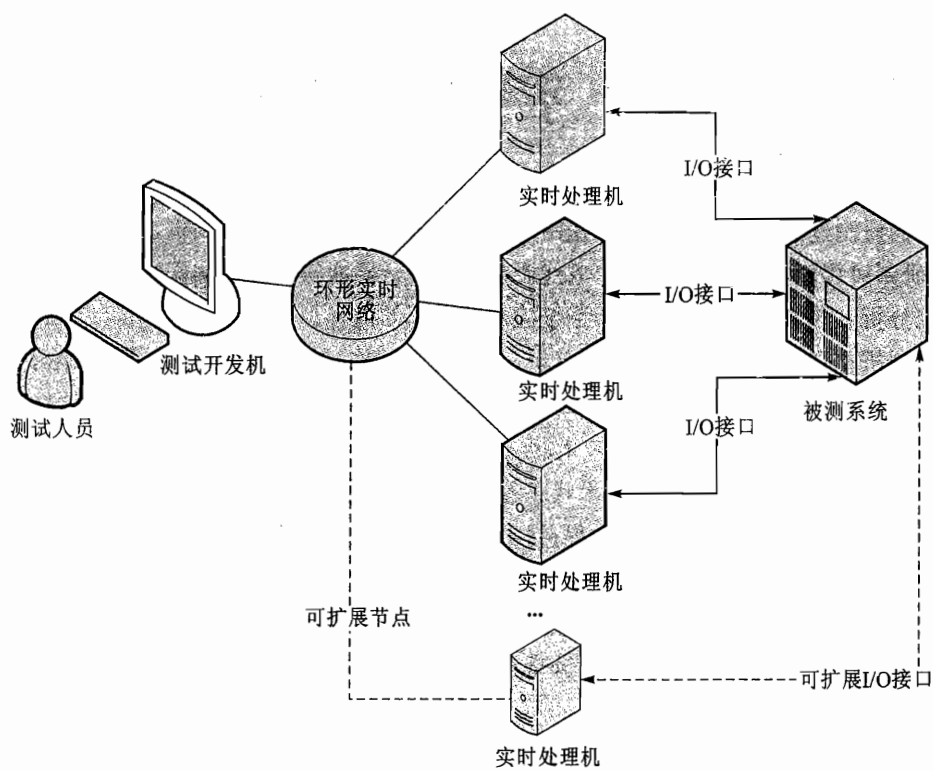


图 5.29 分布式实时仿真测试系统的整体框架结构

第6章 系统分析和控制

飞行器研制过程中的系统分析和控制是使飞行器研制过程从输入到输出的可追溯性得以保持,其研制过程能得到控制,以确保最终研制出的飞行器能符合用户的需求。

系统分析和控制包含系统分析和系统控制两方面的内容。

系统分析包括权衡分析、体系结构分析(工作分解结构)和效能分析。它用来评价备选方案及其技术途径是否能恰当地(如果不是最优)满足系统要求和指标,为选择最终的体系结构和设计方案提供依据。

系统控制包括技术性能测量、技术状态控制和技术评审。其中,技术性能测量是对飞行器系统的技术性能进行控制和验证的过程;技术状态控制是对已确立的基线进行控制;技术评审则是对飞行器研制各阶段实施里程碑控制的重要手段。

鉴于效能分析在前述章节中已有阐述,技术状态控制和技术评审早已为广大飞行器设计和管理人员所熟知,故本章将阐明权衡分析、体系结构分析(工作分解结构)和技术性能测量这3部分内容。

6.1 权衡分析方法

6.1.1 权衡分析的方法和步骤

飞行器设计中的权衡分析的方法如下:

(1) 首先要确定权衡的目标。

(2) 设想实现该目标的各种可能途径,亦即提出可供选择的各种候选方案。在每个候选方案中又可提出许多子方案,即在分系统或部件一级的更小范围内加以权衡。可以利用权衡树清晰地表达出系统间的相互关系,并展示如何经各种权衡而实现总体方案的途径。

(3) 根据已确定的目标制定出评判候选方案优劣的准则和各个方案必须满足的约束条件,并据此评判准则给每个候选方案评分。

(4) 在评分中,可以按照准则中各特征参量的不同重要程度给予加权,并得出加权后的加权评分。

(5) 将这些候选方案及加权评分结果排出等级,并列成权衡表,以供选择。

(6) 鉴于在加权与评分中带有一定的主观性,因此必须进一步分析各方案对已定准则中各特征参量的灵敏度。

(7) 根据加权总评分和灵敏度分析的结果选出符合目标要求的最优方案。

图 6.1 形象地给出了用上述方法进行权衡分析的示意图。

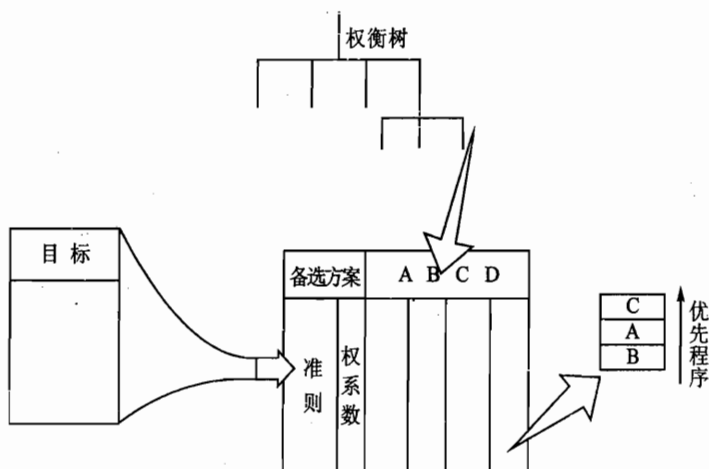


图 6.1 权衡分析示意图

权衡研究方法为评估各个备选的系统概念和设计方案提供了一个结构化的分析模式。图 6.2 所示的权衡研究过程的流程图就是典型的进行权衡研究的基本框架。

下面根据图 6.2 给出的权衡研究流程图,进一步阐述权衡研究的各个具体步骤。

(1) 确定目标和要求

应当以准确、明白的术语确定权衡研究的目标和要求,并以此作为正确决策的基础。同时也作为制定择优准则和加权系数的依据。

权衡研究的目标和要求应当来自使用部门或用户给定的任务需求和约束条件。这些任务需求经过系统工程过程已转化为功能框图、技术要求分配单和系统规范等文件。这些文件则是确定决策准则和权衡目标的主要基础。

作为权衡研究的例子,某军事部门要求研制一种能快速、经济地进行远距离军用物资运输的运载工具。目标要求是:

- 运输距离为 5 000 km;
- 运货量为 350 t;
- 从接到命令到把货物运送到目的地必须在 48 h 内完成;
- 在保证达到上述要求的前提下,力求减少执行任务的费用。

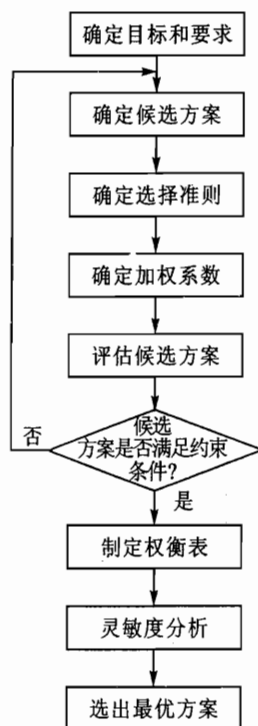


图 6.2 权衡研究流程图

(2) 确定候选方案

确定候选方案的第一步就是根据目标要求提出广泛的、各具特色的设计方案和概念。应当充分发挥设计人员的积极性,要求每一个设计者充分发挥自己的创造才能,大胆设想各种能达到目标要求的设计方案。实践证明,设想的、可能的候选方案越多,从中找到一个最优方案的可能性也就越大。

为了能在候选方案的形成中考虑到所有可行的途径,可以借助权衡树来形成各种候选方案。权衡树是一种树形结构图,每根树枝代表一种方案。它可以简明而又形象地确定达到给定目标的所有可行的途径。图 6.3 表示的是为了达到快速、经济地进行远距离军用物资运输目的而采用的一种运输工具的权衡树^[12]。

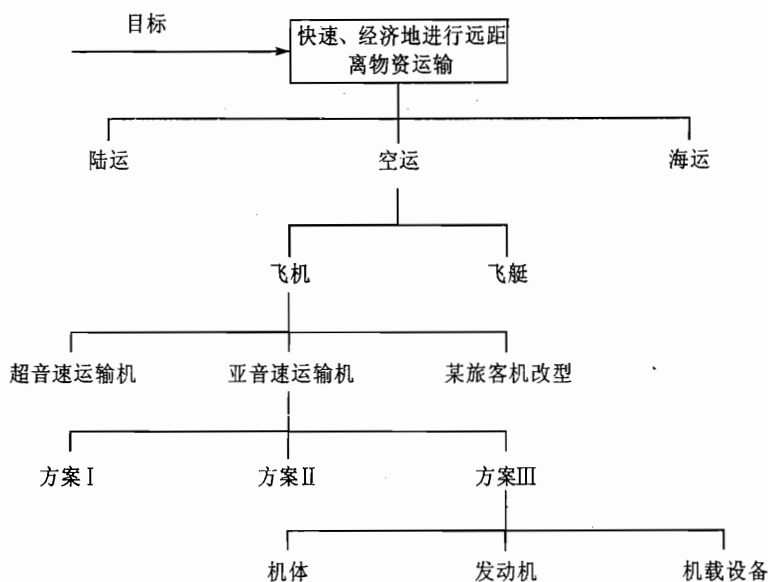


图 6.3 一种运送军用物资运输工具的权衡树

在图 6.3 所示的权衡树上,列出了达到目标的各种可行途径(包括陆运、空运、海运等),但其中某些方案(途径)很快即可排除。例如,根据目标要求,应在 48 h 内从 5 000 km 外运送军用物资,因此要求该运输工具的速度至少达到 $V=104 \text{ km/h}$,故海运和空运中的飞艇方案即可排除。陆运中的载重汽车车队或火车在高速行驰中可以达到这一速度,但在 5 000 km 的长途行驰过程中,需要补充燃料和冷却用水等,再考虑到中途可能遇到的交通堵塞及其他障碍,故陆运方案也可排除在外。最后只剩下空运中的飞机运载方案。

通过权衡树的初步权衡可以排除掉相当一批方案,剩下的可行方案就成为供决策的候选方案。这些方案必须是足够详细的,以便能通过权衡研究判断出每个可达到目标要求的候选方案的相对价值。

(3) 确定选择准则

选择准则是判断候选方案是否能达到目标要求的标准,是选择、比较候选方案的依据。这些准则应当包括定量的指标值。若有可能,还应包括这些指标的门限值(超出门限值的方案不能满足目标要求)。

必须慎重地确定准则,一个合理的选择准则应当是:

- 能在不同的候选方案之间,无偏向地并且有意义地加以区分;
- 与权衡研究的目的直接相关,亦即应包括已确定的重要任务要求和各方(使用方、承制方、政府)都极为关注的关键问题;
- 能尽可能以定量的方式明确给出;
- 能以合理的费用予以测定和估算;
- 在所有层次上,彼此之间都是相互独立的;
- 能为评估人员广泛了解和掌握。

一般,选择准则可以从下述主要的设计文件中选择并确定:

- 研制任务书或战术技术要求文件;
- 系统规范(或系统性能要求文件);
- 试验与鉴定总计划;
- 技术性能测量计划;
- 寿命期费用管理计划;
- 零级总进度表;
- 风险管理计划等。

通常,典型的权衡研究选择准则由准确度、灵敏度、寿命、重量、尺寸、功率、可靠性、维修性、安全性、生存性、费用、进度和风险等特征参量组成。

(4) 确定加权系数

在上述列举的权衡选择准则中,其重要程度并非是同等的。因此,决策者在决定各候选方案时,可按各准则的相对重要程度给予加权。为了保证权衡分析的客观性,加权系数是由决策者制定的,而且一般不让进行权衡研究的分析人员知道,以便使决策者对候选方案获得客观的估计。

给准则加权的目的是为了降低权衡分析中评估人员分散程度的影响。此外,也可以简化彼此之间各不相关的准则之间重要性的比较。

加权系数的数值范围为0~1,越是重要的准则给予的加权系数越高。各准则的加权系数之和应当为1.0,即满足:

$$\sum_{i=1}^n W_i = 1.0 \quad (6.1)$$

式中: n 为选择准则的数目。

(5) 评估候选方案

为了评估候选方案的优劣,首先应当估算出每个候选方案与每项准则相对应的性能。性能估计可以由参加评估分析人员通过参数分析、试验、仿真、比较,或从原准设计的数据外推等各种可靠、有效而又经济的方法来完成。

然后,对该类工程项目富有经验的专家根据权衡准则和各候选方案的性能予以评分。评分值的范围为0~10分,即给最能满足准则要求的候选方案评10分,而0分则是不可接受的最差等级。

在专家评分之后,再填写加权值并将评分值和加权值相乘,得出每个方案各特性参量的加权后评分值。最后可得到每个候选方案的总加权评分值,并可据此选出基准方集。

(6) 判断候选方案是否满足约束条件

候选方案除了应当符合准则要求之外,还必须满足一定的约束条件,亦即满足若干规定的最小(或最大)的可接受值。小于(或大于)该值就会达不到预定的目标。不满足约束条件的方案不能入选。

(7) 制定权衡表

权衡表的格式可参见表6.2,有关评价准则的各特征量列在该表的最左列,第二列是各准则的加权系数,其余各列是各候选方案的性能以及有关专家就各方案的优劣给予的评分。这样就可得出各候选方案总的加权评分值,然后按得分多少给出中选方案的名次。最后在权衡表最下面一栏中概要地列出已选定方案的关键特性。

(8) 灵敏度分析

进行灵敏度分析的目的是为了检验上述的权衡结果,以便向决策者提供合理的决策依据。特别是在各个候选方案的总加权评分较接近时,对任何方案对应的任一准则的性能估计稍有变化,就可能改变彼此之间的总评分次序,从而影响决策。尤其是在进行性能估计且可利用的数据并不十分充分时,更是如此。因此,必须对已初步选定的方案进行灵敏度分析,指出被评估的性能在一定范围内变化时,对总加权评分的影响。此外,通过评估得出的中选方案是在一定的给定条件和情况下作出的,当这些条件和情况发生变化时,选中的最优方案就未必是最优的。灵敏度分析就是检查权衡结果对设计条件发生变化时的敏感性。如果在设计条件的全部可能变化范围内,发现中选方案并非最优,则应采取必要的措施找出在设计条件的全部变化范围内都有效的方案组合,或可制定应急措施,以便在设计条件发生变化时采用。

(9) 选定最优方案

根据评分和灵敏度分析的结果,可以选出能满足任务目标的“最优”方案。应当指出,该“最优”方案并非真正理想的最优方案,而是考虑了性能、费用、进度、风险等因素进行综合权衡的结果。它只是在相对意义上的“最优”。以表6.2中的方案Ⅲ为例,虽然它的加权总评分最高,但并非每一项指标都高于其他方案(如该方案的有效载荷环境就不如其他两个方案)。

还应当指出的是,不能单纯依照总加权评分值的大小来选定方案。因为给各方案评分的



虽然都是富有经验的专家,但他们毕竟或多或少地会带有一定的主观与偏好。因此,为了使各候选方案评分间的差别有意义,各方案间至少应有 10% 的差值,才能表明这两个方案确有较明显的差异。

倘若,各个候选方案之间的差异不明显而难以决策时,可采取以下措施:

- 推迟决策,直到获得进一步的附加信息;
- 评审并进一步修正选择准则和加权系数;
- 修订候选方案的性能估算结果,减少性能估算中的不确定性,给出更为正确的性能估计值及其他有关的附加数据;
- 暂时保留两个供平行发展的设计方案,作进一步的分析、验证和评估。

随着飞行器系统设计进程的发展,不同方案之间的差异将进一步地显示出来。

6.1.2 权衡分析的若干方法

权衡分析的核心内容是对不同的候选方案进行定量估计,亦即根据各候选方案的性能给出各方案的合理评分,作为供决策的依据。常用的定量权衡分析方法有加权评分法和效用函数法。

1. 加权评分法

加权评分法的步骤是:

- (1) 由决策者确定各选择准则和加权系数;
- (2) 由进行权衡分析的人员利用各种方法估算各候选方案的性能;
- (3) 请有关专家或有经验的设计师根据每个方案的各项性能逐项评分;
- (4) 将各方案所得评分与加权系数值相乘,得加权评分;
- (5) 将各项加权评分累加,得出每个方案的总加权评分;
- (6) 比较每个方案的总加权评分值,得出各个候选方案入选的优先次序,并从中选出“最优”的设计方案。

在上述各步骤中,关键在于:

- (1) 合理地确定各项选择准则的加权系数;
- (2) 恰当地根据各候选方案的性能予以评分。

确定准则加权系数的基本依据是工程系统的任务能力(mission capability),应当根据各项准则对任务能力贡献的大小来确定加权系数值。例如 6.1.1 小节涉及的快速、经济地进行长距离军事物资运输的例子,通过图 6.3 权衡树的初步分析,已初步选择亚音速运输机作为入选的方案,而对亚音速运输机的 3 个候选设计方案则应进行仔细的权衡研究。对于亚音速运输机,其任务能力包括运载能力、飞行性能、可用性、适应性、经济性等;而飞行性能又可分解为飞行速度、续航距离、起飞滑跑距离等;可用性则可分解为可靠性和维修性等。因此,加权系数可以遵循一个逻辑分解体系向下分配,如图 6.4 所示。

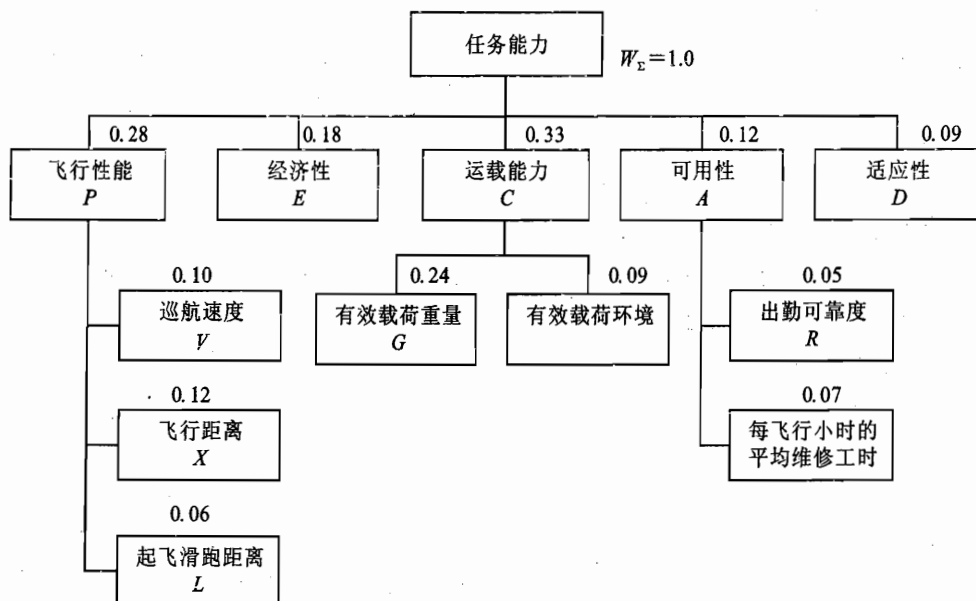


图 6.4 加权系数分配体系

如前所述,加权系数应当根据各准则对实现系统目标(即任务能力)的贡献大小或重要程度来合理确定,而不是只凭决策者的主观臆断给出。

为了客观地给出加权系数,可以利用准则优先程度评估矩阵来确定加权系数值。

准则优先程度评估矩阵是一交叉影响矩阵:

	P_1	P_2	P_3	$\cdots$	P_n
P_1		$k_{1,2}$	$k_{1,3}$	$\cdots$	$k_{1,n}$
P_2	$k_{2,1}$		$k_{2,3}$	$\cdots$	$k_{2,n}$
P_3	$k_{3,1}$	$k_{3,2}$		$\cdots$	$k_{3,n}$
$\vdots$	$\vdots$	$\vdots$			$\vdots$
P_n	$k_{n,1}$	$k_{n,2}$	$k_{n,3}$	$\cdots$	$k_{n,n}$

上述交叉影响矩阵中, $P_1, P_2, \cdots, P_n$ 为待比较的 n 个准则; k 为比例系数,表示准则 P_i 与 P_j 相比的重要程度。例如,准则 $P_1 : P_2 = 7 : 3$,表示准则 P_1 比 P_2 重要,则比例系数为 $k_{1,2} = 7, k_{2,1} = 3$ 。

根据上述交叉影响矩阵,可列出准则优先程度评估矩阵,如表 6.1 所列。

表 6.1 准则优先程度评估矩阵

准 则		准 则					S_i
		$k_{i,j}$					
准 则		P	C	A	E	D	
1	飞行性能 P	0	4	8	8	8	28
2	运载能力 C	6	0	9	9	9	33
3	可用 性 A	2	1	0	3	6	12
4	经 济 性 E	2	1	7	0	8	18
5	适 应 性 D	2	1	4	2	0	9

表 6.1 中,将要比较的各项准则(如飞行性能、运载能力、可用性等)分别列在矩阵的第一行和第一列上。然后,按照彼此之间的相对优先(即重要)程度一一加以对比。各准则自身相比不得分,故评估矩阵对角线上各元均为零。当准则之间一一对比时,较重要的那个准则得分应大于 5 分,而两者之和应为 10 分,即

$$k_{i,j} + k_{j,i} = 10 \quad (6.2)$$

例如,运载能力 C 与飞行性能 P 相比,运载能力重要,故 $k_{C,P} = 6$ 而 $k_{P,C} = 4$,在表 6.1 中应分别在对应 P 、 C 的位置上填写上 4 与 6;而飞行性能 P 与可用性 A 相比,飞行性能重要,故 $k_{P,A} = 8$, $k_{A,P} = 2$ 。如是,即可列出表 6.1 所列的评估矩阵。

将该矩阵中每一行的数值相加得 S_i ,最后便可求得与该行相对应准则的加权系数 W_i :

$$W_i = \frac{S_i}{\sum_{i=1}^n S_i} \quad (6.3)$$

式中: S_i 为第 i 行各比例系数值之和; $\sum_{i=1}^n S_i$ 为所有各行比例系数值之和; n 为准则的数目。

由表 6.1 可得运载能力 C 的加权系数 W_C 为:

$$W_C = \frac{33}{100} = 0.33$$

同理可得,飞行性能 P 的加权系数 $W_P = 0.28$; 可用性 A 的加权系数 $W_A = 0.12$; 适应性 D 的加权系数 $W_D = 0.09$; 经济性 E 的加权系数 $W_E = 0.18$ 。

上述各准则还可继续向下分解为层次更低的参数(例如飞行性能 P 可分解为飞行速度 V 、续航距离 X 等)。这些较低层次的准则之间也同样可以应用优先程度评估矩阵进行彼此之间优先程度的比较,并得到各自的加权系数数值。例如,经过比较后,飞行性能加权系数可分解为图 6.5 所示的较低层次准则(V, X, L_{TO})的加权系数。可用性加权系数可分解为图 6.6 所示的可靠性和维修性加权系数。

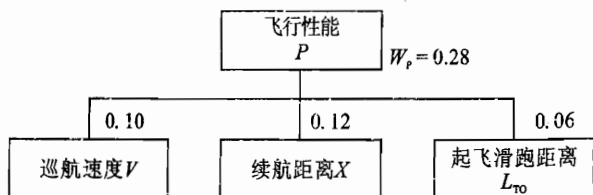


图 6.5 飞行性能加权系数的分配图示

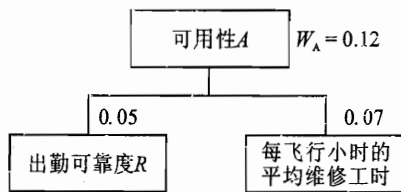


图 6.6 可用性加权系数的分配图示

如是,可得到该亚音速运输机各选择准则的加权系数分配体系及其各加权系数值,如图 6.4 所示。

加权评分法示例

本示例对如前所述的快速、经济地进行远距离军用物资运输的亚音速运输机设计方案进行定量的权衡研究。

(1) 确定目标要求,其主要目标是:

- 根据运货距离要求,运输机的续航距离不能小于 5 000 km。
- 飞行速度要求大于 800 km/h。
- 起飞滑跑距离要求小于 2 000 m。
- 运载能力要求包括:
 - 载货量(即有效载荷重量)为 30 t,因为任务要求的总运货量为 350 t,据飞机飞行速度,能在 48 h 内运货 3 次,故每架飞机可运货 90 t,有 4 架飞机即可满足任务要求;
 - 有效载荷环境:货仓内环境应不高于 5 000 m 高度。
- 经济性:要求低的使用费用。
- 维修性:要求每飞行小时的平均维修工时 $\text{MMH}/\text{FH}=20\sim 25$ 。
- 可靠性:要求飞机的出勤可靠度不低于 98%。
- 适应性:要求对任务环境应有适应能力。

(2) 确定候选方案

如前所述,经过权衡树的初步筛选,确定用飞机作为运载工具。在采用飞机的 3 种方案中,超音速运输机的使用费用过高,不符合经济性要求,此外载货能力也较小,可予以删除;用现有旅客机改装的方案则因现有旅客机的运载能力及快速装卸货物的能力不能满足要求,故初步选定亚音速运输机。对亚音速运输机,提出 3 种设计方案(I、II、III 方案)供定量权衡分析。

(3) 确定选择准则

根据目标要求确定了评价候选方案的选择准则,它们是:续航距离 X 、巡航速度 V 、起飞滑跑距离 L_{T0} 、有效载荷重量 G_P 、有效载荷环境、每飞行小时的平均维修工时 MMH/FH 、出勤可靠度 R 、飞行适应性和飞机使用费用。

上述 9 个选择准则列于权衡表(见表 6.2)的最左列。

表 6.2 权衡表示例

准 则	加权 系数	方案 I			方案 II			方案 III		
		性能	评分	加权 评分	性能	评分	加权 评分	性能	评分	加权 评分
续航距离/km	0.12	5 000	8	0.96	6 000	10	1.2	5 500	10	1.2
巡航速度/(km/h)	0.10	810	8	0.8	820	8	0.8	900	10	1.0
起飞滑跑距离/m	0.06	1 800	8	0.48	1 800	8	0.48	1 500	10	0.6
有效载荷环境	0.09	有环境控 制系统	10	0.9	有环境控 制系统	10	0.9	无环境控制 系统,相当于 5 000 m 高度	5	0.45
每飞行小时的平均维修 工时(MMH/FH)	0.07	25	7	0.49	25	7	0.49	20	10	0.7
出勤可靠度 R	0.05	0.98	8	0.4	0.98	8	0.4	0.99	10	0.5
飞机适应性	0.09	可适应	9	0.81	可适应	9	0.81	可适应	8	0.72
有效载荷重量/t	0.24	30	8	1.92	30	8	1.92	32	10	2.4
使用费用	0.18	中等,4 乘 员,4 发动机	6	1.08	中等,4 乘 员,4 发动机	6	1.08	低,2 乘 员,2 发动机	10	1.8
总加权评分 (名 次)		7.84(3)			8.08(2)			9.37(1)		
中选方案的关键特性(中选方案为方案Ⅲ): • 续航距离可满足任务要求 • 巡航速度高 • 有效载荷重量大 • 使用费用低 • 可靠性、维修性好 • 有效载荷环境和适应性也可满足要求										

(4) 确定准则的加权系数

通过准则优先程度评估矩阵和加权系数分配体系,得到以上各选择准则的加权系数,并列于表 6.2 的左起第 2 列。

(5) 评估候选方案

首先给出 3 个候选方案对应各项准则的性能数据,并分别列于权衡表的相应各行和各列,

然后请专家根据各方案的性能水平给予评分。评分范围为 0~10 分。最后计算对应各项准则的加权评分值,并得出各个方案的总加权评分值。

(6) 检查各方案是否满足约束条件

由表 6.2 中的性能数据可知,上述 3 个候选方案均满足约束条件,故均可列入备选的方案序列。假若有某个方案的续航距离小于 5 000 km 或有效载荷重量小于 30 t,则不能满足最低的目标要求值,这样的方案应予剔除。

(7) 制定权衡表

权衡表如表 6.2 所列。

(8) 选择最优方案

从权衡结果看,方案Ⅲ具有明显的优势,除了有效载荷环境和适应性两项指标外,其他如性能、费用、可用性指标等均优于其他两个方案。该方案的总加权评分值为 9.37 分,分别比方案Ⅰ和方案Ⅱ高出 20%和 16%,故选定方案Ⅲ为进一步发展的基准方案。

最后应将中选方案的关键特性概要地列于权衡表的最下方。

2. 效用函数法

效用函数法是利用效用函数对不同方案进行权衡分析的方法。

效用函数法能将各种不同量纲的准则转化为一种共同的衡量尺度。例如,可以把续航距离(以 km 表示)、巡航速度(以 km/h 表示)、有效载荷重量(以 kg 表示)等通过效用函数这一中介转化为共同的统一尺度。而且,它还能把原来不可能准确测定的非技术性参数(例如费用、研制周期、市场销售前景等)进行相对的量化与比较。

效用函数是某些被评价特征量的函数,可记为:

$$U = \sum_{j=1}^n W_j u_j \{X_j(K)\} \quad (6.4)$$

式中: U 为被评价系统的总效用; n 为被评价系统的特征量数目; $X_j(K)$ 为系统某特征量所具有的特征值; u_j 为效用函数,是某特征量 X_j 的函数; W_j 为加权系数。

效用函数与被评价的特征量有以下性质:

- 每一个被评价的特征量都由一个特征值所确定;
- 被评价特征量的优劣次序根据该特征量对应的效用函数值大小排定;
- 效用函数并不给出被评价系统的绝对值,而只给出各种不同备选方案相比较的相对值大小,然后根据相对值大小即可定出各备选方案的优劣顺序。

一般,效用函数值 $U = \langle 0, 1 \rangle$ 。

效用曲线能与某个特定准则相关的特征量的数值转换成某个效用值。效用值的范围从 0 到 1,通常定义规范要求的某特征量的最低限度值对应效用函数的 0 值($u_j = 0$),而该特征量的最高值则对应效用函数的 1 值($u_j = 1$)。因此,效用曲线的范围包含了备选方案特征量的可接受的范围。例如,前述的运输机方案,其巡航速度的范围为 800~900 km/h,则 $V = 800$ km/h

对应效用曲线上的0值,而 $V=900\text{ km/h}$ 则对应效用曲线上的1,如图6.7所示。

假如飞机的生存概率与其飞行速度有关,则还可以进一步地利用生存概率是速度的函数来制定生存概率的效用曲线。

一般常用的效用曲线种类如图6.8所示。

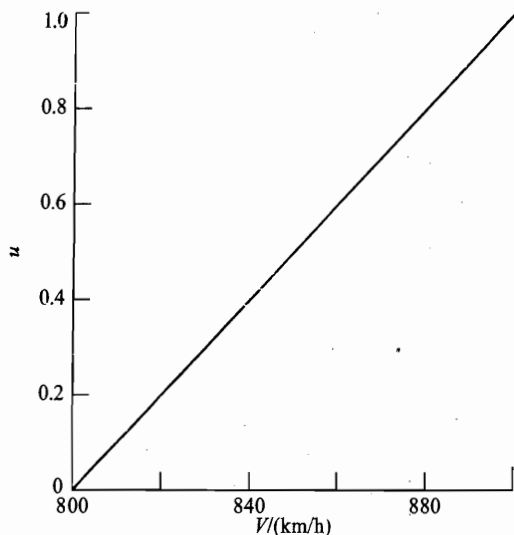


图 6.7 飞机巡航速度的效用曲线示例

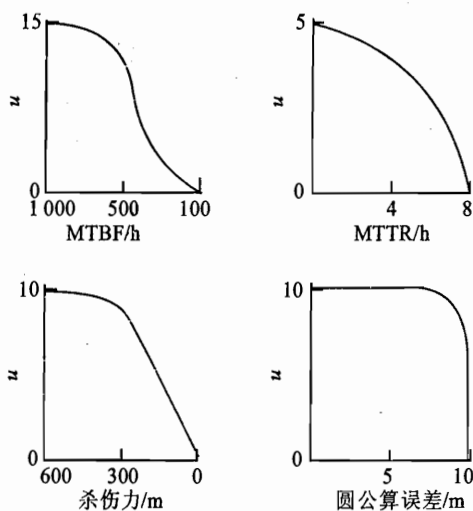


图 6.8 效用曲线示例

对某个给定系统的权衡研究,效用曲线必须使用统一的尺度(例如从0到1)。每条曲线的零点可以定得略低于规范规定的该特征量最小的可接受值。因为最小的可接受值也是某方案能否入选的界限值。

不必为每一个特征量绘制效用曲线。假如效用与特征量之间呈线性关系(见图6.7),则可以用简单的表格或比例因子表示出两者之间的关系。

在完成了上述工作之后,即可利用效用函数法对备选方案进行权衡。具体步骤是:

(1) 确定每个特征量对应的效用曲线(效用曲线的尺度统一从0到1)。

(2) 估算出每个备选方案与每项准则相关特征量的数值大小。

(3) 将特征量的数值根据其相应的效用曲线转换成效用值大小。例如估算出第Ⅱ方案的特性量之一,巡航速度 $V_{\text{II}}=850\text{ km/h}$,根据其相应的效用曲线(见图6.7)可查得相应该速度的效用值 $u_{V_{\text{II}}}=0.50$ 。若第Ⅲ方案的速度 $V_{\text{III}}=900\text{ km/h}$,则相应的效用值 $u_{V_{\text{III}}}=1.0$ 。

(4) 确定各准则的加权系数 W_j (加权系数确定方法参见6.1.2小节)。

(5) 制定如表6.3所列权衡表,将各权衡准则、准则加权系数以及与各方案特征量数值大小相对应的效用值均列在表内相应的栏目中。



(6) 选择最优方案。将各候选方案对应每项准则的效用值 u_j 与加权系数 W_j 相乘后累加,便可得到每个候选方案的总加权效用值。比较这些总效用值 U_j 的大小,就可排列出候选方案的优劣次序,并从中选出最优方案。

表 6.3 所列的是利用效用函数法进行权衡研究的例子。供权衡的候选方案共有 4 个。权衡准则为 A、B、C、D、E 共 5 项。相应的加权系数和每项准则对应的特征量的效用值如表 6.3 所列。最后比较各方案总加权效用值的大小。显然,方案Ⅲ具有较明显的优势(比方案 I、Ⅱ、Ⅳ分别高出 19%、21%、14%),故选择方案Ⅲ为进一步发展的基准方案。

最后,应将中选方案的关键特性概要地列于表 6.3 的最下面一栏。

表 6.3 效用函数法权衡表示例

准则	加权系数	候选方案							
		I		II		III		IV	
		u_j	$W_j u_j$	u_j	$W_j u_j$	u_j	$W_j u_j$	u_j	$W_j u_j$
A	0.15	0.35	0.0525	0.59	0.088 5	0.65	0.097 5	0.45	0.675
B	0.20	0.48	0.092	0.82	0.164	0.92	0.184	0.81	0.162
C	0.10	0.60	0.06	0.60	0.06	0.82	0.082	0.95	0.095
D	0.35	0.95	0.333	0.73	0.256	0.90	0.315	0.60	0.210
E	0.20	0.67	0.134	0.45	0.09	0.60	0.120	0.82	0.164
$\sum_{j=1}^5 W_j u_j$		0.671		0.658		0.799		0.699	
名 次		3		4		1		2	
中选方案的关键特性：略									

6.2 工作分解结构

工作分解结构(Work Breakdown Structure, WBS)是一个以产品为中心的层次体系。该层次体系由研制和生产一个工程系统所必须的产品项目、服务项目、资料项目所组成。工作分解结构完全确定了要研制和生产的产物,并表示出要完成的各个工程单元之间以及它们与最终产品之间的关系。通过工作分解结构可以完全确定一个系统。

6.2.1 系统层次体系

鉴于系统的复杂性不断增加,它所包括的分系统、技术状态项目、部件、组件的数目也越来越多,因此,应当制定系统的层次体系。



一个标准的系统层次体系如图 6.9 所示。它以系统最高层次向较低层次不断分解和延伸,组成一个有序的层次体系。

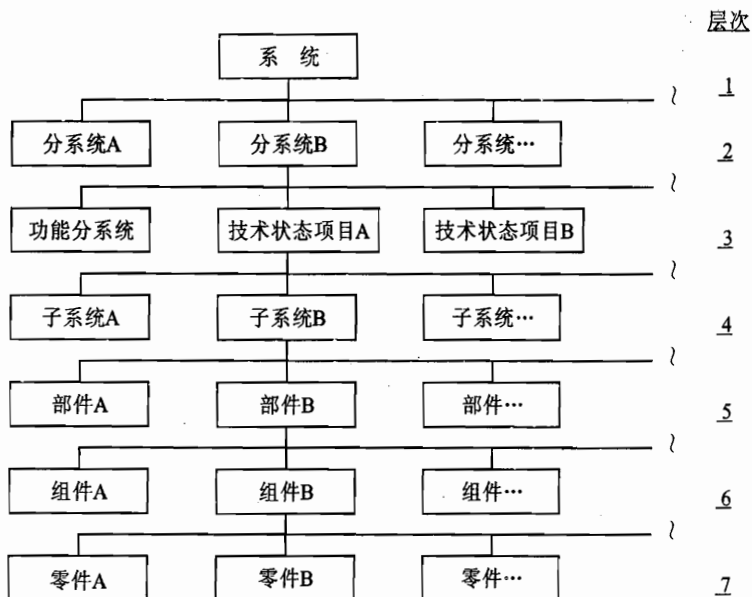


图 6.9 系统层次体系

在系统层次体系中,各组成部分的定义是:

- 系统: 由硬件、软件、人员、设施、数据和劳务组成的整体,用来完成给定的任务,例如一个军用飞机系统。
- 分系统: 是一组密切相关并可实际对接的技术状态项目(或称功能分系统)的组合。它由几家承包单位生产并由一家主承包单位组装的各技术状态项目所组成,例如一架歼击机、一辆汽车等。
- 技术状态项目(功能分系统): 是一套完整的、综合的、能够完成某种功能的子系统的组合,如飞行控制系统、火力控制系统等。技术状态项目是由单家承包单位交付的项目。
- 子系统: 是一组部件的功能性组合,用以完成某功能分系统中的一项重要功能,例如飞行控制系统中的姿态控制子系统或火力控制系统中的雷达子系统等。
- 部件: 是一个功能性单元,作为供分析、制造、试验、维修用的实体,例如蓄电池。
- 组件: 由两个或两个以上的单元组成的可储备件,它可以拆卸或更换零件,例如装有元件的印刷电路板。
- 零件: 除非受到损害,否则一般不能分解的单件,如电阻、晶体管、齿轮等。

6.2.2 工作分解结构的组成和分类

以系统的层次体系为依据,可以建立飞行器系统的工作分解结构(WBS)。它以系统整体作为最高层,然后逐层划分工作单元,直到最低层次的不可再分解的工作单元为止。以图 6.10 所示的某军用飞机系统的 WBS 为例,军用飞机系统为最高层次,它由航空飞行器(飞机)、系统试验与鉴定、项目管理、保障分系统等工作单元组成。而飞机又可分解为机体、发动机、航空电子系统等单元,如此逐层分解便可建立一个飞行器系统的 WBS。

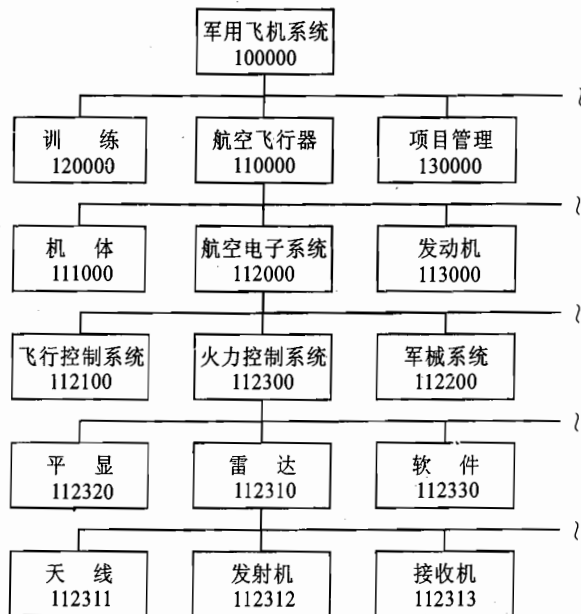


图 6.10 某军用飞机系统的工作分解结构

通常,对某一个给定的工程系统,其 WBS 的最上面 3 个层次由政府型号办公室确定,并称之为“项目纲要性 WBS”。该 WBS 应当在方案设计阶段开始时拟定,并包括在方案设计阶段的招标书中。

然后,由研制部门制定“合同工作分解结构”,它应当将 WBS 继续分解到“工作包”的层次为止。这大约是在 WBS 的第五或第六层次上。

最后,在生产阶段开始之前,应当制定出“项目工作分解结构”,它应当包括工程系统的全部工作单元,亦即应分解到最低层次的不可再分解的工作单元(如零件)为止。

研制部门利用合同工作分解结构对工程系统的技术性能、费用、进度和风险进行综合的控制。

在建立 WBS 中,应当注意下述问题:

- (1) WBS 应当包括与该工程系统有关的全部硬件、软件、劳务和数据等工作,不应漏掉任

何项目。

(2) WBS 的每一方框代表一个工作单元,每个工作单元都有自己的代号。该代号应采用十进制代码,使每个工作单元都具有可跟踪性,以便对其进行费用、进度和性能的综合监控。

(3) 每个工作单元都应建立一份“工作单元工作说明”,在该工作说明中应当包括:

- ① 该工作单元的代号、名称及它在工作分解结构中所处的层次;
- ② 该工作单元的任务;
- ③ 来自较低层次工作单元的输入;
- ④ 向较高层次工作单元的输出;
- ⑤ 该工作单元的负责部门与负责人;
- ⑥ 为完成该工作单元任务所需的时间(进度)、费用、工时和人员需求。

6.2.3 工作分解结构的基本功能

工作分解结构的功能是为工程系统确立有合同效力的系统层次体系,使技术性能、进度、费用等要素通过工作分解结构相互关联。工作分解结构的基本功能是:

- 为技术性能的规划和控制提供依据;
- 为制定进度计划提供各项工作先后次序的逻辑框架;
- 累计工程费用;
- 进行风险辨识;
- 在工作分解结构的基础上,发展出工程项目进展状况的报告系统,使技术性能、进度、费用信息具有可跟踪性。

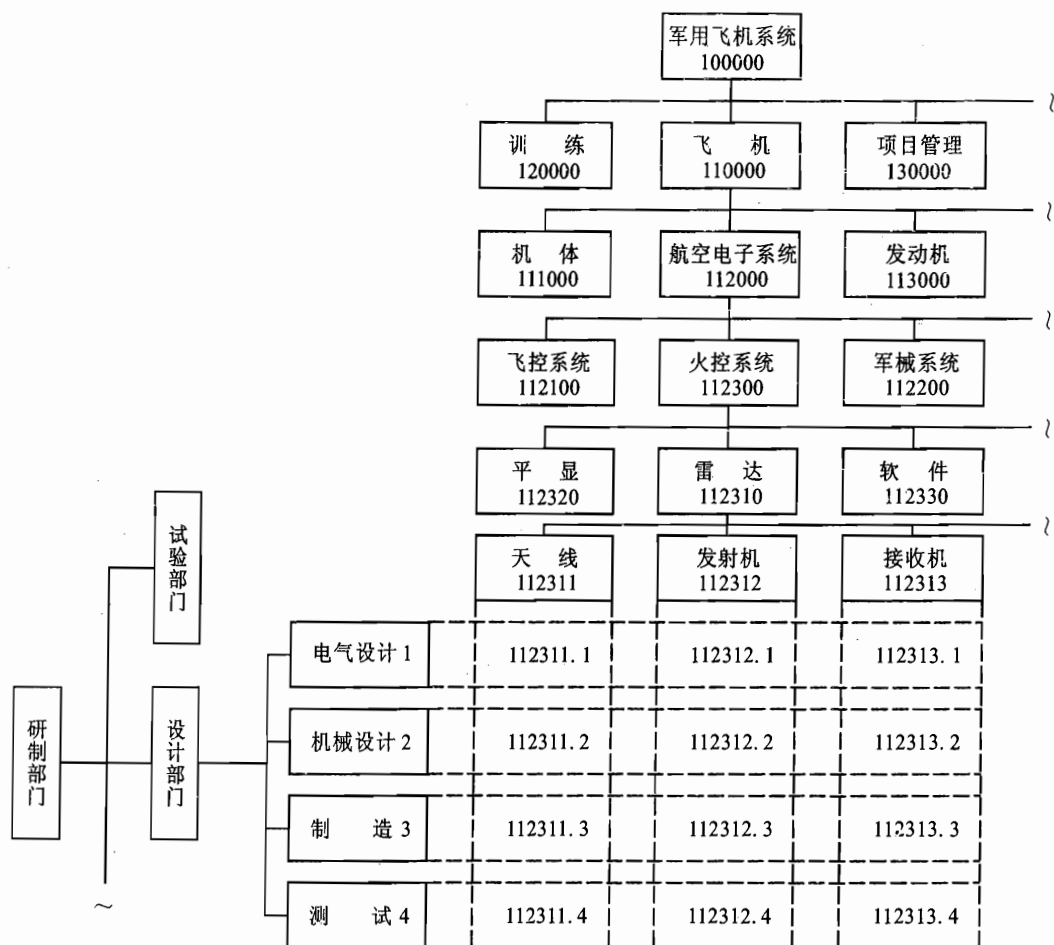
此外,工作分解结构作为技术和管理活动的框架,为所有的项目参与者提供了沟通渠道,并使各项工作能在明确定义的基础上进行。

为了便于对技术性能、进度、费用进行综合管理,应当在合同工作分解结构中确立“工作包”的概念。工作包是处在合同工作分解结构最低层次的工作单元,是研制部门对其进行进度、费用、性能日常管理的最基本的单元。根据管理的方便和习惯,工作包一般是在 WBS 的第五或第六层次上。

要完成每个工作包的任务,必须与研制部门中的各个职能部门发生联系。图 6.11 所示的合同工作分解结构中,各个工作包(图 6.11 中的工作包确立在第六层次上,即“天线”、“发射机”等合同工作分解结构的最低层次上)与下述一个或若干个部门有工作联系:

- 设计部门(又可分为机械设计、电气设计等分部门);
- 制造部门;
- 装配部门;
- 试验与鉴定部门等。

每一个工作包都应对上述各有关部门提出进度、费用和性能指标要求。然后,对其实际进

图 6.11 合同工作分解结构与工作包示意图^[1]

展状况进行日常的监控、管理,并通过管理信息系统向有关部门报告每个工作包及其较高层次工作单元的进展状况,以实现对工程系统技术性能、进度、费用的综合管理。

6.2.4 利用工作分解结构制定进度计划

要制定一项成功的计划,必须对完成工程项目所需的各项工作及其先后次序都有一个系统、全面的了解。

而工作分解结构(WBS)作为一种以产品为中心的层次体系,确定并用图形显示了要开发的产品,以及要完成此产品所需进行的所有工作及其相互关系。WBS 可以为型号管理部门提供安排进度计划的基准。

WBS 与计划网络图或条线图之间有着明显的联系：

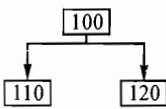
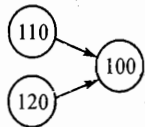
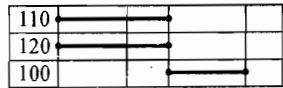
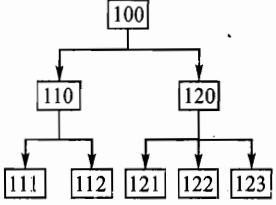
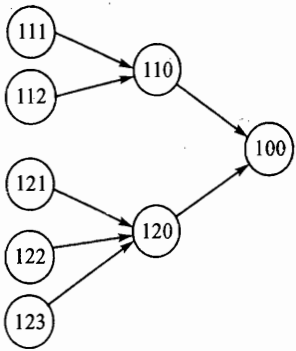
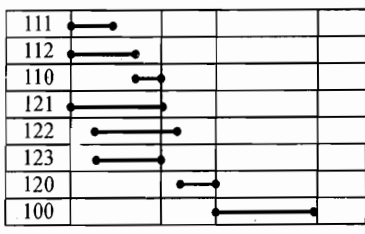
(1) WBS 中的每一个工作单元, 应对应网络图中的一个活动(节点), 或条线图中的一条直线。

(2) 对网络图中的每一个节点或条线图上的每一条直线, 必须提供期望的时间和费用。而 WBS 中每一工作单元的工作说明书中提供了完成该工作单元所需的时间和费用。为了制定计划的方便, 各工作单元的时间、费用的单位必须统一。

(3) 某些活动, 若不属于 WBS 的工作单元(例如, 政府或使用部门自身设施的改建等), 但对工程系统的研制或使用有影响的, 也应包括在网络图或条线图中。

WBS 与计划网络图及条线图的一一对应关系如表 6.4 所列。

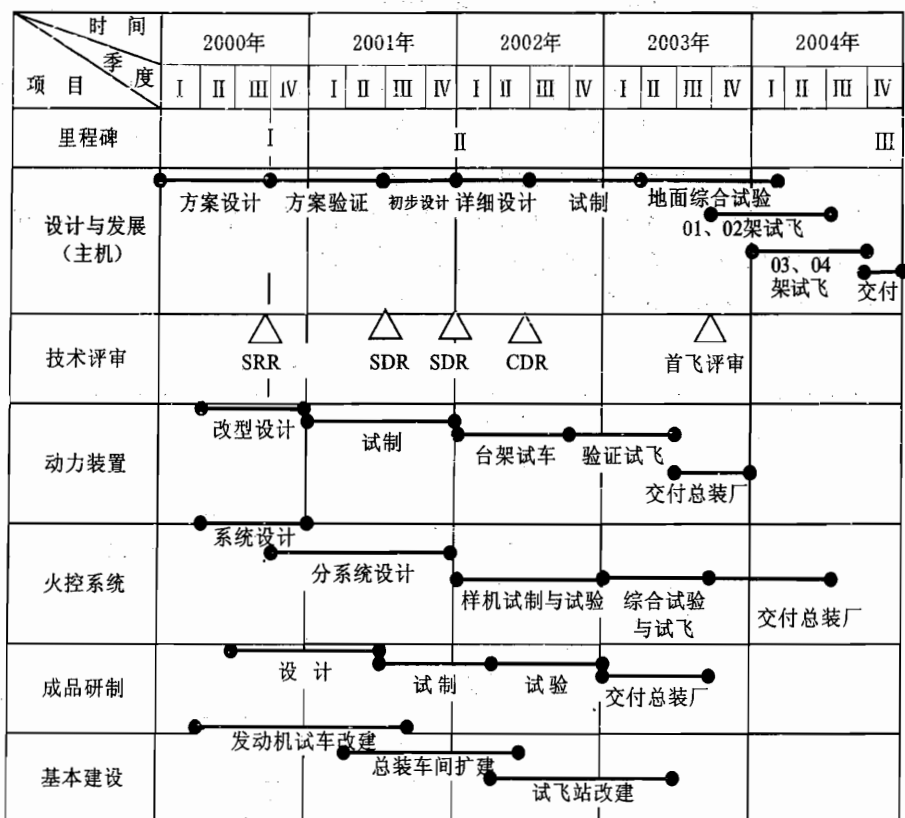
表 6.4 WBS 与计划网络图及条线图的关系

工作分解结构	网络图	条线图
		
		

通常, 可以根据 WBS 第 1~3 这 3 个层次的活动, 制定最高层次的计划网络图或条线图, 并称之为“零级进度表”或“总进度表”。这种进度计划一般只包括 20~40 项活动(具体项目多少将根据工程系统规模大小而定)。表 6.5 是零级进度表的例子。图中包括大约 30 项左右的活动。

零级进度表把整个系统的设计、分析、试验、评审、生产等工作按其流程连为一体。它不仅确定了合同规定应交付的所有硬件、软件的主要研制工作和重大里程碑, 而且还确定了 WBS 所有较低层次活动的进度轮廓。因此, 工程系统的零级进度表也是该工程系统的基本规划文件。

表 6.5 某工程系统的零级进度表



根据表 6.5 所列的零级进度表,便可制订出下一层次的进度计划,亦即一级进度计划和二级进度计划。

一级进度计划(网络图或条线图)通常包括 100~200 个活动。它应当包括整个工程项目的重要活动,尤其应当突出各个领域(如工程设计、试验、成品采购、基建)中的关键活动。

6.2.5 利用工作分解结构估算费用

利用 WBS 来累计工程费用。以图 6.11 所示的军用飞机系统为例,计算费用的步骤如下:

(1) 确立该工程系统的 WBS,将 WBS 的工作单元分解到便于计划和控制的工作包一级,亦即图 6.11 所示的 WBS 第六层次上的发射机、接收机等。再把这些工作包的研制任务分派给相应的职能部门(如图示的电气设计、机械设计等部门)。

(2) 对 WBS 中每个工作单元编码,以此编码作为每个工作单元费用的汇总号。

(3) 对 WBS 中每个工作单元相应的职能部门建立费用账号。例如,图 6.11 中接收机的工作单元编码为 112313。该接收机相应的电气设计部分的费用账号为 112313·1。

(4) 估算每个工作包的费用,该费用一般由以下各部分组成:

- 工时费用(包括设计、制造、试验工时);
- 材料费用;
- 动力、设施和设备费用;
- 间接费用(如管理费用等)。

以发射机为例,可列出如表 6.6 所列的工作包累计费用表。

表 6.6 工作包累计费用表示例

项目名称： <u>发 射 机</u>		工作包编码： <u>112312</u>			日 期： <u> </u>					
费用账号	任务	工 时		单位工时费 (元/月)	工期 (月)	工时费 (元)	材料费 (元)	动力费 (元)	间接费 (元)	总计 (元)
		人员类别	人-月							
112312·1	电气 设计	工程师	10	3 000	1	42 000	30 000	20 000	9 000	101 000
		技师	6	2 000						
112312·2	机械 设计	工程师	10	3 000	1	40 000	30 000	20 000	9 000	99 000
		技师	5	2 000						
112312·3	制造	技术员	5	1 500	2	63 000	150 000	60 000	30 000	303 000
		工人	20	1 200						
112312·4	测试	技师	6	2 000	1	18 000	50 000	50 000	11 000	129 000
		工人	5	1 200						
项目总费用： <u>632 000</u> 项目进度： <u> </u>										
项目负责人： <u> </u> 批准部门： <u> </u>										

(5) 累计总费用将由第(4)步得出的各个工作包的费用汇总到上一层次(即第五层次)工作单元的费用汇总号中。上一层次工作单元的费用是它所属的各工作包费用之和再加上将这些部件组装在一起的组装费、管理费。以此类推,将 WBS 各工作单元由下向上逐层累加,使得到工程总费用。

仍以上述军用飞机系统为例,第五层次中雷达的累计费用如表 6.7 所列。表中所列的组装费包括将下属层次工作单元组装成该工作单元所需的接口、连接件、组装工具、设备、人工、

质量保证、测试等项费用。

表 6.7 工作单元费用累计表示例

项目名称: 雷 达			工作单元编码: 112310		日 期: _____	
下属层次的工作单元			组装费 (元)	管理费 (元)	累计费用 (元)	
工作单元	编 码	费用(元)				
天 线	112311	303 000				
发 射 机	112312	632 000				
接 收 机	112313	500 000	545 000	60 000	2040 000	
项目总费用: 2 040 000			项目负责人: _____		批准: _____	

(6) 将累计的费用分配给不同的职能部门,便于各职能部门独立核算。可以用下述方法累计出职能部门的费用。

凡涉及各职使部门的工作单元费用分别汇集到各部门的费用账号中,如电气设计部门费用账号为“1”,则在与该部门有关的工作单元的费用账号后面标上“1”,如“112311·1”中的“112311”表示雷达天线的工作单元编码,而“112311·1”表示雷达天线费用中与电气设计部分相关的费用。此外,再加上该部门的间接费用(如行政管理费),即可累计出各个职能部门的费用,如表 6.8 所列。

表 6.8 中各工作包相应各职能部门的费用由表 6.6 中得到。其他工作单元相应各职能部门的费用均包含在组装费中,其估算方法与步骤(4)中估计各工作包费用的方法相类似。

通过上述各步骤便可累计出工程系统的总费用以及各个职能部门在该工程系统研制中所需的费用。

表 6.8 职能部门费用累计表示例

职能部门 费用 工作单元编码	电气设计部		机械设计部		制造部		测试部		...
	费用账号	费用(元)	费用账号	费用(元)	费用账号	费用(元)	费用账号	费用(元)	...
112311	112311·1		112311·2		112311·3		112311·4		...
112312	112312·1	101000	112312·2	99000	112312·3	303000	112312·4	129000	
112313	112313·1		112313·2		112313·3		112313·4		
112310	112310·1		112310·2		112310·3		112310·4		
⋮	⋮		⋮		⋮		⋮		
110000	110000·1		110000·2		110000·3		110000·4		
⋮	⋮		⋮		⋮		⋮		

续表 6.8

职能部门 费用 工作单元编码	电气设计部		机械设计部		制造部		测试部		...
	费用账号	费用(元)	费用账号	费用(元)	费用账号	费用(元)	费用账号	费用(元)	...
100000	100000 · 1		100000 · 2		100000 · 3		100000 · 4		
各职能部门间接费									
总 计									
制表人_____ 负责人_____ 批准_____ 日期_____									

6.2.6 制定飞行器系统费用-进度曲线的方法

为了便于在系统寿命期内费用的分配和控制,可结合工程进度来累计费用,并制定费用-进度曲线。方法如下:

- (1) 根据 WBS 绘制工程进度日程表(具体方法可参见第 6.2.4 小节)。
- (2) 根据 6.2.5 小节中的方法(如表 6.6 和表 6.7)得出各工作包费用和工作单元费用。
- (3) 确定费用斜率。为简单起见,假设各工作单元在其工期内的费用是均衡的,故可利用表 6.9 来确定各工作单元的费用斜率。

表 6.9 工作单元费用斜率

工作单元		费用(元)	工期(月)	费用斜率(元/月)
项 目	工作单元编码			
天线	112311	303 000	2	151 500
发射机	112312	632 000	4	158 000
接收机	112313	500 000	3	166 667
雷达	112310	605 000	6	100 833
⋮	⋮	⋮	⋮	⋮
航空飞行器	110000	1 000 000	12	133 323
军用飞机系统	100000	210 000	5	42 000
制表人_____ 批 准_____				
负责人_____ 日 期_____				

应当注意在计算各工作单元费用时不应重复计算同一项费用,因此表 6.9 中的各工作包(如天线、接收机等)费用是其研制费用,而工作单元(如雷达、航空飞行器等)则只计入其组装费和管理费。

(4) 确定统计费用的时间段,时间段大小由各研制单位视其管理需要而定。一般,可以1个月或1个季度为统计费用的时间段。本例取1个季度为一时间段。

(5) 列表计算各个时间段内各工作单元的费用、费用斜率和工程项目总费用,如表 6.10 所列。

表 6.10 各时间段费用的计算表

时间段		工作单元		工期(月)	费用斜率(元/月)	费用(元)	时间段总费用(元)
季 度	月 份	项 目	编 码				
1	1~3	天线	112311	2	151 500	303 000	1 541 633
		发射机	112312	3	158 000	474 000	
		接收机	112313	3	166 667	500 000	
		平显	112320	1	83 833	83 833	
		火控软件	112330	2	90 400	180 800	
2	4~6	发射机	112312	1	158 000	158 000	1 327 081
		雷达	112310	2	100 833	201 666	
		平显	112320	3	83 833	251 500	
		火控软件	112330	3	90 400	271 200	
		导航设备	112100	1	101 857	101 857	
		通信与识别	112300	3	114 286	342 858	
∴	∴	∴	∴	∴	∴	∴	∴
累计总费用 14 518 004 元 制表人 _____ 批 准 _____							

(6) 把各个时间段内的总费用累加,即可得到工程系统总费用。

(7) 在费用-时间分布图上标出每个时间段内的总费用,即可得到如图 6.12 所示的费用分布图。该费用分布是以最早开工时间计算的。

(8) 在允许的进度松弛时间内,调整进度安排,使费用-进度分布更为均匀合理。利用松弛时间调整后的费用分布图如图 6.13 所示。图 6.13 将工程项目刚开始时的费用需求降低,整个费用分布较图 6.12 的费用分布均匀、合理。

(9) 绘制累积的费用-进度曲线,如图 6.14 所示。利用费用-进度曲线可以在各个里程碑决策点控制费用,并将实际消耗的费用与期望的费用-进度曲线相比较,作出有关控制决策。利用费用-进度曲线进行控制的方法可参阅第 7 章。

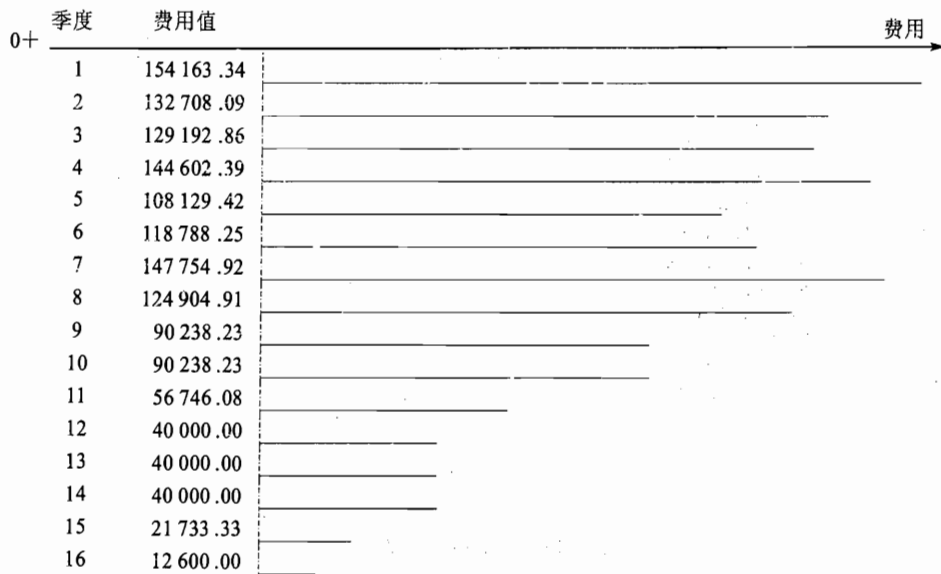


图 6.12 费用-时间分布图(以最早开工时间计算)

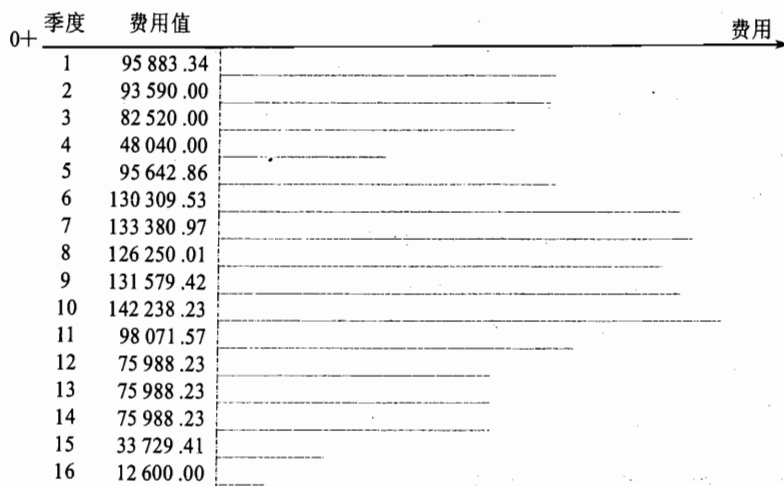


图 6.13 利用进度松弛时间调整后的费用-时间分布图

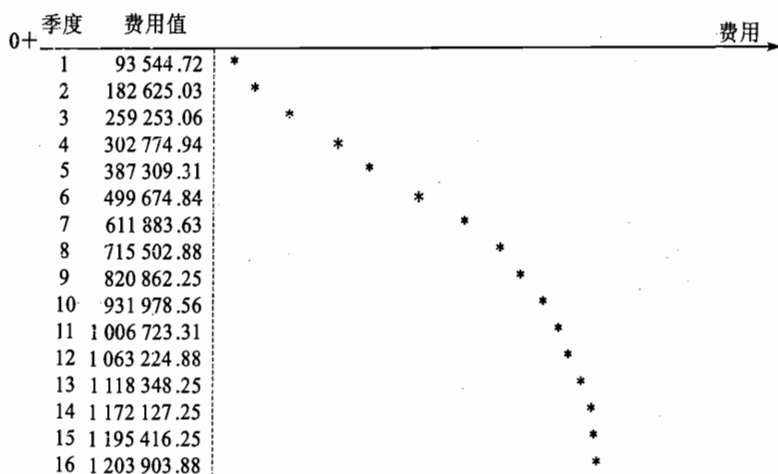


图 6.14 费用-进度累积曲线

6.3 技术性能测量

技术性能测量(Technical Performance Measurement, TPM)是对工程系统的技术性能指标进行控制和对设计进行评估的有效手段。它通过定期的工程分析和试验来验证和预测主要的技术性能参数的量值,提供产品性能是否满足规范要求的情况。这样,就能在产品研制的早期,在其对费用、进度造成重大影响之前,就确认出潜在的性能缺陷。

技术性能测量的主要目的是:

- (1) 提供工程系统实际性能和预期性能的比较。
- (2) 向管理部门及早提供在工程项目研制中应予以注意的关键问题的估计和预测。
- (3) 对建议的设计方案更改带来的影响进行评估。

利用技术性能测量,能在工程系统的潜在性能缺陷对费用和进度造成不可挽回的影响之前,就向项目管理人员发出警告,以便及时采取纠正措施。

6.3.1 技术性能测量的主要任务

技术性能测量工作主要是在方案确定阶段与工程研制阶段中进行的,其工作流程可参见图 6.15。

(1) 在方案确定阶段应制定技术性能测量的方案,包括:

- ① 利用工作分解结构确定产品的关键技术领域;
- ② 选择并确定应予以监控和测量的主要性能参数清单;
- ③ 在产品发展的各里程碑和关键点上,确定各个主要性能参数的预期值,并根据参数预

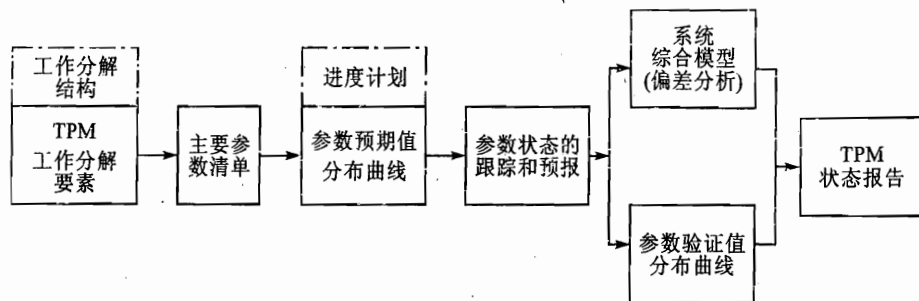


图 6.15 技术性能测量工作流程

期值的分布状况给出参数的预期值分布曲线；

④ 确定用以进行性能参数估计和预测的综合模型。

(2) 在工程研制阶段应实施技术性能测量,包括:

① 通过定期评估和测量(一般至少 3 个月一次),测出各主要性能参数的实际量值,即验证值的大小;

② 算出各主要性能参数的验证值与预期值之间的偏差大小,并通过系统综合模型分析该偏差对技术性能、进度、费用的影响;

③ 根据验证结果修正原来的参数预期值分布曲线;

④ 对每一个主要性能参数提交一份技术性能测量报告,该报告应包括问题分析和建议的改正措施。

(3) 在生产阶段,如果要求对原设计进行更改,或者对原有工程系统进行改型设计,则可利用技术性能测量对设计更改(或改型)可能产生的影响进行评估,并对更改后的技术性能状况实施监控。

6.3.2 选择技术性能测量的主要参数

被选择用来进行技术性能测量的参数必须是能代表工程系统成功与否的关键技术参数。为了使技术性能测量工作经济而又有效,只应当选择少数几个系统最高层次的关键参数进行监控。这是因为为了得到这些关键参数而需要加以跟踪的相关参数的总数可能 10 倍于关键参数。

例如图 6.16 所示的某型号武器的作战分系统,为了要测量该系统的主要参数之一——可用度 A ,需要跟踪较低层次上每个与此相关的分系统、部件和组件的可用度 A_i (或 $MTBF_i$ 和 $MTTR_i$)。只有对所有较低层次工作单元的可靠性和维修性参数均予以跟踪后,才能测定火控系统整机的可用度参数值。

图 6.16 所示的参数树是由工作分解结构发展而来的。工作分解结构确定了飞行器系统应完成的全部任务,并体现了该系统的层次体系和规范体系。因此可根据工作分解结构的工

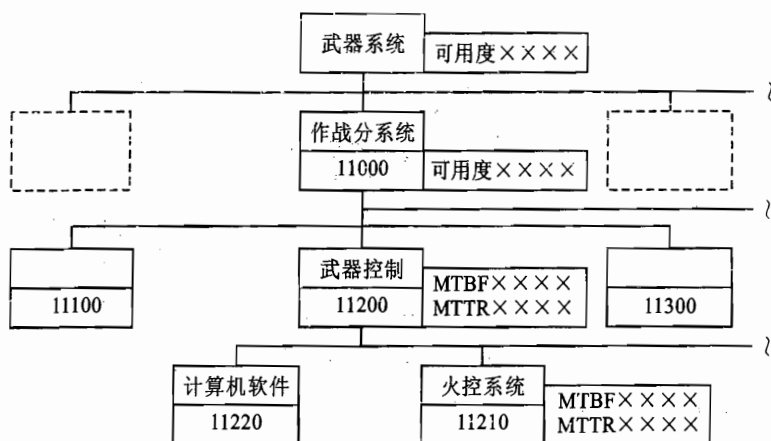


图 6.16 技术性能测量的参数树

作单元来制定技术性能测量参数树。从工作分解结构发展出技术性能测量参数树的另一作用是可据此作为估计或测定工程系统技术性能、费用和进度的重要手段,并确保该系统的技术性能对进度和费用的可跟踪性。有关工作分解结构与制定进度计划的关系,以及利用工作分解结构估算费用的方法,我们已在前面阐述过。如是,就能将测得的技术性能偏差(例如,重量超重 5%)及其对费用、进度的影响联系在一起(例如,为了消除该超重现象而采用复合材料,结果使费用增加 3%,进度延后 3 个月等)。

较低层次的参数可以通过技术要求分配过程来确定。这些参数代表了系统最高层次的技术要求向较低层次的分配。

被选择用来进行技术性能测量的主要参数应当与硬件、软件、后勤保障等要素相关,也就是与工程系统的关键领域相关。

选择技术性能测量主要参数的准则是:

- (1) 这些参数应当是考核或确定整个飞行器系统性能时,最具代表性的、影响最大的参数;
- (2) 这些参数应当是能从工程分析或试验结果中直接测定其量值的参数;
- (3) 这些参数在各个给定的时间点上,其数值及允差带是可以预测的,并能在设计、研制和试验过程中得以证实;
- (4) 在以往类似的工程系统研制中被证明是关键的参数;
- (5) 能表示系统效能的重要参数。

一般,工程系统典型的主要参数是重量、功率、计算机总处理能力、计算机存储规模、命中精度、测距精度、射程、速度、机动性、有效作用范围、有效载荷(重量和体积)、通信能力(频带宽度、通信线路数目)、可靠度/维修性/可用度、使用和维护人员要求。

对飞机来说,其典型的主要参数有:最大飞行设计重量、飞机空重、有效载荷(内部与外挂)、航程、巡航速度、突防速度、起飞及着陆距离、机动性、飞机的任务可靠度、每飞行小时的平

均维修工时、乘员安全性和救生系统性能。

对航空发动机,其典型的主要参数有:最大推力(海平面)、额定推力(海平面)、单位耗油率(海平面)、发动机总重、可靠性(MTBPL——平均推力损失间隔时间)、维修性(每飞行小时的平均维修工时)。

在确定工程系统的主要参数时,必须利用系统工程过程的程序,在分析系统总的技术要求、技术风险和参数灵敏度的基础上,为系统、分系统和每个关键的技术状态项目选择关键的参数或参数的组合。

最后,应将选定予以技术性能测量的参数列成技术性能测量主要参数清单。

6.3.3 技术性能测量的方法和步骤

为了测量每个主要参数在整个工程研制过程中的变化情况并预测其变化趋势,必须给出每个主要参数的预期值分布曲线,如图 6.17 所示。

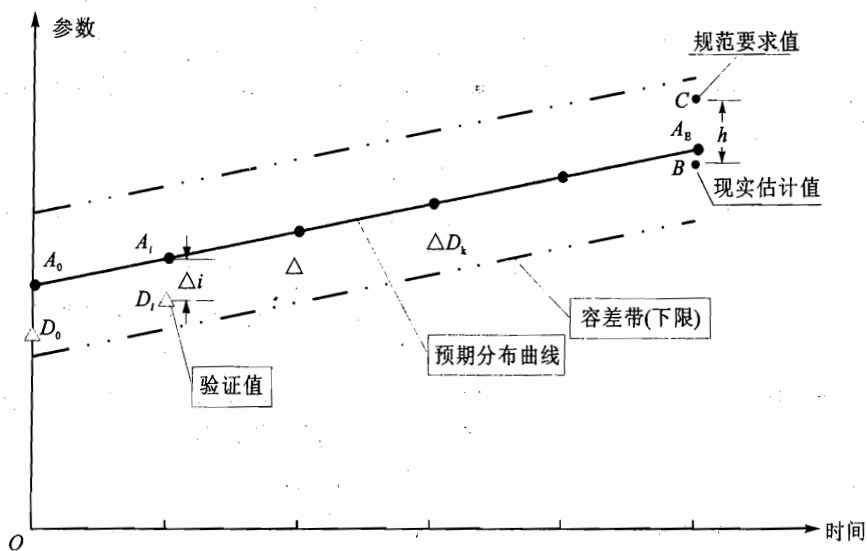


图 6.17 参数的预期值分布曲线

图 6.17 所示的参数预期值分布曲线是一条等斜率的直线。图中各参数的意义为:

- 规范要求值(图 6.17 中的 C 点): 具有合同效力的, 系统性能规范中规定的产品最终应达到的值。
- 预期值(图 6.17 中的 $A_0, A_i, \dots, A_E$ 点): 参数在研制过程中的某个给定时刻的期望值。参数各预期值的连线即为该参数的预期值分布曲线(图 6.17 中的 $\overline{A_0 A_E}$)。
- 验证值(图 6.17 中的 $D_0, D_i, \dots, D_K$ 点): 通过特定的工程分析或试验实际测定或估算出的参数值。

- 验证偏差: 参数的预期值与验证值之间的偏差, 如图 6.17 中的 Δ_i :

$$\Delta_i = |A_i - D_i|$$

- 现时估计值: 在某个给定时刻, 根据现有数据对产品最终交付时, 参数量值的预测值 (如图 6.17 中的 B 点)。

- 预测偏差: 某参数的规范要求值与现时估计值之间的偏差, 如图 6.17 中的 h :

$$h = |C - B|$$

- 参数容差带: 若参数存在可接受的门限值, 则该门限值的分布区间即为参数的容差带 (图 6.17 中上下两条点划线之间的区域, 即为该参数的容差带)。

下面介绍进行技术性能测定的步骤。

(1) 确定参数的规范要求值

参数的规范要求值可以从研制合同、研制任务书或系统规范、研制规范等文件中得到。

(2) 确定参数的预期值和预期值分布曲线

参数的预期值代表被跟踪参数期望的变化, 参数的变化趋势随不同参数而异。图 6.17 中表示的是某参数预期值的生长状况, 例如重量这一主要参数的变化趋势。通常, 在方案设计阶段, 重量是根据统计数据、设计经验和设计草图粗略估算的, 一般均低于最终的产品重量。而在研制过程中, 各部件、成品的超重现象往往会使产品最终重量比最初的预期值增加 15%~20%。对其他参数来说, 也存在负增长状态或零增长状态。

在一般情况下, 当研制工作刚开始进行时, 常常得不到可靠的数据。这时可根据历史资料、统计数据和以往类似工程系统研制中的经验, 并结合现行工程的特点进行工程分析和估算, 从而推断出研制过程各关键点上的参数预期值, 并得到参数的预期值分布曲线。因此, 在这一阶段制定的参数预期值分布曲线并不能十分准确地反映该参数在整个研制过程中变化的客观规律。随着研制工作的进展, 尤其是通过仿真和实物试验后, 在大量可靠的验证值基础上, 可以对最初的参数预期值分布曲线进行修正, 并得到新的预期值分布曲线。图 6.18 展示了某参数原定的预期值分布曲线, 以及根据验证结果修正后的预期值分布曲线。

在确定了参数的预期值分布曲线后, 应当给出该分布曲线的容差带。容差带的边界表示该参数的上、下门限。只要参数落在容差带以内, 即表示该参数的规范要求值可以在给定的时间内用给定的费用达到。

容差带的范围通常由研制部门确定, 并经政府有关主管部门 (或使用部门) 评审和认可。对容差带的评审至少每年进行一次。

容差带的范围并非固定值, 通常在研制初期, 鉴于对工程系统的技术特性尚未很好掌握, 容差带范围可以宽些, 一般可允许到 10%~15%。在研制后期, 随着对在研工程系统的技术特性的逐渐了解, 容差带范围应逐渐缩小。到工程研制阶段后期, 容差带应限制在 5% 以内。

(3) 确定参数的验证值

参数的验证值可以通过工程分析、仿真、试验得到。在研制阶段初期, 工程分析方法是基

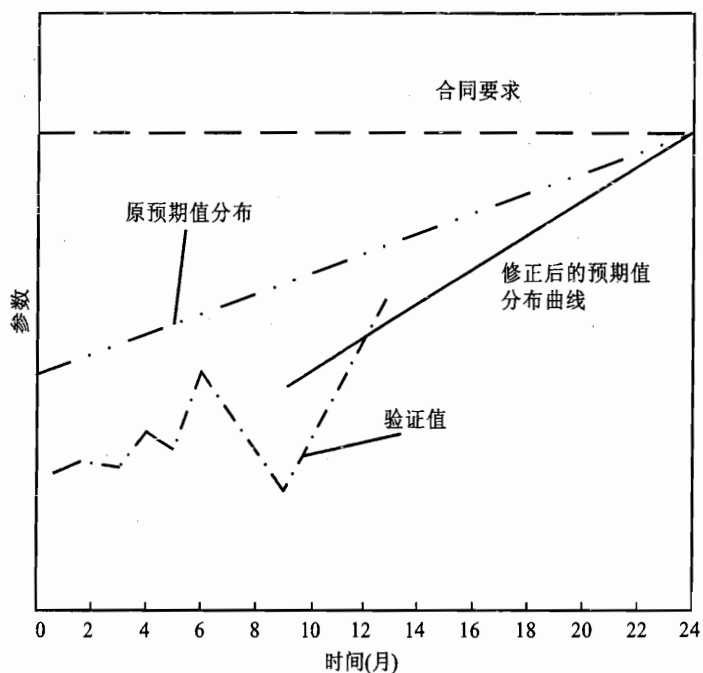


图 6.18 技术性能测量状况

本的方法。因为那时,大量的设计工作还停留在纸面设计阶段。随着研制工作的进展,可通过仿真或缩比模型试验(如风洞模型试验等)得到较为准确的验证数据。当进入工程研制阶段后,尤其进入该阶段后期,随着各种地面试验、原型机试验的进行,就可获得更为准确的验证值数据。

在工程系统各发展阶段,用以估计主要参数验证值的各种可行方法如表 6.11 所列。

表 6.11 技术性能测量主要参数的验证方法

阶 段 验证方法		初步设计	工程研制(前期)	工程研制(后期)	生 产
工程分析		✓	✓		
仿真		✓			
模型 试验	缩比模型	✓	✓		
	数学模型		✓	✓	
	部件试验		✓		
	分系统试验			✓	✓
	系统试验			✓	✓



(4) 绘制参数的验证值分布曲线

根据各阶段测得的验证值,可逐段绘制出验证值分布曲线(图 6.18 中的单点划线所示曲线)。这是每个主要参数实际数值的分布曲线。设计人员可以根据验证值分布曲线的情况来修正原来的参数预期值分布曲线(见图 6.18)。

(5) 求得参数的验证偏差

根据参数的预期值和验证值即可求得验证偏差的大小,并检查该验证偏差是否落在容差带允许的误差范围之内。

(6) 根据参数的预期值分布曲线、验证值分布曲线和表 6.12 所列的各种预测方法,预测出该参数的现时估计值大小。

在工程系统各发展阶段,用以预测主要参数现时估计值的方法如表 6.12 所列。

表 6.12 技术性能测量主要参数的预测方法

阶 段 预测方法	初步设计	工程研制(前期)	工程研制(后期)	生 产
比较法	√	√		
数学模型		√	√	
仿真模型		√	√	√
经验方法		√	√	√
决策方法	√	√		

(7) 确定预测偏差

根据参数的规范要求值和现时估计值即可确定该参数的预测偏差 h 的大小。

(8) 检查预测偏差是否满足合同或规范中规定的参数指标及其门限值,并确认出每一个超出容差带范围或门限值的参数。

(9) 对超出容差带范围或门限值的参数进行偏差分析和评估,包括:

- ① 分析偏差对其他性能参数及整个系统性能的影响;
- ② 分析偏差对预期值分布曲线及容差带的影响;
- ③ 分析由于要消除该偏差而需要进行的设计更改对技术状态变更和生产的影响;
- ④ 估计由于采取消除偏差的纠正措施对费用和进度的影响。

分析和评估结果应写成《参数偏差评估报告》,并上报有关部门。《参数偏差评估报告》的标准格式如表 6.13 所列。

表 6.13 参数偏差评估报告

参数偏差评估报告			
项目 发动机	WBS 项目件号 011200		
性能参数偏差			
主要参数	预期值	验证值	偏差
推力/kg	12 000	10 500	-1 500
偏差对系统性能的影响			
性能特征值	规范规定值	现时估计值	偏差
最大飞行速度/(km/h)	2 000	1 800	-200
消除偏差对费用的影响			
计划费用	预期费用	费用增长	用途
6 200 万元	6 350 万元	150 万元	• 提高增压比 • 提高涡轮后温度
消除偏差对进度的影响			
计划项目	计划进度	预测进度	进度变化
发动机首件交付	2002. 9. 30	2002. 12. 30	推迟 3 个月

(10) 采取纠正措施

当参数的现时估计值低于规范要求值并超出其门限值时,可采取的纠正措施为:

① 重新分配较低层次的参数,以保证系统级最高层次的参数仍能达到规范要求值。但这一途径是否能实现将取决于工程系统是否留有足够的储备(余量)。这些储备包括时间、资源以及技术的储备。

② 若上述途径不可行,则应考虑采用替代方案(包括修改原来的设计方案,采用其他的硬件和软件等)。

如上所述,技术性能测量的作用就是及早提出飞行器系统技术性能的预报,及时发现技术缺陷,并在其对费用、进度造成的影响尚小时就采取对策,使未来交付的飞行器系统的性能能满足规范的要求。

6.3.4 技术性能测量报告

技术性能测量报告是获得工程系统的技术性能信息的主要途径。必须对每一个被监控的主要参数定期地提交技术性能测量报告。

1. 报告内容

技术性能测定报告应报告如下数据:

- 主要参数名称及其对应的 WBS 项目件号；
- 参数的预期值；
- 参数的验证值；
- 验证偏差；
- 参数的规范要求值；
- 参数的现时估计值；
- 预测偏差；
- 报告时间及其对应的阶段里程碑或评审点。

此外,还可包括取得验证值的手段及其置信度的分析以及对现时估计值预测的分析。

2. 报告格式

技术性能测量报告可以采用表格形式(见表 6.14)或图表形式(见图 6.19)。一般,每一个列入技术性能测量主要参数清单中的参数都应有一份单独的技术性能测量报告。

表 6.14 技术性能测定报表

项目 发动机		WBS 项目件号 011200					
参 数	本次测定的里程碑日期			最终产品			
	预期值	验证值	验证偏差	现时估计	规范要求值		预期偏差
					最初	现在	
推力/kg	12 000	10 500	-1 500	12 200	13 000	13 000	-800
单位耗油率/(kg/h—kg)	0.05	0.045	-0.005	0.04	0.04	0.04	0
重量/kg	2 200	2 150	-50	1 950	2 000	2 000	-50
压气机增压比	7.5 : 1	7.8 : 1	+0.3	8 : 1	8 : 1	8 : 1	0
涡轮效率	87	85	-2	87	87	87	0
报告日期 2002. 9. 30 阶段里程碑(或评审点) _____					报告部门		
					发动机工程部		

如果验证偏差超出了容差带范围,或现时估计值低于规范要求指标的门槛值,则应同时提交一份该参数的《参数偏差评估报告》。该报告的格式与内容如表 6.13 所列。

3. 报告频度

技术性能测量报告应定期提交。在可能的情况下,报告提交的时间应当与阶段里程碑和预期值分布曲线中对应的评估点日期相一致。一般,技术性能测量的评估点应与研制阶段中的技术评审点(从系统设计评审开始到功能技术状态审核)、重要试验任务完成点(如功能试验完成、硬件验证试验完成、环境试验完成、可靠性试验完成、计算机软件测试完成等)和阶段决策里程碑相一致。因为在这些关键时刻,也是工程系统管理部门要求提交各种鉴定、评估、评

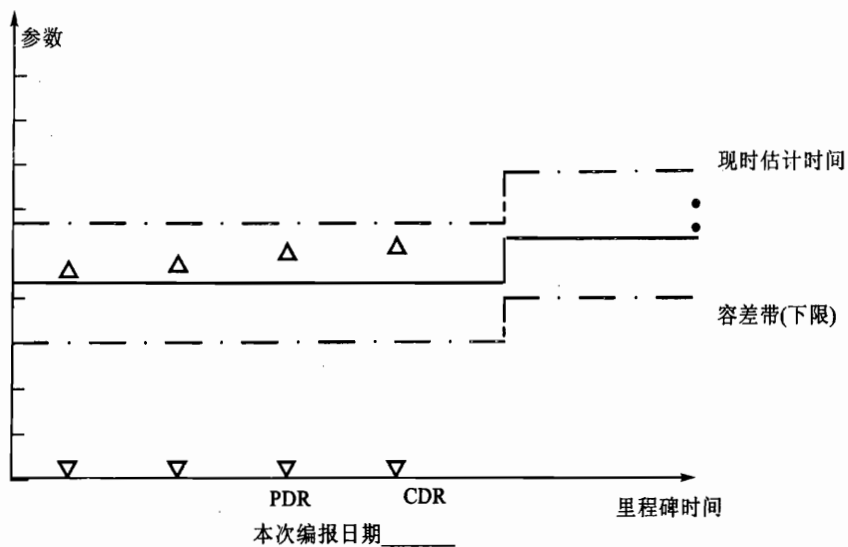


图 6.19 技术性能测量报告的图表格式

审报告的时间。

如果上述关键点之间的时间间隔过长,则应至少每 3 个月进行一次技术性能测量并提交相应的报告。

第7章 风险分析

7.1 概 述

现代工程系统(特别是飞行器系统)的研制,既要采用先进的技术,又需要耗费数以亿计的资金和长达数年的时间。因此,实现工程系统任务目标的风险是工程系统规划和设计中必须予以考虑的重要因素。现代工程系统的规划和设计中已经把风险作为与技术性能、费用、进度同等重要的型号要素。在工程系统的各个发展阶段,都必须对风险予以分析和管理的。

在工程系统的规划和设计中需要考虑风险因素是由于现代工程设计的目的是研究和创造新的以前尚未存在过的对象、过程或系统。虽然在设计中可以利用已知、现有的技术和产品,但设计人员仍需进行创造性的探索,用新的概念和方法设计出优异的系统或产品。因而,工程设计的过程是创造性的过程,也是探索未知的过程。在工程系统的研制过程中,未知量向已知量的转化过程可参见图 7.1。图中所示的“已知的未知量”是指在工程设计开始时,就已经预料或预测到的会突然发生问题的量;“未知的未知量”是指尚不能预料或预测到的会突然发生问题的量。随着工程研制工作的深入开展,出现了由未知的未知量向已知的未知量的转化和由已知的未知量向已知量的转化。如果在设计的工程系统中采用了过多的尚未经过验证的新技术、新成品、新工艺,亦即未知量或不确定的因素过多,就有可能达不到预定的任务目标;或者是达不到预定的性能要求;或者是研制费用大大超过预算;或者是研制周期拖延。这种达不到预定目标的概率就叫做风险。

风险是普遍存在的。在工程系统的研制、科学研究、军事活动、工商业投资,甚至在人们的日常生活中都存在风险。因而风险分析也就发展成为一门新的、独立的学科。

在工程设计中,风险分析由 3 个步骤组成,如图 7.2 所示。

(1) 辨识潜在的风险项目,亦即判断在所研制的工程系统中,哪些方面存在风险。

(2) 定量地估计风险,并将风险项目分类(一般分为高风险、中等风险、低风险 3 类),以确定其中最关键的项目。

(3) 确定各种使风险最小的途径和判定何时开始或结束这些活动。

如图 7.2 所示,风险分析是一个迭代的过程,该过程要确定的是:什么地方有风险? 这些风险有多大? 它的影响或后果如何? 对此应当采取什么行动? 因此,风险分析要解决的是:

- 辨识有潜在问题的领域;
- 量化与这些潜在问题有关的风险;
- 估计这些风险影响的大小并采取各种适当的行动来降低风险。

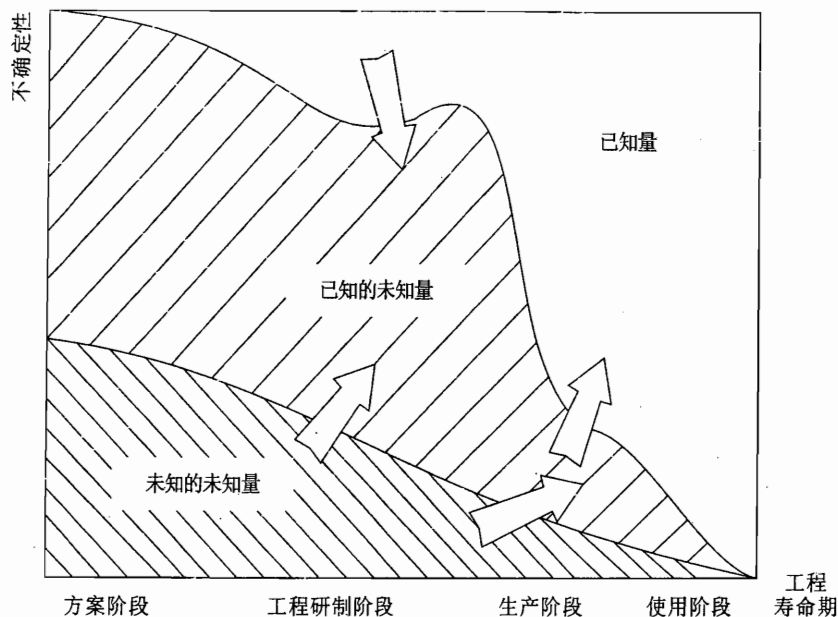


图 7.1 工程系统研制过程中未知量向已知量的转化过程

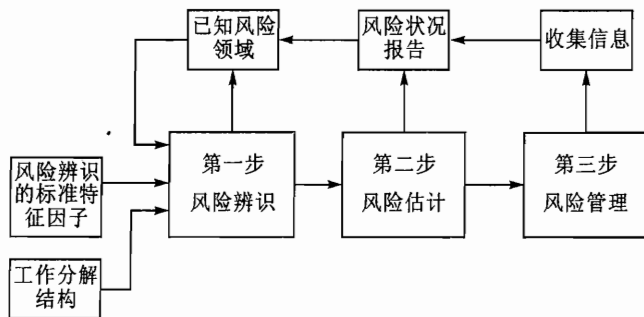


图 7.2 风险分析流程图

本章将分别阐述与风险有关的术语和定义、风险的辨识、风险的估计和风险管理。

7.2 与风险有关的术语和定义

7.2.1 风险和风险因子

风险是达到一个基准目标的不确定性。所谓基准目标，可以是“一项规范规定的技术要求”、“一个性能指标”、“某个 LCC 指标值”等。通俗地说，风险就是发生不幸事件的概率。

风险内涵两个相互影响的变量：失败的概率(P_f)和失败造成的后果(C_f)。因此，风险 $R=$



$f(P_i, C_i)$ 。

失败的概率 P_i 是指所研制的系统、分系统、部件或元件的工作不成功的概率, 即:

$$P_i = 1 - P_s$$

式中: P_s 是成功的概率 ($P_i + P_s = 1$)。

失败的后果 C_i 是指由于所研制的系统、部件或元件的工作不成功而导致其失去效用的程度, 即

$$C_i = 1 - C_s$$

式中: C_s 是成功的结果或效用 ($C_i + C_s = 1$)。

由此, 可以导出风险因子 RF 的表达式:

$$RF = \text{失败的似然估计} = 1 - \text{成功的似然估计} = 1 - P_s C_s = 1 - (1 - P_i)(1 - C_i)$$

即

$$RF = P_i + C_i - P_i \cdot C_i \quad (7.1)$$

式(7.1)表示风险因子 RF 是 P_i 和 C_i 两个集的并。因此, 风险因子在 P_i 和 C_i 均为最大值时达到最大。

7.2.2 风险分类

通常, 风险可划分为低风险、中等风险和高风险 3 个级别。它们的定义为:

- 低风险: 是指可以辨识并可以监控其对工程目标影响的风险。这种风险发生的概率相当低, 其后果也无关紧要, 故除了通过正常的设计部门对其加以监控外, 不需要采取其他的专门措施来处理该类风险。
- 中等风险: 是指可以被辨识的并对工程系统的技术性能、费用或进度将产生较大影响的风险。这类风险发生的概率相当高, 以致需要对其进行严密的监视和控制。应当在各个设计阶段的设计评审中对这类风险进行评审, 并应采取适当的手段或行动来降低风险。
- 高风险: 是指发生的概率很高, 其后果将对工程有极大影响的风险。这种风险只有在单纯的研究工作, 或工程研制中的方案阶段中方可允许存在, 而对一个进入工程研制阶段的工程项目则是不能允许的。必须严密地监控每一个高风险领域, 必须强制性地执行使风险降低的计划。对这类风险应当定期地报告和评审。

一般规定: $RF < 0.3$ 为低风险; $0.3 \leq RF \leq 0.7$ 为中等风险; $RF > 0.7$ 为高风险。

7.2.3 等风险曲线

图 7.3 所示是一族等风险曲线。它是由给定的一系列 P_i 和 C_i 值组成的。

用式(7.1)的风险因子 RF 表达式可以得出常值的 RF。在一个直角坐标系上, 将给定的 P_i 和 C_i 值画出。其中, 横坐标为 C_i , 其值从 0 到 1; 纵坐标为 P_i , 也从 0 到 1。将求得的 RF 等值点相连, 即组成了风险因子 RF 的等值线。利用风险等值线可以把所有的风险项表示在一

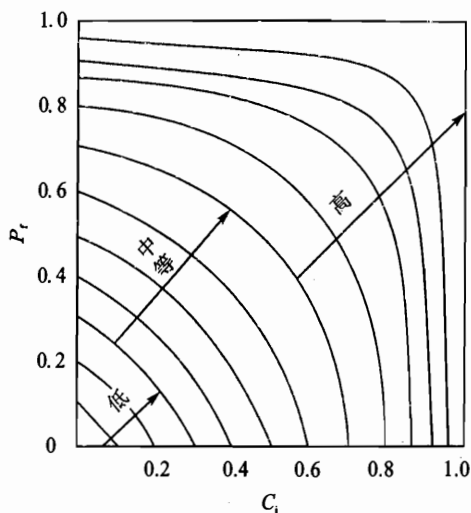


图 7.3 等风险曲线族

张图上。可以用它来确定与工程发展有关的各风险项目的风险级别及各类风险之间的边界。亦即可以从图中看出所有有关的风险项目处于哪个风险区域之中。

7.3 风险的辨识

风险分析的第一步就是进行风险辨识。风险辨识的任务是要确定整个工程系统中有潜在问题的项目,亦即要回答:在所研制的工程系统中,哪些地方有风险?引起这些风险的主要因素是什么?这些风险造成的后果如何?

完成这种辨识是否充分、彻底将决定风险分析和 Risk 管理的效率。

工程系统的寿命期内,在计划、技术、试验、生产、后勤等方面总是存在一定的风险。

计划风险包括资金、进度、合同履行情况等风险。

技术风险包括可能采用的新技术和性能要求方面的风险(如高精度、高速度、高机动性等要求),同时还涉及设计概念的可行性,或采用新成品和软件带来的风险。

工程风险与可靠性、维修性、安全性等方面的风险有关。

因此,在进行风险辨识时,应当从任何一个能辨认出潜在问题的信息源中进行风险辨识。这些信息源是:

- 类似工程项目中的经验和教训;
- 规范和技术要求文件;
- 技术性能测量(TPM)计划及其分析结果;
- 进度计划及其执行情况;
- 寿命期费用(LCC)的分析和基线费用估计。

在进行风险辨识时,应对工程系统所有的方面进行评估,以便发现其潜在的风险。对风险的搜索必须彻底,以免漏掉重要的项目。为了确保考察并详细地调查到所有的领域,通常可以采用“分解方法”,亦即利用工作分解结构(WBS)这一手段来帮助进行风险辨识。工作分解结构的概念可参见本书第6章。下面仍以图6.11所示的武器系统的WBS为例进行风险辨识。

在进行风险辨识时,可利用该WBS逐层逐项地进行风险评估。一般的工程系统,应对WBS中从第一层到第四层的所有项目都要进行风险评估,对某些重大的工程系统则应分析到第五层的所有项目。

具体的方法是,首先将WBS中的项目按次序列到一张风险评估清单中(参见表7.1,鉴于篇幅所限,该清单中只列出了第一到第四层次中的部分项目)。然后,请该类工程项目富有经验的专家来评估各个项目的风险状况。为了提高评估的准确性,避免完全由专家的主观判断来评估风险,可以在清单中列出每个项目的原准设计产品是什么?现在的设计在该原准基础上作了哪些改进?这样就可以从特性相似的现有事件的客观概率外推而得到较为准确的估计。

表 7.1 风险评估清单

工程项目名称: 军用飞机系统										
WBS 层次				WBS 工作单元号	工作单元 名称	设计原准	与原准相 比的变化	风险评估		
I	II	III	IV					P_i	C_i	风险 级别
√				100000	军用飞机系统					
	√			110000	飞机					
		√		111000	机体					
		√		112000	航空电子系统					
		√		113000	发动机					
			√	112100	飞控系统					
			√	112200	军械系统					
			√	112300	火控系统	F-16	少量改型	0.2	0.1	低
				⋮	⋮					
填表人 _____ 职务/职称 _____ 专 业 _____										

请专家评估工程项目的风险时,应事先找出对该工程项目适用的风险特征因子,以使用一个统一的标准来评估各个项目的风险级别。风险特征因子可参见表7.2和表7.3。专家们应当评估清单中每个工作单元(项目)的风险级别。最好能给出定量的失败概率 P_i 和失败的后果 C_i 值,以使用7.4节中给出的风险估计流程图算出风险因子的大小。

由上述方法得出的专家评估意见不是直接基于大量试验、分析的客观结果,但也不是完全由某个人主观估计的结果,而是从特性相似的现有事件的客观概率外推而得到的估计。为了使专家评估的意见更为合理,可以采用现在国内外常用的一些专家意见调查法,如“德尔菲方法”。

7.4 风险估计

风险估计就是对风险进行定量的分析,也就是回答:所研制工程系统的风险有多大?本节将分别阐述技术风险、进度风险和费用风险的定量估计方法。

7.4.1 技术风险的估计

估计技术风险应当考虑两个因素:失败的概率 P_f 和失败的后果 C_f 。 P_f 可视为是与硬件、软件的成熟性、复杂性等特性有关。此外, P_f 还取决于与它相关联的接口项目。表 7.2 给出了与之有关的 5 个特征因子,对这 5 个不同的特征因子取均值即可得到 P_f 值。

$$P_f = \frac{P_{Mh} + P_{Ms} + P_{Ch} + P_{Cs} + P_D}{5} \quad (7.2)$$

式中: P_{Mh} 为与硬件成熟程度有关的失败概率; P_{Ms} 为与软件成熟程度有关的失败概率; P_{Ch} 为与硬件复杂程度有关的失败概率; P_{Cs} 为与软件复杂程度有关的失败概率; P_D 为与其他项目相关程度有关的失败概率。

若所研制的工程项目中不包括软件,则与软件有关的两个因子 P_{Ms} 、 P_{Cs} 可以略去,并将式(7.2)的分母改为 3。

例 7.1 某一工程项目的设计采用了现有的硬件,但软件的主要部分需要重新设计,使其适合现在的工程要求。硬件属于简单设计,软件的复杂程度中等,且该软件委托给某软件开发公司研制,因而工程进度与软件性能将取决于该合同单位。如是,可根据表 7.2 取各特征因子值如下:

$$P_{Mh}=0.1; P_{Ms}=0.5; P_{Ch}=0.1; P_{Cs}=0.5; P_D=0.5$$

故

$$P_f = \frac{0.1+0.5+0.1+0.5+0.5}{5} = 0.34$$

表 7.2 特征因子对失败概率 P_f 的影响

特征因子 量 值	成熟性因子 P_M		复杂性因子 P_C		相关因子 P_D
	硬件 P_{Mh}	软件 P_{Ms}	硬件 P_{Ch}	软件 P_{Cs}	P_D
0.1	现有硬件	现有软件	简单设计	简单设计	与现有系统、设施 或 相 关 承 包 商 无 关



续表 7.2

特征因子 量 值	成熟性因子 P_M		复杂性因子 P_C		相关因子 P_D
	硬件 P_{Mh}	软件 P_{Ms}	硬件 P_{Ch}	软件 P_{Cs}	P_D
0.3	局部重新设计	局部重新设计	局部增加复杂性	局部增加复杂性	进度取决于现有系统进度、设施或相关承包商
0.5	主要部分重新设计,但技术可行	主要部分重新设计,但技术可行	复杂性有中等程度增加	复杂性有中等程度增加	性能取决于现有系统性能、设施或相关承包商
0.7	技术可行的复杂设计	新软件,但与现有软件有类似之处	复杂性显著增加	复杂性显著增加,程序块数目有中等程度增加	进度取决于新系统的进度、设施或相关承包商
0.9	最新技术	最新技术,以前从未做过	极其复杂	高度复杂,数据库极大,使用操作复杂	性能取决于新系统的性能、设施或相关承包商

若上述 5 个特征因子的重要程度不同,则可采用加权系数表示其不同的重要性。这时,失败概率 P_f 的表达式为

$$P_f = W_{Mh} \cdot P_{Mh} + W_{Ms} \cdot P_{Ms} + W_{Ch} \cdot P_{Ch} + W_{Cs} \cdot P_{Cs} + W_D \cdot P_D \quad (7.3)$$

式中: W_{Mh} 、 W_{Ms} 、 W_{Ch} 、 W_{Cs} 、 W_D 分别为硬件和软件的成熟程度、硬件和软件的复杂程度以及相关程度的加权系数。这 5 个加权系数之和应等于 1,即

$$W_{Mh} + W_{Ms} + W_{Ch} + W_{Cs} + W_D = 1 \quad (7.4)$$

例如例 7.1 中,分别取 $W_{Mh} = 0.2$, $W_{Ms} = 0.2$, $W_{Ch} = 0.1$, $W_{Cs} = 0.3$, $W_D = 0.2$,则由式(7.3)可得 $P_f = 0.38$,与取平均值时的 P_f 略有不同。

下一步应考虑失败的后果 C_f 。当风险发生时,必然会对工程系统的技术性能、费用和进度产生影响,同样可用特征因子的方法来确定 C_f 值。表 7.3 中给出了 3 个与失败的后果有关的特征因子,通过计算其平均值即可求得 C_f 的大小。

$$C_f = \frac{C_t + C_c + C_s}{3} \quad (7.5)$$

式中: C_t 为由技术因素引起的失败后果; C_c 为由费用的变化引起的失败后果; C_s 为由进度的变化引起的失败后果。

例 7.2 前述的工程项目,经评估后认为存在下述原因引起的失败后果 C_f : 技术上存在某些问题,但只对技术性能有很小影响;解决上述技术问题使费用增加 8%,使进度延后 5 周。如是,可根据表 7.3 取各特征因子值为

$$C_t = 0.3; \quad C_c = 0.5; \quad C_s = 0.5$$

故

$$C_i = \frac{0.3 + 0.5 + 0.5}{3} = 0.43$$

表 7.3 特征因子对失败后果 C_i 的影响

特征因子 量 值	技术因子 C_t	费用因子 C_c	进度因子 C_s
0.1(低)	对技术性能影响极小或无影响	费用略有变动,但不超出预算估计	对进度的影响极小,可忽略不计,或可由进度松弛时间补偿
0.3(较次要)	技术性能下降很小	费用超出预算 1%~5%	部分进度延后,但小于 1 个月,需对进度计划作某些调整
0.5(中等)	技术性能有某种程度下降	费用估计增加 5%~20%	进度延后 1 月以上,3 个月以下
0.7(显著)	技术性能显著下降	费用估计增加 20%~50%	进度延后超过 3 个月
0.9(高)	技术性能指标不能达到	费用估计增加 >50%	进度大为推迟,并影响系统的里程碑进度

同理,在计算 C_i 值时,也可根据对技术性能、费用、进度影响的重要程度不同予以加权。考虑加权后的 C_i 计算公式为

$$C_i = W_t \cdot C_t + W_c \cdot C_c + W_s \cdot C_s \quad (7.6)$$

式中: W_t 、 W_c 、 W_s 分别为技术性能、费用、进度的加权系数。

这 3 个加权系数之和应等于 1,即:

$$W_t + W_c + W_s = 1 \quad (7.7)$$

例如,例 7.2 中,取 $W_t = 0.4$, $W_c = 0.4$, $W_s = 0.2$,则根据式(7.6)可得 $C_i = 0.42$ 。

确定了 P_i 和 C_i 后,即可由式(7.1)和图 7.4 中所示的风险估计流程图计算出风险因子 RF 的数值。

$$RF = 0.34 + 0.43 - 0.34 \times 0.43 = 0.624$$

此 RF 值表明该工程项目的风险级别为中等。

图 7.4 中还指出了不同的风险级别所应采取的风险管理方法。

从图 7.3 的等风险曲线族图上可查得上述算例中 $P_i(0.34)$ 和 $C_i(0.43)$ 对应的 RF 点介于 0.6 和 0.7 这两条等风险曲线之间,属于中等风险范围。

如上所述,可根据表 7.2、表 7.3 中给出的特征因子值,并利用图 7.4 中的风险估计流程图定量地计算出风险的取值。

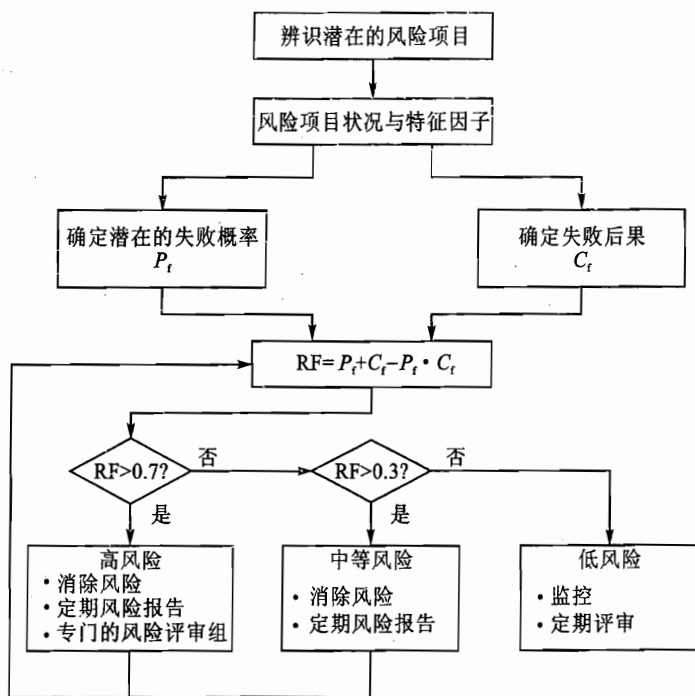


图 7.4 风险估计流程图

7.4.2 进度风险的估计

进度风险取决于下述诸因素。

(1) 技术风险的大小

技术风险会对进度产生影响，一般认为：

- 若工程项目的技术风险中无高风险项目，而且中等风险的技术项目不超过 2 个，则对进度的影响很小。
- 若技术风险中无高风险项目，但有 3 个以上的中等风险项目，则对进度有中等程度的影响。
- 若技术风险中有 1 个以上的高风险项目，则会对进度有严重影响。

(2) 计划安排的合理性

- 若进度计划按工程进展的需要安排，切实可行且留有一定的时间裕度，以防意外情况发生，则这类计划属于低风险。
- 若进度计划可靠，但对意外发生的情况未事先留有裕度，则进度计划为中等程度风险。



- 若进度计划不可靠,且不是根据每项研制工作的实际需要时间来安排,而是根据竞争的需要或上级指令,先定下最后完工日期,再倒过来分配每项工作应完成的时间。这种进度计划属于高风险。

经验和统计结果表明:如果人为地将进度压缩,使研制时间少于其额定进度的80%,则该研制项目通常会失败。

(3) 资源的充分性

这里所指的资源包括资金、人力、器材、设备、设施等项。

- 若资源充足且可供使用,则进度计划为低风险。
- 若现有资源充足,但与其他工程项目之间存在潜在的冲突和矛盾,从而可能会因某些意想不到的问题而影响进度。在这种资源条件下,进度计划有中等程度的风险。
- 资源不足且与现有的其他工程项目之间存在严重的潜在冲突,则对进度有极大影响。

(4) 人员的经验

- 若参与该工程项目的人员在类似的工程项目中已积累了经验,有足够的知识储备可用于该项目,则进度计划成功的可能性很大。
- 若参与人员在类似的项目中已有一般性的经验,但在某些关键部门缺乏有经验的人员,则进度有中等程度的风险。
- 若参与人员普遍没有在类似项目中工作的经验,关键部门可用的有经验人员很少,则进度有高风险。

(5) 成品厂(与分承包单位,下同)的交货历史

- 若成品厂交货信誉很好,管理水平较高,则对工程进度影响很小。
- 若成品厂交货状况较好,管理水平尚可,但交货状况有时不稳定,则对进度会产生中等程度的影响。
- 若成品厂交货状况不好,管理混乱,则对进度会产生严重影响。

(6) 企业的管理状况与水平

- 若企业管理人员有很强的进取精神,且采用科学的管理和决策方法,则计划成功的可能性很大。
- 若企业管理人员认真负责,但在复杂情况下处理问题的能力和决策水平较差,则进度有中等程度的风险。
- 若企业管理人员素质差,管理混乱,则进度有高风险。

根据上述分析结果,可以用表7.4所列的进度风险估计表来定性或定量地估计进度风险因子 RF_s 的大小。

表 7.4 进度风险估计表

工程项目名称 _____				
序 号	进度风险因子	风险级别	本项目状况	
			定性	定量
1	技术风险状况 R_{ST} • 无高风险,且中等风险项目不超过 2 个 • 无高风险,但中等风险项目为 3 个以上 • 有 1 个以上高风险项目	低 中 高	✓	0.3
2	计划安排的合理性 R_{SS} • 进度计划可靠并留有裕度 • 进度计划可靠但未留裕度 • 进度计划不可靠	低 中 高	✓	0.1
3	资源的充分性 R_{SC} • 资源充分且可供使用 • 资源充分但与其他项目的潜在矛盾 • 资源不足,或可用性很差	低 中 高	✓	0.5
4	人员的经验 R_{SP} • 参与人员富有经验 • 参与人员有一般性经验 • 参与人员普遍无经验	低 中 高	✓	0.2
5	成品厂的交货状况 P_{SV} • 成品厂交货信誉好,管理得力 • 成品厂交货状况较好,管理认真 • 成品厂交货信誉不好,管理混乱	低 中 高	✓	0.4
6	企业管理状况 R_{SM} • 管理人员有进取精神,采用科学管理 • 管理人员认真、负责,但决策能力不够强 • 管理混乱,管理人员不可靠	低 中 高	✓	0.3
备 注	填表人 _____ 填表日期 _____			

进度风险因子 RF_s 为

$$RF_s = \frac{R_{ST} + R_{SS} + R_{SC} + R_{SP} + R_{SV} + R_{SM}}{6} \quad (7.8)$$

式中: R_{ST} 为技术风险影响因子; R_{SS} 为计划合理性因子; R_{SC} 为资源充分性因子; R_{SP} 为项目人员的经验因子; R_{SV} 为成品厂状况的影响因子; R_{SM} 为企业管理状况的影响因子。

以表 7.4 中所估计的各风险因子值,由式(7.8)可求得该工程项目的进度风险因子为

$$RF_s = \frac{0.3+0.1+0.5+0.2+0.4+0.3}{6} = 0.30$$

故该工程项目的进度风险为低风险。

7.4.3 费用风险的估计

费用风险取决于下述诸因素。

(1) 工程系统任务要求的明确性

- 若任务要求很明确,且使用方和承制方对任务要求有共同的理解,则费用为低风险。
- 若任务要求基本明确,但在某些细节上尚需进一步确定,则费用风险为中等。
- 若任务要求不明确,使用方可能不断提出新的要求,或双方对任务要求有不同的理解,则费用风险属高风险。

(2) 技术风险对费用的影响

- 若技术风险中无高风险项目,中等风险项目也不超过 2 个,则对费用影响很小。
- 若技术风险中无高风险项目,中等风险项目在 3 个以上,则对费用有中等程度影响。
- 技术风险中有 1 个以上的高风险项目,则对费用将产生严重影响。

(3) 进度风险对费用的影响

进度的拖延必然会使费用增加,故进度风险会对费用产生影响。

- 若进度风险中无高风险项,中等风险的进度指标(如 R_{ST} 、 R_{SS} 等)不超过 2 个,则对费用影响很小。
- 若进度风险中无高风险项,但中等风险项目在 3 个以上,则对费用有中等程度的影响。
- 若进度风险中有 1 个以上的高风险项目,则对费用有严重影响。

(4) 成本预算准确性的影响

- 在作成本预算时,有充分的类似工程项目的历史数据可供参考,且成本估算部门有足够可用的合格人员,则费用风险较小。
- 在作预算时,有足够可用的合格人员,但仅有部分历史数据可供参考,则认为费用风险为中等。
- 在作预算时,缺乏可用的合格人员,且无类似项目的历史数据供参考,则费用风险属高风险。

(5) 工程项目合同的类别

- 若采用固定价格合同,则费用风险最低。
- 若采用成本加奖励费合同,则费用风险中等。
- 若采用拨款性合同,则费用风险最高。



(6) 合同报价的状况

- 若合同报价与其他竞标单位的报价和预测成本基本相符,则费用风险为低风险。
- 若合同报价略低于其他竞标单位报价和预测成本,则费用风险为中风险。
- 若合同报价显著低于其他竞标单位报价和预测成本,则费用风险为高风险。

根据上述分析结果,可利用表 7.5 所列的费用风险估计表来估计费用风险。

表 7.5 进度风险估计表

工程项目名称 _____				
	费用风险因子	风险级别	本项目状况	
			定性	定量
1	任务要求明确性因子 R_{CR} • 任务要求明确 • 任务要求基本明确 • 任务要求不明确	低 中 高	✓	0.4
2	技术风险影响因子 R_{CT} • 无高风险,中等风险项目不超过 2 个 • 无高风险,但中等风险项目有 3 个以上 • 有 1 个以上高风险项目	低 中 高	✓	0.3
3	进度风险影响因子 R_{CS} • 进度风险中无高风险项,中等风险项目不超过 2 个 • 进度风险中无高风险项,但中等风险项目有 3 个以上 • 有 1 个以上高风险项目	低 中 高	✓	0.3
4	成本预算准确性因子 R_{CC} • 有历史数据参考,成本估算人员合格 • 有部分数据参考,成本估算人员合格 • 无类似项目数据参考,缺乏成本估算合格人员	低 中 高	✓	0.4
5	合同类型影响因子 R_{CN} • 固定价格合同 • 成本加奖励费合同 • 拨款性合同	低 中 高	✓	0.2
6	合同报价影响因子 R_{CB} • 与预测成本及其他单位报价基本相符 • 略低于预测成本及其他单位报价 • 显著低于预测成本及其他单位报价	低 中 高	✓	0.1
备注	填表人 _____ 填表日期 _____			

由表 7.5 可得费用风险因子 RF_c 为

$$RF_c = \frac{R_{CR} + R_{CT} + R_{CS} + R_{CC} + R_{CN} + R_{CB}}{6} \quad (7.9)$$

式中: R_{CR} 为任务要求明确性因子; R_{CT} 为技术风险影响因子; R_{CS} 为进度风险影响因子; R_{CC} 为成本预算准确性因子; R_{CN} 为合同类型影响因子; R_{CB} 为合同报价影响因子。

以表 7.5 中所估计的各风险因子值, 由式(7.9)可求得该工程项目的费用风险因子为

$$RF_c = \frac{0.4 + 0.3 + 0.3 + 0.4 + 0.2 + 0.1}{6} = 0.28$$

故该工程项目的费用风险为低风险。

7.5 风险管理

在风险已被辨识和定量估计之后, 必须对已辨识的风险项目进行风险管理。风险管理的主要任务是采用控制和降低风险的技术与方法使风险降低到可以接受的程度。如果没有用来控制已辨识风险的措施, 那么也就没有风险管理。

7.5.1 降低或控制风险的技术

降低或控制风险技术有以下几类。

1. 回避风险(avoiding risk)

回避风险的方法就是将风险的影响尽量降低到最低限度。为了回避风险, 就应当避免发生失败的可能性和潜在的失败造成的后果。在工程项目管理中, 回避风险可以反映在系统设计概念的选择上和对承制部门(承包单位)的选择中。亦即尽可能地选择风险最低的系统设计方案和承包单位。在选择承包单位的过程中, 工程项目负责人应当确认各个投标单位的缺点, 从而避免选定的承包单位具有不可接受的风险。但是, 并不是每一个风险都能完全回避的。回避风险的措施可以简单地将其转移到其他领域或其他部门中去。

2. 控制风险(controlling risk)

控制风险是一个连续地判定工程的情况, 制定供选择的降低风险的方案, 再监控其风险状况的反馈过程。其目的是要在多个方案中选出风险较低的方案。例如, 制定不同的生产方案, 对关键的部件进行平行的开发, 对关键器材的研制给予优先等。制定这些方案的时间和费用要求决定了管理部门选择的监控手段所需的灵敏度。一般的监控手段可以由技术性能测量(TPM, 详见第6章)和费用-进度控制系统提供。

3. 承担风险(assumption of risk)

承担风险是在对存在的风险有透彻的了解后, 决定接受这些风险发生后所造成的后果。一般, 在工程项目的研制中都必须承担某些风险。但是, 承担的风险必须恰当。工程项目负责人应当分析并确定在该部门的能力范围内能承担的恰当风险。如果承担的风险过多, 则应以

适当的费用把一部分风险转移给别的部门或其他承包单位。

4. 转移风险 (transfer of risk)

转移风险是辨识与潜在风险有关的领域并与其他部门分担风险的过程。其方式一般是通过协议书使各有关部门同意分担风险。通常应由风险最高的部门支付一定费用,通过担保、费用刺激、成本加奖励合同等,将由于失败造成的损失由承制部门以某种方式的“代价”承担下来。这种“代价”可以是费用的增加、进度的改变或产品性能的更改。

7.5.2 减小风险的途径

工程系统的风险取决于:

(1) 工程系统任务要求的明确性。这是指工程项目所要完成的任务要求是否已规定得很明确,使用方和承制方对这些要求是否有一致的理解。注意:决不能被迫接受一项任务要求不明确、没完没了、带有很大风险的合同。

(2) 技术性能的风险。亦即是否有能力根据以往完成类似工程的经验和现有的技术储备,用给定的资源满足本工程系统的技术性能要求。

(3) 进度风险。

(4) 费用风险。

针对上述可能发生的风险,将在下文中分别给出减小上述风险的途径。

1. 及早明确工程系统的任务要求

(1) 工程系统任务要求完全确定的时间不能晚于方案确定阶段结束的时间。

(2) 使用部门必须以下述文件形式提出明确的任务要求:

- 工作说明(SOW): 阐述研制部门的任务和工作范围;
- 技术要求文件(TRD): 提出系统的功能与性能技术要求。

在上述任务要求中,应包括完整的质量要求和环境要求。

(3) 主要的任务要求或设计要求应当以可度量的参数的形式加以表述。这些参数在设计过程中可以通过试验或模拟加以测量。概率统计类型的要求(如可靠性、维修性要求等)应通过扩展的系统级的试验来验证它们是否满足设计要求。

(4) 系统、分系统、部件的设计技术要求一定要分别以系统规范、分系统规范、部件规范等文件的形式加以确定。规范是一系列技术要求的陈述,在规范中对设计要求的规定应当明确、清晰,不允许有含糊不清、模棱两可、不确定、重复的陈述。

(5) 主承制单位有责任保证使分承包单位和成品厂商获得完整、确定的设计要求。这些设计要求应当符合使用方的要求。

(6) 在方案确定阶段结束时,应由使用方会同承制方制定工程系统正式的研制任务书,以确定工程系统的任务、要求、数量、进度和经费,并上报国家有关部门批准。研制任务书一旦批准,使用方的任务要求即行冻结。在以后的研制过程中,对任务要求可进行补充和修改,但这

些补充和修改必须经过充分论证,经使用方、承制方协商同意并报国家有关部门批准后方可更改。

追加的任务要求应追加相应的经费,并应允许对原订的研制进度作适当调整。

2. 减小工程项目技术风险的途径

减小工程项目技术风险的途径在于恰当地确定工程项目技术性能的风险百分数。工程项目的技术性能风险是指在工程设计中需解决重大技术关键的难易程度。要研制一个新的工程项目,必须基于过去的经验和现有的技术,但也必须采用新的、先进的技术。将新技术与过去积累的经验相比,新的要求、新的信息就构成了新刺激要素。显然,旧的经验不能全部用上,因此新、旧之间应当有一个合理的关系。可以用一个参数——风险百分数来表示新、旧成分的比例,亦即表示风险与把握之比:

$$\text{风险百分数} = \frac{\text{新技术(新刺激)}}{\text{旧经验}}$$

这个参数也就是新概念、新技术、新要求、新信息与原有旧经验的关系。这个风险百分数的大小,可以表明技术的难易程度。为形象地说明这一点,用一个塔形图来加以描述,如图 7.5 所示。

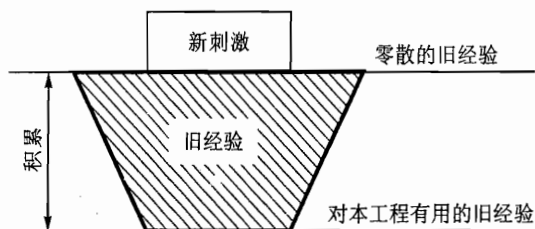


图 7.5 表示新刺激和旧经验比例的塔形图

图中的梯形代表旧经验,矩形代表新刺激,两者面积之比可以表示它们的数量关系。梯形的高度表示旧经验积累的程度,梯形上底的长度表示旧经验的总和。但并非任何旧经验和旧知识都对本项工程有用,有用的部分有多少,可用梯形下底的长度表示。因此梯形上、下底长度之比就表示在总的经验中有多少可利用。好的工程技术人员应当尽量把旧的、零散的经验转变为可利用的经验,亦即使梯形的下底尽可能的长,使塔形的基础更为稳固。以改进产品的性能为例,应当考虑改进产品的哪一部分可以提高产品性能?这一部分的比例应当占多大?有多少旧经验可利用?怎样以最小的风险取得最大的改变比率?为了讨论这一问题,可作出如下“极限情况讨论”(见图 7.6)。图 7.6 给出了塔形图的 6 种极限情况。从图中可以清楚地看到新刺激与旧经验基础之间的几种典型关系。

经验与统计结果表明,新、旧要素之间的比例,亦即风险百分数应在 15%~20% 最为合适;达到 20%~25% 则要付出相当的代价;而超过 25% 的工程,风险很大。上述经验与统计数

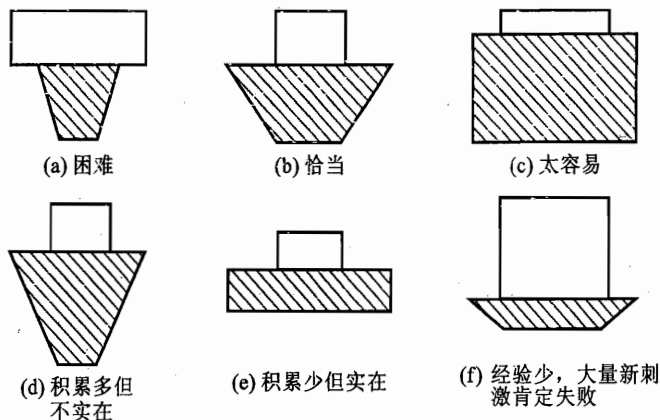


图 7.6 塔形图的 6 种情况

据是符合渐进规律的, 并已从大量的工程实践中得到证明。它已成为系统工程中的一条重要定律——渐进定律。这条定律不仅适合于工程项目的设计, 而且对一个国家的经济发展、农业中的土壤改良、儿童识字(生字率的控制)都可应用这条定律。又如读者阅读外文文献, 若生字率超过 20%, 亦即每 4、5 个字中就有 1 个生词要查一次字典, 就会觉得非常困难, 甚至读不下去了。

在工程项目设计中, 采用新技术和成品应遵循下述准则:

- 尽可能地采用现有的并证明是有效的技术和成品。
- 任何新技术、新成品、新材料和新工艺的采用都必须经过充分验证(包括工程项目的模拟试验和验证试验), 只有在确认该项新技术(新成品等)成熟后才能采用。
- 在采用某些重大新技术项目时, 应当考虑后备方案。
- 对某些先进技术, 若预计在本项目交付时还不能采用, 则可放到在该项目进一步改型时再采用。这时应在该项目的方案确定阶段, 就将这种准备在未来采用的新技术预先给予考虑(如预先留出空间、重量、电源功率、计算机容量等)。这种设计方法在国外称之为 P³I(Pre-Planned Product Improvement), 即预规划产品改进。

如果采用的新技术中包括有中等风险和高风险的项目, 则要根据它们相应的风险级别确定出减小风险的途径, 这包括:

- 着手进行平行的研制工作(即准备应急的后备方案);
- 尽早开始发展(研制)工作;
- 进行广泛的发展试验;
- 进行仿真研究, 以作出性能预测;
- 请有关专家评审设计;
- 加强研制过程的管理和评审。



3. 减小工程项目进度风险的途径

工程项目的进度要考虑到工程起点、总目标、途径、障碍等因素,这些可用图 7.7 表示。图中的左端是准备工作,其中要分析工程所需的后备(包括后备技术和资源),图中的右端是目标,它可能在中途为另一个目标所代替。从出发到目标的路线不只一条,而且有改变的可能,其中还存在最佳路线。在计划执行过程中,影响路线的因素是障碍。图 7.7 中表示过程的线与表示障碍的倒三角形相交部分的长度就表示风险大小。这种风险大小是随时间变化的,故可用风险随时间的变化率来表示。在图 7.7 中,则用障碍三角形的升降来代表。例如图中第三个障碍三角形较大,在开始时,认为风险率只有 10%,但随着时间的进展,发现它增长到了 50%(例如天旱对农作物的影响可能随着时间的进展而急剧恶化)。

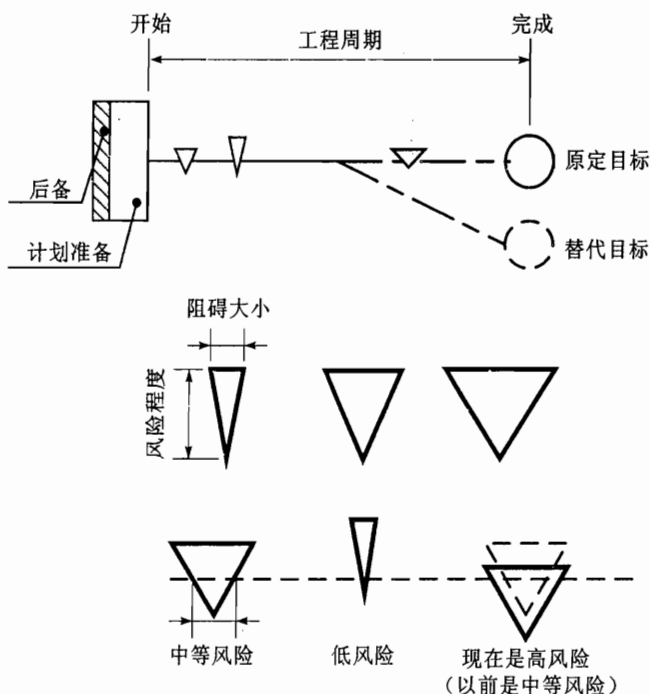


图 7.7 工程项目进度风险

出现障碍可能有内因或外因,在工程项目设计时应考虑到,若发现有无法在规定时间内克服的障碍,就要认真地考虑何时转向代替目标。为此应对下述问题进行权衡:为克服障碍而使整个工程受到的影响是什么?耽误的时间有多长?资源的重新分配是否能减轻障碍?风险太大时,有何后备可以利用?何时需要采用这些后备?什么是可以替代的目标?何时需要做转向的准备?一般地说,对进度的考虑要涉及所有这些因素,但其中最重要的是对风险的考虑、后备的准备与目标的转向。

根据上面的分析,进度风险具体取决于技术风险、工程计划安排的合理性、资源的充分性、

人员的经验、成品厂的交货历史和企业的管理状况。

因此,为了减小工程项目的进度风险,应当:

- 不允许在研制阶段存在技术性能属高风险的项目,中等风险的项目也应尽可能减小;
- 合理安排工程的进度计划,且应留有适当的时间裕度,以防意外情况发生时对计划的冲击;
- 保证用于工程项目的资源是充分、可供使用的;
- 参与工程项目的人员,尤其是在工程的关键部门工作的人员应在类似的工程项目中积累了经验;
- 选择交货信誉好,管理水平高的企业作为成品厂;
- 提高企业管理水平,采用科学的管理方法和决策方法。

4. 减小工程项目费用风险的途径

在工程系统的研制和生产中,造成费用风险的主要原因有以下几种:

(1) 资金的投资总量不足,尤其在下述情况下更为严重:

① 不是根据工程系统研制的实际需要来确定投资总额,而是把国家或使用部门的现有资金供应能力作为该项工程的预期投资总量。

② 在合同投标过程中,为了中标而明显地压低合同报价,使该报价严重地偏离工程系统实际的预期费用值。

在这些情况下,费用超支的现象必然严重。

(2) 资金分配不合理,这主要表现在:

① 对早期的研究(包括预研)和研制费用投资不足。早期投资不足的后果是:对工程项目中采用的新技术不能进行充分的预先验证,从而增加了技术风险。在工程系统的方案探索和确定过程中,只能采用纸面的验证或少量的实物验证,而没有能力进行样机和全尺寸原型机验证。工程系统早期研制工作的先天不足将大大增加后期研制工作的困难,即使采用追加投资的方法,也无法使型号恢复到理想状况。这是因为工程系统寿命期费用的85%以上是在研制的早期阶段就已决定了的。

研制阶段合理的资金分配应当如图7.8中的理想投资“样板曲线”所示。

② 在研制阶段后期,由研制向生产的过渡中,早期的生产资金准备不足。理想的生产阶段投资应当是:在研制工作还正在进行时,就为早期的生产准备好资金,亦即为工装、生产线启动和加工周期很长的部件(如飞机起落架等)制造准备资金。理想的生产阶段投资如图7.8中的“样板曲线”所示。当采用该“样板曲线”投资时,在研制阶段后期就已积累了早期的生产所需的资金(即“样板曲线”下的交叉线所表示的阴影部分)。而过分滞后的生产阶段投资(例如等研制工作全部结束后再开始投资)则会产生投资风险。

为了减小研制和生产阶段的费用风险,应当按照图7.8中的“样板曲线”进行合理的资金分配。

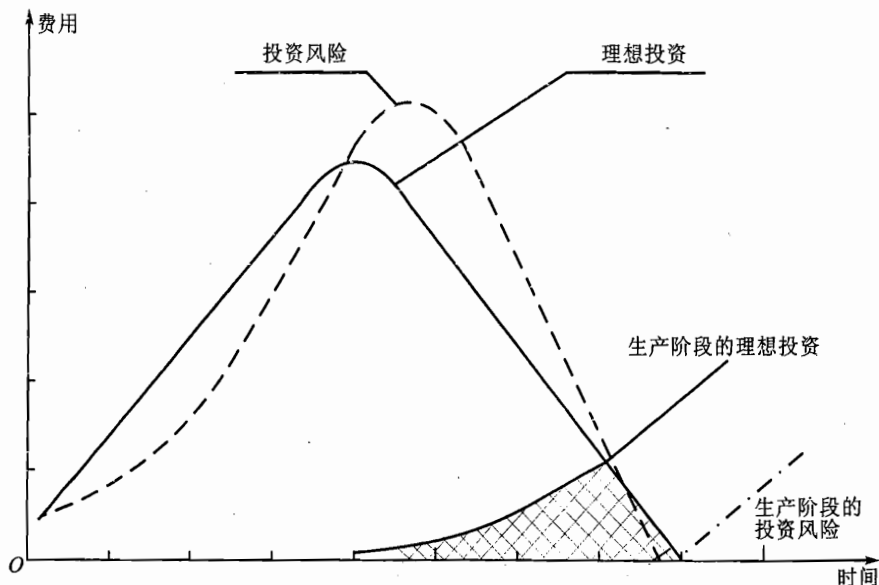


图 7.8 理想投资的“样板曲线”与投资风险

(3) 实际拨款日期滞后于计划拨款日期。

实际拨款日期滞后于计划拨款日期现象在工程系统的研制中经常发生，其后果是使费用-进度曲线的起始点后移，或使该曲线的阻尼系数加大。

图 7.9 所示是一条根据理想投资曲线绘制的费用-时间曲线。由于实际拨款时间的推迟，使该曲线的起始点后移，亦即使工程系统研制工作的启动时间推迟。

如果在工程系统的研制工作已经启动以后，实际的分期拨款时间（如按季度或年度拨款）仍然滞后于计划的分期拨款时间，则将使费用-进度曲线由计划状态（阻尼系数 $\xi=1$ ）变为过阻尼状态（阻尼系数 $\xi>1$ ），如图 7.10 所示。

过阻尼的费用-进度曲线通常会造成计划完成时间的拖延。

(4) 在计划的总费用中没有考虑合理的风险费用。

工程系统的风险费用是与该项目的风险大小密切相关的。应当根据工程系统风险的大小测算风险费用。

根据风险因子大小测算风险费用可采用以下方法：

设工程系统各工作单元的基线费用估算值为 BCE_i ，风险费用估计为 RCE_i ，则项目总费用 TCE 为

$$TCE = \sum_{i=1}^N (BCE_i + RCE_i) \quad (7.10)$$

式中： $RCE_i = RF_i \cdot BCE_i$ 。

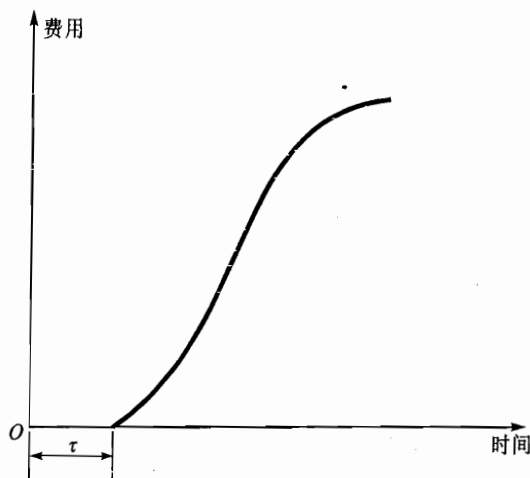


图 7.9 实际拨款的滞后对费用-进度曲线的影响

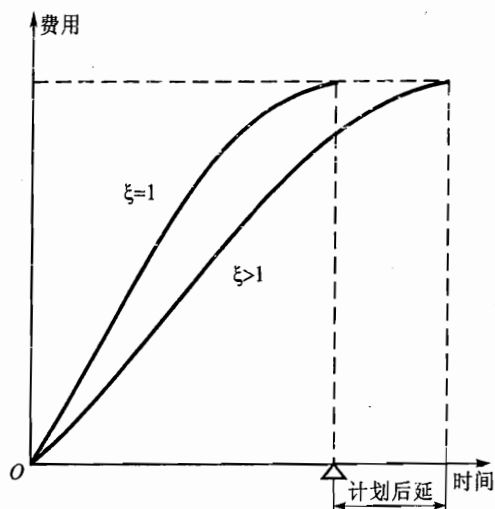


图 7.10 费用-进度曲线的计划状态和过阻尼状态

故式(7.10)可改写为

$$TCE = \sum_{i=1}^N (1 + RF_i) \cdot BCE_i \quad (i = 1, 2, \dots, N) \quad (7.11)$$

式中： RF_i 为第 i 个工作单元的风险因子； BCE_i 为第 i 个工作单元的基线费用估计值； N 表示该工程项目由 N 个工作单元组成。

如前所述，在工程系统研制中，可以用工作分解结构(WBS)进行工程费用的估算和风险的辨识和估计。现在，则可利用 WBS 估算风险费用。

如表 7.6 所列的某武器系统的 WBS，已知该系统各工作单元的风险因子 RF_i 和基线费用估计值 BCE_i ，即可根据式(7.11)算出该武器系统跟踪雷达的风险费用为 80 万元。同理，可根据 WBS 其他工作单元的情况测算出该武器系统总的风险费用为 400 万元。

表 7.6 某武器系统的风险费用估算表

工程项目名称：某武器系统									
WBS 层次					项 目	项目代码	工程费用 (百万元)	风险费用	
I	II	III	IV	V				风险因子 RF_i	风险费用 RCB_i (百万元)
√					武器系统	000	40.0	0.10	4.00
	√				雷达	400	4.0	0.20	0.8
		√			天线	401	0.5	0.10	0.05
		√			接收机	402	0.5	0.30	0.15

续表 7.6

工程项目名称: 某武器系统									
WBS 层次					项 目	项目代码	工程费用 (百万元)	风险费用	
I	II	III	IV	V				风险因子 RF_i	风险费用 RCB_i (百万元)
		√			发射机	403	0.8	0.20	0.16
		√			伺服系统	404	1.0	0.25	0.25
		√			显示器	405	0.3	0.20	0.06
		√			雷达车	406	0.1	0.10	0.01
		√			训练	407	0.5	0.20	0.10
		√			资料	408	0.3	0.07	0.02

采用本法测算风险费用的关键是应当正确估计各工作单元的风险因子 RF_i 。在估计风险因子时不可避免地带有风险估计人员的主观性。因此,应当用“德尔菲”法等专家意见调查法认真地确定风险因子,尤其对于基线费用估算值较高或风险因子较高的工作单元更应慎重处理。

在估算出工程系统的风险费用后,就可以计算出一旦风险发生后,工程研制部门遭受的经济损失及还能获得的最小利润。

假若预计的全部风险均已发生,并且全部损失要由研制和生产部门自己承担。则基本最小利润率 BMP(Basic Minimum Profit)为

$$BMP = \frac{P \cdot V_c - RCE}{V_c} \times 100\% \quad (7.12)$$

式中: P 为按百分数计算的基本合同利润率; V_c 为合同总价值; RCE 为风险费用。

以表 7.6 所列的武器系统为例,其合同总价值 V_c 为 4 000 万元,基本合同利率 P 为 15%, 风险费用 RCE 为 400 万元。据式(7.12)可计算出当风险发生后该武器系统的基本最小利润率为

$$BMP = \frac{0.15 \times 4\,000 - 400}{4\,000} \times 100\% = 5\%$$

但风险分析一般不应完全局限于当前的工程计划,还应考虑它对未来的影响,也就是要考虑在本项目取得成功后,可能会获得的后续合同。这样便可增加总的实际利润。当可获得后续合同时,总的实际利润 ORP 可由下式计算:

$$ORP = \frac{(P \cdot V_c - RCE) + \sum_{i=1}^m V_{Ai} \cdot P_i \cdot R_i}{V_c + \sum_{i=1}^m V_{Ai} \cdot R_i} \times 100\% \quad (7.13)$$



式中: V_{Ai} 为第 i 个后续合同的价值; P_i 为按百分数计算的第 i 个后续合同的利润率; R_i 为可能获得第 i 个后续合同的概率; m 为可能得到的后续合同的总数。

仍以上述武器系统为例,若在该武器系统研制完成后,又获得 600 万元的备件合同,利润率为 20%,签订合同的概率为 100%;此外,又有 80% 的概率得到武器系统的改型合同,合同价值为 800 万元,利润率为 15%,则由式(7.13)可求得该系统的总实际利润率 $ORP=7.9\%$ 。亦即考虑了后续合同后,利润率从原来的 5% 提高到 7.9%。

(5) 减少工程项目费用风险的途径

根据以上分析,为了减少工程系统的费用风险,应当采取如下措施:

- 根据工程系统的实际情况,合理预测费用,并提出所需的预期投资总量。
- 根据图 7.8 所示的“样板曲线”合理分配研制阶段的资金,防止研制阶段早期投资不足引起的费用风险。
- 根据图 7.8 所示的“样板曲线”合理进行生产阶段的投资,防止早期的生产资金准备不足引起的费用风险。
- 实际拨款日期应与计划拨款日期相符,防止由于实际拨款日期滞后引起的进度拖延。
- 根据工程系统的风险辨识和风险估计结果合理测算风险费用,并把风险费用包括在型号总费用之内。
- 在与使用部门签订研制或生产合同时,必须明确该项目的任务要求,任务要求应以量化的参数形式提出。
- 不允许在初步设计之后的研制阶段中存在任何高风险项目,中等风险项目的数量也应尽可能减少。
- 用多种方法合理估算工程系统的费用,并在研制过程中不断予以修正。
- 根据风险状况确定合同类型:对低风险项目采用固定价格合同;对中等风险项目采用成本加奖励费合同;对高风险项目采用拨款性合同。
- 在投标过程中,合同报价应与预测的研制费用基本一致。

(6) 资金的合理分配和控制

一个工程系统所需的资金如何投入,是一次性投入,还是分阶段投入;分阶段投入时的资金分布(投资强度)情况如何,这些对工程系统的研制进程会产生重大影响。必须科学地给出工程系统研制期间资金分配的合理规律。

以往在资金投放中,常常存在两种不合理的情况:

① 资金投放的速率过快。这虽然可以保证工程系统研制的资金需求,但拨款部门将遭受资金现时价值的损失。

② 资金投放的速率过慢,亦即投资不足。这会导致产生费用风险,影响工程系统研制工作的正常进行。

在工程系统的研究和发展阶段(包括方案探索、方案确定、工程研制 3 个阶段)中,理想的

研究和发展费用(RDC)投资包线应当如图 7.11 所示。这是美国国防部在其国防部指令 4245·7M 中给出的理想的资金样板。

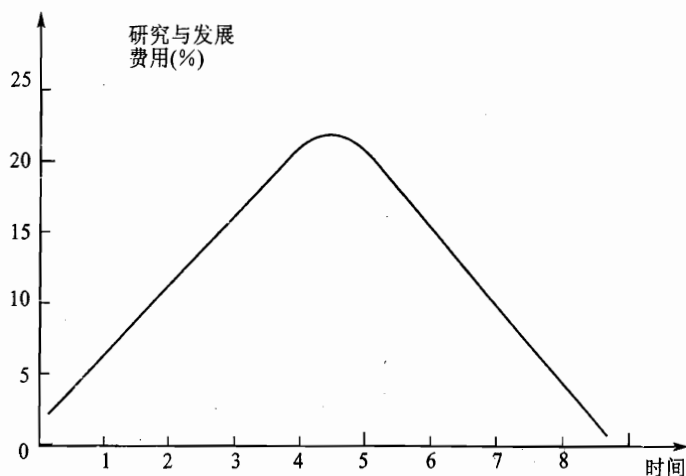


图 7.11 理想的研究和发展费用投资包线

图 7.12 所示的是投资速率过快的情况，其阴影部分为现时价值损失。

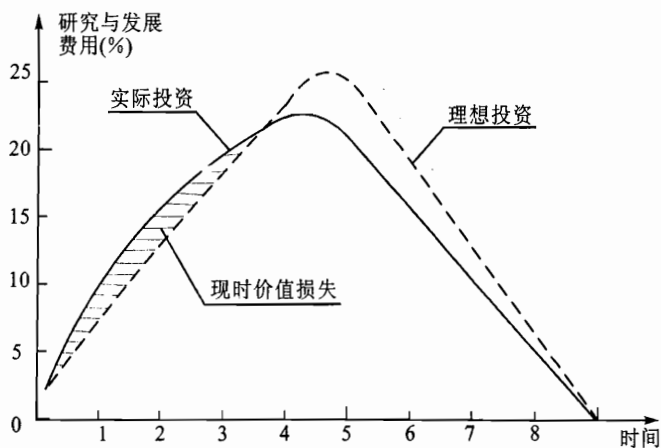


图 7.12 投资速率过快造成的现时价值损失

而图 7.13 所示的是投资不足的情况，其阴影部分表示由于投资不足引起的费用风险。

下面将给出现时价值损失和费用风险的定量计算方法。此外，还将给出理想投资情况下的资金分布曲线计算方法。

(1) 理想投资分布曲线(资金样板)的计算

设投资的资金总额为 A ，拨款区间数为 K (拨款区间可以以年、季度、月为单位)，首区间的

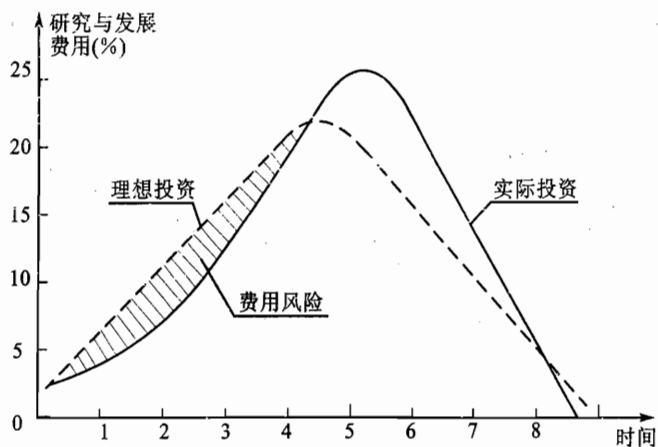
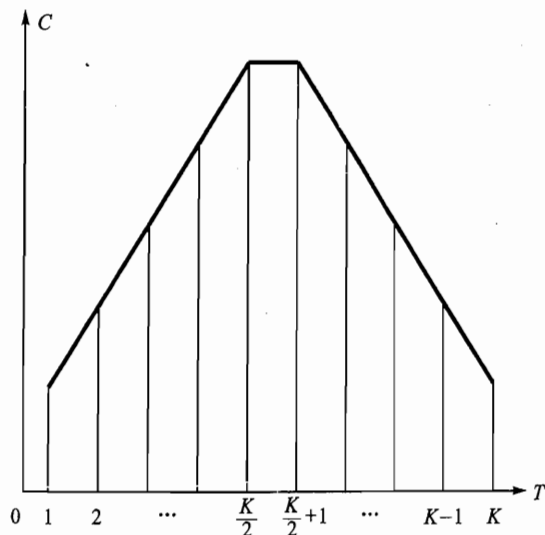


图 7.13 投资不足引起的费用风险

投资额为 α , 相邻区间的投资差额为 d 。

① 当 K 为偶数时, 即以图 7.14 所示的形式拨款。鉴于拨款区间对称, 前 $K/2$ 项与后 $K/2$ 项的付款区对称, 则有:

$$a_i = a_{K-i}$$

图 7.14 K 为偶数时的理想投资曲线

式中: α_i 为第 i 个区间的理想投资额。

根据已知条件和图 7.14 可列出:

$$A = 2 \left[a + (a + d) + \cdots + a + \left(\frac{K}{2} - 1 \right) d \right]$$

则有

$$d = \frac{4(A - aK)}{K(K-1)} \quad (7.14)$$

② 当 K 为奇数时, 投资曲线如图 7.15 所示。

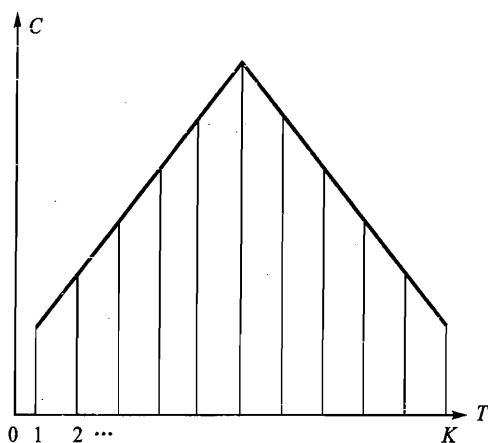


图 7.15 K 为奇数时的理想投资曲线

根据已知条件和图 7.15 可列出:

$$d = \frac{4(A - aK)}{(K-1)^2}$$

③ 累积费用:

$$b_i = \sum_{j=1}^i a_j \quad (7.15)$$

式中: b_i 为至第 i 区间时的总费用。

④ 累积资金的现时价值:

$$b_{ipv} = \sum_{j=1}^i \left[\frac{a_j}{(1+D)^j} \right] \quad (7.16)$$

式中: b_{ipv} 为至第 i 区间时的总资金现时价值; D 为计算现时价值用的贴现率。

(2) 资金现时价值损失的计算

① 由于资金投放过快造成的现时价值损失为

$$LS_{PV} = \sum_{i=1}^K \frac{\bar{a}_i - a_i}{(1+D)} \quad (7.17)$$

式中: LS_{PV} 为资金的现时价值损失值; $\bar{a}_i$ 为第 i 区间的实际投资额; a_i 为第 i 区间的理想投资额。

② 对于某个给定的区间 i 内, 资金的现时价值损失为

$$LS_{PVi} = \bar{a}_i - a_i \mid \bar{a}_i \geq a_i$$

且

$$\sum_{j=1}^i \frac{(\bar{a}_j - a_j)}{(1+D)^j} \geq 0 \quad (7.18)$$

如是,就可以在累积的费用-进度曲线上标出存在现时价值损失的投资区,如图 7.16 所示。

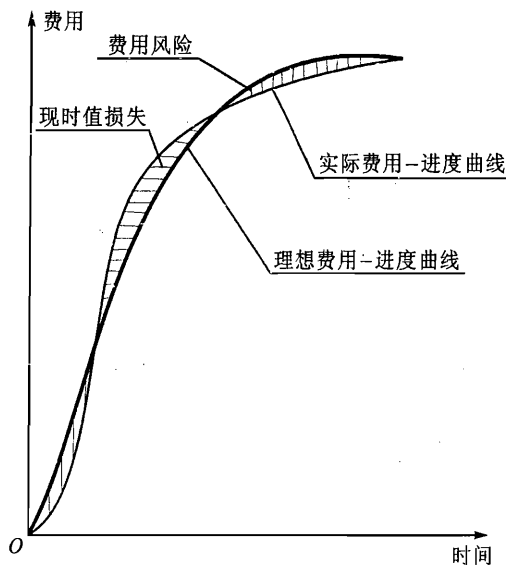


图 7.16 现时价值损失和费用风险的示意图

(3) 费用风险的计算

投资速率过慢将导致投资风险,亦即造成资金的短缺,从而影响工程进展。

费用风险可根据下列式子进行计算:

至第 i 区间的风险为

$$\sum_{j=1}^i \frac{a_j - \bar{a}_j}{(1+D)^j} \Big|_{\sum_{j=1}^i (a_j - \bar{a}_j) > 0}$$

将各区间费用占总费用的比例以 k_i 表示,即:

$$k_i = \frac{a_i}{A(1+D)^i} = \frac{a_i}{(1+D)^i} \sum_{l=1}^K \frac{(1+D)^l}{a_l}$$

从而有

$$\sum_{i=1}^K k_i = 1$$

同理

$$\bar{k}_i = \frac{\bar{a}_i}{A(1+D)^i} = \frac{\bar{a}_i}{(1+D)^i} \sum_{l=1}^K \frac{(1+D)^l}{a_l}$$

和

$$\sum_{i=1}^K \bar{k}_i = 1$$

若有

$$\sum_{l=1}^i (\bar{k}_l - k_l) < 0$$

则表明存在费用风险(到第 i 区间的实际总投资额小于理想总投资额)。总的费用风险为:

$$R_c = \sum_{l=1}^K (k_l - \bar{k}_l) \mid_{k_l > \bar{k}_l}, \text{ 且 } \sum_{l=1}^i (\bar{k}_l - k_l) < 0 \quad (7.19)$$

如是,即可在累积的费用-进度曲线上标出存在费用风险的投资区,如图 7.16 所示。

(4) 各个投资区间的费用状况估计

利用上面阐述的判别原则,可以判定任一投资区间(第 i 区)的费用状况。亦即存在投资不足的风险状况,还是投资过多的现时价值损失状况,并且可以给出这些状况变化的趋势。这将十分有助于计划管理人员对计划进行评估和调整。各个投资区的费用状况及其判别准则已综合在表 7.7 中。该表表明了各投资区可能的费用状况。

表 7.7 第 i 投资区间的费用状况

1. • 无风险 • 现时价值的损失继续上升 $\bar{a}_i \geq a_i,$ $\sum_{l=1}^i \frac{\bar{a}_l - a_l}{(1+D)^l} \geq 0$	3. • 有费用风险 • 通过在第 i 区间追加投资,使风险回降,但仍存在风险 $\bar{a}_i \geq a_i,$ $\sum_{l=1}^i \frac{\bar{a}_l - a_l}{(1+D)^l} < 0$
2. $\bar{a}_i < a_i,$ $\sum_{l=1}^i \frac{\bar{a}_l - a_l}{(1+D)^l} \geq 0$ • 无风险 • 通过在第 i 区间减少投资额,使现时价值损失减小,但仍存在损失	4. $\bar{a}_i < a_i,$ $\sum_{l=1}^i \frac{\bar{a}_l - a_l}{(1+D)^l} < 0$ • 有费用风险 • 第 i 区间的投资仍然不足,风险继续增加

注: $\bar{a}_i$ 为第 i 区间的实际投资额; a_i 为第 i 区间的理想投资额; $\sum_{l=1}^i \frac{\bar{a}_l - a_l}{(1+D)^l}$ 为至第 i

区间为止,实际与理想的累积现时价值之差。

显然,最理想的投资状况应当是既不存在费用风险,又不存在现时价值损失,亦即:

$$\begin{cases} \bar{a}_i = a_i \\ \sum_{l=1}^i \frac{\bar{a}_l - a_l}{(1+D)^l} = 0 \end{cases} \quad (7.20)$$

(5) 生产阶段的理想投资

对于生产阶段的投资,同样也存在理想投资、过于超前和过于滞后的投资状况,如图 7.17 所示。

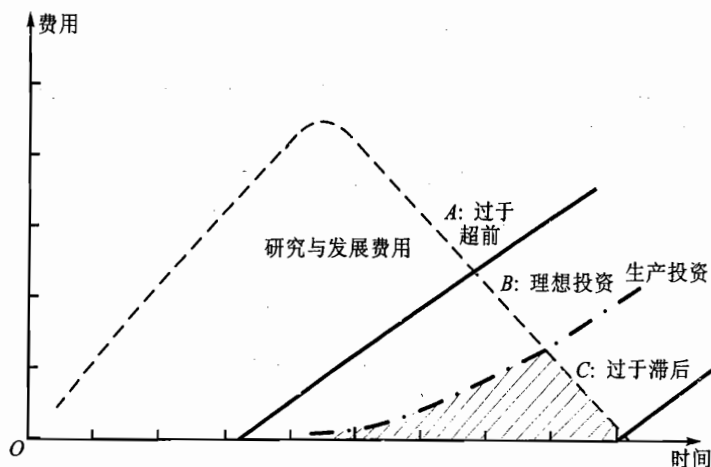


图 7.17 生产阶段的投资状况

图 7.17 中的实线 B 代表生产阶段的理想投资,即在研究和发展阶段中期就开始为早期的生产准备,亦即为工装、长周期器材、生产线的启动等准备资金。实线 C 表示在研究和发展阶段趋于结束时,所需的生产资金的形成过程。

曲线 A 表示的是设计与生产平行进行的过于超前的情况,这将导致额外的支出(例如设计更改对已准备好的工装、器材的影响)和现时价值损失。

曲线 C 则代表了等研制工作全部结束并通过设计鉴定后再开始进行生产投资的过于滞后的情况,这显然会造成生产阶段开始时的资金不足而影响生产进度。

曲线 B 代表了理想的中间状态。由此可得到低风险的、有控制的从研制到生产的过渡。图 7.17 所示的阴影部分,即是从研制到生产的低风险的过渡区域。

风险管理是系统工程管理的一个重要组成部分。其核心是选择控制已辨识风险的方案,拟定降低风险的措施,并在整个研制过程中对此进行连续的监控和管理。

第 8 章 并行工程

国内外几十年来飞行器研制的经验和教训是：由于设计—制造—使用过程之间的不协调和不沟通，在整个产品研制过程中产生了大量的设计更改和返工，以致造成研制中的周期延长，费用增加，质量下降。

根据系统工程的理论，得出了一条已为广大技术人员和管理人员所熟知的原理：在仅花费寿命周期费用(LCC)1%~3%的产品初始研制阶段，就已决定了 90%~95%的 LCC。显然，巨大的设计潜力应当在设计过程中，特别是在设计一开始就应予以发挥，即在产品研制一开始就应考虑决定 LCC 的基本要素，并把它们设计进产品中去。

并行工程第一次使上述理论变成了现实。

实施并行工程的效益是显著的，国外大量权威性的统计数据表明：

- 并行工程能降低研制费用 30%~60%。
- 并行工程能缩短研制周期 30%~60%。
- 并行工程能显著提高产品质量，例如：
 - 提高设计质量，使工程更改量减少 2/3；
 - 提高制造质量，使制造缺陷下降 80%，废品和返工减少 80%；
 - 提高使用与保障质量，使外场故障率下降 80%，备件储存量减少 60%。

8.1 并行工程的发展背景和概况

并行工程(Concurrent Engineering, CE)是 20 世纪 80 年代后期由美国首先提出的新观念，并于 90 年代蓬勃发展成为一项新的工程技术，属于系统工程范畴。

并行工程是一些大公司在总结了各自产品研制实践中的经验和教训的基础上，几乎是同时提出并加以发展的一项新技术。例如：美国普惠公司，在研制发动机 F-100 的改型产品 F-100-PW-200(用于 F-16 战斗机)的过程中，于 1987 年提出了建立“设计到加工”的多功能小组的概念，在发动机设计过程中，吸收从材料到制造和质量管理等方面的工程人员参与，从一开始设计就全盘考虑材料供应、制造技术和使用等问题，使产品能很快地从设计转入制造及投入使用。1988 年，美国防务分析研究所(IDA)调查了工业界在 13 个产品研制中开展全面质量管理(TQM)的情况后，写出了“并行工程在武器系统采办中的作用”的研究报告，指出并行工程在武器系统采办中对工程设计工作有重大效益。1989 年，美国国防部接受了并行工程的概念，把它作为一项应在 20 世纪 90 年代予以重点发展和推广的新观念，并称并行工程是 90 年代的挑战。而美国空军在普惠公司的“设计到加工”的多功能小组这一新观念基础上，于

1990年采用了更为广泛的多功能小组概念,提出了“综合产品研制(Integrated Product Development, IPD)”,并认为 IPD 更能准确地反映 CE 的内涵,同时可避免与项目管理中的常用术语“并行”相混淆。因此,后来 IPD(综合产品研制)和 CE(并行工程)这两个名词同时使用,其内涵相同。

自从提出 CE(或 IPD)后,美、英等国在 20 世纪 90 年代新飞行器系统和民用产品的研制中竞相采用这一新概念。例如,在美国 ATF 计划的动力装置 F119 的研制中,普惠公司设置了综合产品研制(IPD)机构,负责 F119 发动机的设计、制造、培训和后勤保障工作;通用电气公司在美国国防部支持下,在 GE90 发动机的研制中开展了并行工程(CE);英国罗罗公司在遑达 800 发动机研制中采用了同期工程(SE);波音公司在波音 777 飞机研制中采用了“设计制造工作组(Design/Build Team, DBT)”方法。近期,波音公司在新一代波音 737-700 飞机研制中,洛克希德公司在新一代先进战斗机 F-22 的研制计划中,又推行一种“综合产品工作组(Integrated Product Team, IPT)”方法。IPT 方法是目前最新、最有效的并行工程工作方法,据斯坦福大学研究人员调查,目前已有近 500 家美国公司在尝试 IPT 工作法。此外,麦道公司、IBM 公司等也竞相推行 IPD。

8.2 并行工程的定义与特点

8.2.1 定义

虽然各国对并行工程技术所采用的名称和具体方法在形式上有所差别,但其实质大同小异。为统一认识,目前人们大多采用美国 R.L. Winner 在防务分析研究所 R-338 研究报告中的定义:“并行工程是对产品及其相关过程(包括制造过程与保障过程)进行并行的、综合设计的一种系统化方法。这种方法力图使开发者从一开始就考虑到产品全寿命周期(从概念形成到产品报废)中的所有因素,包括质量、费用、进度与用户需求”。

8.2.2 特点

并行工程与传统的工程研制过程相比有如下特点:

(1) 将传统的序列化研制过程转变为并行的、相互间不断有信息交流的交互作用的研制过程。并行研制过程允许不同的研制阶段并行进行,并且可有一段搭接时间(见图 8.1),即前阶段尚未结束,后续阶段就可开始。图中双向传输的信息流表示两个阶段的人员之间发生了信息交流。

(2) 强调协作精神。并行研制过程中要求不同研制阶段的所有成员都要了解产品的总目标和技术要求,在产品开发之初,设计、制造、使用、维修等各方面的人员要协同工作,统一考虑,共同解决各类问题。

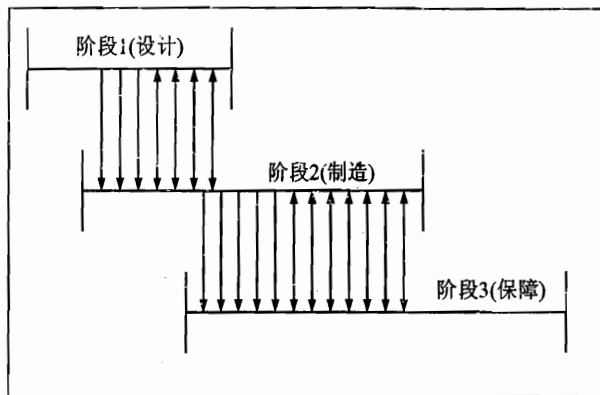


图 8.1 并行研制过程

(3) 强调人在产品研制中的主动精神。在传统的序列方法中,各种问题的发现大多是随机的,人对问题的反应是被动的,问题的解决是滞后的,往往当问题被发现时,已对研制工作造成无法挽回的损失。而并行工程则通过协同工作、并行发展,可主动发现问题。在研制过程中,当后续阶段发现以前的设计存在问题时,应主动、及时地反馈信息给前面阶段的设计人员,以便及时对现行设计进行修改。同样,前阶段设计人员应当将已定的设计方案提交给后续的设计人员,以便观察是否存在矛盾和不协调的问题,如果发生问题,则可在产品的并行研制组内由各方有关成员共同解决。

8.3 实施并行工程的巨大效益

(1) 美国国防部对 CE 进行了详细而全面的调查,调查表明,采用 CE 的效益是巨大的,具体体现在:

① 降低研制费用

对麦道、波音及 IBM 公司等单位的调查表明,采用综合产品研制可使费用下降 30%~60%,其中包括:

- 降低设计阶段费用;
- 减少了加工、制造和装配阶段的费用;
- 由于减少了返工和废品及降低了零、部件库存而省下费用。

② 缩短研制周期

对麦道公司等单位调查结果表明,采用综合产品研制可使研制周期缩短 35%~60%。对复杂的武器系统研制来说,效果尤为明显。

③ 显著提高产品质量,包括:

- 减少了重新设计(re-design)和工程更改;

- 减少了返工(re-work)和废品;
- 减少了缺陷和产品故障率。

美国各大公司的设计人员和管理人员通过综合产品研制懂得了提高质量并不意味着增加费用;相反,如果质量是在产品的研制过程中得以提高,则成本往往会下降。

如前所述,美国 PW 公司在 F119 研制中推行并行工程后,与 F-100-PW-220 相比, F119 的返厂率下降 74%,发动机非计划拆卸率减少 33%,每飞行小时所需的维修工时减少 63%,这对于提高武器装备的效能和节省寿命周期费用有着重大影响。

美国国防部对并行工程效益的调查结果可参见表 8.1。

表 8.1 并行工程效益调查结果

成 果 公 司	费 用	进 度	质 量
麦道公司	在反应堆项目和导弹项目上节省 60%	• 某高速飞机的初步设计阶段由 45 周减少到 8 周 • TAV-8B 设计周期缩短 18 个月	TAV-8B 飞机:废品率下降 58%,返工率下降 29%,由不协调发生的费用下降 38%,每单元缺陷下降 70%,图纸更改下降 68%
波音公司弹道导弹分部	工时费用下降 \$28/工时,报价费用下降 30%	部件、器材的生产准备时间下降 30%,部件的设计分析时间下降 90%	器材缺陷从 12%下降到 0, 99%部件无缺陷运行
AT & T	新电路包产品的修理费用下降 40%	总过程时间缩短 46%	缺陷降低 30%~87%
Deere & Company	研制费用实际节省 30%	研制时间节省 60%	检验员减少 2/3
Hewlett-Packard 仪器部分	制造成本下降 42%	研制周期缩短 35%	产品外场故障下降 60%,废品率和返工率下降 75%
IBM	产品直接装配工时费下降 45%	设计周期下降 40%	工程更改减少,可靠性和可测试性得以保证

图 8.2 显示了采用 IPD 后的工程更改数量比传统工程方法有明显减少,而且 90%以上的工程更改已在产品交付前完成。

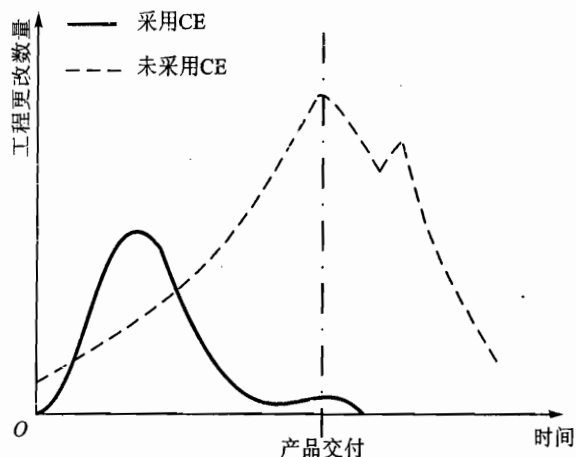


图 8.2 采用与未采用 CE 的工程更改量比较

8.4 并行工程的实施要素

通过对国外大量有关并行工程的文献、资料与实施范例的综合分析与研究,可得出并行工程的实施要素是:

(1) 把用户需求转化为完整的产品和过程要求的规范化方法

并行工程特别强调用户(使用部门)的需求,并以用户需求作为产品设计和过程开发的依据。它提供了把用户需求转化为完整的产品和过程要求的规范化方法和程序。

(2) 交互作用的相协调的并行研制过程

交互作用相协调的并行研制过程是将设计过程、制造过程与保障过程的研制协调地予以综合,将传统的序列化的研制过程转化为并行的、相互间不断有信息交流的交互作用的研制过程。

(3) 多学科(多专业)的综合产品研制机构

多学科、多专业的综合产品研制机构是指打破原来设计部门、制造部门、保障部门之间的界限;组建成自上而下的多学科的综合产品研制组织。从确定产品要求到整个过程的研制均由综合产品研制小组协同进行。小组内和小组间经常交流信息,以确保产品研制成功。

(4) 综合计算机辅助工程环境

综合计算机辅助工程环境是在原来已开发的计算机辅助设计与制造(CAD-CAM)的基础上,综合进计算机辅助采办和后勤保障(CALS),形成设计—制造—保障的一体化计算机辅助工程环境。其中重点是开发 CALS,其目的是加速将诸如计算机辅助可靠性、维修性和保障性等设计工具综合到已有的 CAD 和 CAM 系统中去,促进数字化技术信息的使用和综合,以便进行产品的综合设计、制造和保障。CALS 将使现行的大量纸面工作状态转向一个综合的、

高度自动化的工作状态,从而极大地提高了产品的可生产性、保障性和质量。美国国防部要求新研制的武器系统将以数字形式而不是纸面形式提交技术资料,亦即用综合数据库来代替纸面资料,因而有可能极大地提高产品质量,降低采办周期和费用。

8.5 并行工程的组织方式

组建综合产品研制组织机构并明确其职责是实施并行工程管理的核心问题,也是对传统的、由各职能部门进行产品研制方法的重大变革。

综合产品研制组织一般由一个顶层的综合产品研制小组和下属的各分系统、主要项目的综合产品研制小组组成。

首先组建顶层的综合产品研制小组。顶层的综合产品研制小组一般由产品研制的技术总负责人(如型号总设计师)任组长,由传统工程、专门工程、生产工程部门的技术主管(如各有关工程部门的副总设计师等)、质量主管、合同主管、技术状态管理主管等组成。

然后,根据工作分解结构(WBS)选定重要的分系统、主要项目和部件组建各分系统和主要项目等综合产品研制小组。

目前,实施并行工程(或综合产品研制)的组织有两种基本形式:一种是各有关人员集中在一处工作;另一种是借助先进的通信工具,通过在工程研制人员进行系统的通信,使他们能够及时获得有关的纵向和横向的交互信息,能够同时考虑产品所有方面的问题。前一种形式的典型例子是组建“综合产品研制小组(Integrated Product Team, IPT)”,成功的例子是美国洛克希德飞机公司在美国空军支持下研制新一代战斗机 F-22 的计划中推行的 IPT 新方法,最为成功的例子当推美国波音飞机公司在研制新一代波音 737-X 系列项目中采用的 IPT 方法。后一种形式的典型例子是洛克希德公司的“臭鼬工程队”工作法,此法早于并行工程观念的提出,现已逐步被 IPT 所取代。

8.5.1 综合产品研制小组的两个案例

(1) 洛克希德公司在 F-22 战斗机研制中组建的“综合产品研制小组”机构

美国洛克希德飞机公司在研制第四代战斗机 F-22 时组建了 IPT 机构,按几个层次(飞行器系统、6 个主要分系统及下属产品,共组建了 84 个 IPT 机构,如图 8.3 所示)。该公司 F-22 计划办公室总经理加里·赖利说:“这些小组一旦开始工作就会给 F-22 计划一个极大的推动力,这是我们能如期使 F-22 计划成功的关键之一。”推行 IPT 管理的主要目标之一,是发展一种在初始阶段就能达到最佳设计的管理方法,从而大大减少传统方法中在计划后期需做的设计更改工作。

(2) 波音公司在新一代波音 737-700 飞机研制中组建的“综合产品研制小组”机构

波音飞机公司在新一代波音 737-700 飞机研制工作即将启动前,及时总结了波音 777 研制工作中采用“设计制造工作小组(DBT)”方法的管理经验,在对 DBT 方法进行补充和完善

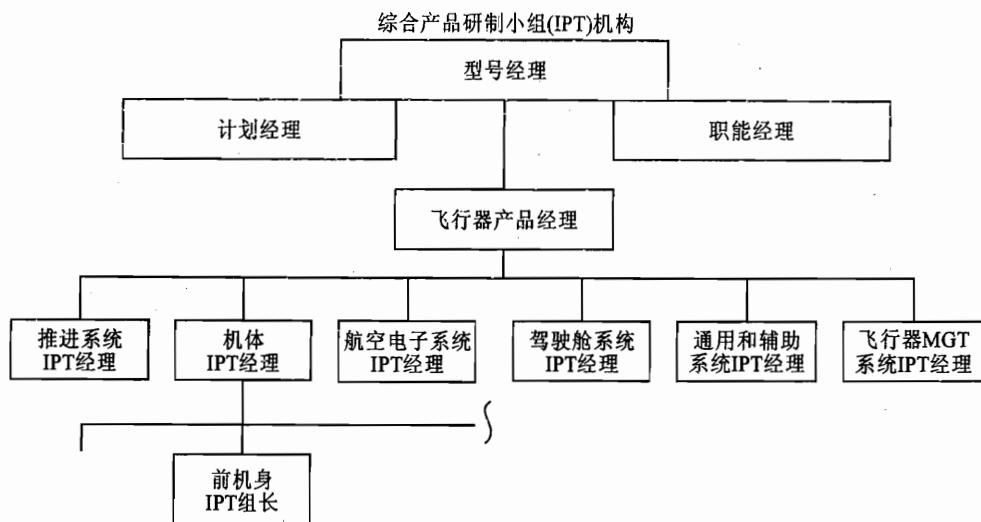


图 8.3 F-22 综合产品研制组织机构

的基础上,决定组建更为完善的“综合产品研制小组(IPT)”来开展波音 737-700 研制工作。

波音公司在研制波音 777 时运用并行工程观念,组建了“设计制造工作小组(DBT)”。DBT 方法是一次管理模式的重大改革,波音 777 项目全部采用计算机辅助设计与制造(CAD-CAM)技术和三维 CAD 绘图系统(CATIA),简化了管理层,打破了专业界限,密切了专业之间、部门之间的联系,形成了“大总体、大结构、大系统”格局。波音 777 研制的成功证明了 DBT 方法是成功的,但是 DBT 方法还没有完全超出传统的管理模式,还没有把激发和调动员工的主观能动性放在管理的中心,仍然强调以专业和部门的职能为主,而把产品设计的需要放在次要位置。波音公司及时总结了研制波音 777 的管理经验,在新一代波音 737 的研制中采用了优于 DBT 的新管理方法,组建了“综合产品研制小组”机构,推行新的 IPT 工作法。

新的 IPT 工作法有以下 3 个特点:

① 认为人是生产力中最活跃、最富创造力的因素,把营造一种和谐、融洽、充满真诚和理解的工作环境作为重要任务,以使员工心情舒畅地奉献个人的聪明才智,创造出最出色的工作业绩。

② IPT 工作中不再强调专业和部门的功能,而是强调以产品为中心,IPT 小组对产品全面负责。

③ IPT 工作法提出按产品实行矩阵管理的思路,完全打破了部门之间、专业之间的界限,缩短了所有协调和信息交流的路径,精简和疏通了妨碍 IPT 小组提高工作效率的环节,使人员和资金得到最佳配置。不但提高了设计质量,加快了研制进度,还降低了研制成本。



8.5.2 典型的“综合产品研制小组”的组织方式

一般,应根据工作分解结构(WBS)建立3个层次的综合产品研制小组(IPT)。

波音商用飞机公司根据并行工程观念组建的 IPT 组织颇具典型性,它共有 3 层 IPT 组织,如图 8.4 所示。图中,波音飞机公司总裁、执行副总裁、经理、737-700 项目总经理依次为直线领导,下级只对自己的直接上级负责。利顿厂的各职能部门受总经理领导。737-700 项目总经理对产品全面负责,下属共有 3 层 IPT 组织,总经理对由工程发展部、产品定义部、生产部、质量保证部、动力 IPT、机翼 IPT 和机身/尾翼 IPT 经理组成的第一层 IPT 实施领导,第二层 IPT、第三层 IPT 仅受其上一层 IPT 经理的领导,每个层次 IPT 成员之间不存在领导与被领导的关系。

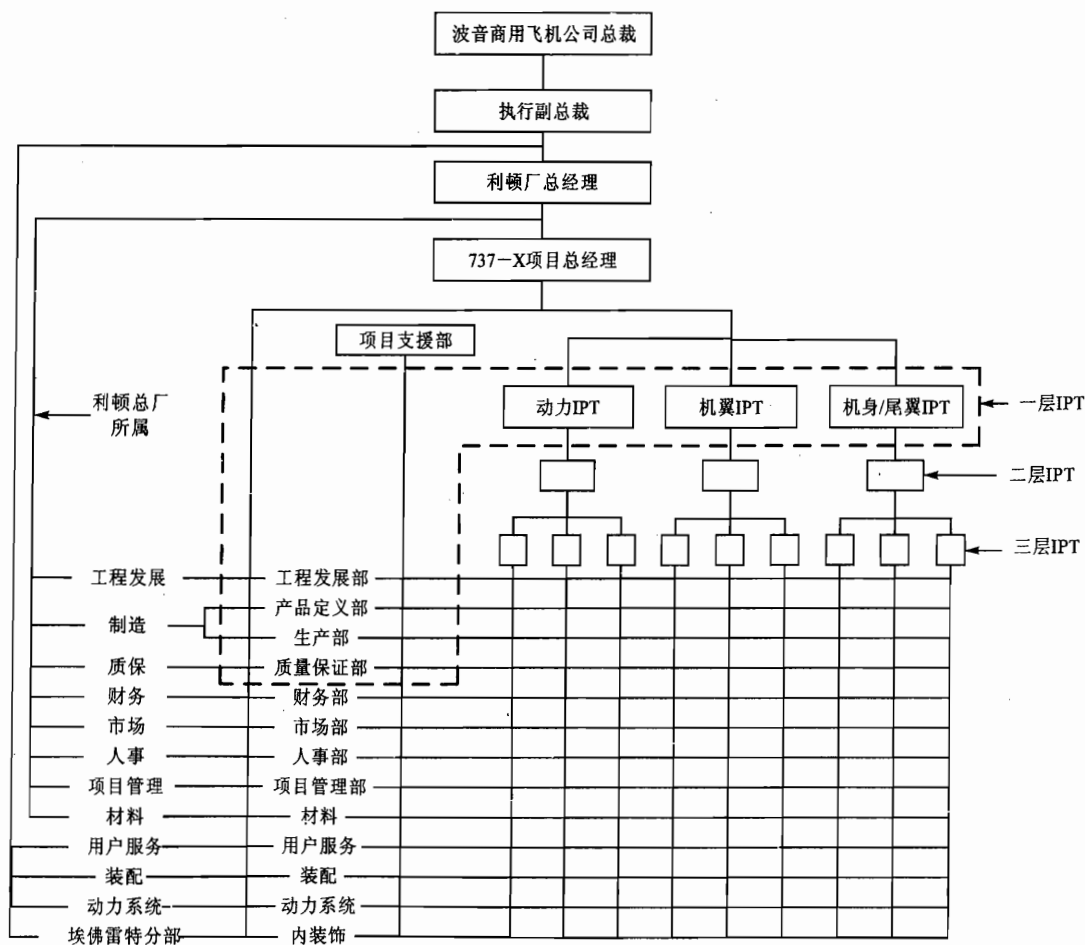


图 8.4 波音公司 IPT 组织机构

这3个IPT层次的划分是以所承担任务的类别和研制顺序的先后为依据的。

第一层IPT在公司下达的737-X的总体方案、结构布局、载荷分布等参数基础上,分解任务,协调和组织研制工作。主要任务包括:

- ① 全机数字化预装配审查;
- ② 全机打样设计审查;
- ③ 全机关键件、重要件的控制;
- ④ 全机关键件、重要件的设计审查;
- ⑤ 机翼机身对接方案;
- ⑥ 发动机舱和机翼连接方案;
- ⑦ 全机配平和调整;
- ⑧ 全机安全性审查;
- ⑨ 飞机生存能力分析;
- ⑩ 飞机试飞、飞机取证等。

第二层IPT承担全机重要零、部件及连接设计。例如翼身对接接头、机翼前后梁、机身重要承力件、加强框、对接框的设计等。

第三层IPT承担详细设计任务,任务要分解到人,其IPT数量较多。例如波音737-700,它以737-300为原准机,为保持飞机有良好的结构继承性,在结构上未作大的改变,虽然其第三层IPT小组数量相对较少,但也有约150个。

8.5.3 IPT的人员组成及对成员的要求

在组织IPT时,要特别强调IPT是多功能和多专业的小组。多功能是指包括与产品相关的各个功能领域(或部门),多专业是包括与该产品研制的各相关专业,特别是可靠性、维修性、测试性、安全性、保障性(R. M. T. S. S)专业的人员。

IPT人员组成一般包括设计、制造、工艺、强度、重量、R. M. T. S. S部门和三维CAD绘图系统(如CATIA)等方面的专家,材料、成本控制及各部门代表,甚至还包括市场、航空公司、供应商、承包商代表。根据任务完成情况或根据需要,对IPT人员还要不断调整。

IPT在工作中以设计人员为中心,其他专业人员和部门代表配合设计,协助把来自各方面的要求和意见反映到设计中,使设计更加完善。

IPT组内除设计人员外,其他成员是各自专业部门的全权代表,他们技术水平较高,实践经验较丰富,公司赋予这些代表独立处理与其他专业或部门有关问题的权力。IPT组内需要其他专业或部门解决的问题,由代表同自己的部门去讨论,限期答复,设计人员只同该专业或部门代表打交道,而不必在各部门之间往返协调。由于所有IPT成员同室办公,各专业代表对设计意图和要求比较清楚,对给定的外部条件十分明白,因此,完成本专业任务相对容易,反复性工作大大减少。



运用 IPT 方法进行产品研制管理,对员工来说是一次深刻的观念转变,必须改变过去那种“每个设计部门只管自己专业或部门内部的设计任务,将设计结果推给下个部门了事,很少为后续工作着想,结果使设计中存在许多错误,最后造成惊人的浪费”的做法。IPT 工作法要求小组研制人员把每一道后续工作作为自己的用户,必须千方百计为用户提供最佳服务,即以产品为中心,把提高工作效率和提供优质服务作为主要目标。

IPT 要求成员关心设计进展情况,及时发现问题,及时解决问题,不得将问题留到设计后期再解决。IPT 成员十分清楚,问题发现愈晚,造成的损失愈大。根据“十进位规则(rule of tens)”,设计更改成本将以 10 倍的量级进入研制计划的后续阶段,例如:在初步设计阶段更改成本若定为 1 元,到详细设计阶段将为 10 元,到制造和装配阶段将为 100 元。

8.5.4 IPT 的指导原则和工作程序

为了规范各级 IPT 的工作,一般应制定统一的指导原则和工作程序。

(1) IPT 的指导原则如下:

- IPT 通过并行的产品设计来综合其产品,并利用优选的过程来完成其综合。
- IPT 要对产品负责。每个 IPT 要对按合同规定生产的综合产品负责,工作组的各个成员主要关心如何生产出最高质量的产品,并关心总的技术设计和生产过程。另外 IPT 还必须负责对来自各产品供应方与用户之间的信息交流。
- IPT 是一个团队。IPT 成员为团队工作,对该组织负责,每个成员都献出各自的专业技能,并确保将其专业综合到产品之中。
- IPT 成员在同一地点工作。其目的是为了简化为完成并行产品研制所需的大量交互功能之间的通信。IPT 成员被授权在 IPT 中可全权代表其专业或功能部门作决策。
- IPT 的成员、领导者和小组规模将随任务的需要而适时地变化,在 IPT 中已经具体化了的关于产品功能的概念在生产和保障过程将继续有效。被批准的研制计划的更改将统一并入 IPT 的路线图(roadmap)中。

(2) IPT 小组具有交叉功能,同室办公加强了专业之间的交流,避免了设计上的片面性,使产品设计更加完善;更改、返工和不必要的重复性工作大为减少。

(3) IPT 小组对其设计的产品全面负责,成员的权限和职责明确,关系融洽,减少了相互扯皮和推诿。

(4) IPT 以产品为中心,任务落实到人,并有严格的检查和约束机制,有效地保证了研制进度。

(5) IPT 方法能合理配置人才和资源,强化成本控制和经费管理,保证了人才和经费的有效利用。



8.6 并行工程中确定产品与过程要求的规范化方式

8.6.1 运用 QFD 将用户需求转化为产品与过程要求的规范化方法

QFD(Quality Function Deployment)是一种把用户需求转化为产品开发过程各阶段技术要求的系统方法。

运用 QFD 可将用户需求逐层分解展开为产品系统级要求;分系统、部件、零件技术要求;加工过程的工艺要求、生产要求;最后输出操作指令单。如是,则可做到由用户需求驱动产品研发、生产的全过程。

运用 QFD 将用户需求转化为产品与过程要求的规范化流程如图 8.5 所示。在这一规范化流程中,需要经历产品规划,部件、零件展开,工艺过程规划和生产规划 4 个阶段,相应需开发以下 4 类矩阵(实践中可根据具体情况增减)。

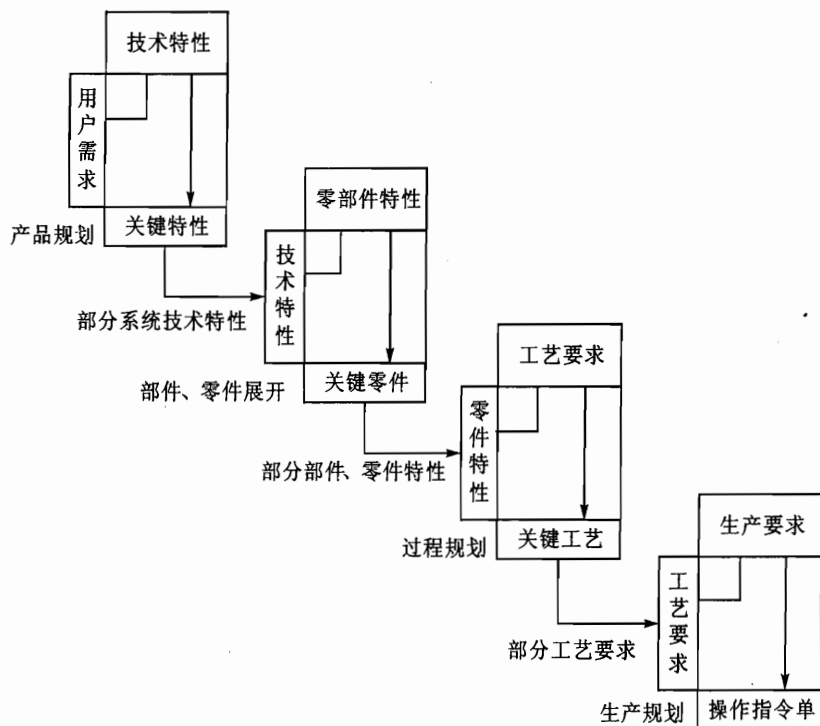


图 8.5 QFD 的规范化流程

(1) 产品规划矩阵

产品规划矩阵又称“质量屋”，其功能是将用户需求转化为对应的产品系统级的技术特性（或设计特性），并确定关键技术特性，准备转入部件、零件展开矩阵。其转化过程可通过一系

列规范化的步骤来完成。

(2) 部件、零件展开矩阵

部件、零件展开矩阵的功能是把前一阶段传下来的技术要求(或设计特性),特别是关键技术要求,转化为对应的零件特性与技术要求,并确定关键的零件特性,然后将其转入过程规划矩阵。

(3) 过程规划矩阵

过程规划矩阵的功能是把前一阶段传递下来的零件特性转化为对应的工艺要求,并为每个零件特性各准备一份工艺计划图表。该阶段要确定关键的工艺要求。对于这些关键的工艺要求,首先要在该阶段采取措施,如果问题在本阶段不能解决,则还要向下一个步骤转移,作为生产规划矩阵的输入。

(4) 生产规划矩阵

生产规划矩阵的功能是把工艺要求转化为相应的生产要求,并通过操作指令单传递到生产现场。

8.6.2 QFD 的规范化流程

(1) 确定用户需求

① 了解用户需求,亦即设法获取用户需求的原始信息。在飞行器的研制和使用中,人们越来越认识到飞行器的可靠性、维修性、测试性、安全性和保障性(R. M. T. S. S)对提高装备的作战能力、生存能力、部署的机动性,以及减少维修人力,降低研制和使用保障费用均有重大影响。在现代装备设计中,R. M. T. S. S 作为产品质量的重要特性已成为与性能要求同等重要的设计要求。因此用户必然会提出对装备性能和质量特性(包括 R. M. T. S. S)等各方面的要求,研制人员必须掌握用户需求的第一手资料,全面了解用户真正的原始需求是什么。这将是产品研制的出发点和整个研制过程的依据。

② 将原始的用户需求规范化、工程化

原始的用户需求一般是概要性的、非量化的,有些甚至是模糊的,研制人员必须对原始用户需求进行分类、推演、细化,以形成规范化、工程化的用户需求,作为产品规划过程的输入。其推演举例如表 8.2 所列。

③ 用户需求重要度的排序

用户需求重要度是表示用户需求重要程度的指标。重要度可以这样计算:当根据原始信息转换为质量要求时,如果某个要求重复出现的次数越多,则表示大多数用户重视该要求,该要求的重要度就高;反之则低。也可对用户用调查表进行调查,要求用户对各项性能和质量要求的重要度打分,然后根据返回的调查表数据进行统计处理后得到用户需求重要度的排序。一般用百分数表示,全部用户需求重要度的总和为 100%(或 1.0)。



表 8.2 用户需求的推演

用户原始要求	第二级	第三级
起落架性能良好	使用可靠	• 能够锁定 • 不允许泄漏 • 防盐雾、防振动、耐腐蚀 • 满足强度要求
	寿命长	• 可使用 1 000 飞行小时 • 收放 500 次
	经济性好	• 成本低 • 工艺简单

确定用户需求是 QFD 的第一步,也是最关键、最困难的一步。因为它要求了解什么是用户真正的需要,并且要以用户的要求而不是产品研制部门自己的想法来阐述产品要求。

(2) 产品规划阶段

① 根据用户需求确定产品相应的技术特性,并构成规划矩阵

各专业技术人员应根据用户需求确定相应的产品技术特性。这些技术特性应包括性能特性和可靠性、维修性、测试性、安全性、保障性等质量特性。将用户需求作为规划矩阵的垂直列,将相应的产品特性作为该矩阵的水平行,构成产品规划矩阵,如图 8.6 中(1)和(2)所示。在矩阵水平行所列的产品技术特性应是直接反映用户需求的,并必须在产品整个设计、制造、安装和使用过程中予以开发的产品要求。因此,这些特性必须以可验证的参数来表达。这是因为这些特性在研制过程中要予以监控,并且要定期地与目标值相比较。如果产品实现了技术特性所对应的目标值,则用户对产品就会满意。

② 确定用户需求与产品技术特性的关系矩阵

用户需求和产品技术特性之间的关系可以用图 8.6 中(3)所示的关系矩阵来表示。鉴于用户需求与产品技术特性之间的相关程度不同,因此,要用不同数字来表示这种关系的强弱程度,一般用 9、3、1 分别代表强相关、中等相关、弱相关。用适当的数字填到关系矩阵中去的优点是能很快指出产品技术特性是否充分地覆盖了用户的需求或期望。没有填入数字(或填入弱相关数字)的则表示用户没有阐述这种特性或表示用户对某些产品技术特性的要求不强烈。产品技术特性可以修改或补充,以保证所有的用户要求已被充分地了解和表述。后面还将利用这些数字计算技术特性的重要度。

③ 确定产品技术特性之间的相关关系

在产品技术特性之间必然存在一定的相关关系,亦即某个技术特性的变化会影响到其他的特性。若是两个特性相互支持(或加强)就是正相关,若是相互矛盾或冲突就是负相关。这

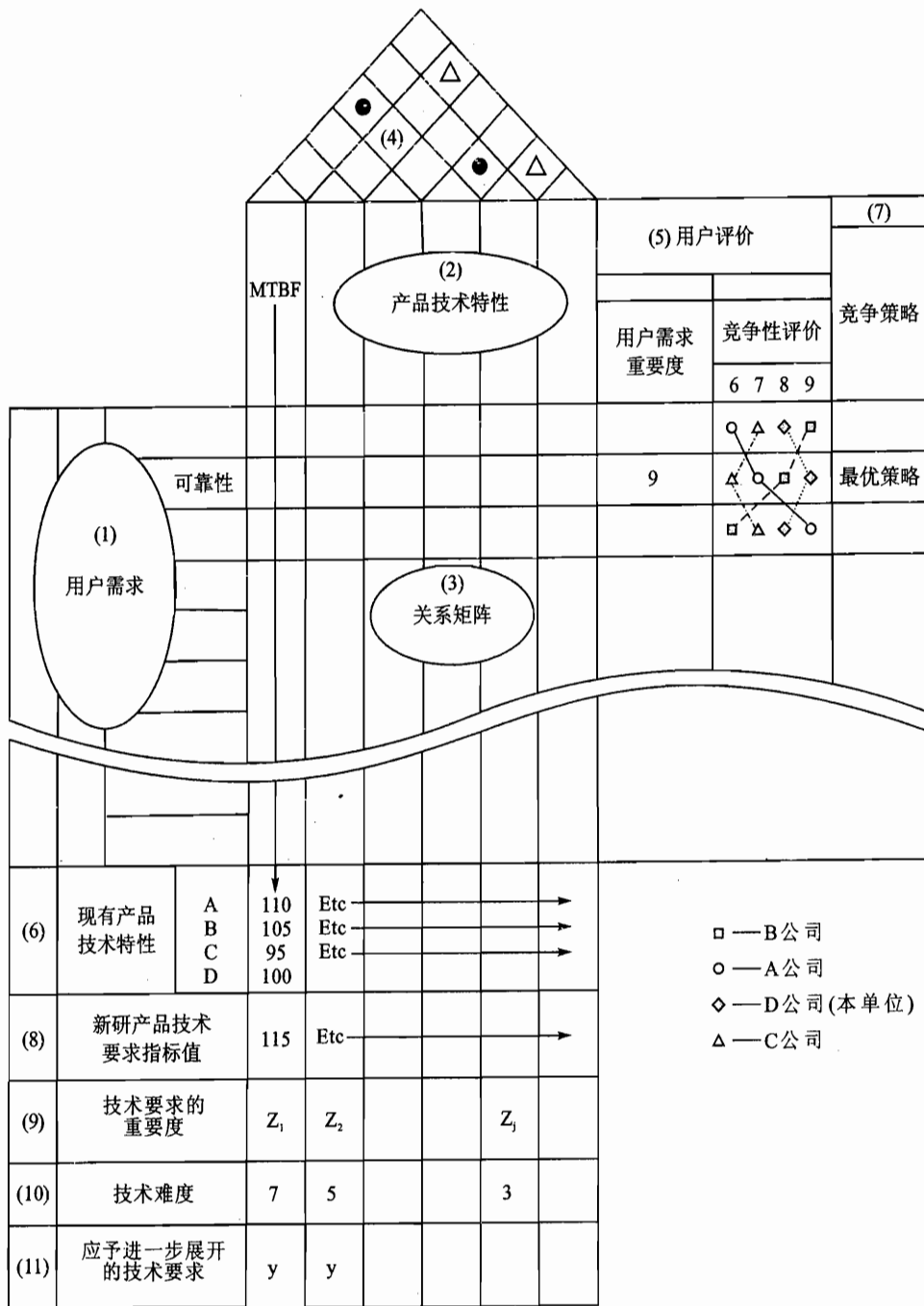


图 8.6 产品规划矩阵



将有助于各技术特性之间进行权衡。这些关系可以容易地用图 8.6 矩阵中的屋顶(4)来表示,以数字 3、-3、1、-1、0 分别表示正、负强相关,正、负弱相关与不相关。这样,就可以非常直观地看到各技术特性之间的相关关系,这也是这种“质量屋”方法的优点之一。该屋顶所示的关系也可称为“相关矩阵”。

④ 与其他产品的比较分析(市场评价)

在确定产品特性的指标值之前,应将本单位产品的现状与其他单位类似产品的现状进行比较分析。如果可能,最好对用户采用调查表调查,让用户来评价本单位产品和其他单位产品在产品特性上的优劣,从而了解本单位产品在质量方面满足用户的程度。一般用 1~5 之间的评分来区别,5 分为最好。市场评估结果列于图 8.6 所示规划矩阵的第(5)栏中。

⑤ 列出现有产品和对手产品技术特性的量值并进行技术评价

因为被评价的都是已有的产品,所以很容易列出在比较分析中进行过竞争能力评价的、各项用户需求相对应的、产品技术特性的量值。这些技术特性量值应以客观的、可测度的参数来表示,并将它们列在图 8.6 之规划矩阵的第(6)栏中。

⑥ 确定竞争策略

确定竞争策略,就是在考虑了现有产品(包括本单位产品和其他单位类似产品)的竞争能力评价后,研制单位在新研制的产品中所选择的、在竞争中要采取的方针。

在选择竞争策略时,应考虑的因素是:

- 用户需求的重要度大小;
- 在这些领域中,本单位过去和现在的状况;
- 与产品特性相关连的成本与进度;
- 竞争对手(对武器装备而言,则是要对付的威胁方)的潜在能力等。

最后,选定的竞争策略(或称“目标产品”)应记录在图 8.6 所示规划矩阵的第(7)栏中。

⑦ 为每个技术要求确定指标值

在确定了竞争策略后,便可为每个技术要求确定具体的指标值。这些指标值的确定主要取决于竞争策略、用户需求重要度和本单位现有产品的长处和弱点等因素。

技术要求指标必须是可度量的量化值,这些指标要在产品研制的各个阶段予以测定和验证,并最后通过验收。技术要求指标值填写在图 8.6 所示规划矩阵的第(8)栏中。

⑧ 计算每个技术要求的重要度

$$Z_j = \sum_{i=1}^m W_i r_{ij} \quad (j = 1, 2, \dots, n)$$

式中: Z_j 为第 j 个技术要求的重要度; W_i 为第 i 个用户需求的重要性评分; r_{ij} 为关系矩阵中第 i 个用户需求与第 j 个技术特性之间的关系所对应的数字,即加权系数,对应强、中等和弱相关的加权系数分别为 9、3 或 1; m 为用户需求项的总数; n 为产品技术特性项的总数。

通过计算所得的技术要求重要度填入图 8.6 所示规划矩阵的第(9)栏中。

⑨ 确定技术难度

技术难度是指达到各个技术要求的困难程度。技术难度一般可分为 5 等,数值越大表示难度越高。各个技术要求的技术难度可由确定要求的综合产品研制小组商定。建议通过对技术特性作两两比较,确定各特性的相对难度,可用 1、3(或 $1/3$)、5(或 $1/5$)、7(或 $1/7$)、9(或 $1/9$)来分别表示特性 A 与特性 B 相比难度为相当、稍大、较大、很大、非常大,最后通过均值化,以便于比较的百分值表示各技术特性的难度,并填入图 8.6 所示规划矩阵的第(10)栏中。

⑩ 选择应予进一步展开的技术要求

在众多的技术要求中,应当选择若干重要的技术要求,使其从规划阶段直到生产的全过程再继续展开并得以控制,以保证用户的需求在产品及其过程的开发中始终都能正确地予以反映,并一直继续到产品交付用户使用。

选择应予控制的关键技术要求应当考虑到该技术要求的重要度和难度,应当选择重要度高、难度大的技术要求予以控制,并进一步展开。对选定的关键技术要求,在图 8.6 所示规划矩阵的第(11)栏中填入“Y”。

至此,制定产品规划矩阵的规范化步骤已完成。

(3) 部件、零件展开阶段

本阶段工作主要是通过构成部件、零件展开矩阵来完成,该矩阵的组成内容和步骤与规划矩阵相似,区别仅在于没有规划矩阵中的市场评价、竞争策略和技术评价这 3 项。

在规划矩阵中,确定的都是产品系统级的技术要求,应将其转化为分系统、部件,直至零件的技术要求。可以通过一系列的矩阵转换来实现这一过程,如图 8.5 所示。将原规划矩阵水平行中所列的系统级技术要求转移到下一个部件、零件展开矩阵中的垂直列,将从相应的系统级技术要求分解出来的部件或零件特性列在该矩阵的水平行。如是,通过一系列的转换,就可以完成技术要求的分配工作,并确定出关键的零件特性。详细步骤和矩阵图形可参考产品规划阶段及图 8.6 内容。

(4) 工艺过程规划阶段

本阶段将上一阶段被选中的关键部件、零件特性转换为对应的工艺要求(两者构成过程规划矩阵的垂直列和水平行),分析其相关程度,填写关系矩阵,参考规划阶段有关步骤,确定关键工艺要求。另外,还应为每个关键零件各准备一份工艺计划图表,表中列有关键零件的各特性、对应的工艺过程、对关键零件的重要参数建立的控制点及过程监控计划等。

(5) 生产规划阶段

本阶段进行产品生产规划,分析工艺要求,特别是关键工艺要求转化为相应的生产要求,对上阶段的重要工艺过程编写操作指令单作为输出,以指导操作人员进行具体的产品加工和检验。

根据图 8.5 所示的流程,采用与制定产品规划矩阵相似的步骤,即可逐个完成部件、零件展开矩阵、过程规划矩阵等。通过上述一系列步骤,可将用户需求转换成系统、分系统和零件

的技术要求,并一直转化为生产该产品的过程流和工厂操作指令。因而可以设计和生产出能真正满足用户需求的产品。

在利用 QFD 确定产品过程时,应当根据不同的研制阶段,及时开发出相应的 QFD 矩阵。与各个研制阶段对应的 QFD 矩阵如图 8.7 所示。亦即要求在方案探索阶段完成系统级的产品规划矩阵;在方案确定阶段(简称方案阶段)完成分系统与部件、零件的展开矩阵;在工程研制阶段的详细设计完成时,完成过程规划矩阵;在工程研制阶段结束时,完成生产规划矩阵。

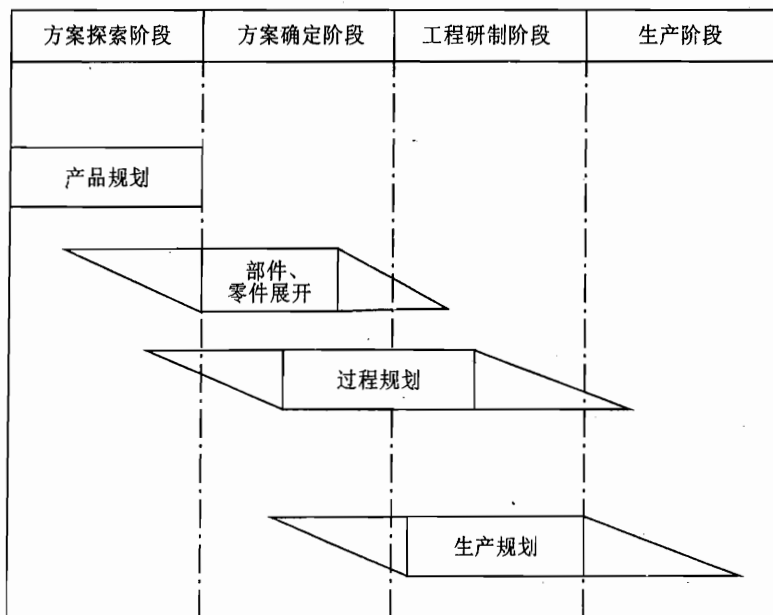


图 8.7 各研制阶段对 QFD 的管理

8.7 并行的研制过程

20 世纪 90 年代以来,在系统工程技术的基础上发展的并行工程方法对产品及其相关过程(包括制造过程和保障过程)进行并行交叉的综合设计;并在研制初期就考虑到产品全寿命期内所有因素,包括性能、质量、费用、进度、风险以及用户需求等,其中特别重视用户的需求,运用“质量功能展开(QFD)”技术,使用户需求得以驱动产品的研制和生产过程,并通过综合权衡技术使产品及其过程均予优化,保证用户能获得满意的产品。

8.7.1 研制过程并行运作方式的特点

(1) 在研制过程中,对产品自身的设计及其相关的制造过程和保障过程的设计同样重视,采取并行运作。

根据并行工程的定义,并行工程就是将传统的产品设计、制造和保障过程综合在一起进行

并行、综合研制,使产品及其相关过程两方面都得以优化,并能显著降低其 LCC 和研制周期。

传统的产品和制造过程设计方法如图 8.8 所示^[9],即首先在产品的设计完成后,根据产品的设计方案及现已具有或可能新开发的生产系统技术和投资决策状况,制定出制造产品所适用的生产系统规范;然后根据该规范进行生产系统的设计、制造过程的设计;最后根据生产成本模型来检验该设计规范和方案是否合理、可行。若在生产系统和制造过程的设计中发现产品设计有问题,就要进行产品设计更改,甚至产品性能规范的更改,造成返工。更有甚者,在传统的产品研制过程中,由于设计人员与工艺人员之间缺乏及时的信息交流与反馈,以致设计中存在的问题直到装配阶段才暴露出来,于是不得不修改原设计直至重新设计。而并行的产品设计与生产系统和制造过程的综合研制方法则可避免或减少上述弊病。

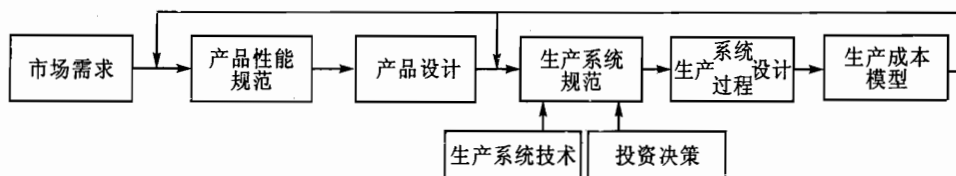


图 8.8 传统的产品设计和制造过程

图 8.9 所示的就是产品设计与生产系统和制造过程的并行研制方法。亦即在根据市场需求制定产品性能规范的同时,就应根据生产系统技术和资源约束制定相应的生产系统规范;在根据产品规范进行产品设计时,也要并行地进行生产系统的设计和制造过程的研制。由于这两者之间是并行进行的,并且由设计与生产部门的成员所组成的综合产品小组参与,所以当生产系统设计中发现问题或不协调时,可随时将信息传输给产品设计人员,及时进行协调,从而避免了大量的工程更改工作。

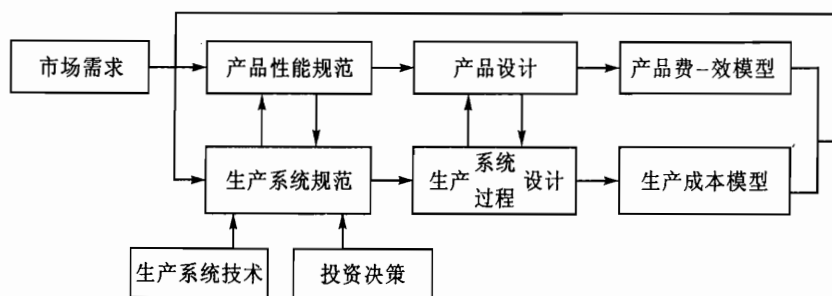


图 8.9 产品设计过程与生产系统和制造过程的并行研制

保障性分析和保障系统的设计过程也必须与产品设计过程同步进行。现代战争的特点是战争准备时间短,战斗节奏快,战场投入的装备数量大,损伤量也大大增加。这就对作战飞行器的保障特性提出了更高的要求,既要求所设计的飞行器便于使用和维修(便于保障),又要求在飞行器使用中能提供强有力的使用与维修保障(得到保障)。要使飞行器设计得便于保障,



在飞行器立项和论证时,就应同时进行保障问题的研究和论证工作,并在整个研制过程中,保障系统的分析与设计应与飞行器设计同步进行,相互协调。这样不仅能大大提高武器装备的战备完好性和任务成功性,还能大大降低保障费用。

(2) 传统工程、生产工程和专门工程综合、并行运作。

现代先进飞行器研制中包括的工程专业越来越多,一般分为3大类,即传统工程专业、生产工程专业和专门工程专业。

- ▶ 传统工程专业是指与传统的产品设计相关的工程专业,如空气动力学、机械学、力学、结构学、电子学等。
- ▶ 生产工程专业是指各种生产系统设计和各类制造工艺技术门类。
- ▶ 专门工程是指与飞行器战备完好性和任务成功性密切相关的工程专业,如可靠性、维修性、测试性、保障性、安全性、人素工程等特殊专业。这些专业均与产品故障的发生、发展、预防与消除密切相关。其核心是可靠性、维修性和保障性(即 R & M & S)专业。这是因为它们对飞行器系统的设计影响最大。

并行工程的实质就是将传统工程、生产工程和专门工程(R & M & S等工程)专业相综合,使这3项工程的专业技能并行地、协调地、交互作用地融入研制过程,使产品的传统性能与质量特性同时得以满足并予以优化。

鉴于传统工程和生产工程的技术已较为成熟,特别是与之相应的计算机辅助工具 CAD-CAM 也日趋完善,所以在产品研制中的设计—生产一体化技术的应用已较广泛。因而有人误将这种一体化技术当作是“并行工程”,而忽视了更为重要、更有挑战性的 R & M & S 综合技术。在寿命周期的所有要素中,R & M & S 等要素都是对寿命周期费用和装备效能有决定性影响的。就现代工程项目而言,其 LCC 中,使用与保障费用占 60%,而 R & M & S 对使用与保障有重大影响。倘若在飞行器研制中没有充分考虑 R & M & S 要素,则使用、保障费用将进一步增加到占 LCC 的 80% 左右。因此,并行工程就是要从产品研制一开始就考虑 R & M & S 要素。上世纪 90 年代,卓有成效地实施并行工程的现代工程项目,如波音 777 客机, F-22 第四代战斗机等均在产品研制一开始就将 R & M & S 综合作为并行工程的重要组成部分而予以特别关注。F-22 在方案阶段将 40% 的工作量放在 R & M & S 综合上;波音 777 则从设计一开始就全力开发用于维修性、保障性设计的计算机虚拟现实技术和 CALS 技术,从而成为世界上第一架没有样机,也没有图纸的飞机。因此,R & M & S 综合是并行工程中不可分割的重要组成部分。从这个意义上来说,没有 R & M & S 综合就没有并行工程。

图 8.10 展示的是较为典型的将传统工程与专门工程相综合的并行研制过程;在进行工程系统功能分析时,同时进行 R & M & S 分析;在进行性能设计时,同时进行 R & M & S 设计和保障方案的设计;随后通过系统综合权衡选出优化的设计方案和保障方案,并据此进行详细设计。

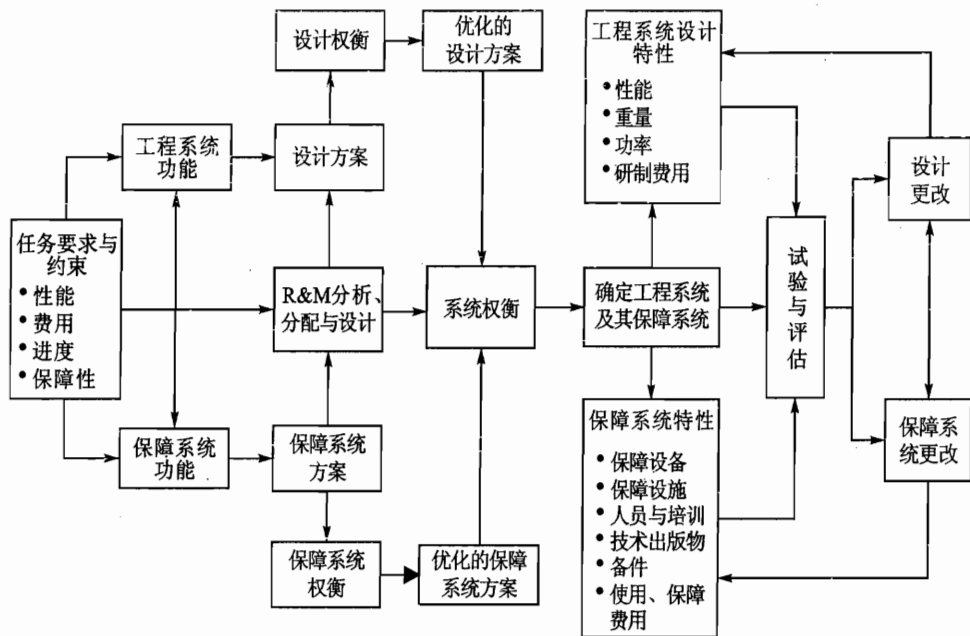


图 8.10 传统工程与专门工程的并行研制

8.7.2 并行研制过程是一个不断改进的过程

并行研制过程是一个不断改进的过程。这一过程也称为“规划、执行、检查、更改过程 (PDCA 过程)”。

过程改进的出发点是更好地满足用户的需求,所以首先必须以用户的观点选择应予改进的问题,并确定改进的目标和指标;然后依次进行规划、执行、检查和更改。

(1) 规划(Plan): 规划阶段的工作主要包括以下 4 部分。

- ① “什么”问题是应予改进的,即确定问题;
- ② “为什么”选择了这些问题,即分析原因;
- ③ “如何”解决这些问题,即提出解决方案;
- ④ “何时”解决这些问题,即制定行动规划。

(2) 执行(Do): 执行最佳的解决方案。

(3) 检查(Check): 对照原定目标和指标测定改进结果,并检查解决方案的效果如何。

(4) 更改(Act): 在需要时对解决方案进行更改,在更改后确立新的工作方式,并不断积累经验。

最后,应确保过程的改进是持续的,直至产品整个寿命周期的结束。

上述不断改进的 PDCA 研制过程的示意图如图 8.11 所示。

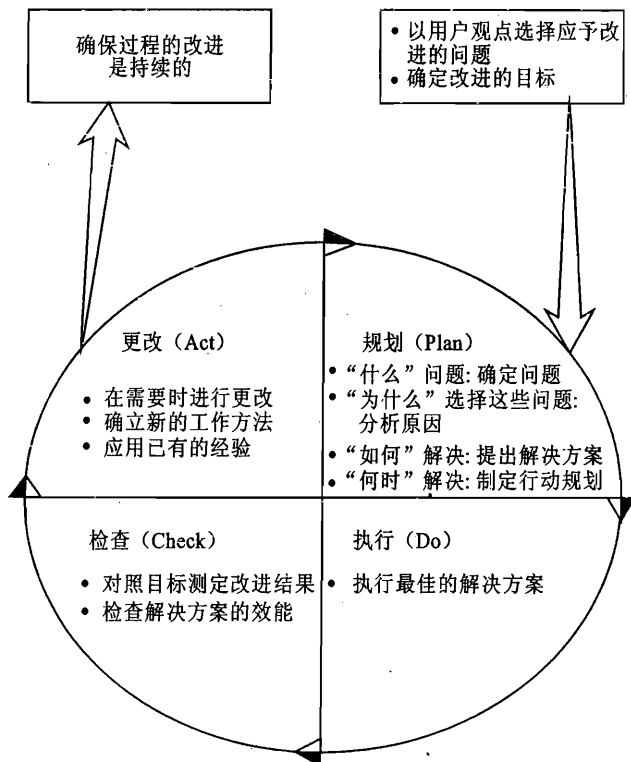


图 8.11 不断改进的 PDCA 研制过程的示意图

8.8 并行工程的环境

并行工程的环境是综合的计算机辅助工程环境。该计算机辅助工程环境是实施并行工程的关键要素之一,它能极大地提高产品设计和制造的效率,减少设计更改和返工,从而显著改善产品质量,缩短研制周期。

8.8.1 并行工程环境概述

并行工程要求在综合计算机辅助工程环境中研制产品,以达到产品零件、部件和整机的产品定义相容。其主要的特点是:

- 有效地综合利用现有的和新的计算机应用软件和数据库;
- 各工程专业能在并行工程过程中进行综合;
- 利用在线数据库(on-line databases)实现对设计、制造和保障信息的管理、交换和重复使用;
- 计算机辅助设计系统应适应本单位产品设计、制造和保障的具体环境和有关规定,并应

符合相应的国家与工业标准；

- 具有良好的、兼容的人员界面(human interfaces)和人机界面。

综合计算机辅助工程环境的作用是在产品研制中减少由于设计更改、错误和返工带来的质量下降和费用上升。

应用综合计算机辅助工程环境可以在发图之前,进行零件、部件乃至整机的三维图形的协调、配合和预安装,以便及时发现零、部件、设备、管路相互之间的干涉与配合问题。利用该工程环境还可以同时进行产品性能及制造、保障过程的同步设计,以协调性能、可生产性与可靠性、维修性、保障性诸特性之间的关系,并进行综合权衡,从而可以显著地提高产品质量,降低成本。

从国外现已发表的实施并行工程的案例分析,在并行工程中应用的综合计算机辅助工程环境大致包括:

- 100%的三维数字化产品设计;
- 电子样机;
- 计算机辅助可靠性、维修性、保障性分析与设计;
- 交互通信与数据传输技术等。

8.8.2 100%的三维数字化产品设计

100%的三维数字化产品设计(3D digital product definition)是指所有的产品零部件以三维形式在 CAD-CAM 系统中进行设计,并利用计算机传送设计数据集的数字化信息。

产品的数字化设计是利用 CAD-CAM 工具设计产品零部件的三维图形。在特定的 CAD-CAM 系统中,零件可设计为三维实体模型,并可在计算机上装配,以检查干涉、配合状况,还可用计算机进行重量、应力等分析和计算。此外,还可以方便地从实体图像中得到剖面图。应用数字化设计数据,可驱动 CAM 进行数控机床加工零件,还能方便地建立产品工程图册,并利用 CAD 数据集编制技术出版物。在设计中,所有零件在 CAD-CAM 系统中只形成唯一的数据集,并可提供给其他设计人员使用。如果有更改要求,则只对数据集进行修改,而不需要在“图纸”上更改。每个零件数据集中均有一个三维的模型及其相应的二维视图。

在数字化产品设计中还包括数字化预安装(digital preassembly)。

数字化预安装是指在计算机上设计和安装零部件的过程,并在成套图纸发图之前和零件生产前检查零部件乃至整机安装配合的情况。实现数字化预安装的前提是进行数字化产品设计,其目的是为了减少设计更改和返工。

数字化预安装是在计算机上模拟装配过程,它根据零部件设计、分析人员和工装设计人员的要求,利用飞机系统各层次中的零部件模型实施预安装。零件以三维图像显示,以检查相互间的配合、协调、干涉状况。通过预安装过程,可以暴露出全机存在的干涉问题,并予以解决。数字化预安装可利用 CAD-CAM 系统,还可根据需要建立临时装配图。

数字化预安装的主要工作是:

- (1) 利用数字化产品设计中得到的零件三维实体建立装配模型。
- (2) 设计人员进行数字化预安装,以发现零件装配中存在的配合、协调、干涉、同轴度等问题。
- (3) 发现问题后,设计人员相互协调,利用联机反馈文件传送存在问题的零件模型,并修改设计,解决存在问题。
- (4) 将修改后的零件模型重新存入数字化预安装共享文件。
- (5) 通过飞机系统各层次中相关零部件的安装、配合,不断发现问题,修改设计,直到所有零部件的安装配合均满足设计要求为止。

在并行工程产品研制中,要求在发图前完成并审核通过所有零部件和工装的数字化预安装,以最大限度地减少在生产阶段中发生的工程更改和返工。

8.8.3 电子样机

电子样机(electronic mockup)是在计算机上产生的三维图像的样机,可用来替代传统的物理样机(木质或金属样机),以协调结构及其内部设备、管路的布局。电子样机与物理样机流程图如图 8.12 所示。波音公司在飞机研制中的实践证明:用电子样机协调结构及其内部设备、管路、缆线等被验证的对象可以达到前所未有的 92% 的合格率,而用物理样机只能达到 50% 的合格率。其主要原因在于:

- (1) 电子样机提供了三维物体的点、线、曲面及其整体的图像空间,避免了两个物体占有同一空间的错误。

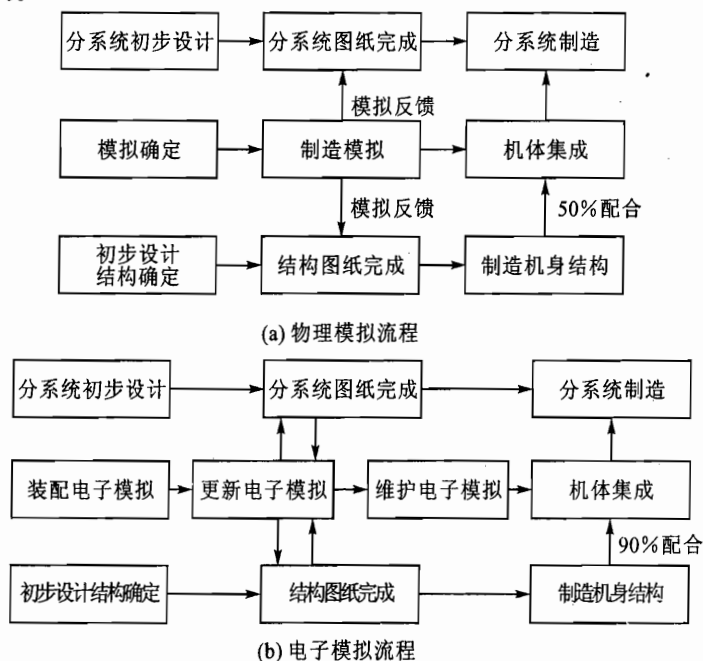


图 8.12 电子样机与物理样机流程图



(2) 物理样机一般是在技术状态冻结之前提供的,由于型号研制经费和周期的限制,不可能随着研制阶段的进展提供不同的物理样机。而进入详细设计阶段后,由于不断的设计更改,物理样机的技术状态并不能与详细设计阶段中飞机真正的技术状态充分地保持一致,因此潜藏了不少不协调的问题。而电子样机则可以由设计数据库持续地更新其技术状态,以反映产品当前的真实状态。

在现代的飞机研制中,在电子样机的基础上还开发了虚拟现实(VR)技术。即除了在计算机终端上显示物体的三维图像外,还将人的图像迭加进去;并使操作人员和维修人员能身临其境地用手等人的器官去“触摸”、“拆装”机舱内的设备、管路等,以更真实地验证设计中的可达性、维修性等特性。波音 777 就是应用了虚拟现实技术而成为世界上第一架不需要任何物理样机的飞机。

8.8.4 计算机辅助可靠性、维修性、保障性分析与设计

计算机辅助可靠性、维修性、保障性(R&M&S-CAD)是用来进行可靠性、维修性、保障性的综合权衡和分析、设计的重要工具。它一般由数据库、模型库、用户接口等组成。其中既包括综合权衡的工具(如战备完好性模型、LCC 模型等),又包括可靠性、维修性、保障性等各“性”分析和设计软件包。这样,在产品设计中可方便地使“专门工程”专业与传统工程专业进行并行的开发。

图 8.13 所示的是一个应用于并行工程的计算机辅助 RMST 工作站^[11]。该工作站由数据库、模型库和用户接口组成。

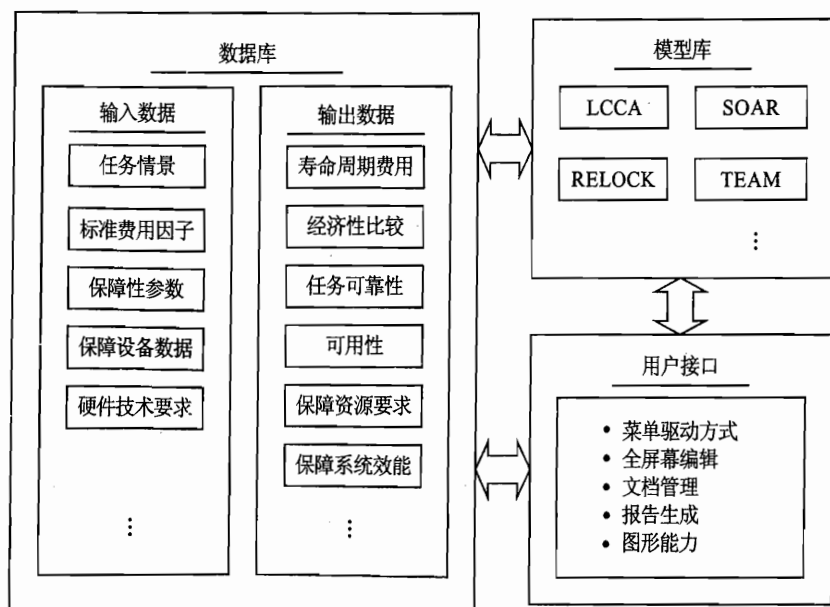


图 8.13 计算机辅助 RMAT 工作站

表 8.4 机上 LRU 数据

LRU 序号	LRU 类型	单机拥有的 LRU 数量	外场不能维修比率	故障率	MTTR	备件水平	备件短缺量期望值	备件补偿期望值	每天需求率 D	费用 (元)	每次供货时间 (天)
1	1	2	0.600	0.000 50	2.000	5	0.05	2.336 0	0.160 0	20 000.0	21.0
2	1	18	0.300	0.000 07	0.500	20	0.00	0.947 5	0.201 6	8 000.0	14.0
3	1	10	0.950	0.001 80	4.000	15	34.82	49.824 0	2.880 0	2 500.0	14.0
4	1	2	0.100	0.000 06	0.500	5	0.00	0.036 5	0.019 2	5 000.0	14.0
5	1	1	0.650	0.000 40	2.000	5	0.00	1.292 8	0.064 0	30 000.0	28.0
6	1	2	0.750	0.001 75	2.000	9	4.09	12.880 0	0.560 0	9 000.0	28.0
7	1	2	0.200	0.000 30	2.000	5	0.00	0.595 2	0.096 0	6 000.0	21.0
8	1	36	0.950	0.000 20	2.000	30	10.71	40.608 0	1.152 0	15 000.0	35.0
9	1	2	0.700	0.000 55	2.000	5	0.12	2.939 2	0.176 0	10 000.0	21.0
10	0	2	0.990	0.004 10	4.000	20	12.54	32.524 5	1.312 0	75 000.0	21.0

表 8.5 战备完好性性能

MFMC 飞机指标 ($P_r=0.15$)	3	NFMC 飞机期望值	7
概率 $P(\text{NFMC}<3)$	0.030 71	机群平均出动率(架次/天)	37
FMC 飞机架数 (conf. = 0.90)	10	单机出动率(次/架一天)	2.828
概率 $P(\text{出动率}=\text{指标值})$	0.503 3	LRU 短缺量总数	62

显然,上述的战备完好性不能满足要求,为了使战备完好性指标达标,可以采用如下措施:

- 降低 LRU 的故障率;
- 减少 LRU 的 MTTR;
- 增加 LRU 的备件数量等。

鉴于篇幅,下面只分析备件数量对提高战备完好状态的影响。通过模型的动态仿真,发现影响飞机完好状态的 LRU 是第 3、8、10 号 LRU,由于这 3 类备件数量不足,致使 EBO 曲线呈上升态势(见图 8.14),并影响飞机状态(以 FMC 代表)和战备完好性状态(以 SGR 代表)呈恶化趋势。通过模型的分析,可得到为了使战备完好性指标达标所要求的备件水平,如表 8.6 所列。表 8.6 同时还给出了为增加备件所增加的费用。



表 8.6 为战备完好性指标达标所要求的备件水平

LRU 序号	原来的备件水平	提高后的备件水平	备件增加量
1	5	5	0
2	20	20	0
3	15	42	27
4	5	5	0
5	5	5	0
6	9	17	8
7	5	5	0
8	30	30	0
9	5	5	0
10	20	36	16
Total cost for increased LRU = 1 339 500.0(增加备件所需的费用)			
ptc = 0.950 565 46 (NFMC 飞机<目标值的概率)		cc = 0.949 999 99 (置信度)	

在增加备件之后,该飞机的战备完好性指标值可全部达标,如表 8.7 所列。

同样,可利用该模型分析可靠性、维修性等特性对战备完好性目标的影响,并在进行综合权衡后确定装备的 R & M & S 指标值。

表 8.7 战备完好性达标的性能

MFMC 飞机指标($P_r=0.15$)	3	NFMC 飞机期望值	1.7
概率 $P(\text{NFMC}<3)$	0.950 6	机群平均出动率(架次/天)	40
FMC 飞机架数(conf. =0.90)	17	单机出动率(次/架一天)	2.19
概率 $P(\text{出动率=指标值})$	0.995 8	LRU 短缺量总数	20

8.8.5 交互通信与数据传输技术

计算机辅助采办与综合保障(CALS)的一个作用就是在飞行器整个寿命周期内用数字化方式传输产品的数据和信息。与之相适应的还有综合飞行器数据库。在该数据库中,包括了飞行器及其部件的三维实体模型、图样、数据、设计单、测试报告、可生产性和保障性措施等有关设计信息。例如,麦道直升飞机公司开发的 IDEA 数据库,利用产品工作分解结构(WBS)的层次结构作为框架,建立起全系统的树状数据库,即层次数据库模型(hierarchical data model)。该模型为树状结构形式,用户可在 WBS 中的任一层次中的任一单元进入数据库。例如,



用户根据需要打开“机体”单元和“发动机”单元,则该数据库就会把与该分系统有关的产品定义的数据呈现给用户。

系统的层次数据模型允许用户访问数据库中的产品特定属性(如重量)。在子系统层次,数据库的层次结构反映了研制工作的树状图。在层次结构中,一个或多个层次或单元拥有全部的产品定义数据。在各工作单元之间可建立数据联系,以便在研制过程中实现数据与信息的共享。各 WBS 层次单元中均含有产品属性的信息和数据,并可以进行换算、映射、判别、反馈等。例如,每个工作单元的文本记录中均有“重量”数据,将各单元的重量加起来就可确定部件、分系统乃至整个系统的重量。

必须指出,国外型号研制的大量实践表明:在飞行器研制中仅靠应用大量的计算机辅助设计工具并不能明显地提高设计质量和加快研制进度。研究表明,先进的计算机辅助设计工具必须与产品研制的管理模式和组织形式相适应。在未来的飞行器(和其他产品)研制中,真正的改革应该是组织机构和管理模式的改革。因此,在并行工程中,综合计算机辅助工程环境必须与管理模式相适应。

8.9 下一代飞行器研制中的系统工程/并行工程管理问题

8.9.1 实施系统工程/并行工程管理的要点

在现代飞行器的研制中,采用先进技术是完全必要的,但这还不够,还必须采用先进的管理模式和方法。在我国以往的飞行器研制中,往往对采用和引进先进技术很重视,但管理方法仍很陈旧,不注意,甚至不愿意采用先进的管理方法,结果使新技术的引进事倍功半。例如,我们在阐述并行工程时,强调并行工程的4个要素是一个整体。但在我国的不少飞行器研制单位,推行并行工程时只注意技术层面的事,例如引进先进的综合计算机辅助工程环境,采用无纸设计、100%的三维数字化产品设计等,但却不采用 IPT 组织方式。

下面是我国两个飞行器研制中推行并行工程的不同效果:

一个是我国某直升机研制项目,这是一个与国外合作的项目。当时国外主承包商与我国某研制单位的合作条件就是:必须在该项目研制中推行并行工程。其中最困难的是组建 IPT,因为我国的设计与制造是分别由不同单位承担的,要把这不同单位的人组合在一个团队(Term)里甚为困难。尤其在研制初期,由制造厂来的工艺、制造技术人员不习惯在设计初期就天天和设计人员在一起,但渐渐就进入了状态,对设计中的工艺和制造问题提出了改进的意见,因此,他们在设计图纸只完成70%时,就已将制造周期长的部件投产,并大大降低了返工现象。

对并行工程效果进行评价的一个指标是设计图纸的更改量,这是设计质量好坏的一个标志。该直升机项目研制与同类项目相比,设计图纸更改量减少了2/3。

另一个是我国某民用飞机项目,也推行了并行工程,但只重视应用计算机辅助工程环境

(如 CATIA 等),忽视了组织模式的改变,仍是设计部门只管设计,制造部门只管工艺、生产,没有组建综合产品研制组织。因此设计与生产之间不协调的现象仍大量存在,设计图纸更改量大,返工现象严重。

正如在 8.8.5 小节中已指出的那样:“在并行工程中,综合计算机辅助工程环境必须与管理模式相适应”。

在现代飞行器研制中,实施系统工程/并行工程管理的要点有:

- (1) 应采用整体规划、综合协调、分散决策的管理模式,尽可能减少管理决策的层次。
- (2) 坚持采用综合产品研制团队(IPT)的研制组织,打破条条块块分割的型号研制体制。
- (3) 各部门(团队)的负责人应做到责、权、利一致,不能出现负责任的人没有权,有权的人不负责任的现象。
- (4) 型号主管部门或订购方必须科学地拨发研制费用,尤其必须注意早期研制费用拨发的充分性。须知,一个型号的命运是在研制阶段中的方案确定时就大致决定了。因此充足的早期费用能支持多方案的论证和验证,从而选出较合理的设计方案。
- (5) 在采用和引进先进技术的同时,必须采用和引进先进的管理模式与之相适应,方能事半功倍。
- (6) 应当设立系统工程管理部门,因为系统工程管理不同于一般的项目管理,项目管理主要是管理成本、进度、分承包单位的合同管理等内容,而系统工程管理涵盖了技术和管理两个方面的综合、协调,是将系统工程的原理、方法全面贯彻于飞行器研制全过程的管理。
- (7) 应当对飞行器研制,特别是下一代先进飞行器研制的技术骨干、管理人员,尤其是高层领导成员进行系统工程管理的培训,只有使他们掌握了系统工程管理知识,具有相应的系统工程管理能力,才能确保飞行器研制成功。

8.9.2 系统工程能力成熟度模型

为了使飞行器研制单位对本单位飞行器研制中实施系统工程的能力有一个较为清晰的了解和客观的评价准则,我们提出了一个飞行器研制中的系统工程能力成熟度模型。该模型分为 5 级,共 24 个关键过程域。

(1) 初始级

该级的特征是未引入系统工程的方法,或只引入了个别的方法,如设计评审、技术状态管理等,基本上仍然以传统的、陈旧的管理模式进行飞行器研制。因此,其产品和过程的质量以及进度、成本等不可预测或不受控。

(2) 可重复级

该级的特征是已引入了相当的系统工程方法,但尚未形成体系,已进行了可靠性、维修性、软件工程等工程专业的设计,但尚未进行工程专业综合,更没有将工程专业与传统专业有机地综合。处于该级能力成熟度的飞行器研制单位有能力研制同类或相似类型的飞行器,并能研



制成功,即其能力是可重复的。

可重复级的关键过程域有 8 个:

- ① 设计评审;
- ② 技术状态管理(对硬件)/配置管理(对软件);
- ③ 工作分解结构(WBS);
- ④ 计算机辅助工程环境;
- ⑤ 试验与评价(T & E);
- ⑥ 需求管理;
- ⑦ 软件工程;
- ⑧ 组织培训。

(3) 已定义级

该级的特征是已全面引入系统工程过程、工程专业综合、系统分析和控制,并以并行工程进行飞行器研制。处于该能力级的飞行器研制单位,其飞行器研制过程是可控的,研制过程与产品的质量是可预期的。该能力级的关键过程域有 8 个:

- ① 采用综合产品研制组织(IPT);
- ② 运用系统工程过程;
- ③ 进行工程专业综合;
- ④ 风险分析与管理;
- ⑤ 运用并行工程;
- ⑥ 开展软件可靠性工程;
- ⑦ 开展综合保障工程;
- ⑧ 里程碑控制。

(4) 已量化级

该级的特征是对性能、质量、研制过程均进行定量的管理,并以规划(P)—执行(D)—检查(C)—改进(A)的 PDCA 过程对研制过程进行自觉、持续的改进。其关键过程域有 5 个:

- ① 技术性能测量;
- ② 研制过程的量化控制;
- ③ 研制费用(RDT)拨发的合理性;
- ④ 采用 PDCA 过程持续改进;
- ⑤ IPPD。

(5) 已优化级

该级的特征是以技术创新和过程创新作为飞行器研制的动力,并以效能和全寿命周期费用作为飞行器设计优化和权衡的准则。该级的关键过程域有 3 个:



- ① 效能;
- ② 全寿命期费用(LCC);
- ③ 技术创新和过程创新。

系统工程能力成熟度模型示意图如图 8.15 所示。

24 个关键过程域为:

- ① 设计评审;
- ② 技术状态管理/配置管理;
- ③ 技术性能测量;
- ④ 工作分解结构;
- ⑤ 采用综合产品研制组织(IPT);
- ⑥ IPPD;
- ⑦ 计算机辅助工程环境;
- ⑧ 里程碑控制;
- ⑨ 试验与评价;
- ⑩ 研制过程的量化控制;
- ⑪ 研制费用(RDT)拨发的合理性;
- ⑫ 需求管理;
- ⑬ 运用系统工程过程;
- ⑭ 进行工程专业综合;
- ⑮ 效能;
- ⑯ 全寿命期费用 LCC;
- ⑰ 风险分析与管理;
- ⑱ 运用并行工程;
- ⑲ 软件工程;
- ⑳ 开展软件可靠性工程;
- ㉑ 开展综合保障工程;
- ㉒ 采用 PDCA 过程持续改进;
- ㉓ 组织培训;
- ㉔ 技术创新与过程创新。

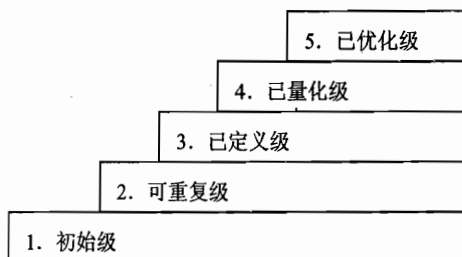


图 8.15 系统工程能力成熟度模型示意图

飞行器研制系统工程能力成熟度的级别,其特征和关键过程域如表 8.8 所列。

表 8.8 系统工程能力成熟度的级别,其特征和关键过程域

等 级	特 征	关键过程
1 初始级	以传统方式进行飞行器研制,或只引入了个别的系统工程方法	无
2 可重复级	已引入了相当的系统工程方法,但未形成体系 已进行了可靠性、维修性等工程专业的的设计,但是分别进行的没有将工程专业综合,更没有将工程专业与传统专业有机综合	① 设计评审 ② 技术状态管理/配置管理 ④ 工作分解结构 ⑦ 计算机辅助工程环境 ⑨ 试验与评价 ⑫ 需求管理 ⑬ 软件工程 ⑭ 组织培训
3 已定义级	全面引入包括系统工程过程、工程专业综合、系统分析和控制的系统工程方法,并以并行工程进行飞行器研制	⑤ 采用综合产品研制组织(IPT) ⑧ 里程碑控制 ⑬ 运用系统工程过程 ⑭ 进行工程专业综合 ⑰ 风险分析与管理 ⑱ 运用并行工程 ⑳ 开展软件可靠性工程 ㉑ 开展综合保障工程
4 已量化级	自觉地、持续地以 PDCA 进行过程改进,对飞行器研制过程进行量化控制	③ 技术性能测量 ⑥ IPPD ⑩ 研制过程的量化控制 ⑪ 研制费用(RDT)拨发的合理性 ㉒ 采用 PDCA 过程持续改进
5 已优化级	采用效能 E 和全寿命周期费用 LCC 等作为飞行器设计优化和权衡的准则,以技术创新和过程创新作为飞行器研制的动力	⑮ 效能 ⑯ 全寿命期费用(LCC) ㉓ 技术创新与过程创新

注: ①~㉓是指 24 个关键过程域的顺序编号。

我们认为:一般的飞行器研制单位至少应达到 2 级,即可重复级水平,而现代先进飞行器研制单位则应达到 3 级(已定义级)的能力成熟度。

参考文献

- [1] Booz, Allen & Hamilton Inc. 美国系统工程管理[M]. 北京: 航空工业出版社, 1991.
- [2] [美]NASA. NASA 主要系统的工程项目周期.
- [3] [美]防务系统管理学院. 系统工程概论[M]. 军用标准化中心, 译. 2001.
- [4] [美]防务系统管理学院. 系统工程管理指南[M]. 北京: 宇航出版社, 1992.
- [5] 何新贵, 王伟, 等. 软件能力成熟度模型[M]. 北京: 清华大学出版社, 2000.
- [6] 郑人杰, 等. 实用软件工程[M]. 北京: 清华大学出版社, 1997.
- [7] Watts S. Humphreg. 个体软件过程[M]. 北京: 北京航空航天大学出版社, 2001.
- [8] Watts S. Humphreg. 小组软件开发过程[M]. 北京: 人民邮电出版社, 2000.
- [9] James L. Nevins. Concurrent Design of Products and Processes. McGraw-Hill Publishing Company, 1989.
- [10] [美]洛克希德导弹和空间公司. 系统工程管理指南[M]. 北京: 航空工业出版社, 1988.
- [11] M. H. Awtry. Logistics Engineering Workstation for Concurrent Engineering Applications. IEEE, 1991.
- [12] 阮镡, 章国栋. 工程系统的规划与设计[M]. 北京: 北京航空航天大学出版社, 1991.
- [13] 阮镡, 陆民燕, 韩峰岩. 装备软件质量和可靠性管理[M]. 北京: 国防工业出版社, 2006.
- [14] J. T. Brewer. Implementation of Georgia Tech's Concurrent Engineering Methodology To A High Speed Civil Transport (HSCT). AIAA - 94 - 4328 - CP, 1994.
- [15] T. L. Kopra. Application of a Concurrent Engineering Methodology to the Design of a Dual Use VTOL Aircraft. AIAA - 94 - 4329 - CP, 1994.
- [16] R. R. Hill. Engineering Research Technology For Concurrent Engineering Form Unified Life Cycle Engineering. Wright-Patterson Air Force Base.
- [17] Ruan Lian. Concurrent Engineering And It's Practice In the Program Development In China. IRMS - 96, 1996.
- [18] S. Virtanen. Reliability in product design-Specification of dependability requirements. Proceedings Annual Reliability and Maintainability Symposium, 1998.
- [19] J. Krammer, 等. Concurrent Engineering in Design of Aircraft Structures. JOURNAL OF AIRCRAFT VOL. 32 NO. 2. March-April, 1995.
- [20] 杨为民, 等. 可靠性、维修性、保障性总论[M]. 北京: 国防工业出版社, 1995.
- [21] 章文晋, 孙宇锋, 马麟. 数字化平台下可靠性、维修性、保障性 CAD 技术应用[J]. 北京: 可靠性系统工程, 2007, 6(2).
- [22] 马绍民, 章国栋. 综合保障工程[M]. 北京: 国防工业出版社, 1995.
- [23] 陈学楚, 等. 装备系统工程[M]. 北京: 国防工业出版社, 2005.
- [24] 王轶辰, 蒋崇武. 实时嵌入式软件可靠性仿真测试工程化环境——模型驱动的嵌入式软件仿真测试方法. 国防科学技术报告, 2008.