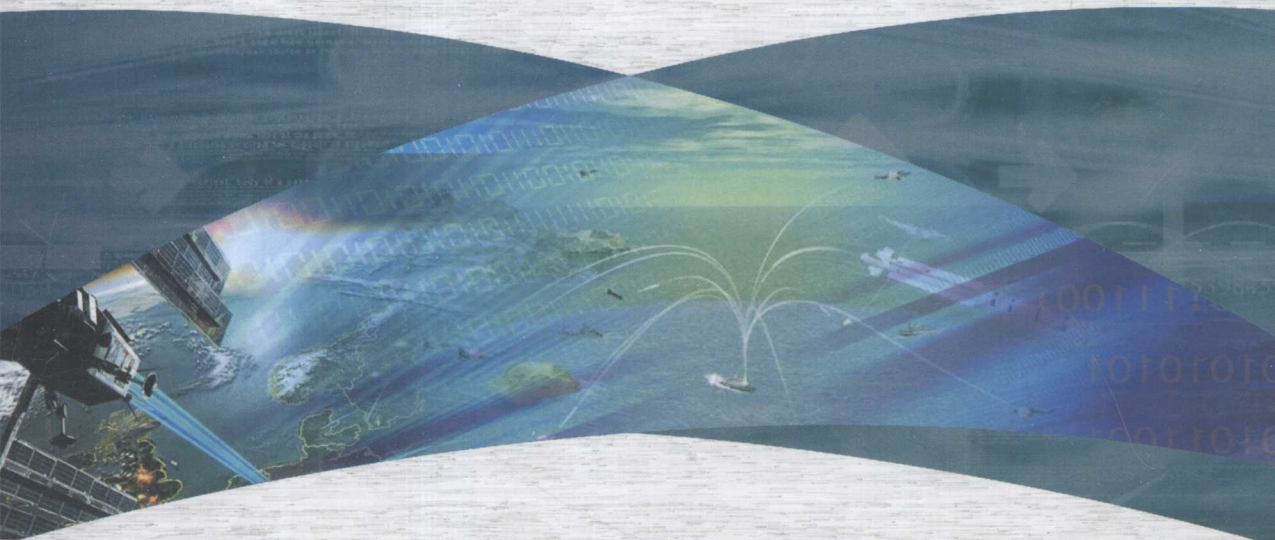


高等学校十二五规划教材·国防科技类

# 军用软件工程

JUNYONG RUANJIAN GONGCHENG

叶雪梅 陈菁 主编



西北工业大学出版社

# 军用软件工程

叶雪梅 陈 菁 主编

叶雪梅 陈 菁 蔡艳宁 安 利 范青刚 编著

西北工业大学出版社

图例(PI)目录别寄并图

军用软件

软件工程

ISBN 978

1. 019

1. 019

中国版本

西北工业大学出版社

地址: 西安市太白南路

电话: (029) 88493611

网址: www.nwpu.edu.cn

印刷: 西安元元印刷厂

尺寸: 787mm × 1092mm

张数: 14.25

印数: 3000

日期: 2015年1月

定价: 29.00元

**【内容简介】** 本书在对军用软件及其相关概念介绍的基础上,阐述了基于项目的面向对象的软件开发方法、基于生命周期的迭代式软件研发过程、军用软件项目管理及军用软件过程改进,并对军用软件工程相关标准进行了介绍。

本书主要用做软件工程课程的教材,适用于计算机科学与技术专业、信息研究与安全专业本科生或硕士研究生、各类继续教育人员,亦可作为高等院校计算机科学与技术专业、软件工程专业或其他相关专业的教学参考书,或作为从事软件开发、项目管理及软件过程改进等工作人员的参考书、培训教材等。

### 图书在版编目(CIP)数据

军用软件工程/叶雪梅,陈菁主编. —西安:西北工业大学出版社,2012.1  
高等学校规划教材. 2110 系列教材  
ISBN 978-7-5612-3282-8

I. ①军… II. ①叶…②陈… III. ①军用计算机—软件工程—高等学校—教材  
IV. ①E919

中国版本图书馆 CIP 数据核字(2012)第 003401 号

出版发行:西北工业大学出版社

通信地址:西安市友谊西路 127 号 邮编:710072

电 话:(029)88493844 88491757

网 址:www.nwpup.com

印 刷 者:陕西天元印务有限责任公司

开 本:787 mm×1 092 mm 1/16

印 张:14.375

字 数:349 千字

版 次:2012 年 1 月第 1 版 2012 年 1 月第 1 次印刷

定 价:29.00 元

# 前 言

军用软件工程是个新兴的学科,它是与软件开发与管理所用军用标准及工程化的思想和方法来指导军用软件开发和管理的整个过程的学科,它具有鲜明的实践性。近年来在军用计算机和军用信息管理等领域中,它的地位显得越来越重要,不仅大型软件项目离不开它,一般的、甚至小型软件项目也必须运用它的概念、原则和方法。软件经理,软件分析人员,软件设计、编程、测试、维护人员和软件配置管理人员以及质量保证人员,都必须很好地掌握软件工程、软件项目管理及软件过程改进知识,才能适应岗位工作的要求。特别是在当前我军信息化建设的形势下,军用软件工程更加受到重视。事实也一再表明,不掌握软件工程知识,不按照软件工程的要求去做软件项目,提供给部队的产品只能是低质量的、难以维护的。

本书系统地介绍了与军用软件相关的软件工程的有关概念、原理、方法、技术、标准和相关管理技术,突出了面向对象技术在软件开发过程中的应用以及软件项目管理、软件过程改进的重要性。全书共分6章,第1章主要对军用软件工程的产生、概念、有关标准以及开发过程标准进行概述;第2章主要阐述面向对象技术和统一建模语言,并对建模工具 Rational Rose 进行详细介绍;第3章主要对军用软件研发过程进行详细论述;第4章主要对软件项目管理有关内容进行介绍;第5章主要论述军用软件过程管理和过程改进相关内容;第6章主要介绍军用软件开发、军用软件项目管理及过程改进中的几个主要的军用软件标准。

全书从方法介绍的角度出发,内容紧凑,每章之后都配有习题,讲述力求理论联系实际,深入浅出,循序渐进。本书由叶雪梅、陈菁主编,具体编写分工如下:第1章、第3章及第6章由叶雪梅编写,第2章、第4章由陈菁编写,第5章由蔡艳宁编写,全书习题及小结部分由安利、范青刚编写。全书由叶雪梅统稿,汪洪桥、陈柏松审阅了书稿。

在本书的编写过程中,参考和引用了许多专家、学者的著作和论文,在正文中未一一注明,在此,谨向相关参考文献的作者表示衷心的感谢。

由于水平有限,书中不当之处,恳请读者批评指正。

编著者

2011年10月

# 目 录

<b>第 1 章 军用软件工程的基本概念</b> .....	1
1.1 软件的发展及软件工程的产生 .....	1
1.2 军用软件及军用软件工程 .....	8
1.3 军用软件工程标准 .....	10
1.4 军用软件工程方法 .....	18
1.5 军用软件开发过程模型 .....	21
1.6 小结 .....	30
习题 1 .....	31
<b>第 2 章 面向对象的软件开发方法</b> .....	32
2.1 面向对象技术 .....	32
2.2 统一建模语言 .....	44
2.3 Rose 简介 .....	54
2.4 小结 .....	59
习题 2 .....	59
<b>第 3 章 军用软件研发过程</b> .....	61
3.1 迭代式软件研发过程 .....	61
3.2 需求分析 .....	65
3.3 系统设计 .....	79
3.4 软件编码 .....	97
3.5 软件测试 .....	101
3.6 软件维护 .....	110
3.7 小结 .....	122
习题 3 .....	123
<b>第 4 章 军用软件项目管理</b> .....	126
4.1 概述 .....	126
4.2 进度管理 .....	128
4.3 质量管理 .....	133
4.4 软件成本估算 .....	137
4.5 人员管理 .....	142
4.6 配置管理 .....	145

4.7 风险管理 .....	149
4.8 小结 .....	151
习题 4 .....	151
<b>第 5 章 军用软件过程管理和过程改进</b> .....	<b>153</b>
5.1 基本概念 .....	153
5.2 经典的软件过程管理方法论 .....	156
5.3 军用软件能力成熟度模型 .....	172
5.4 软件过程改进 .....	173
5.5 SPI 的实施 .....	174
5.6 小结 .....	178
习题 5 .....	178
<b>第 6 章 军用软件工程标准</b> .....	<b>179</b>
6.1 军用软件开发通用标准及其修订 .....	179
6.2 军用软件开发文档通用标准及其修订 .....	195
6.3 军用软件研制能力成熟度模型(GJB 5000A—2008) .....	204
6.4 小结 .....	223
习题 6 .....	223
<b>参考文献</b> .....	<b>224</b>

# 第1章 军用软件工程的基本概念

GJB 2786A—2009《军用软件开发通用要求》对计算机软件(或软件)进行了下述定义:软件是指能使计算机硬件完成计算和控制功能的有关计算机指令和计算机数据定义的组合。它包括程序、相关数据及其说明文档。其中,程序是按照事先设计的功能要求执行的指令序列,数据是程序能正常操纵信息的数据结构,文档是与程序开发维护和使用有关的各种图文资料。

软件工程是针对软件这一具有特殊性质的产品的工程化方法。它涵盖了软件生存周期的所有阶段,并提供一整套工程化方法来指导软件人员的相关工作。

军用软件工程是结合军用软件标准与军用软件特点提出的适用于军用软件的工程化方法。

本章通过介绍软件、软件危机和软件工程的基本概念,讲解军用软件工程的发展及现状,并介绍军用软件工程标准和军用软件工程方法等。

## 1.1 软件的发展及软件工程的产生

### 1.1.1 软件的发展

20世纪50年代,软件伴随着第一台电子计算机的问世诞生了。以编写软件为职业的人员也开始出现,到60年代美国大学里开始出现授予计算机学位的专业,设置相关课程教人们编写软件。在短短50年的时间里,软件产业迅速发展成为推动人类社会发展的支柱产业之一,并成为人类社会信息化发展不可或缺的重要组成部分。从总体上讲,计算机软件的发展经历了3个时期。

#### 1. 程序设计时期(1946—1956年)

这个时期的软件生产方式主要是个体手工劳动。使用的工具是机器语言、汇编语言;开发的方法是追求编程技巧,追求程序运行效率,使得程序难读、难懂、难修改;硬件特征是价格贵,存储容量小,运行可靠性差;软件特征是只有程序、程序设计概念,不重视程序设计方法。其主要有以下特点。

- (1) 只有改善硬件结构,才能提高计算机系统的性能。
- (2) 编程只是计算机运行时必须进行的准备工作。
- (3) 程序是为解决特定问题而专门设计的,缺乏通用性。
- (4) 编程是个人行为,是个体生产方式。
- (5) 程序只有清单,无文档说明和设计资料,程序的设计文档不是必需的。

以上这些特点的核心就是:只有程序概念,没有软件概念。

#### 2. 程序系统时代(1956—1968年)

从20世纪50年代末到70年代初,计算机硬件技术有了较大发展,其特点是:

- (1)出现了通道技术和中断技术,计算机速度有了很大提高。
- (2)外存设备和人机交互设备得到了较大改进,计算机开始普及。
- (3)计算机应用从科学计算扩展到了数据处理和实时控制。
- (4)出现了各种应用软件和系统软件,如编译软件、操作系统等。

(5)软件规模增大,通用性增强,其表现为:程序的生产由个体生产方式变成了多人分工,协作编制完成,也就是说程序的生产已基本完成了从个体方式向作坊方式的转变。软件的维护由多人承担。软件维护是指软件在使用之后,为了改正错误或满足新的需求而修改软件的过程,除了改正性维护和预防性维护外,一个重要的维护工作就是适应性维护。也就是说,大约每36个月就有新一代的软件出现,时常会增加或修改外部设备或其他系统部件,所以操作系统和其他系统需要更新版本,同样部分应用软件也需修改部分软件和参数,所以维护量较大。软件不再是计算机硬件的附属成分,而成为计算机系统中与硬件相互依存的不可缺少的部分,而且这时软件在计算机成本中的比例超过硬件。

(6)软件技术有了较大的发展,出现了多道程序技术、多用户的人机交互系统和文件管理系统。

以上这些表明,这时的程序如果没有说明,别人将无法应用,因而出现了软件工程师和软件产业,而且在当时形成了这样的状况:大型软件不断出现,大型软件的需求不断增加,而软件的生产却是作坊式的;软件的大型化和作坊式生产之间的矛盾,导致了程序错误的增多。各类软件及OS的更新,需要修改部分程序和程序参数,软件维护费用急剧增加;开发大型软件需几十、几百甚至几千人合作,作坊式的生产方式不仅满足不了软件的需要,而且使程序变得不可维护,于是就出现了软件危机。

最为典型的例子是IBM公司于1963—1966年开发的IBM360系列机的操作系统。IBM是世界上著名的计算机大公司,IBM360是计算机发展史上的一个里程碑,当人们谈论计算机发展时,都会提到IBM公司IBM360计算机系统。就是这样一个著名的IBM360计算机的操作系统,虽然花费了数千人3年的时间,历尽了艰辛,其研制也令人失望。该项目的负责人Fred Brooks在总结该项目时生动地描述了研制过程的困难和混乱,他说:“就像巨兽在泥潭中作垂死挣扎,挣扎得越猛,就陷得越深,最后没有一个野兽能逃脱被淹没的命运。程序设计就像这样一个泥潭,一批批程序员被迫在泥潭中挣扎,谁也没有料到问题竟会变得这样棘手。”

这个例子形象地形容了当时的“软件危机”现象。

### 3. 软件工程时期(20世纪70年代至今)

为了摆脱软件危机的困境,北大西洋公约组织成员国于1968年和1969年连续召开软件工程研讨会,研究摆脱软件危机的对策,提出了“如何以规范化、工程化的方法、手段和技术来开发软件”的问题,进而提出了软件工程的观念。后来便出现了软件工程学这样一个软件方面的学科分支。

几十年来,软件工程已取得重大进展,各种软件工程技术得到了较为广泛的应用,对于推进软件技术的进步和软件产业的发展起到了十分积极的作用。但从总体上讲,软件的供需矛盾仍然很突出。关于软件工程时期软件的一些功能和特点,将在下文中作详细的介绍。

#### 1.1.2 软件危机

所谓软件危机,简单地说,就是指在软件开发和软件维护过程中所存在的一系列严重问



题。具体地说,软件危机具有下述一些表现。

(1)软件开发没有真正的计划性,对软件开发进度和软件开发成本的估计常常很不准确,计划的制定带有很大的盲目性,因此工期超出、成本失控的现象经常困扰着软件开发者。

(2)对于软件需求信息的获取常常不充分,软件产品往往不能真正地满足用户的实际需求。

(3)缺乏良好的软件质量评测手段,从而导致软件产品的质量常常得不到保证。

(4)对于软件的可理解性、可维护性认识不够,软件的可复用性、可维护性不如人意。有些软件因为过于“个性化”,甚至是难以理解的,更谈不上进行维护。缺乏可复用性引起的大量重复性工作极大地降低了软件的开发效率。

(5)软件开发过程没有实现“规范化”,缺乏必要的文档资料或者文档资料不合格、不准确,难以进行专业维护。

(6)软件开发的人力成本持续上升,如美国在1995年的软件开发成本已经占到了计算机系统成本的90%,如图1-1所示。

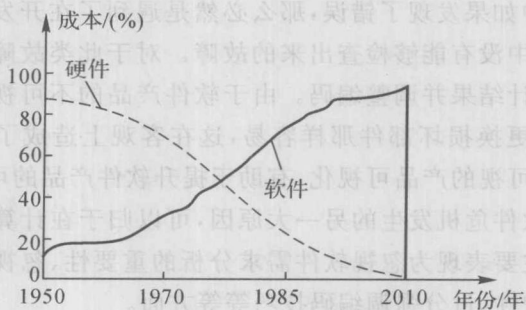


图 1-1 计算机系统硬件、软件成本比例变化

(7)缺乏自动化的软件开发技术,软件开发的生生产率依然低下,远远满足不了急剧增长的软件需求,如图1-2所示。

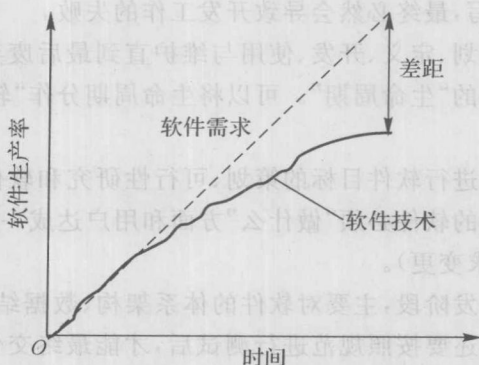


图 1-2 软件技术发展的滞后性

软件危机曾经是历史上的阴影,目前软件工程界也仍然在一定程度上受到它的影响。软件工程概念的提出,正是为了克服软件危机。自1968年以来,随着软件工程学的不断发展,软

件危机得到了一定程度的遏制,但还远远没有被彻底解决。《The Standish Group. Chaos. 1995》一文报告了20世纪90年代中期美国商用软件产业的情况:1995年美国公司取消了810亿美元的软件项目;在所考察的软件项目中,完成前就取消了其中的31%;53%的软件项目进度拖延,通常拖延的时间超过预定工期50%以上;只有9%的大型软件项目能够及时交付且费用不超支(对中型和小型软件公司来说这一数据为16%)。从上面的统计数据不难看出,在软件开发过程中,危机的影响依然存在。再回顾一下世纪之交时,为防治软件“千年虫”问题所投入的巨大人力与物力,也反映了软件维护工作的艰巨与困难。

软件危机的存在是不争的事实,产生软件危机的原因可以归纳为主、客观两个方面。

从客观上来看,软件不同于硬件,它的生产过程和产品都具有明显的“不可视”特征,这就导致在完成编码并且上机运行之前,对软件开发过程的进展情况较难衡量,软件产品的质量也较难进行先期评价,因此,对开发软件的过程进行管理和控制比较困难。在软件工程的早期,制定详细的开发计划并且进行全程跟踪调控,对于所有的阶段产品和阶段工作进展进行技术审查和管理复审,可望在一定程度上克服“开发过程不可视”造成的消极影响。

此外,软件运行过程中如果发现了错误,那么必然是遇到了在开发时期(分析、设计、编码过程)引入的,在检测过程中没有能够检查出来的故障。对于此类故障的维护,通常意味着要修改早期的分析结果和设计结果并调整编码。由于软件产品的不可视特征,维护过程不像硬件产品维护时只要简单地更换损坏部件那样容易,这在客观上造成了软件难以维护的结果。利用足够的文档资料使不可视的产品可视化,有助于提升软件产品的可理解性和可维护性。

从主观上分析,导致软件危机发生的另一大原因,可以归于在计算机系统发展的早期,软件开发的“个体化”特点,主要表现为忽视软件需求分析的重要性、忽视软件的可理解性、文档不完备、轻视软件的可维护性、过分强调编码技巧等等方

只有软件的用户才真正了解他们自己的需求。而且应当承认,用户一开始并不见得能够清晰、准确地表达自己的需求。软件开发人员需要做大量的、深入细致的调研工作,引导用户逐步准确、具体地描述对软件的需求,才能够得到对问题、目标的正确认识,从而获得解决问题的恰当出发点,有望开发出真正能够满足用户需求的软件产品。在对用户的需求没有清楚的认识时就仓促地进行程序编写,最终必然会导致开发工作的失败。

一般来说,软件产品从策划、定义、开发、使用与维护直到最后废弃,要经过一个漫长的时期,通常把这个时期称为软件的“生命周期”。可以将生命周期分作“软件定义”、“软件开发”和“运行与维护”3个阶段。

在软件定义阶段中,主要进行软件目标的策划、可行性研究和软件的需求分析工作,通过和用户多次交流,在所要开发的软件必须“做什么”方面和用户达成一致(当然在开发过程中也允许在严格的控制下进行需求变更)。

软件被定义之后,进入开发阶段,主要对软件的体系架构、数据结构和主要算法进行设计和编码实现。对于编码结果,还要按照规范进行测试后,才能最终交付使用。如前所述,在开发阶段也可能对于此前不够准确的软件定义结果进行调整。统计数据表明,在典型的软件工程中,编码工作量大约只占软件开发全部工作量的15%~20%。

软件的运行与维护阶段在软件生命周期中占据的时间最长。在软件运行过程中,分析和设计阶段的一些遗留缺陷可能会逐步暴露;运行环境的变化也会对运行中的软件提出变更要求;用户新需求的提出则常常要求扩充现有软件的功能或者改进其性能,所有这些要求与问题

都必须通过“软件维护”工作去解决。在维护过程中,必须注意保持所有软件产品之间的一致性。针对不同的需求,维护工作一般可以分为纠错性维护、适应性维护、扩充性维护和预防性维护等不同类型。

作为软件,应当有一个完整的配置。美国著名的软件工程专家,加州州立大学教授 Boehm 指出,软件是程序以及开发、使用、维护程序所需要的所有文档。所以,软件产品除包括程序之外,还应当包括完整、准确、翔实的文档资料。主要的文档应当包括“需求规格说明书”、“体系结构设计说明书”、“详细设计说明书”、“安装手册”、“操作手册”、“系统管理员手册”等。缺乏必要的配置文档,将严重影响软件的可理解性,从而给软件的维护造成严重障碍。

做好包括项目策划、可行性研究、需求分析三项内容的软件定义工作,是提高软件质量、降低软件成本、保证开发进度的关键环节。值得注意的是,在软件开发的各个阶段进行修改所付出的代价是极其不同的。在早期引入变动,涉及的面比较小,因而代价也比较低;在开发的中期,因为许多配置项已经完成,所以引入一个变动,就要对它所涉及的所有已经完成的配置项进行变更,不仅工作量大,而且逻辑上也更复杂,因此付出的代价剧增;如果在软件“已经完成”时再引入变更,更是要付出高得多的代价。根据美国一些软件公司的统计资料,软件开发后期引入一个变动比在早期引入相同变动所需付出的代价高 2~3 个数量级。图 1-3 所示定性描绘了在不同时期引入一个变动需要付出的代价的变动趋势。图 1-4 所示是美国贝尔实验室统计得出的定量结果。

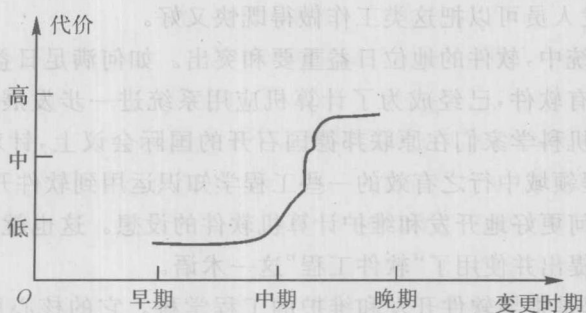


图 1-3 变更代价随时间变化的趋势示意图

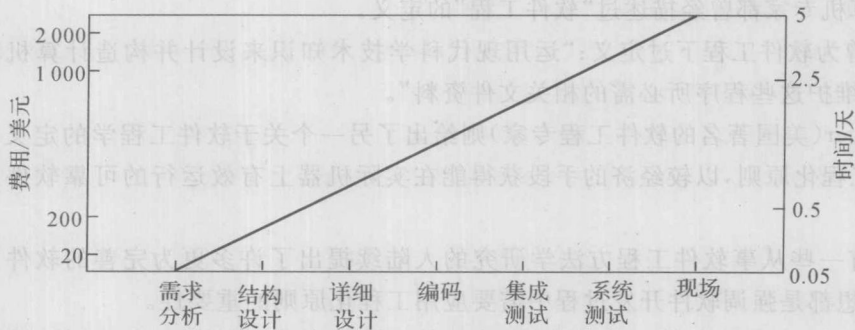


图 1-4 更正一个问题需要付出的代价

### 1.1.3 软件工程的产生

如何解决软件危机呢?可以借鉴其他工程领域的成功经验,基于软件危机产生的主、客观原因,从软件工程技术和软件工程管理两方面来采取措施,防范软件危机的发生。

软件开发不是某种个体劳动的神秘技巧,而应当是一种组织良好、管理严密,分析、设计、编码、测试、质量保证等各类人员协同配合,共同完成的工程项目。在软件开发过程中,必须充分吸收和借鉴人类长期以来从事各种工程项目所积累的行之有效的原理、概念、技术和方法,特别要注意吸收几十年来在计算机硬件研究和开发中积累的经验教训。

从管理层面上考虑,应当注意使用和推广在实践中总结出来的成功的技术和方法,并且探索更好的、更有效的技术和方法,注意积累软件开发过程中的经验数据财富,逐步消除在计算机系统早期发展阶段形成的一些错误概念和做法。建立适合于本组织的软件工程规范,制定软件开发中各个工作环节的流程文件、工作指南和阶段工作产品模板,实施针对软件开发全过程的计划跟踪和品质管理活动,为每一项工程开发活动建立配置管理库,实施严格的产品基线管理并建立组织的软件过程数据库和软件财富库,为各类员工及时提供必要的培训等都是加强软件开发活动管理工作的有效手段。

从技术角度考虑,应当开发和使用更好的软件开发工具,提高软件开发效率和开发工作过程的规范化程度。在计算机软件开发的各个阶段,都有大量的烦琐重复的工作要做,在适当的软件工具的辅助下,开发人员可以把这类工作做得既快又好。

现代计算机应用系统中,软件的地位日益重要和突出。如何满足日益增长的软件需求,如何维护应用中的大量已有软件,已经成为了计算机应用系统进一步发展的瓶颈。1968年,北大西洋公约组织的计算机科学家们在原联邦德国召开的国际会议上,针对软件危机的严峻形势,提出了把在其他工程领域中行之有效的一些工程学知识运用到软件开发过程中来,从管理和技术两个方面研究如何更好地开发和维护计算机软件的设计。这也就是软件工程的基本思路。在这次会议上首次提出并使用了“软件工程”这一术语。

简单地说,软件工程是指导软件开发和维护的工程学科。它的核心思想是采用工程的概念、原理、技术和方法来开发和维护软件,把经过实践考验证明是正确的管理技术和当前能够得到的最好的技术方法结合起来,从而大大提高软件开发的生產率和成功率。

许多计算机专家都曾经描述过“软件工程”的定义。

Boehm 曾为软件工程下过定义:“运用现代科学技术知识来设计并构造计算机程序及为开发、运行和维护这些程序所必需的相关文件资料”。

Fritz Bauer(美国著名的软件工程专家)则给出了另一个关于软件工程学的定义:“建立并使用完善的工程化原则,以较经济的手段获得能在实际机器上有效运行的可靠软件的一系列方法”。

后来又有一些从事软件工程方法学研究的人陆续提出了许多更为完善的软件工程的定义,但主要思想都是强调软件开发过程中需要应用工程化原则的重要性。

GB/T 11457—1995《软件工程术语》中将软件工程定义为:“软件开发、运行、维护和引退的系统方法”,目的就是为软件全生存周期活动提供工程化手段,从而提高软件质量、降低软件成本和缩短开发周期等。

相比之下,IEEE 给出了关于软件工程的一个更加综合的定义:

(1)将系统化的、规范的、可度量的方法应用于软件的开发、运行和维护过程,即将工程化方法应用于软件开发与维护过程中。

(2)对上述方法的研究。

就内容来看,软件工程应当包括3个要素:方法、工具和过程。

软件工程方法为软件开发提供了“如何做某项工作”的技术指南。它包括了多方面的任务,例如项目策划和估算方法、软件需求分析方法、体系结构的设计方法、详细设计方法、软件测试方法等等,使得整个开发过程的每一种阶段任务都能够“有章可循”。

软件工程工具为软件工程方法提供了自动的或半自动的软件支撑环境。目前这样的工具已经有许多种,而且已经有人把诸多软件工程工具集成起来,使得一种工具产生的信息可以为其他工具所使用,形成了一种称之为计算机辅助软件工程(CASE)的软件开发支撑环境。CASE把各种软件工具、开发机器和一个存放开发过程信息的工程数据库组合起来,形成了一个完整的软件工程环境。

软件工程中的“过程”是将软件工程的方法和工具综合起来以达到合理、及时地进行计算机软件开发的目的。可以将软件工程过程理解为软件工程的工艺路线。过程定义了各种方法使用的顺序、各阶段要求交付的文档资料、为保证质量和控制软件变更所需要的管理环节和在软件开发各个阶段完成的里程碑。

针对软件工程的基本要件,有许多计算机科学家进行了诠释,先后提出了100多条有关软件工程的相关原则。著名软件工程专家B. W. Boehm集众家所长,并总结了TRW公司多年开发软件的经验,在1983年提出了软件工程的7项基本原则,作为保证软件产品质量和开发效率的最小集合。具体包括:

(1)用分阶段的生命周期计划严格管理软件工程过程。

(2)坚持在软件工程过程中进行阶段评审。

(3)实行严格的产品控制。

(4)采用现代的开发技术进行软件的设计与开发。

(5)工作结果应当是能够清楚地审查的。

(6)开发小组的人员应该“少而精”。

(7)承认不断改进软件工程实践的必要性。

这7条原则是互相独立的、缺一不可的最小集合,同时又是相当完备的。可以证明,其他已经提出的100多条软件工程原理都可以由这7条原则的任意组合蕴含或派生。

从首次提出“软件工程”的概念开始,迄今已经经过了近半个世纪,在此期间,计算机硬件、软件技术领域都有了长足的发展,各种新产品、新技术、新方法、新工具不断问世。伴随着计算机科学与技术的进步,软件工程作为一门新兴学科也同样有了很大的发展。从传统的软件工程到面向对象的软件工程,从一般的软件工程到净室软件工程,从软件工程到软件再工程,从人工软件工程到计算机辅助软件工程,整个软件工程学正在日趋走向成熟,并在计算机应用领域中发挥着越来越大的作用。

## 1.2 军用软件及军用软件工程

### 1.2.1 军用软件的概念

军用软件是现代武器装备的灵魂,军用软件建设是军队实现信息化的关键。随着军队机械化、信息化水平的逐步提高,多种类型的、形形色色的军用软件将用于从单兵到全军的各个层次的指挥作战系统,用于以计算机作信息处理和系统控制的武器装备系统。

军用软件通常指用于军事目的的软件,一般可分为两大类:一类是武器系统软件,另一类是非武器系统软件(称为自动化信息系统软件)。武器系统软件包括为武器系统专门设计或专用的并成为整个系统不可缺少的一部分的嵌入式软件,指挥、控制和通信软件,对武器系统及其完成任务起保障作用的其他武器系统软件,如任务规划软件、战斗管理软件、后勤保障软件、演习分析软件、训练软件、飞行计划软件、应用测试软件、程序管理软件、模拟器软件等。非武器系统软件主要指执行与武器系统无关的系统使用和保障功能的软件,例如科学计算软件、人员管理软件、资源控制软件、地图管理软件、设备维修软件、仿真软件和人工智能软件等。

随着军队信息化程度的不断提高,军用软件已经渗透到军事应用的各个方面,成为武器装备体系中不可或缺的组成部分。现代武器系统称为“智能”武器,是因为软件为其提供了大脑。软件通过专用硬件的运行,可以完成许多的功能,如作战飞机的每一次使用基本上都依赖于软件,包括战略和战术行动,监视、探测、评估和预警等。在不少情况下,光有硬件是不行的,软件的功能甚至超出其他部件,如软件控制飞机的航向稳定度,使飞机的隐身技术成为可能。软件的重要性已在高科技的战争中得到证明,主要体现在下述几个方面。

(1)软件是高新武器装备的灵魂。在高新武器装备中,由软件实现的功能越来越多,有些装备由软件实现的功能甚至大大超过了硬件。软件不但能执行以前由硬件执行的许多功能,还能执行光靠硬件几乎无法执行的功能,例如,为减少雷达截面积,B-2轰炸机没有航向稳定面,飞机的垂直稳定度全靠软件来控制,从而满足了B-2轰炸机隐身的要求。软件的应用不仅极大地提高了武器装备系统原有的许多性能,而且已成为整个军事系统的控制中枢和威力倍增器,成为高新武器装备的灵魂。

(2)软件是构筑信息化装备体系的关键。现代战争是体系和体系的对抗,单一武器、单一系统的决胜作用已经逐渐弱化,由大量嵌入芯片和软件构成的各种武器装备形成的信息化装备体系已成为战争制胜的基础。构筑信息化装备体系绝不是各种装备的简单堆积,也不只是各种武器装备和系统之间的物理连通,而是通过软件的控制,使各种作战信息按照作战要求有序流动,满足体系内各部分之间互连互通互操作的要求,实现不同武器系统的功能互补、协同行动和互相支援。因此,软件已成为构筑信息化装备体系的关键。

(3)软件可有效提升武器装备的整体作战效能。由于软件能极大提高武器系统的信息获取、传输、处理、存储、管理、分发及其数字化、智能化、网络化水平,利用软件技术对已有武器装备进行改造已成为部分武器装备升级换代所依靠的主要模式之一。利用软件技术进行改造不仅能大大提高武器装备作战效能,而且还具有成本低、周期短和效果好的优点。例如,伊拉克战争前夕,美国海军F-14战斗机主要通过软件升级的方式具备了投放精确制导武器的能力,使改造周期从以往的几个月甚至几年缩短至3个星期。另外,美军的全球指挥控制系统近几

次的改进也主要是通过软件升级实现的。

(4) 软件是信息战中攻防对抗装备发展的焦点。信息战已成为现代战争的主要模式。一方面,作为一类特殊软件的计算机病毒已成为信息战进攻的重要手段,其作用主要是通过破坏敌方信息系统和计算机网络中的软件,达到使其瘫痪的目的。在海湾战争中,美军利用开发的计算机病毒“预埋”技术,在芯片中置入计算机病毒,给伊方造成了重大的损失。另一方面,信息安全软件、网络防护软件成为信息战防御的关键工具。为对抗计算机病毒和其他针对信息系统及网络中软件的攻击,保证其正常运行,各国都在开发以软件为主要形式的对抗措施。

### 1.2.2 军用软件的特点

软件本身的复杂性、抽象性和易变性使软件难以研制,质量难以保证。军用软件的特殊应用又使其开发和质量保证难度增大,主要体现在以下几个方面。

(1) 军用软件要面临复杂、不确定和恶劣的作战环境,必须具有抗毁和容错能力。因此,军用软件应具有高可靠性、高安全性和高生存性。

(2) 在信息对抗环境下,要求军用软件具有较高的安全防护能力。因此,军用软件应具有高保密性。

(3) 作战使命任务对军用软件的信息传输处理速度、对外部事件的快速响应能力提出了更高的要求。因此,军用软件应具有高的实时性要求。

(4) 军用软件不少是嵌入式的,受到严格的硬件和软件条件的约束,被硬件及软件体系结构、操作系统特性、应用需求和编程语言的变化所制约。

(5) 军用软件的开发还要纳入武器装备研制过程,这决定了军用软件开发方法要与硬件研制中采用的自下而上的方法相一致。

(6) 军用软件规模巨大,如一架现代战斗机所包含的软件已经超过 2 500 万行源代码,一艘现代化战舰或潜艇所包含的软件有 5 000 万行源代码。软件规模越大,就越复杂,这不仅带来了技术问题,而且为软件的管理带来了很大的困难。

(7) 现代化战争是一体化的联合作战,需要进行数据交换、信息共享、应用协同,这些均对军用软件提出了高互操作性要求。

上述军用软件的特点对军用软件研制管理的方法和技术都有特殊要求,同时还要求开发人员具备相关应用领域的知识,另外,军用软件对开发过程、工具和安全保密等方面都有较高的要求。

### 1.2.3 军用软件工程

以信息技术为核心的现代高技术局部战争中,无论在作战指挥、武器装备还是在后勤保障等方面,军用软件都将发挥越来越重要的作用。随着我军信息化建设步伐的加快,军用软件开发的高峰正在到来,但是目前我军军用软件开发在很多方面还不规范、不科学,从而导致了军用软件的质量不高、性能不强、适应性差。鉴于此,采用工程化的方法对军用软件进行开发和管理,即军用软件工程技术在军用软件的开发管理中逐渐形成。

军用软件工程定义为军用软件开发、运行、维护和引退的系统方法,是软件工程理论和方法在军事领域中的应用。针对军用软件的特殊应用和特点,军用软件工程包括军用软件的开发方法与技术、工具与环境、管理以及军用软件工程标准 4 个要素。

### 1.3 军用软件工程标准

#### 1.3.1 军用软件工程标准的发展

我国软件工程标准化工作 20 世纪 80 年代初才开始起步,经过 20 多年的努力,颁布了多项国标和国军标,另如航空、航天、电子、机械等部门也基于行业软件管理的需要分别制定了若干行业标准,国内大型工程项目也制定了自己的软件工程标准规范。

1983 年军用标准化工作实行统一管理以后,软件工程标准化工作受到各有关方面的重视,取得了可喜的成绩,迄今已颁布的近 50 项国军标中的大多数标准已被军内外广泛应用,对一些大型信息系统工程及重点武器型号的研制、生产及使用起到了积极的促进作用。按照“积极采用国际标准及国外先进标准”的技术政策,我国军用软件工程标准绝大多数都是参考美国军用标准,并结合我国具体情况制定的,但由于共知的原因,我国军用软件工程标准整体水平还是落后于国外先进国家,整体上我国同类标准的出台,一般落后美国军标近 10 年左右。然而相对于目前我国软件工程化水平而言,这些标准仍具有先进性和指导性,至少在今后的相当时期内仍将会发挥重要的作用。

军用软件的开发与维护都应该在 GJB 的指导下进行,现将重要的国家军用标准列举见表 1-1。

表 1-1 军用软件工程标准举例

标准号	标准名称	发布部门
GJB 5000A—2008	军用软件研制能力成熟度模型	总装电子信息基础部
GJB 438B—2009	军用软件开发文档通用要求	总装电子信息基础部
GJB 2786A—2009	军用软件开发通用要求	总装电子信息基础部
GJB/Z 115—1998	GJB 2786《武器系统软件开发》剪裁指南	原国防科工委司令部
GJB 2694—1996	军用软件支持环境	原国防科工委司令部
GJB/Z 102—1997	软件可靠性和安全性设计准则	原国防科工委司令部
GJB 2434A—2004	军用软件产品评价	原国防科工委司令部
GJB 439—1988	军用软件质量保证规范	原国防科工委司令部
GJB 3181—1998	军用软件支持环境选用要求	原国防科工委司令部
GJB/Z 142—2004	军用软件安全性分析指南	总装电子信息基础部
GJB 1091—1991	军用软件需求分析	原国防科工委司令部
GJB 1267—1991	军用软件维护	原国防科工委司令部
GJB 2115—1994	军用软件项目管理规程	原国防科工委司令部
GJB 2255—1994	军用软件产品	原国防科工委司令部
GJB 5234—2004	军用软件验证和确认	总装电子信息基础部
GJB 5235—2004	军用软件配置管理	总装电子信息基础部



续表

标准号	标准名称	发布部门
GJB 5236—2004	军用软件度量	总装电子信息基础部
GJB 2434A—2004	军用软件产品评价	总装电子信息基础部
GJB 1268A—2004	军用软件验收要求	总装电子信息基础部
GJB/Z 141—2004	军用软件测试指南	总装电子信息基础部
GJB 6143—2008	军用软件评审和审核	总装电子信息基础部

### 1.3.2 重点标准分析

本小节介绍了几个常见的军用软件工程标准,有些标准虽然已进行重新修订,但鉴于其在军用软件工程发展历史上的重要作用,这里也一并作介绍。

#### 1. GJB 437《军用软件开发规范》

《军用软件开发规范》是第一个软件工程国军标,它规定了软件生命周期中软件需求分析、软件设计、软件实现和软件测试的基本要求,同时它还涉及这些阶段中的软件质量保证、软件配置管理、软件开发管理和软件文档编制等方面的内容。GJB 437 为军用软件开发规定了统一的最低要求,而 GJB 438, GJB 439, GJB 1090, GJB 1267 和 GJB 1268 则是对 GJB 437 要求的补充和细化,在 GJB 437 的实施过程中,常常需要与这些标准配套使用。GJB 437, GJB 438 和 GJB 439 等标准在我国军用软件的开发中曾起到十分重要的作用,然而随着软件工程技术的迅速发展以及计算机在军事领域中更广泛深入的应用, GJB 437 已难以满足当前软件开发等各方面的需要。首先,现代武器装备特别是大型武器系统的软件往往嵌入到系统中,与设备或其他分系统密不可分,因此在系统研制一开始就必须考虑软件问题。然而 GJB 437 则是将软件系统作为单纯软件来考虑,淡化了软件与整个武器系统的关系,因此导致系统的要求较难向下分解,软件与设备或其他分系统的接口较难定义,系统集成和测试也较难实现。其次,采用 GJB 437 意味着按照瀑布式开发模型进行软件开发,限制了软件开发人员对目前较先进的开发模型、开发方法的使用,因此也给军用软件的开发工作带来了很大不便。另外, GJB 437 的内容也不够完善,对开发过程中涉及的风险管理、安全性等问题都未做出规定。

#### 2. GJB 2786—1996《武器系统软件开发》

该标准规定了武器系统软件开发和保障的基本要求,适用于软件生存周期的全过程,为软件的订购方(使用方)了解承制方的软件开发、测试和评价工作提供了依据。GJB 2786 克服了 GJB 437 的不足,充分体现了系统工程和软件工程的思想,具有鲜明的特点。特点之一就是提供了承制方在满足合同或任务书要求前提下的灵活性。在标准中,它定义了软件开发的 8 项主要活动,并特别说明这些活动可以重叠,也可以交叉或循环进行,因此 GJB 2786 对许多软件开发模型来说都是可接受的,即其本身可接受多种不同的软件开发方法。标准中唯一带有约束性的是承制方应使用有充分的文件证明的、系统化的软件开发方法,且该方法应支持合同要求的正式审查和审核。

GJB 2786 规定了软件开发的 8 项主要活动。

##### (1) 系统要求分析和设计。