



普通高等教育“十一五”国家级规划教材

● 高等院校计算机专业及专业基础课系列教材

软件工程

(第三版)

王立福

孙艳春

刘学洋

编著

北京大學出版社





普通高等教育“十一五”国家级规划教材

软件工程

(第三版)

王立福 孙艳春 刘学洋 编著



北京大学出版社
PEKING UNIVERSITY PRESS

内 容 提 要

本书是在北京大学计算机科学技术系使用的《软件工程》(第二版)教材的基础上,结合 IEEE 最新发布的软件工程知识体系 SWEBOK(Software Engineering Body of Knowledge)和 IEEE/ACM 软件工程学科小组公布的软件工程教育知识体系 SEEK(Software Engineering Education Knowledge),由主讲、主考教师编写而成的。本书既是北京大学计算机科学系本科生指定教材,也可作为其他高校的本科生教材及软件从业人员的参考书。

本书注重基础知识的系统性,并注重选材的先进性及知识的应用,有助于提高读者求解软件的能力,特别是提高读者直接参与软件开发实践和工程管理能力。

图书在版编目(CIP)数据

软件工程/王立福等编著. —3 版. —北京:北京大学出版社,2009.10
(高等院校计算机专业及专业基础课系列教材)
ISBN 978-7-301-15913-2

I. 软… II. 王… III. 软件工程—高等学校—教材 IV. TP311.5

中国版本图书馆 CIP 数据核字 (2009) 第 172987 号

书 名: 软件工程

著作责任者: 王立福 孙艳春 刘学洋 编著

责任编辑: 沈承凤

封面设计: 张 虹

标准书号: ISBN 978-7-301-15913-2/TP·1059

出版发行: 北京大学出版社

地 址: 北京市海淀区成府路 205 号 100871

网 址: <http://www.pup.cn> 电子信箱: zpup@pup.pku.edu.cn

电 话: 邮购部 62752015 发行部 62750672 编辑部 62752038 出版部 62754962

印 刷 者: 北京大学印刷厂

经 销 者: 新华书店

787 毫米×1092 毫米 16 开本 22.5 印张 562 千字

1997 年第 1 版 2002 年第 2 版

2009 年 10 月第 3 版 2009 年 10 月第 1 次印刷

定 价: 38.00 元

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有,侵权必究

举报电话: (010)62752024 电子信箱: fd@pup.pku.edu.cn

前 言

编写一本适合本科生学习的软件工程教材,实在是一件很困难的事情。其原因主要有三:一是软件工程这门课程所涉及的内容十分宽泛,既涉及技术层面,又涉及管理层面;既关联实际问题的理解和描述,又关联软件工具的使用;二是在社会需求的拉动下,软件工程技术发展非常迅速,新概念、新技术、新方法不断出现;三是作为一门技术学科,其内容具有很强的技术特征,而且仅仅走过了40余年的发展历程,与其他学科相比,例如数学、物理、化学以及建筑等,还是相当“年轻”的一门学科。因此,在教材内容的选取与组织方面,在有关概念的表述方面,实在是一种挑战。

通过参与杨芙清院士主持的国家科技攻关项目,通过参与张效祥院士主编的《计算机科学技术百科全书》,通过参与国家有关标准规范的制定,特别是通过几年来的教学实践,对软件工程有关的知识还有些领悟,有所积累。

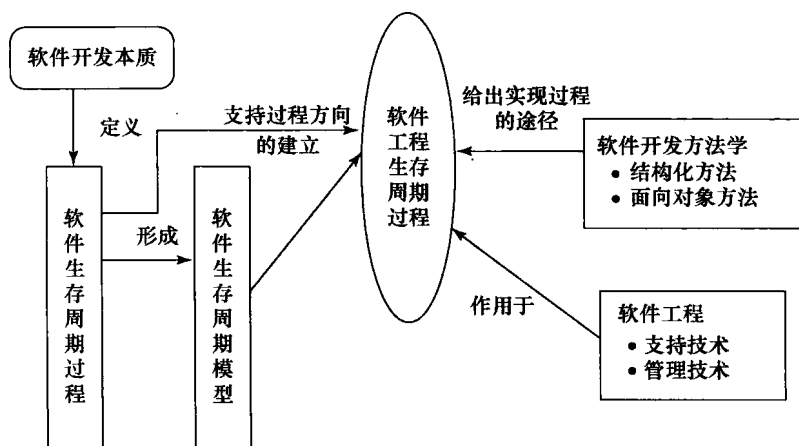
在教材内容的选取方面,基本遵循以下两条原则:

一是选取的内容能够有助于提高读者求解软件的能力,特别是提高读者直接参与软件开发实践和工程管理能力;

二是选取的内容基本上为基础性的,是比较“稳定”的,尽量介绍有关软件工程的国际标准,尽量讲解成熟技术。

在教材内容的组织方面,依据内容选取的基本原则,基于对软件开发本质的认识,紧紧围绕软件开发,主要讲解了软件工程的两大技术问题,一是开发逻辑,二是开发途径。其中,开发逻辑涉及软件生存周期过程、工程中常用的软件生存周期模型——有关过程、活动和任务的组织框架,以及项目软件生存周期的规划与监控;开发途径涉及结构化方法和面向对象方法,以及支持软件评估所需要的软件测试技术等。并且简单介绍作用于开发活动上的一些管理活动,其中重点介绍支持管理活动的一些基础性技术,例如规模、成本、进度估算等。

本书的内容组织可图示如下:



在概念的表述方面,依据内容组织的特定层面,尽量引用《计算机科学技术百科全书》中有关的条目和相关国际标准中的定义,并注重其中概念语境和语义的讲解。在不失严谨的条件下,尽量采用“图-文”形式,使所表达的概念容易理解。

在第一章绪论中,通过软件工程概念的提出以及发展历程,试图回答软件开发的本质。

在第二章软件过程中,通过讲解 ISO/IEC 12207 标准,试图回答开发需要进行哪些映射;通过讲解一些典型的软件生存周期模型,试图回答组织一个软件工程中活动的框架;通过应用软件生存周期过程和软件生存周期模型,试图回答如何形成一个软件工程项目的生存周期过程,即如何正确给出求解软件的逻辑。

在第三章需求与需求规约中,讲解了软件开发中的最重要的概念——需求和需求规约,基本性质以及作用,试图回答问题定义在软件开发中的重要性。

在第四章结构化分析和第五章结构化设计中,从“如何做”这一视角,详细讲解了结构化分析和结构化设计,以期回答如何开展软件工程中的需求和设计活动。

在第六章和第七章中,从“方法学”这一视角,详细讲解了 UML(统一模型化语言)和 RUP(统一软件开发过程),还是以期回答如何开展软件工程中的需求、设计等活动。

在第八章软件测试中,通过讲解软件测试技术,包括基于程序结构的路径测试技术和几个基于软件规约的功能测试技术,试图回答如何支持对软件系统/产品质量的评估。

在第九章中,基于一个典型的管理模型,通过讲解软件工程项目管理的基本活动,以及其中具有基础性的管理技术(例如规模、成本和进度估算方法等)和过程评估准则——CMM,试图回答在开展软件工程中的诸多活动中,如何对之进行有效管理。

在第十章计算机辅助软件工程中,通过讲解软件工具、开发环境以及相关模型,试图回答实现软件开发自动化的基本途径。

在第十一章内容总结一章中,对本书中主要内容进行了系统的归纳和总结,以帮助读者把书读薄。

由于时间仓促,更主要的是由于水平问题,本书中依然还会存在很多不足和错误,真诚地希望读者提出,并通过电子邮件(wlf@sei.pku.edu.cn)和其他方式,进行有意义的讨论。

目 录

第一章 绪论	(1)
1.1 软件工程概念的提出与发展	(1)
1.2 软件开发的本质	(2)
1.3 本章小结	(5)
习题一.....	(5)
第二章 软件过程	(6)
2.1 软件生存周期过程	(6)
2.1.1 基本过程	(6)
2.1.2 支持过程	(19)
2.1.3 组织过程	(27)
2.1.4 软件生存周期过程以及角色和关系	(32)
2.2 软件生存周期模型	(35)
2.2.1 引言	(35)
2.2.2 瀑布模型	(36)
2.2.3 增量模型	(38)
2.2.4 演化模型	(39)
2.2.5 螺旋模型	(40)
2.2.6 喷泉模型	(42)
2.3 软件项目生存周期过程的规划与控制	(42)
2.3.1 概念:软件项目生存周期过程	(42)
2.3.2 软件项目生存周期过程的规划	(43)
2.3.3 软件项目生存周期过程的监控	(54)
2.4 本章小结.....	(56)
习题二	(57)
第三章 软件需求与软件需求规约	(58)
3.1 需求与需求获取	(58)
3.1.1 需求定义	(58)
3.1.2 需求分类	(59)
3.1.3 需求发现技术	(61)
3.2 需求规约(SRS)及其格式	(63)
3.2.1 定义	(63)
3.2.2 基本性质	(63)
3.2.3 需求规约(草案)格式	(63)

3.2.4	表达需求规约(规格说明书)的三种风格	(64)
3.2.5	需求规约的作用	(65)
3.3	本章小结	(66)
	习题三	(66)
第四章	结构化分析方法	(67)
4.1	基本术语	(68)
4.2	模型表示	(70)
4.3	建模过程	(70)
4.4	实例研究	(76)
4.5	应用中注意的问题	(80)
4.6	需求分析的输出	(81)
4.7	需求验证	(83)
4.8	本章小结	(84)
	习题四	(84)
第五章	结构化设计	(86)
5.1	总体设计的目标及其表示	(86)
5.2	总体设计	(89)
5.2.1	数据流图的类型	(89)
5.2.2	变换设计与事务设计	(91)
5.2.3	模块化及启发式规则	(98)
5.2.4	实例研究	(105)
5.3	详细设计	(108)
5.3.1	结构化程序设计	(108)
5.3.2	详细设计工具	(110)
5.4	结构化方法小结	(116)
	习题五	(119)
第六章	面向对象方法——UML	(121)
6.1	表达客观事物的术语	(123)
6.1.1	类与对象	(123)
6.1.2	接口	(133)
6.1.3	协作	(135)
6.1.4	用况	(135)
6.1.5	主动类	(135)
6.1.6	构件	(136)
6.1.7	制品	(136)
6.1.8	节点	(136)
6.2	表达关系的术语	(137)
6.2.1	关联	(137)
6.2.2	泛化	(142)

6.2.3 细化	(144)
6.2.4 依赖	(145)
6.3 组织信息的一种通用机制——包	(148)
6.4 模型表达工具	(149)
6.4.1 类图	(150)
6.4.2 用况图	(151)
6.4.3 状态图	(156)
6.4.4 顺序图	(165)
6.5 UML 小结	(167)
习题六	(169)
第七章 面向对象方法——RUP	(170)
7.1 RUP 的作用和特点	(170)
7.2 核心 workflow	(173)
7.2.1 需求获取	(174)
7.2.2 需求分析	(184)
7.2.3 设计	(199)
7.2.4 RUP 的实现和测试	(220)
7.3 RUP 小结	(222)
习题七	(225)
第八章 软件测试	(226)
8.1 软件测试目标与软件测试过程模型	(226)
8.1.1 软件测试目标	(226)
8.1.2 测试过程模型	(227)
8.2 软件测试技术	(228)
8.2.1 路径测试技术	(228)
8.2.2 基于事务流的测试技术	(234)
8.2.3 其他功能测试技术简述	(238)
8.3 静态分析技术——程序正确性证明	(242)
8.4 软件测试步骤	(246)
8.4.1 单元测试	(246)
8.4.2 集成测试	(247)
8.4.3 有效性测试	(249)
8.5 本章小结	(249)
习题八	(250)
第九章 软件工程项目管理概述	(253)
9.1 软件工程管理活动	(253)
9.2 软件规模、成本和进度估算	(258)
9.2.1 软件系统/产品规模估算	(258)
9.2.2 成本和进度估算	(262)

9.3	能力成熟度模型 CMM	(268)
9.3.1	CMM 产生的背景	(268)
9.3.2	CMM 的成熟度等级	(268)
9.3.3	CMM 的结构	(273)
9.3.4	CMM 的使用以及对相关标准的影响	(279)
9.4	ISO 9000 系列标准简介	(282)
9.4.1	八项质量管理原则	(283)
9.4.2	质量管理体系基础	(283)
9.4.3	ISO 9001 和 ISO 9004 标准的关系	(287)
9.5	CMM 与 ISO 9000 系列标准的比较	(287)
9.6	本章小结	(287)
	习题九	(288)
第十章	软件开发工具与环境	(289)
10.1	基本概念	(289)
10.2	工具集成模型	(295)
10.3	大型软件开发环境青鸟系统概述	(301)
10.4	本章小结	(305)
	习题十	(305)
第十一章	内容总结	(306)
11.1	关于软件过程范型	(307)
11.1.1	软件生存周期过程	(308)
11.1.2	软件生存周期模型	(314)
11.1.3	软件项目生存周期过程规划	(317)
11.2	关于软件设计方法	(322)
11.2.1	结构化方法	(322)
11.2.2	面向对象方法	(326)
	参考文献	(352)

第一章 绪 论

正确认识软件开发,是从事软件开发实践和软件工程项目管理的思想基础。

1.1 软件工程概念的提出与发展

软件工程这一术语首次出现在 1968 年的 NATO 会议上。20 世纪 60 年代以来,随着计算机的广泛应用,软件生产率、软件质量远远满足不了社会发展的需求,成为社会、经济发展的制约因素,人们通常把这一现象称为“软件危机”。

当时,软件开发虽然有一些工具支持,例如编译连接器等,但基本上还是依赖开发人员的个人技能,缺乏可遵循的原理、原则、方法体系以及有效的管理,使软件开发往往超出预期的开发时间要求和预算。

一般而言,工程是将科学理论和知识应用于实践的科学。在理解“工程”这一概念的基础上,可以把软件工程定义为:软件工程是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工程,或以此为研究对象的学科。

软件工程概念的提出,其目的是倡导以工程的原理、原则和方法进行软件开发,以期解决出现的“软件危机”。

软件工程作为一门学科至今已有 30 余年的历史,其发展大体可划分为两个时期。

20 世纪 60 年代末到 80 年代初,软件系统的规模、复杂性以及在关键领域的广泛应用,促进了软件的工程化开发和管理。这一时期主要围绕软件项目,开展了有关开发模型、开发方法和支持工具的研究。主要成果体现为:提出了瀑布模型,试图为开发人员提供有关活动组织方面的指导;开发了诸多过程式语言(例如 PASCAL 语言、C 语言、Ada 语言等)和开发方法(例如 Jackson 方法、结构化方法等),试图为开发人员提供好的需求分析和设计手段,并开发了一些支持工具,例如调试工具等。在这一时期,开始出现各种管理方法,例如费用估算、文档复审等;开发了一些相应支持工具,例如计划工具、配置管理工具等。因此这一时期的主要特征可概括为:前期主要研究系统实现技术,后期则开始关注软件质量和软件工程管理。

20 世纪 80 年代以来,基于已开展的大量软件工程实践,围绕对软件工程过程的支持,开展了一系列有关软件生产技术,特别是软件复用技术和软件生产管理的研究和实践。主要成果是提出了《软件生存周期过程》等一系列软件工程标准;大力开展了计算机辅助软件工程(CASE)的研究与实践(例如我国在“七五”、“八五”、“九五”期间,均把这一研究作为国家重点科技攻关项目),各类 CASE 产品相继问世。其间最引人注目的是,在工程技术方面,出现了面向对象语言,例如 Smalltalk、C++、Eiffel 等;提出了面向对象软件开发方法;在工程管理方面,开展了一系列过程改进项目,其目标是在软件产业的实践中,建立一种量化的评估程序,判

定软件组织和过程的成熟度,提高组织的过程能力。

近几年来,围绕网络,特别是 Internet 网的广泛应用,以软件复用技术为基础,在软件构件技术、软件平台技术(包括应用框架)、需求工程技术、领域分析技术以及应用集成技术等研究方面,均取得了非常有影响的成果,有力地促进了软件工程学科和软件产业的发展。

1.2 软件开发的本质

计算机软件一般是指计算机系统程序及其文档。其中,程序是计算机任务的处理对象和处理规则的描述;文档是为了解程序所需的阐述性资料。

由以上定义可知,软件是对一个特定问题域的抽象,是被开发出的一种逻辑实体,而不是一种“有形”的元件。该问题域有自己的术语空间和业务程序,例如一个图书管理系统,有“图书管理员”、“图书”、“书库”以及有关业务的“借书”、“还书”、“新书入库”等术语,有“借书”或“还书”等特定的业务程序,或称业务处理逻辑。软件开发的目的是将问题域中概念映射为运行平台层面的概念,例如变量、常量、表达式以及语句等,把问题域中的处理逻辑映射为运行平台层面的处理逻辑,例如顺序语句、选择语句和循环语句等,如图 1.1 所示。

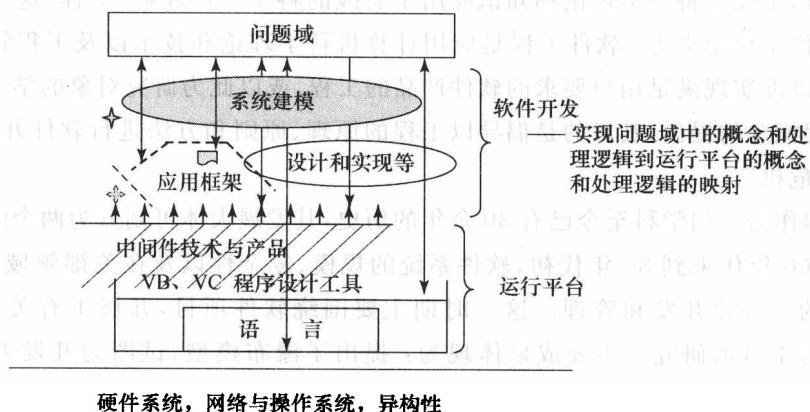


图 1.1 软件开发的含义

因此可以说,软件开发就是要“弥补”问题域与运行平台之间的“距离”,其中这一距离是通过问题域中的概念和处理逻辑不同于运行平台中的概念和处理逻辑而体现的。另外,尽管随着软件技术的进步,问题域与运行平台之间的“距离”会越来越小,但几乎很难实现“彻底”的软件自动化,换言之,问题域与运行平台之间的“距离”将长期存在。

从问题域向运行平台直接进行映射,势必存在一定的复杂性。为了控制这一复杂性,需要确定多个抽象层,例如需求、设计、实现和部署等,每一抽象层均由自己特定的术语定义的,形成该抽象层的一个术语空间。

如果按照自顶向下的途径进行软件开发的话,首先就是通过需求建模,把问题域的概念和处理逻辑向需求这一抽象层进行映射,再把需求层的概念和处理逻辑向设计层进行映射,依次进行,直至映射到运行平台这一抽象层为止,如图 1.2 所示。

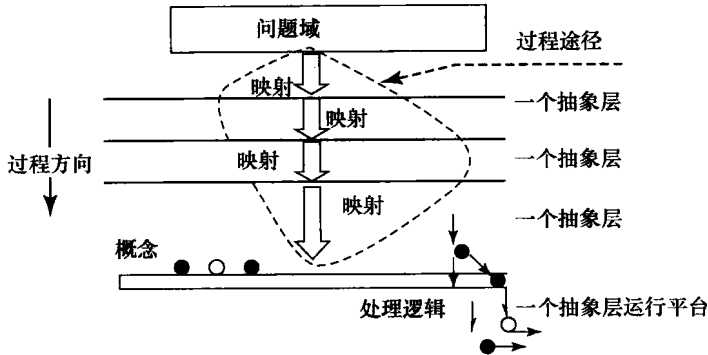


图 1.2 软件开发本质示意

因此可见,软件开发的本质可概括为:不同抽象层术语之间的“映射”,不同抽象层处理逻辑之间的“映射”。

软件开发既然是实现多个不同抽象层之间的映射,而且是由开发人员来做这样的映射,因此自然就要涉及两个方面的问题:一是如何实现这样的映射;二是如何管理这样的映射,以保障映射的有效性和正确性。

关于如何实现这样的映射,这是技术层面上的问题。这一问题又涉及到如下两方面的内容:

一是过程方向,即求解软件的开发逻辑。这是一件十分重要的问题,事关项目的成功与否。诸如瀑布模型、演化模型和螺旋模型等,给出了有关活动的组织框架,为设计开发逻辑提供了基础,如图 1.3 所示。

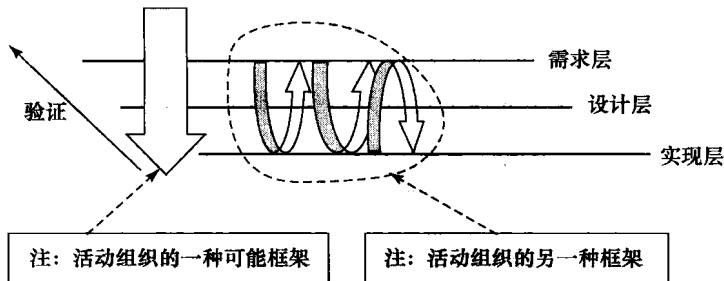


图 1.3 过程方向所涉及的主要内容

二是过程途径,即求解软件的开发手段。一般来说,由于人们的认知能力,问题的结构化谱系如图 1.4 所示。为了求解其中的非结构化和半结构化问题,其基本手段是建模,即运用所掌握的知识,通过抽象,给出该问题的一个结构。在软件开发领域,实际工程中采用的建模手段主要包括结构化方法、面向对象方法以及诸多面向数据结构方法等。

何谓模型?简单地说,模型是任一抽象,其中包括所有的基本能力、特性或一些方面,而没有任何冗余的细节。进一步地说,模型是在特定意图下所确定的角度和抽象层次上对物理系统的描述,通常包含对该系统边界的描述,给出系统内各模型元素以及它们之间的语义关系。例如图 1.5 中的信用卡确认系统的功能模型,其中采用 UML 作为建模工具。

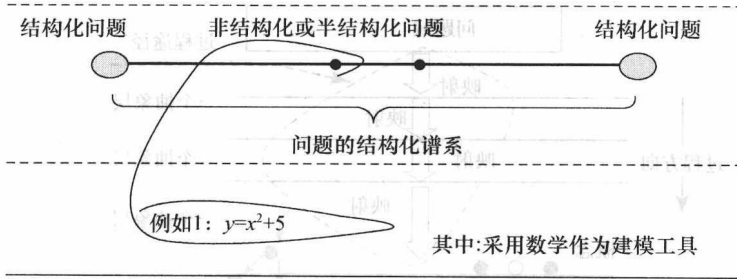


图 1.4 问题求解的基本手段

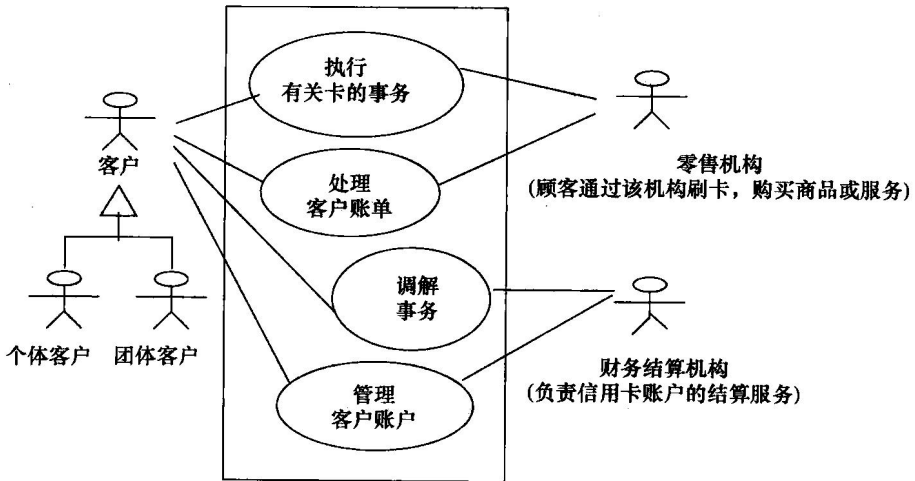


图 1.5 模型示例

在软件开发中,软件系统模型大体上可分为两类:概念模型和软件模型,如图 1.6 所示。

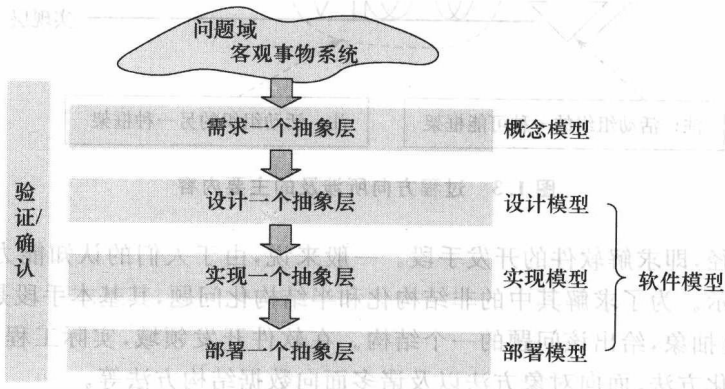


图 1.6 软件系统的一种模型分类

分层的基本动机是控制开发的复杂性。其中,在需求层面上创建的系统概念模型是对客观事务系统的抽象,即标识要解决的问题,或称问题定义。软件模型依据所在的抽象层,可进一步分为设计模型、实现模型和部署模型等,给出了相应概念模型的软件解决方案。

关于如何管理这样的映射,是管理层面上的问题,其主要功能包括软件项目的规划、组织、人员安排、控制和领导等。

从以上的论述中可以了解,软件开发既有技术上的问题,又有管理上的问题。因此,软件工程作为一门研究软件开发的学科,其主要内容包括:

- (1) 做哪些映射,即要完成哪些开发任务(第二章);
- (2) 如何根据软件项目特点、环境因素等,选择并组织这些开发任务(第二章);
- (3) 如何实现不同抽象层之间的映射(第四章,第五章,第六章);
- (4) 如何进行测试(第八章),如何支持整个软件开发(第十章);
- (5) 如何管理一个软件项目,主要包括如何进行项目规划,如何控制开发过程质量,如何控制产品质量等(第九章)。

下面,分别就以上五个方面的知识进行介绍。但由于软件开发管理所涉及的内容比较多,而且更具有“人文”特点,因此在概要介绍软件项目管理概念的基础上,就其中几个典型的管理技术进行了较为详细的讨论。

1.3 本章小结

本章简要阐述了软件开发的本质,即问题空间的概念和处理逻辑到解空间的概念和处理逻辑的映射。在此基础上,介绍了实现这一映射的基本途径,即系统建模。所谓系统建模,是指运用所掌握的知识,通过抽象,给出该系统的一个结构——系统模型。因此,模型是一个抽象,该抽象是在特定意图下所确定的角度和抽象层次上对物理系统的一个描述,描述其中的成分和成分之间所具有的特定语义的关系,还包括对该系统边界的描述。

在软件开发领域,系统模型分为两大类:一类称为概念模型,描述了系统是什么;另一类统称为软件模型,描述了实现概念模型的软件解决方案。软件模型又可进一步分为:设计模型、实现模型和部署模型等。

正确认识软件开发的本质,认识建模的意义,了解模型概念以及模型分类,直接关系到对软件工程开发逻辑、开发途径有关知识的理解、掌握和正确应用。正如本章首语所言:“正确认识软件开发,是从事软件开发实践和软件工程项目管理的思想基础”。

习 题 一

1. 解释以下术语:
 - (1) 软件;
 - (2) 软件工程;
 - (3) 软件危机;
 - (4) 模型。
2. 简单回答软件开发的本质是什么。

第二章 软件过程

开发逻辑,是获取正确软件的关键。

2.1 软件生存周期过程

早在 20 世纪 70 年代中期,人们就提出了“软件生存周期”这一概念,开始注重编程之前的工作。简言之,“软件生存周期是软件产品或系统的一系列相关活动的全周期。从形成概念开始,历经开发、交付使用、在使用中不断修订和演化,直到最后被淘汰,让位于新的软件产品”。国际标准化组织于 1995 年就其中“一系列相关活动”发布了一个国际标准,即《ISO/IEC 软件生存周期过程 12207-1995》。这一标准是软件工程标准中一个基础性文件,系统化地给出了软件开发所需要的任务,即回答了软件开发需要做哪些基本“映射”。

为了有效地组织和表述软件生存周期中的任务,在该标准中使用了三个术语,即过程、活动和任务。过程是软件生存周期中活动的一个集合,活动是任务的一个集合,而任务是将输入变换为输出的操作。可见,这里所说的任务,是一个“原子”映射,而过程和活动均是“复合”映射。

随着该标准的不断应用,以及软件复用技术的发展,并结合 CMM(能力成熟度模型)和 ISO/IEC/TR15504 的推进,国际标准化组织于 2002 年给出了 ISO/IEC 12207-1995 的补篇 1,主要包括:

① 增加了一些新的软件过程,例如测量过程、资产管理过程、复用程序管理过程以及领域软件工程过程等;

② 增加了一些有关增进该标准应用效果的内容,例如给出了每一过程的目标以及成功实现过程的基本判定准则等。

继之又于 2004 年给出了 ISO/IEC 12207-1995 的补篇 2,主要对补篇 1 的内容做了一些修改。

该标准按照承担软件开发工作的主体,将软件生存周期过程分为三类:基本过程、支持过程和组织过程。并在该标准的附录中,给出了剪裁过程以及相关的指导,以便当把软件过程运用到相关组织、具体应用领域或具体项目时,可以根据特定情况,对各种过程和活动进行剪裁,形成特定项目所需要的软件生存周期过程。

2.1.1 基本过程

基本过程是指那些与软件生产直接相关的活动集,又可分为五个过程,即获取过程、供应过程、开发过程、运行过程和维护过程。

例如 1: 获取过程

获取过程是获取者所从事的活动和任务,其目的是获得满足客户所表达的那些要求的产品和/或服务。该过程以定义客户要求开始,以接受客户所要求的产品和/或服务结束。

该过程包括以下基本活动:

- (1) 启动;
- (2) 招标[标书]准备;
- (3) 合同编制和更新;
- (4) 对供方的监督;
- (5) 验收和完成。

供应过程的每一基本活动又包含一组特定的任务。例如“启动”活动包括下述任务:

- ① 描述获取、开发或增强一个系统、软件产品或软件服务的概念或要求,以此开始这一活动。
- ② 定义并分析该系统需求。系统需求一般应包括业务、组织和用户的需求,还应包括与设计、测试有关的安全性、保密性和其他关键性需求以及应遵循的标准和规程。
- ③ 需方可以自己定义并分析软件需求,也可委托供方进行这项任务。
- ④ 如果需方委托供方进行系统需求分析,那么需方就要审核并批准所分析的需求。
- ⑤ 为了执行任务②和④,应使用开发过程(见例如 3)。
- ⑥ 依据对有关风险、费用和效益等方面的适当分析,选择获取方案。方案包括:
 - (a) 是否购买满足需求的现货软件产品;
 - (b) 是否在自己组织内部进行软件产品的开发或获得软件服务;
 - (c) 是否通过合同来开发软件产品或获得软件服务;
 - (d) 是否采用上述(a)、(b)、(c)的一个组合;
 - (e) 是否增强现有的软件产品或服务。
- ⑦ 当要去获得一个现成软件产品时,应确保满足以下条件:
 - (a) 满足该软件产品的需求;
 - (b) 文档是可用的;
 - (c) 满足专利权、使用权、拥有权、担保权和许可权;
 - (d) 规划对该软件产品的未来支持。
- ⑧ 制订一个获取计划并执行之,该计划应包括下述内容:
 - (a) 对该系统的需求;
 - (b) 对如何使用该系统进行规划;
 - (c) 准备使用的合同类型;
 - (d) 有关组织的职责;
 - (e) 准备使用的支持(例如验证、质量保证等);
 - (f) 风险以及管理这些风险的方法。
- ⑨ 定义验收策略和条件(准则),并形成文档。

在补篇 1 中,对获取过程增加了以下活动:(a) 合同终结处理;(b) 获取方针;(c) 供应方

关系管理；(d) 用户关系管理；(e) 财务管理。关于该过程的其他活动，可参见有关标准。

总的来说，成功实现该过程的结果是：

- (1) 定义了获取需求、目标、产品/或服务验收准则以及获取策略；
- (2) 制订了能明确表达顾客和供方的期望、职责和义务的协定；
- (3) 获得了满足顾客要求的产品和/或服务；
- (4) 按规定的约束，例如要满足的成本、进度和质量等，对该获取过程进行了监督；
- (5) 验收了供方的可交付产品；
- (6) 对每一接受的交付项，均有一个由客户和供方达成的满意性结论。

例如 2：供应过程

供应过程是供方为了向客户提供满足需求的软件产品或服务所从事的一系列活动和任务，其目的是向客户提供一个满足需求的产品或服务。

该过程的启动，或为应答需方的招标书并决定开始编制投标书，或与需方签订一项提供系统、软件产品或软件服务的合同。启动之后，继之确定为管理和保证项目所需的规程和资源，包括编制项目计划，执行计划，一直到将系统、软件产品或软件服务交付给需方为止。

该过程包括以下基本活动：

- (1) 启动；
- (2) 准备投标；
- (3) 签订合同；
- (4) 规划；
- (5) 执行和控制；
- (6) 复审和评估；
- (7) 交付和完成。

供应过程的每一活动又包含一组特定的任务。例如“规划”活动包括下述任务：

① 供方应复审获取需求，以便定义管理该项目、保证可交付的软件产品或服务质量的框架。

② 如果合同中没有规定采用什么软件生存周期模型，那么供方就应确定或选择一个适合于该项目的范围、规模和复杂度的软件生存周期模型，并应从本标准中所述的过程、活动和任务中进行选择，并将它们映射到所选择的软件生存周期模型。

③ 供方应为关于项目的规划建立适当的需求，以便管理该项目并保证可交付软件产品或服务的质量。这样的需求应涉及资源的需要以及需方的参与。

④ 一旦建立了有关规划的需求，供方就应该考虑：

- (a) 是否利用内部资源来开发该软件产品或提供软件服务；
- (b) 是否通过分包合同来开发该软件产品或提供软件服务；
- (c) 是否从内部或外部来获得现货软件产品；
- (d) 是否采用(a)(b)(c)的组合。

并针对以上每一种选择给出风险分析。