



# 软件工程化与 机载软件适航审查实践

Software Engineering and  
Airworthiness Review Practice of  
Airborne Software

主 编 王焕彬 崔利杰  
副主编 任 博 李 强 李 泽



国防工业出版社  
National Defense Industry Press

# 软件工程化与机载软件 适航审查实践

主 编 王焕彬 崔利杰  
副主编 任 博 李 强 李 泽  
编 委 王 澈 刘文杰 张 磊

国防工业出版社

· 北京 ·

## 内 容 简 介

本书重点介绍了软件工程化的相关理论、方法、过程、模型以及机载软件适航审查的相关标准、过程、方法。其中,软件工程化理论与实践部分主要介绍了软件工程的基本原理、软件生命周期的基本模型以及软件需求分析、设计、测试、维护的相关方法,在实践部分主要对配置管理过程和质量保证过程进行系统介绍,结合具体过程,给出了相应的文档编写模板以及各类过程的评审检查单,便于广大读者在软件开发实际工作中借鉴和应用;机载软件适航审查理论与实践部分主要介绍了机载软件适航的理论、机载软件适航的标准、机载软件适航审查的过程和方法,给出了机载软件适航审查的流程、方法以及各个审查阶段应满足的目标,从而为机载软件适航审查人员在具体的审查工作中提供指导和帮助。

本书的实践性较强、适用面较广,可作为高等院校软件工程、计算机技术及信息技术等专业本科生和研究生学习软件工程的教材,也可以作为软件工程师以及软件适航审查人员从事软件开发管理以及软件适航审查等工作的参考资料。

### 图书在版编目(CIP)数据

软件工程化与机载软件适航审查实践/王焕彬,崔利杰  
主编.—北京:国防工业出版社,2019.1  
ISBN 978-7-118-11689-2

I. ①软… II. ①王… ②崔… III. ①机载计算机—  
软件工程—研究②机载计算机—软件—适航性—检查—  
研究 IV. ①V247.1

中国版本图书馆CIP数据核字(2018)第231226号

※

国防工业出版社出版发行

(北京市海淀区紫竹院南路23号 邮政编码100048)

三河市天利华印刷装订有限公司

新华书店经售

\*

开本 787×1092 1/16 印张 24¼ 字数 564 千字

2019年1月第1版第1次印刷 印数 1—1500册 定价 68.00元

(本书如有印装错误,我社负责调换)

国防书店:(010)88540777

发行邮购:(010)88540776

发行传真:(010)88540755

发行业务:(010)88540717

# 前 言

软件工程化相关理论和方法是软件开发过程中的理论指导和方法指南,而对于在各类飞机上运行的机载软件而言,由于飞行安全的特殊要求,机载软件的开发过程仅仅依靠软件工程化的相关理论指导是不够的,因此,机载软件的开发过程还需遵守软件适航的相关标准。在国外一些发达国家(如美国等),机载软件开发过程中已有成熟的标准和方法,这些标准和方法已涵盖了民航领域和军机机载软件研制领域,而在我国,虽然软件适航在民航领域已有深入的研究和较广泛的应用,但在军机机载软件研制过程中,软件适航才刚刚起步。面对这现状,本书的编写人员从软件工程化和机载软件适航审查的角度进行了相关的研究和探索,总结了相关的理论、方法和经验,编写了本教材。

本书内容共分为 11 章,主要包括两大部分内容:一是软件工程化理论与实践相关内容;二是机载软件适航审查理论与实践相关内容。其中,第 1 章至第 6 章主要介绍软件工程化理论与实践的相关知识,第 7 章至第 11 章主要介绍机载软件适航审查理论与实践的相关知识。

本书是集体劳动的成果,参加编写的人员有王焕彬、崔利杰、任博、王澈、李泽、刘文杰、张磊、李强,其中第 1、2、3、4 章由王焕彬编写,第 5、6 章由王澈编写,第 7、8 章由崔利杰编写,第 9、10、11 章由任博编写,李泽、张磊、李强和刘文杰主要负责全书的统稿和文字校对工作。在本书编写过程中得到了中航工业有关研究所的研究人员、空军有关军事代表局军代表等的帮助,同时,在本书编写过程中还参考了吴军华、臧铁钢、肖孟强、窦万峰、曾强聪、杨丰、吕云翔、郭宁、张家浩、张晓龙、蔡喆、沈小明等老师的著作,在此对上述老师以及书中参考文献的作者表示衷心感谢。由于编者水平有限,书中难免有疏漏和错误,敬请广大读者提出宝贵意见和批评指正,谢谢。

编 者

2018 年 3 月

# 目 录

<b>第 1 章 软件工程化引论</b> .....	1
1.1 软件危机的产生 .....	1
1.2 软件工程基础 .....	2
1.3 本章小结 .....	31
<b>第 2 章 软件生命周期及开发模型</b> .....	32
2.1 软件生命周期概述 .....	32
2.2 软件生命周期模型 .....	34
2.3 软件过程 .....	40
2.4 本章小结 .....	42
<b>第 3 章 软件需求分析方法</b> .....	44
3.1 软件需求分析概述 .....	44
3.2 软件需求获取 .....	48
3.3 结构化需求分析 .....	52
3.4 面向对象需求分析 .....	53
3.5 本章小结 .....	75
<b>第 4 章 软件设计方法</b> .....	76
4.1 软件设计概述 .....	76
4.2 软件设计原理 .....	79
4.3 软件架构设计 .....	81
4.4 软件详细设计 .....	90
4.5 本章小结 .....	99
<b>第 5 章 软件测试与维护技术</b> .....	100
5.1 软件测试概述 .....	100
5.2 软件测试技术 .....	103
5.3 系统测试及方法 .....	110
5.4 软件维护概述 .....	115
5.5 软件维护的类型 .....	117
5.6 软件维护的过程 .....	118
5.7 本章小结 .....	127
<b>第 6 章 软件工程化实践</b> .....	129
6.1 软件过程管理 .....	129
6.2 软件配置管理 .....	160

6.3	软件质量保证管理 .....	189
6.4	军事代表在军用软件质量监督中的职责 .....	238
6.5	本章小结 .....	244
<b>第7章</b>	<b>机载软件适航概述 .....</b>	<b>245</b>
7.1	机载计算机及机载软件 .....	245
7.2	机载软件适航要求 .....	251
7.3	机载软件安全性与可靠性 .....	252
7.4	DO-178B 标准介绍 .....	260
7.5	DO-178B 应用介绍 .....	265
7.6	本章小结 .....	285
<b>第8章</b>	<b>机载软件计划过程及适航审查 .....</b>	<b>287</b>
8.1	计划过程概述 .....	287
8.2	计划文档的适航审查要求 .....	290
8.3	软件合格审查计划 .....	291
8.4	软件开发计划 .....	302
8.5	软件验证计划 .....	308
8.6	软件配置管理计划 .....	319
8.7	软件质量保证计划 .....	325
8.8	本章小结 .....	331
<b>第9章</b>	<b>机载软件验证过程及适航审查 .....</b>	<b>332</b>
9.1	软件验证过程概述 .....	332
9.2	评审和分析 .....	334
9.3	测试过程 .....	336
9.4	结构覆盖 .....	340
9.5	验证独立性 .....	344
9.6	数据耦合和控制耦合 .....	348
9.7	本章小结 .....	350
<b>第10章</b>	<b>机载软件构型管理过程、质量保证过程及适航审查 .....</b>	<b>351</b>
10.1	构型管理过程 .....	351
10.2	质量保证过程 .....	357
10.3	本章小结 .....	362
<b>第11章</b>	<b>机载软件合格审查过程 .....</b>	<b>363</b>
11.1	合格审查过程概述 .....	363
11.2	软件合格审查的实施过程和工具 .....	363
11.3	软件生命周期资料 .....	372
11.4	本章小结 .....	379
<b>参考文献</b>	<b>.....</b>	<b>380</b>

# 第 1 章 软件工程化引论

随着各种不同类型软件的出现以及软件开发规模的不断扩大,出现了软件危机。为有效解决软件危机问题,学者们提出软件工程的思想。软件工程结合了计算机科学和现代工程的原则,它对指导软件开发、提高软件质量和开发效率具有重要的作用。本章将概括性介绍软件危机的产生、软件工程的定义、基本原理、框架以及软件工程方法等内容。

本章学习重点:

- (1) 软件危机以及软件危机产生的原因;
- (2) 软件工程定义及基本原理;
- (3) 软件工程相关方法。

## 1.1 软件危机的产生

### 1.1.1 软件危机的出现

20 世纪 60 年代中期以后,计算机硬件技术日益进步,计算机的存储容量、运算速度和可靠性明显提高,生产硬件的成本不断降低,从而为计算机的广泛应用创造了极好的条件。在这种形势下,迫切要求计算机软件也能与之相适应。因而,一些开发大型软件系统的要求被提了出来。然而软件技术的进步一直未能满足形势发展的需要,在大型软件的开发过程中出现了复杂程度高、研制周期长、正确性难以保证这三大难题。遇到的问题找不到解决办法,致使问题堆积起来,形成了人们难以控制的局面,出现了“软件危机”。

最为突出的例子是美国 IBM 公司于 1963—1966 年开发的 IBM360 系列机的操作系统。该软件系统花了大约 5000 人年的工作量,最多时,有 1000 人投入开发工作,写出近 100 万行的源程序。尽管投入了这么多的人力和物力,得到的结果却极其糟糕。据统计,这个操作系统每次发行的新版本都是从前一版本中找出 1000 个程序错误而修正的结果。可想而知,这样的软件质量糟到了什么地步。IBM360 操作系统的历史教训已作为软件开发项目中的典型事例被记入史册。

如果开发的软件隐含错误,可靠性得不到保证,那么在运行过程中很可能对整个系统造成十分严重的后果,轻则影响到系统的正常工作,重则导致整个系统的瘫痪,乃至造成无可挽回的恶性事故。如,银行的存款可能化为乌有,甚至弄成赤字;工厂的产品全部报废,导致工厂破产。

1963 年,美国用于控制火星探测器的计算机软件中的一个“,”被误写为“。”,致使飞往火星的探测器发生爆炸,造成高达数亿美元的损失。

## 1.1.2 软件危机的原因

造成软件危机的原因很多,一般情况下可简单地归纳为如下几点:

(1) 软件开发无计划性。缺乏软件开发的经验和有关软件开发数据的积累,使得开发工作的计划很难制定,致使经费预算常常超支,进度计划无法遵循,开发完成的期限一拖再拖。

(2) 软件需求不充分。作为软件设计依据的需求,在开发的初期阶段提得不够明确,或是未能得到确切的表达。开发工作开始后,软件人员和用户又未能及时交换意见,造成开发后期矛盾的集中暴露。

(3) 软件开发过程不规范。开发过程没有统一的、公认的方法论和规范指导,参加的人员各行其是。加之设计和实现过程的资料很不完整,或忽视了每个人与其他人的接口,使得软件很难维护。

(4) 软件产品无评测手段。未能在测试阶段充分做好检测工作,提交用户的软件质量差,在运行中暴露出大量的问题。

## 1.1.3 解决软件危机的方法

一般解决软件危机可从以下几个方面着手:

(1) 应该加强软件开发过程的管理,做到组织有序、各类人员协同配合,共同保证项目完成,避免软件开发过程中个人单干的现象。

(2) 推广使用开发软件的成功技术与方法,并不断探索更好的技术与方法;消除在计算机系统早期发展阶段形成的错误概念和做法。

(3) 开发和使用好的软件工具,支持软件开发的全过程,即建立软件工程支持环境。

总之,为了解决软件危机,既需要有技术措施(方法和工具),还要有组织管理措施,从而导致了软件工程的产生。

# 1.2 软件工程基础

## 1.2.1 软件工程定义

1968年,在北大西洋公约组织举行的一次学术会议上,首次提出了“软件工程”的概念,并将其定义为“为了经济地获得可靠的和能在实际机器上高效运行的软件,而建立和使用的健全的工程规则”。这个定义肯定了工程化的思想在软件工程中的重要性,但是并没有提到软件产品的特殊性。

经过40多年的发展,软件工程已经成为一门独立的学科,人们对软件工程也逐渐有了更全面、更科学的认识。我国2006年的国家标准GB/T 11457—2006《软件工程术语》中对软件工程定义为“应用计算机科学理论和技术以及工程管理原则和方法,按预算和进度,实现满足用户要求的软件产品的定义、开发、发布和维护的工程或进行研究的学科”。

软件工程的提出是为了解决软件危机所带来的各种弊端。具体地讲,软件工程的目

标主要包括以下几点：

- (1) 使软件开发的成本能够控制在预计的合理范围内；
- (2) 使软件产品的各项功能和性能能够满足用户需求；
- (3) 提高软件产品的质量；
- (4) 提高软件产品的可靠性；
- (5) 使生产出来的软件产品易于移植、维护、升级和使用；
- (6) 使软件产品的开发周期能够控制在预计的合理时间范围内。

实际上,可以把上述各个目标概括为开发正确、可用和经济的软件产品。当然,在实际的软件开发过程中,软件开发团队很难同时兼顾所有的目标。通常,人们会根据实际项目的情况,对各个目标做优先级排序。

### 1.2.2 软件工程基本原理

自1968年提出“软件工程”术语以来,研究软件工程的专家学者们陆续提出了100多条关于软件工程的准则或信条。经过长期的开发实践和理论研究,著名软件工程专家B. W. Boehm提出了以下几项软件工程的基本原理:

(1) 将软件的生命周期划分为多个阶段,对各个阶段实行严格的项目管理。软件开发是一个漫长的过程,人们可以根据工作的特点或目标,把整个软件的开发周期划分为多个阶段,并为每个阶段制定分阶段的计划及验收标准,这样有益于对整个软件开发过程进行管理。在传统的软件工程中,软件开发的生命周期可以划分为可行性研究、需求分析、软件设计、软件实现、软件测试、产品验收和交付等阶段。

(2) 坚持阶段评审制度,以确保软件产品的质量。严格地贯彻与实施阶段评审制度可以帮助软件开发人员及时地发现并改正错误。在软件开发的过程中,错误发现得越晚,修复错误所要付出的代价就会越大。实施阶段评审,只有在本阶段的工作通过评审后,才能进入下一阶段的工作。

(3) 实施严格的产品控制,以适应软件规格的变更。在软件开发的过程中,用户需求很可能不断发生变化,有时,即使用户需求没有改变,软件开发人员受到经验的限制以及与用户交流不充分的影响,也很难做到一次性获得全部的、正确的需求。可见,需求分析工作应该贯穿整个软件开发的生命周期。在软件开发的整个过程中,需求的改变是不可避免的。当需求变更时,为了保证软件各个配置项的一致性,实施严格的版本控制是非常必要的。

(4) 采用现代程序设计技术。这是提高软件开发和维护效率的关键。现代的程序设计技术,比如面向对象,可以使开发出来的软件产品更易维护和修改,同时还能缩短开发的时间,并且更符合人们的思维逻辑。

(5) 开发出来的软件产品应该能够清楚地被审查。虽然软件产品的可见性比较差,但是它的功能和性能应该能够被准确地审查和度量,这样才能有利于有效的项目管理。一般软件产品包括可以执行的源代码、一系列相应的文档和资源数据等。

(6) 合理地安排软件开发小组的人员,并且开发小组的人员要少而精。开发小组人员的数量少有利于组内成员充分的交流,这是高效团队管理的重要因素。而高素质的开发小组成员是影响软件产品的质量和开发效率的重要因素。

(7) 不断地改进软件工程实践。随着计算机科学技术的发展,软件从业人员应该不断地总结经验并且主动学习新的软件技术,只有这样才能不落后于时代。

### 1.2.3 软件工程框架

相对于其他学科而言,软件工程是一门比较年轻的学科,它的思想体系和理论基础还有待进一步修整和完善。软件工程学科包含的内容有软件开发技术和软件工程管理。其中,软件开发技术包含软件工程方法学、软件工具和软件工程环境;软件工程管理学包含软件工程经济学和软件管理学。

#### 1. 软件工程方法学

通常,把在软件生命周期全过程中使用的一整套技术方法的集合称为方法学,也称为范型。目前使用最广泛的软件工程方法学是传统方法学和面向对象方法学。

##### 1) 传统方法学

传统方法学也称为生命周期方法学或结构化范型,它采用结构化技术(结构化分析、结构化设计和结构化实现)来完成软件开发的各项任务,并使用适当的软件工具或软件工程环境来支持结构化技术的运用。

其特点:①把软件生命周期的全过程划分为若干阶段,然后顺序地完成每个阶段的任务;②前一阶段的完成是后一阶段开始的前提和基础,后一阶段任务是前一阶段提出的解法的进一步的具化;③每一阶段结束之前都必须进行正式严格的技术审查和管理评审;④每一阶段都必须交出高质量的文档资料。

其优点:①把整个软件生命周期划分成若干阶段,每个阶段任务相对独立,有利于不同人员的分工协作;②每一阶段结束之前都必须进行正式严格的技术审查和管理评审,使软件开发工程的全过程以一种有条不紊的方式进行,保证了软件的质量,特别是提高了软件的可维护性。

其缺点:这种技术要么面向行为(即对数据的操作),要么面向数据,还没有既面向数据又面向行为的结构化技术。数据和对数据的处理原本是密切相关的,把数据和操作人为地分离成两个独立的部分,自然会增加软件开发与维护的难度。与传统方法相反,面向对象方法把数据和行为看成同等重要,它是一种把数据和数据的处理紧密结合起来的方法。

##### 2) 面向对象方法学

面向对象方法学包含以下四个要点:①对象包含数据和对数据的处理,用对象分解代替了功能分解;②把所有对象划分成类;③按照父类和子类的关系,把若干个相关类组成一个层次结构的系统(类等级),子类自动拥有父类的数据和操作(类继承);④对象彼此间通过发送消息互相联系。

其优点:①对象所有私有信息都被封装在该对象内,不能从外界直接访问,具有封装性;②模拟人类习惯思维方式,使描述问题的问题空间与实现解法的解空间在结构上尽可能保持一致;③开发出的软件产品由许多小的、相对独立的对象组成,降低了软件产品的复杂度,简化了软件开发和维护工作;④面向对象的继承性和多态性,进一步提高了面向对象软件的可重用性。

## 2. 软件工具

软件工具是指为了支持计算机软件的开发和维护而研制的程序系统。使用软件工具的目的是提高软件设计质量和生产效率,降低软件开发和维护的成本。软件工具可用于软件开发的整个过程。例如,需求分析工具用类生成需求说明;设计阶段需要使用编辑程序、编译程序、连接程序,有的软件能自动生成程序等;在测试阶段可使用排错程序、跟踪程序、静态分析工具和监视工具等;软件维护阶段有版本管理、文档分析工具等;软件管理方面也有许多软件工具。软件开发人员在软件生产的各个阶段可根据不同的需要选用合适的工具。目前,软件工具发展迅速,其目标是实现软件生产各阶段的自动化。

## 3. 软件工程环境

软件工程方法和软件工具是软件开发的两大支柱,它们之间密切相关。软件工程方法提出了明确的工作步骤和标准的文档格式,这是设计软件工具的基础,而软件工具的实现又将促进软件工程方法的推广和发展。

软件工程环境(Software Engineering Environment,SEE)是方法和工具的结合。软件工程环境的设计目标是提高软件生产率和改善软件质量。

计算机辅助软件工程(Computer Aided Software Engineering,CASE)是一组工具和方法的集合,可以辅助软件工程生命周期各阶段进行软件开发活动。CASE是多年来在软件工程管理、软件工程方法、软件工程环境和软件工具等方面研究和发展的产物。CASE吸收了计算机辅助设计、软件工程、操作系统、数据库、网络和许多其他计算机领域的原理和技术。因此,CASE领域是一个应用、集成和综合的领域。其中,软件工具不是对任何软件工程方法的取代,而是对方法的辅助,它旨在提高软件工程的效率和软件产品的质量。

## 4. 软件工程管理

软件工程管理就是对软件工程各阶段的活动进行管理。软件工程管理的目的是为了按预定的时间和费用,成功地生产出软件产品。软件工程管理的任务是有效地组织人员,按照适当的技术、方法,利用好的工具来完成预定的软件项目。

软件工程管理的内容包括软件费用管理、人员组织、工程计划管理、软件配置管理等方面内容。

### 1.2.4 软件工程方法学

方法是工程学的核心,只有采用正确的方法,才能保证工程的有效实施。软件工程方法包括软件开发研发方法、软件管理方法、软件度量方法、软件环境方法等。从方法的层次上可以把软件开发方法分为基础方法和一般方法。基础方法包括分析方法、综合方法、关联方法、抽象方法、模型方法、分类方法、系统方法、集成方法等。一般方法有面向功能方法、面向数据方法和面向对象方法等。

#### 1.2.4.1 概述

##### 1. 软件工程方法的含义

从广义上讲,软件工程方法包括软件开发方法、软件管理方法、软件度量方法、软件环境方法等,其中软件管理方法、软件度量方法和软件环境方法属于软件工程管理、软件工程进度量和软件工程环境等分支学科中的内容,一般所讨论的软件工程方法也就是指软件

开发方法。软件工程方法与软件生存周期、软件过程、软件建模语言等存在着密切的联系,并在很大程度上决定着软件开发的成败和优劣,所以说,软件工程方法是软件工程学科的重要内容。

## 2. 软件工程方法的分类

可以从不同方面对软件工程方法进行分类。

从方法的层次上,可以分为基础方法和一般方法。基础方法指在软件工程过程中人们分析问题和解决问题的通用方法,主要包括分析方法、综合方法、关联方法、抽象方法、模型方法、分类方法、系统方法、集成方法等。一般方法是建立在基础方法之上,根据软件和软件工程的特点和性质,适应软件开发工作的独特性所形成的方法体系。我们日常所谈论的软件开发方法是指软件开发的一般方法,一般方法有面向功能方法、面向数据方法和面向对象方法。

从方法所适应的范围可以分为局部方法和全局方法。局部方法仅适应于软件开发周期某一个或某几个阶段,如程序设计方法、系统分析方法等。全局方法是覆盖软件开发全过程的方法,面向功能方法、面向数据方法和面向对象方法都是全局方法。

从开发风范上可以分为自顶向下的方法和自底向上的方法。自顶向下方法遵循由粗到细、从总体到局部、由框架到细节的开发风范。自底向上方法则是反过来,按照由局部到总体、由细节到框架的开发风范的一类方法。大部分软件工程方法并不单独采用某一种方法风范,而是两种开发风范相结合,但以某一种风范为主。

根据方法的严密性,可以把软件工程方法分为形式化方法和非形式化方法。形式化方法采用严格的数学方法,对软件需求、过程和要素采用形式化描述,对描述的结果可以进行论证,对被证明是正确的描述,可以通过软件生成系统转化为所需的软件结果。非形式化方法在软件工程过程中采用非形式化的自然语言或限定性自然语言进行描述,强调软件开发的工程性,缺乏严密的形式化描述。

从方法侧重的不同,可以分为面向功能方法、面向数据方法和面向对象方法。面向功能方法是从把系统的功能作为分析、设计和实现主线的一类软件工程方法,结构化方法就属于此类方法。面向数据方法是从系统所处理的数据和信息以及数据结构入手,进行软件开发的一类方法,JSD是典型的面向数据的方法。面向对象方法则是直接面向问题领域中的客观事物,以客观事物为对象来进行系统分析、设计和实现。

根据系统开发效率和人工介入的程度,可以分为自动化方法、半自动化方法和手工方法。自动化方法是人工不介入或介入很少,可以由软件生成工具自动生成所需要的软件。半自动化方法则是在人工的控制和参与下,借用辅助开发工具开发软件。手工方法则是没有工具和环境的支持,完全通过人工进行分析、设计、实现工作。目前的软件开发大部分采用半自动化方法,即需要一定软件开发工具和环境的支持,但主要靠开发人员的参与和控制。

根据方法的过渡方式,可以分为过程型方法和演化型方法。过程型方法严格划分工作阶段和工作步骤,并且按照阶段和步骤的顺序从事软件开发工作。演化型方法则是螺旋式的开发方式,先根据需求构造一个初步的系统原型,在原型的基础上逐步完善,软件开发没有明显的过程特征。

### 3. 软件工程的基础方法

软件工程的基础方法是建立在人类思维方法的基础之上,与软件开发相关的,人们认识问题和解决问题的通用方法。任何一种思维方法都直接或间接地决定和影响软件的开发方法。根据其相关程度,软件工程的基础方法主要有分析方法、综合方法、关联方法、抽象方法、模型方法、分类方法、系统方法、集成方法等。下面简要介绍这些方法。

#### 1) 分析方法

分析是人们在认识客观事物过程中使用的一种基本思维方法,意指把事物分解成为多个部分、方面、属性、要素、阶段等,而分别加以思考和认识。根据一个事物的生存演化过程,可以从时间上把一个事物划分为多个生存阶段,对每一个阶段分别进行认识和把握。例如,把人的生命过程划分为幼年、童年、少年、青年、中年和老年几个阶段。也可以按照事物的组成进行分解,例如,医学上把人分为神经系统、消化系统、内分泌系统、生殖系统等部分。

分析方法是软件开发的一个基本方法,体现在软件开发的各个方面。把软件按时间过程划分成为策划、分析、设计、实现、演化等阶段,把一个大型的软件划分为计划、统计、财务、生产、人事、科研、质量、供应、销售等子系统,把一个子系统又分为多个构件,都是分析方法的运用。

所有软件开发都运用到了分析方法,但分析的对象和依据各有不同。面向功能方法是按照功能进行分解,面向对象方法则是按照事物实体进行分解,面向数据的方法则是按照实体静态属性所呈现的结构形态进行分解。

#### 2) 综合方法

综合是把事物的各部分、属性、要素、阶段等联系起来进行整体认识的思维方法。例如,由控制器、运算器、存储器、输入设备和输出设备五部分构成计算机整体,本能、理智和理性等部分构成人的心理部分,而机体和心理两部分则共同构成人。

分析与综合是同一思维活动的两个方面,是矛盾的辩证统一关系。分析是基于对事物整体认识,而分析的结果又会加深对事物的整体认识。综合以分析为基础,分析越精细准确,综合就越全面完美。分析和综合并不是一次完成的,而是一个螺旋过程,经过综合—分析—再综合,逐步接近于事物本质。

综合方法是指导软件开发的一个基本方法。综合策划、分析、设计、实现、演化几个阶段构成完整的软件生存周期,软件的开发综合软件的目标、需求、方法、技术、环境等多种要素,在面向对象方法中,由对象构成类,由类以及类相互之间的联系构成系统都运用到综合方法。

#### 3) 关联方法

事物之间存在着广泛的联系,事物之间的联系影响着事物的行为和特征。所以要全面地分析和认识事物,必须关心事物之间的联系。关联方法就是分析事物时,主动从事物之间的联系上分析和认识事物的思维方法。在软件开发过程中必须考虑软件与其外部环境以及软件内部各要素之间的联系。与软件关系最为密切的是软件所服务的业务领域和其支撑环境。软件开发者必须搞清楚所服务业务领域的业务和管理。另外,软件也需要运行环境。在软件内部,各子系统之间和子系统内部各构件之间也都存在密切的联系。在面向对象方法中,各对象之间的一般—特殊关系、整体—部分关系都是对象之间的关

系,软件开发者必须认识和厘清这些关系。

#### 4) 抽象方法

抽象是抽取出同类事物本质特征而舍弃非本质特征的思维方法。抽象的目的是认识事物,进而反映事物。抽象对事物的反映是一种本质、概要和简化的反映,而不是对事物原貌的复制和照搬。要认识事物本质就需要对事物进行分析和综合,所以,抽象要运用分析、综合等其他思维方法。

抽象是软件开发过程中广泛使用的方法,存在于软件的宏观结构、微观分析和软件开发的各个活动之中。在系统分析过程中,通过抽象来认识问题领域的本质,由需求分析所得出的系统逻辑模型就是抽象的结果。在系统设计阶段,确定软件结构、构件、构件内部算法和流程,都要运用到抽象方法。

#### 5) 模型方法

模型是对事物抽象、概括、本质的反映。在软件开发中,大量使用到模型方法。系统分析中的逻辑模型,系统的设计模型,数据库设计中的概念模型、逻辑模型和物理模型,业务过程的决策模型、库存模型、产供销模型、财务模型、人才需求模型等应用模型,都是模型方法在软件开发中的应用。

#### 6) 分类方法

分类是指按照事物的特征,对事物分门别类地进行组织,使事物有序和系统化。如果没有分类,人类社会和我们的日常生活将处于无序和混乱状态,因此,分类在社会组织、日常生活和科学研究中起着十分重要的作用。分类是人们使用的一种基本思维方法。人按性别可划分为男女,按年龄划分为幼年、童年、少年、青年、中年和老年,按学历划分为小学、中学、大学和研究生,按职称划分为初级、中级和高级,按道德标准划分为好人和坏人等,这都是分类方法的具体应用。

分类方法是软件工程方法的基础,并广泛应用于开发的各个方面。在面向对象方法中,类的概念就是分类方法的运用,把具有相同属性和服务的一组对象称为一个类。

#### 7) 系统方法

系统是由相互联系、相互影响的多个部分结合为具有确定目标和功能,并处于特定环境的有机整体。系统具有整体性、关联性、综合性、结构性和适应性等特征。系统既是描述相互联系的客观统一体,又是一种十分有用的思维方法。系统方法就是从全局出发,对系统内外各种联系及其规律性加以辩证的分析,找出合乎目的的最佳方案,并依此指导人们行动的方法。系统方法一般包括五个步骤:①确定系统目标;②根据目标提出多个可行方案;③设计方案的模型来模拟实际情况;④对模型进行比较,选出最佳方案;⑤确定系统的组成、结构和相互联系。

软件本身是系统,所以在软件开发过程中广泛地运用到系统方法。软件开发阶段的划分就是系统方法的体现,每个软件的开发需要从时序上划分成为多个阶段,每一个阶段有确定的目标、任务和时间要求,各阶段的目标、任务和时限的确定必须服从于系统总目标的需要,这就体现了系统方法的整体性;各阶段的工作既相对独立,又互有联系,共同协调构成软件开发总过程,这又体现了系统方法综合性。另外,阶段的确定,各阶段目标、任务和时限的确定都必须遵循辩证统一的原则。

## 8) 集成方法

集成方法是指把多个相对独立的构件或部分,根据目标的需要构成和谐、兼容和相互联系的整体所遵循的方法。在把多个部分综合成为整体方面,集成与综合有相似之处,但集成方法有其独特性。在集成过程中,各部分的独立性更强,联系更为稀松。例如,大庆、辽河、新疆、大港、胜利等油田的信息系统本来已经是独立的信息系统了,但可以集成为整个中国石油天然气总公司的综合信息系统。集成的各个部分的结构具有多样性。可以把多个结构差异很大的计算机、网络和支撑软件集成为一个一体化的系统平台。集成对时限的要求也比较宽松。例如,一个大型软件的建设,可以规划多年完成,已经完成的部分可以自成系统地投入使用。在随后的时间里,可以把后开发的子系统逐步集成到总系统之中。

除了以上基础方法之外,比较方法、分隔方法、观察方法、数学方法、实验方法也都是软件开发所遵从的基础方法。

### 4. 软件工程的一般方法

软件工程的一般方法是建立在基础方法之上,适应和满足软件开发的特殊要求,所形成的相对一套完整的方法体系。软件工程的一般方法与软件的生存规律、软件过程、软件开发技术、软件的工具和环境都有密切的联系,它在一定程度上由这些因素所决定,并反过来影响这些因素。除此之外,软件工程方法的形成与人们审视软件和软件开发过程的角度也有很重要的关系,这也是造成软件工程方法多样化的一个重要原因。

20世纪60年代中期以来,人们对软件工程方法进行了不懈的研究,先后出了几十种软件工程方法,其中产生重大影响的有面向功能方法、面向数据方法和面向对象方法等。

#### 1.2.4.2 面向功能方法

面向功能是软件工程的一类方法,该类方法把软件开发的侧重点放在软件功能上面,面向功能方法的典型代表是结构化方法。本节以结构化方法为主,介绍面向功能方法的思想和基本方法。

##### 1. 结构化方法的特征

结构化方法(structured method)是20世纪60年代末,在结构化程序设计的基础上发展起来的,遵循系统思想,严格划分工作阶段,充分考虑用户需求,遵循软件生命周期,突出软件各部分之间的结构关系的一类开发软件的全局方法。结构化方法由结构化分析(SA)、结构化设计(SD)以及结构化编程(SP)三部分构成,结构化方法的主要特征表现在以下几个方面。

##### 1) 遵循系统思想

系统思想是结构化方法的灵魂。结构化方法首先按照系统的思想和观点来认识和观察软件。软件以及软件所服务的业务领域本身是复杂的系统,在认识过程中需要运用系统的思想和方法。在认识系统的各个部分和要素及其相互关系时,总是从系统的总体目标出发,把各要素放到系统整体中,认识各部分在总系统中的作用和位置。

软件开发过程需要利用系统的思想和方法来指导开发工作。整个软件开发是一个完整的工程项目,需要按照系统方法来组织和管理项目开发工作。开发工作的组织安排、人员分工、资源调配、技术运用、工程进展都要从系统全局需要出发,符合系统开发的整体要求。在系统开发各阶段的工作开展也离不开系统方法的指导。在系统调研、分析和设计

过程中,从总体到局部、从宏观到微观、由粗到细的自顶向下方法就是系统化思想和方法的体现。系统分析阶段采用的分层数据流图和设计阶段采用的模块结构图都闪烁着系统化思想。

### 2) 严格划分工作阶段

结构化方法把软件开发划分为问题定义、需求分析、设计、实现和测试等阶段。每一个阶段都有明确的目标、任务和预期达到的结果。前一阶段的工作未完成之前,不能跨越到下一阶段,这样才能使整个开发工作有条不紊地进行。结构化方法按时间顺序划分工作阶段也是系统化思想的体现。结构化方法遵循软件生存周期的瀑布模型。

### 3) 充分考虑用户需求

用户既是软件需求的提出者,又是软件的使用者,同时也是软件开发工作的参与者。因此,软件开发必须充分考虑和满足用户的需求。在软件开发中,开发人员必须与用户广泛地进行交流,深入了解用户的想法,并对用户所提出的需求全面深入地进行分析。

### 4) 按照工程化的方法

结构化方法形成了相对完整的工程化方法体系。在分析阶段用自顶向下的分层数据流图和数据字典来进行需求分析,并导出系统的逻辑模型;在设计阶段用系统结构图来描述系统的结构;而在详细设计阶段用判定树、判定表等工具描述模块算法和流程。

### 5) 突出系统各部分的联系

结构化方法强调和凸显系统部分之间的联系。分析阶段采用的分层数据流图就是从数据变换的角度,反映出系统的需求结构框架。在软件中数据是对客观实体的反映,数据的变化过程反映系统的功能处理需求,所以结构化方法把数据流图和数据字典作为新系统的逻辑模型。系统结构图是设计阶段用来描述系统软件结构的工具,它反映构成系统的各个模块之间的调用和数据传输关系。到了实现阶段,采用结构化程序设计,整个软件由多个模块构成,模块之间和模块内部呈现出顺序、选择和循环三种结构形态。突出系统各部分之间的结构框架就成为结构化方法的显著特征。

## 2. 结构化分析

### 1) 概述

结构化分析是结构化方法中的第一个环节,也是系统开发过程中最重要的环节。结构化分析工作的主要任务是运用结构化分析方法和工具,调研现行系统的业务管理过程和新系统的需求,在详细调研的基础上,通过综合系统目标、用户要求、考虑系统的背景和环境,以及资金能力和技术因素,通过客观、认真、全面的分析,确定出合理可行的系统需求,并提出新系统的逻辑方案(也称系统逻辑模型),编写出系统规格说明。系统规格说明经过审查通过之后,可以结束分析工作,把系统规格说明提供给设计阶段,作为结构化设计工作的依据。

结构化分析遵从瀑布模型。在分析工作没有完全完成,系统规格说明未通过审查之前,不能结束分析工作。结构化分析运用的工具是数据流图和数据字典。用数据流图描述数据的传输、加工和处理过程。数据流图既作为现行系统数据加工处理的描述工具,同时又作为新系统逻辑模型的描述工具。

### 2) 结构化分析方法

结构化分析方法基于数据处理的观点,从系统功能的角度,运用分层数据流图对数据

处理流程采用自顶向下、逐步细化求精的方法,分析和描述系统需求。

数据处理观点认为软件是数据和信息的加工处理系统,软件处理的直接对象是数据和信息,软件的作用就是把数据源通过多环节的加工处理,变换成为用户所需的信息。按照数据处理观点,软件的功能是通过软件对数据的加工处理过程表现出来的。基于这种认识,结构化分析方法的基本思想是,根据系统的功能需求,运用数据流图和数据字典等工具,通过对数据流程变换过程的分析,构造和描述出新系统的逻辑模型。

(1) 数据流图。数据流图是结构化分析的工具。数据流图采用图 1.1 中的四个基本符号。



图 1.1 数据流图的基本符号

数据流是在数据加工、数据存储或数据源之间传输的数据。数据流不同于控制流,只有实际存在数据传输时,才能有数据流。在数据流上要标出数据流的名称,数据流名称要能够明确反映数据的确切含义。数据流可以表示一组复杂抽象的数据,如客户数据,也可以表示一个简单的具体数据,如学生的数学成绩。

数据加工表示对数据的加工变换处理,简称为加工。数据加工的作用是把进入加工的数据流变换成为加工流出的数据流。数据加工需要有确定的名称,在数据加工符号的上方还要给出加工编号。

数据源和数据汇表示系统之外数据的来源和终点,简称为数据源。数据源也要有名称。

数据存储是对数据存放的抽象表示,可以是一个文件、数据库或记录簿,也要给出数据存储的名称。

图 1.2 是学生成绩管理数据流图的例子。该数据流图描述了学校对学生成绩管理的数据处理过程。教师阅完卷之后,向教务处提交学生的成绩单,教务处把成绩单中的成绩登记到学籍表中。教务处需要对学生成绩进行汇总,把汇总结果记入成绩汇总表。根据成绩单和成绩汇总表对学生成绩进行分析,并把分析结果填入成绩分析表中。还要从成绩单中摘出不及格的学生,记入学生补考名单,以便通知补考的学生。在这个数据流图中共有成绩单、成绩汇总表、成绩分析表和学生补考名单四个数据流。数据源是教师,数据汇是教务处。学籍表是数据存储。图中有四个数据加工,分别是登记学生成绩、汇总学生成绩、分析学生成绩和抽取补考成绩。

绘制数据流图需要遵循一些基本原则。第一是客观原则。数据流图必须反映数据流程变换的客观实际,主观臆断、违背客观实际的数据流图只能把系统开发引入失败,所以要求开发人员在绘制数据流图之前,要进行深入细致的调查研究工作。第二是数据流原则。在数据流图中的每一股数据流必须是实际存在的数据流。控制流不能作为数据流,