

重点大学软件工程规划系列教材

软件工程项目实训教程

吕云翔 刘瀚诚 刘天毅 编著

清华大学出版社



重点大学软件工程规划系列教材

软件工程项目实训教程

吕云翔 刘瀚诚 刘天毅 编著

清华大学出版社

RFID

清华大学出版社

北京

内 容 简 介

本书依据典型的软件开发过程组织内容,围绕“小型二手货交易平台”这一实际软件项目设计 CASE 工具实验,旨在培养读者应用 CASE 工具进行实际软件开发的实践能力。全书共 9 章,所涉及 CASE 软件包括:软件分析与设计工具 Microsoft Visio、面向对象软件设计工具 Rational Rose、软件数据模型建模工具 PowerDesigner、软件集成编码开发工具 Microsoft Visual Studio、Microsoft 测试管理工具 Test Manager、系统负载测试工具 LoadRunner、分布式版本控制系统 Git 与项目管理工具 Microsoft Project。

本书是高等院校计算机科学、软件工程及相关专业“软件工程”实践课程的理想教材,也可以供程序员、软件测试工程师、系统工程师以及软件项目经理等相关人员阅读参考。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

软件工程项目实训教程/吕云翔等编著. —北京:清华大学出版社,2016

(重点大学软件工程规划系列教材)

ISBN 978-7-302-44341-4

I. ①软… II. ①吕… III. ①软件工程—教材 IV. ①TP311.5

中国版本图书馆 CIP 数据核字(2016)第 166855 号

责任编辑:魏江江 薛 阳

封面设计:常雪影

责任校对:徐俊伟

责任印制:何 芊

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦 A 座 邮 编:100084

社总机:010-62770175 邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质 量 反 馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

课 件 下 载: <http://www.tup.com.cn>, 010-62795954

印 装 者:三河市吉祥印务有限公司

经 销:全国新华书店

开 本:185mm×260mm 印 张:16.75

字 数:418 千字

版 次:2016 年 11 月第 1 版

印 次:2016 年 11 月第 1 次印刷

印 数:1~2000

定 价:34.00 元

产品编号:067988-01

前 言

随着信息时代的飞速发展,计算机和软件已成为生产和生活中极为重要的组成部分。因此,培养熟练掌握计算机科学技术与软件工程相关领域知识的专业人才刻不容缓。

作为计算机和软件工程专业本科学生的必修课程,“软件工程”在国内外都是大学计算机科学教育体系中的核心课程之一。它担负着系统、全面地介绍软件工程的基本理论,为其他专业课程的学习奠定坚实基础,培养学生基本软件工程思维,以及提高学生实践和管理能力的重任。

软件工程是应用计算机科学、数学、逻辑学及管理科学等原理开发软件,将系统化的、严格约束的、量化的方法应用于软件的开发、运行与维护,以实现提高质量、降低成本等目的的一门新兴的、综合性的应用科学。传统的仅通过理论课程讲授软件工程的授课方式显然无法提高学生的操作和实践能力。因此,我们建议在开设“软件工程”的理论课程的同时,开设与之配套的“软件工程实践”课程。

随着软件工程理论的发展,诸如 CASE 工具、自动化测试工具、项目管理工具、配置管理工具相关的自动化工具也在不断涌现。这些工具使得软件开发效率大大提高,降低了软件的开发成本。相应地,这些工具也对软件开发领域的从业者、相关专业的学生提出了新的要求。要培养相关专业的人才,除了重视软件工程理论与实践的教学方法外,同样需要有足够优秀的实践辅导材料作为支撑。

市面上现行的软件工程类书籍更注重理论和概念,对实际中用到的工具介绍不甚详细,因此不能满足软件工程专业教师与学生的日常教学要求。大部分现行教材中使用的软件版本过低,内容也较为陈旧,实验设置只是作为理论内容的补充,不足以使学生实践软件工程中的概念,实践和可操作性不强,课时安排不够合理,因此与当今计算机和软件工程相关专业大学生的实践需求严重脱轨。

针对以上诸多问题,本书作者旨在编写一本适用于计算机和软件工程相关专业学生、内容与当前软件工程理论和工具发展实际情况紧密结合的软件工程实践课程参考书籍。本书在以下几个重要方面具有突出特色。

(1) 目标针对性强:本书针对计算机和软件工程相关专业学生,而不是广泛的高校学生,旨在培养他们的实践能力,加深对软件工程

的理解,为今后的课程学习和实践打下基础。

(2) 内容与与时俱进:本书充分考虑到现今软件与技术使用的实际情况,内容既考虑了软件版本的兼容性,又与最新的技术紧密结合,去除了过于陈旧和不实用的内容,符合软件工程工具发展的最新趋势。

(3) 真实实践项目:本书围绕“小型二手货交易平台”这一现实软件项目的软件生存周期设置实验,从不同角度展现软件生存周期各阶段的工作内容,针对每项工作设置相应CASE工具,使读者能够最大化掌握软件开发中各工具的使用方法。

(4) 配图充实丰富:讲解和实验都配有丰富的插图,清晰易懂,融入了大量具体的实践过程,而非简单介绍原理。

(5) 视角新颖独到:每章附有思考题和实验练习题,鼓励学生对实践过程和理论相联系,延伸思考,开拓学生视野。

(6) 教学操作性强:实验有清晰的步骤提示,易于老师讲授和学生自学,实验难度从浅入深,可按实际需要进行选做和调整,实验课时合理,符合一般教学安排。

本书分为9章,涉及软件工程理论与发展、软件设计与分析、软件开发、软件测试与管理、版本控制等方面的内容。每章分为工具简介、实验操作、案例演示和课后延伸4个部分。工具简介为学生提供所使用的软件的基本介绍、版本信息、功能用途及安装配置等;实验操作部分介绍了软件的基本使用操作方法;案例演示部分以贯穿全书的实际案例,用本章介绍的软件工具加以实际操作演示;课后延伸部分提供了思考题、实验练习题(可参考附录B中的文档做实验)和参考文献,帮助学生消化理解以及发散思维。本书建议的教学课时为:课内24课时,课外12课时,共36课时。具体教学安排可以根据实际教学情况进行调整。

软件工程实践课程应该建立在理论课程的基础上,本书不是取代软件工程理论课程的教材,因此,在学习本课程时,应同时学习软件工程理论课程,并配合理论课程教材一同使用。

本书总结了作者多年软件工程实践与教学的经验。为了使本书更具有可用性,我们以北京航空航天大学软件学院作为试点,用本书的内容进行了实验性教学。在此,感谢北京航空航天大学软件学院在工作上给与的支持,以及在成书过程中所提供的各种宝贵资源。

由于计算机技术发展迅速,软件工程实践课程本身还在探索之中,市场上针对计算机和软件工程相关专业的软件工程实践课程的教材中,能使人耳目一新的并不多见,我们力求使本教材做到完美;但由于学习能力和水平有限,书中难免有疏漏之处,恳请各位同仁和广大读者给予批评指正,也希望各位能将实践过程中的经验和心得与我们交流(yunxianglu@hotmail.com)。

作 者

2016年4月

于北航软件学院

目 录

第 1 章 绪论	1
1.1 软件工程概述	1
1.1.1 软件工程的理论与概念	1
1.1.2 软件工程的发展	2
1.1.3 软件生存周期	2
1.1.4 软件生存周期模型	4
1.1.5 软件工程实用工具	5
1.2 “小型二手货交易平台”案例介绍	6
1.2.1 项目背景	6
1.2.2 需求说明	6
1.2.3 平台要求	7
1.2.4 工期与开发团队说明	7
小结	7
思考题	8
参考文献	8
第 2 章 软件分析与设计工具 Microsoft Visio	10
2.1 概述	10
2.2 基本使用	12
2.2.1 操作面板介绍	12
2.2.2 利用模板进行绘图	12
2.2.3 创建/编辑模具与模板	14
2.3 利用 Visio 进行“小型二手货交易平台”结构化 模型设计	15
2.3.1 数据流模型设计	16
2.3.2 状态图模型设计	17
2.3.3 流程图设计	19
2.4 利用 Visio 进行“小型二手货交易平台”面向对 象设计	20
2.4.1 用例模型设计	21
2.4.2 类模型设计	21

2.4.3 数据模型设计	26
2.5 利用 Visio 进行“小型二手货交易平台”系统部署架构设计	29
小结	31
思考题	32
实验练习题	32
参考文献	32
第3章 面向对象软件设计工具 Rational Rose	34
3.1 概述	34
3.2 基本使用	35
3.2.1 操作面板介绍	35
3.2.2 基本使用介绍	38
3.3 利用 Rational Rose 进行“小型二手货交易平台”面向对象设计	41
3.3.1 构建用例模型	41
3.3.2 构建逻辑模型	44
3.3.3 根据用例模型绘制顺序图	48
3.3.4 构建活动图模型	50
3.3.5 构建包模型	55
3.3.6 构建构件模型	58
3.4 利用 Rational Rose 进行“小型二手货交易平台”系统部署架构设计	60
3.5 双向工程	63
3.5.1 正向工程	63
3.5.2 逆向工程	66
小结	68
思考题	68
实验练习题	69
参考文献	69
第4章 软件数据模型建模工具 PowerDesigner	71
4.1 概述	71
4.2 基本使用	74
4.2.1 操作面板介绍	74
4.2.2 建模过程	75
4.3 构建“小型二手货交易平台”业务处理模型 BPM	78
4.4 构建“小型二手货交易平台”数据模型	87
4.4.1 构建概念数据模型	87
4.4.2 构建物理数据模型	93
4.4.3 构建面向对象模型	104
4.5 模型转换	115

4.5.1 概念数据模型 CDM 转换为逻辑数据模型 LDM	115
4.5.2 概念数据模型 CDM 转换为物理数据模型 PDM	116
4.5.3 物理数据模型转换为数据库	119
4.5.4 面向对象模型转换为代码	120
小结	122
思考题	123
实验练习题	123
参考文献	123
第 5 章 软件集成编码开发工具 Microsoft Visual Studio	125
5.1 概述	125
5.2 基本使用	125
5.2.1 操作面板介绍	125
5.2.2 基本操作	127
5.3 利用 Visual Studio 进行“小型二手货交易平台”软件开发	130
5.3.1 利用 Visual Studio 进行“小型二手货交易平台”编码	131
5.3.2 利用 Visual Studio 进行“小型二手货交易平台”调试	135
5.3.3 利用 Visual Studio 进行“小型二手货交易平台”单元测试	139
小结	144
思考题	144
实验练习题	144
参考文献	144
第 6 章 Microsoft 测试管理工具 Test Manager	146
6.1 概述	146
6.2 基本使用	148
6.2.1 操作面板介绍	148
6.2.2 流程与操作	152
6.3 使用 Test Manager 进行“小型二手货交易平台”测试管理	158
小结	164
思考题	164
实验练习题	165
参考文献	165
第 7 章 系统负载测试工具 LoadRunner	166
7.1 概述	166
7.2 基本使用	167
7.2.1 操作面板介绍	167
7.2.2 VU Script 简介	171
7.3 利用 LoadRunner 进行“小型二手货交易平台”压力测试	172

7.3.1 定义虚拟用户	172
7.3.2 配置测试场景	191
7.3.3 运行与监控测试场景	197
7.3.4 分析测试结果	198
小结	199
思考题	200
实验练习题	200
参考文献	200
第8章 分布式版本控制系统 Git	202
8.1 概述	202
8.2 基本使用	204
8.2.1 Git 指令集与图形化工具	204
8.2.2 添加 Git Gui 操作	209
8.3 利用 Git 进行“小型二手货交易平台”版本控制	211
8.3.1 初始化版本库	212
8.3.2 分支控制	219
8.3.3 其他工作	222
小结	225
思考题	225
实验练习题	226
参考文献	226
第9章 项目管理工具 Microsoft Project	227
9.1 概述	227
9.2 基本使用	228
9.2.1 操作面板介绍	228
9.2.2 视图介绍	230
9.3 利用 Project 进行“小型二手货交易平台”项目管理	234
9.3.1 配置项目	235
9.3.2 任务管理	239
9.3.3 资源管理	244
9.3.4 项目审查	249
小结	251
思考题	252
实验练习题	252
参考文献	252
附录A extend 与 include 关系对应模具创建方法	254
附录B 案例：“小型二手货交易平台”	257

绪 论

1.1 软件工程概述

1.1.1 软件工程的概念与理论

什么是软件工程？简单来说，软件工程是将工程学的思想、方法与技术应用于软件开发过程的一门学科。1968 年，在北大西洋公约组织举行的一次学术会议上，人们为了解决当时出现的软件危机问题，首次提出了软件工程这个概念，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”。经过近半个世纪的发展，软件工程的定义、思想与方法等理论在不断丰富和完善，其相应的技术也在飞速进步。今天，软件工程已成为计算机科学与技术中一门独立的学科，对各行各业产生了深远的影响。

由于软件工程一直以来没有一个统一的定义，在本书中，谨引用 IEEE 对软件工程的定义作为标准：①：将系统化的、严格约束的、量化的方法应用于软件的开发、运行和维护，即将工程化应用于软件；②：在①中所述方法的研究。

软件工程的内容非常丰富，包括软件工程原理、过程、方法、模型、管理、环境及工具等，其中，过程、方法和工具是软件工程的三要素。

具体而言，IEEE 在 2014 年发布的《软件工程知识体系指南》中将软件工程知识体系划分为以下 15 个知识领域。

- (1) 软件需求(Software Requirements)
- (2) 软件设计(Software Design)
- (3) 软件构建(Software Construction)
- (4) 软件测试(Software Testing)
- (5) 软件维护(Software Maintenance)
- (6) 软件配置管理(Software Configuration Management)
- (7) 软件工程管理(Software Engineering Management)

- (8) 软件工程过程(Software Engineering Process)
- (9) 软件工程模型和方法(Software Engineering Models and Methods)
- (10) 软件质量(Software Quality)
- (11) 软件工程职业实践(Software Engineering Professional Practice)
- (12) 软件工程经济学(Software Engineering Economics)
- (13) 计算基础(Computing Foundations)
- (14) 数学基础(Mathematical Foundations)
- (15) 工程基础(Engineering Foundations)

1.1.2 软件工程的发展

软件工程是一门相对年轻的学科,它的发展伴随着软件和软件开发技术的更新。我们知道,软件是由计算机程序的概念发展演化而来的,是在程序和程序设计发展到一定规模并且逐步商品化的过程中形成的。软件开发大体上经历了从程序设计阶段、软件设计阶段到软件工程阶段的演变过程。

1968年,“软件工程”的概念首次被提出。20世纪70年代,人们开始在软件开发中运用一些工程思想,将工程技术与软件开发技术结合在一起,提出了简单的“瀑布模型”。然而随着软件规模的增大和开发难度的日益增加,“瀑布模型”遇到了瓶颈。

20世纪70~80年代,人们在不断的软件实践中逐渐提出了更多的模型,以解决遇到的各种问题。COCOMO、CMM模型等相继被提出,软件体系结构相关的研究和技术日益成熟。

与此同时,需求和设计工具、开发工具、测试工具和配置管理工具等也不断发展。随着CASE(Computer Aided Software Engineering,计算机辅助软件工程)的出现,软件开发的效率得到了进一步的提高。CASE是一套方法和工具,允许开发者在规定的应用规则下,由计算机自动生成合适的计算机程序。CASE工具和技术可提高系统分析员和程序员的工作效率,其重要技术包括应用生成程序、前端开发过程面向图形的自动化、配置和管理以及生存周期分析工具。CASE的出现极大促进了软件工程的发展。

20世纪90年代,面向对象思想的发展渗透到了软件工程中来。基于面向对象的分析和设计模式、方法、技术和工具极大地丰富了软件工程的理论,使其日趋成熟。其中典型的代表是UML设计方法和技术。

20世纪90年代末,随着“敏捷软件开发宣言”的发布,敏捷开发思想逐渐兴起。极限编程XP、Scrum、特征驱动开发等概念和过程也相继被提出,在实践中不断被完善。

21世纪以来,随着互联网时代的高速发展,以大数据、云计算、移动互联网和智能信息技术等为代表的新技术冲击着软件工程,带来了前所未有的挑战。新的技术变革必将引发软件工程领域更多的概念、思想和工具的出现。敏捷开发、精益思想、构件化软件开发、模型驱动体系结构等都可能引领未来的软件工程发展方向。

1.1.3 软件生存周期

同任何事物一样,一个软件产品或软件系统也要经历孕育、诞生、成长、成熟、衰亡等阶段,一般称为软件生存周期(也称软件生命周期)。通过把整个软件生存周期划分为若干阶

段,每个阶段有明确的任务,这样一来,规模宏大、结构复杂的软件及其开发变得容易控制和管理。通常,软件生存周期主要包括以下阶段。

(1) 可行性研究:主要目的是定义问题,确定软件的开发目标和分析其可行性,制定初步的开发计划。

(2) 需求分析:在确定软件开发可行的情况下,对目标软件系统需要解决的问题和要实现的功能进行详细分析,形成需求规格说明书。

(3) 软件设计:根据需求分析的结果,对整个软件系统进行设计,分为概要设计和详细设计。概要设计旨在建立系统的总体架构,详细设计关注每个子系统和模块的内部实现细节。形成的软件设计说明书将为后续编码实现提供依据。

(4) 编码实现:根据软件设计说明书,将设计结果转换成计算机可运行的程序代码。在编码实现过程中必须要制定统一、符合标准的编码规范,以保证程序的可读性、易维护性,提高程序的运行效率和整个系统的稳定性。

(5) 软件测试:主要目的是发现软件产品中存在的缺陷,进而保证软件产品的质量。可以划分为单元测试、集成测试、系统测试和验收测试。

(6) 运行与维护:软件产品交付后,随着用户需求的增长或改变,以及市场环境的变化,软件产品的功能需要不断完善。为了保证软件产品的正常运行,需要进行一定的维护。

ISO 12207 标准为软件从构思到报废的生存周期过程提供了框架。ISO 12207 规定了 5 个“主要过程”: 采办、供应、开发、维护和操作。它把 5 个过程分为各项“活动”,再把各项活动分为各项“任务”,并对“任务”的执行提供了要求。它规定了 8 个“保障过程”: 文档、配置管理、质量保证、验证、确认、联合评审、审查和问题解决,以及 4 个“组织过程”: 管理、基础设施建设、改进和培训。ISO 12207 标准可以由图 1-1 体现。

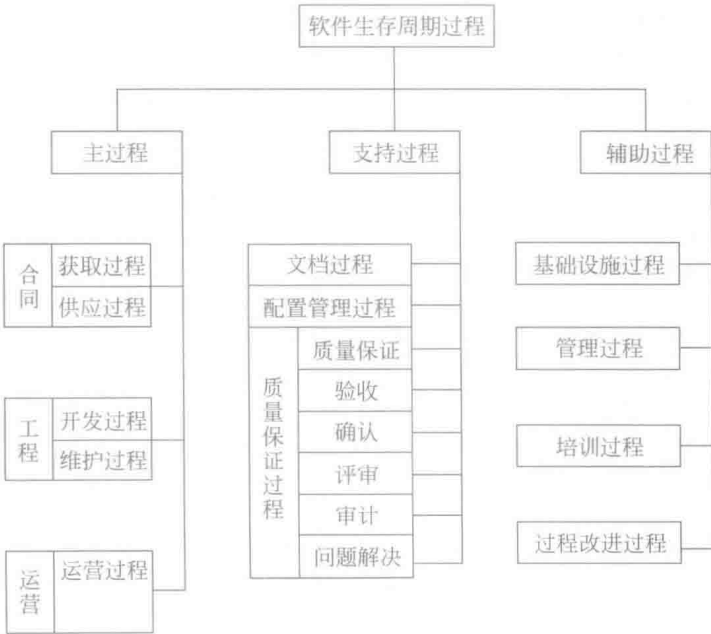


图 1-1 ISO 12207 标准

生存周期也可以用图 1-2 来表示。

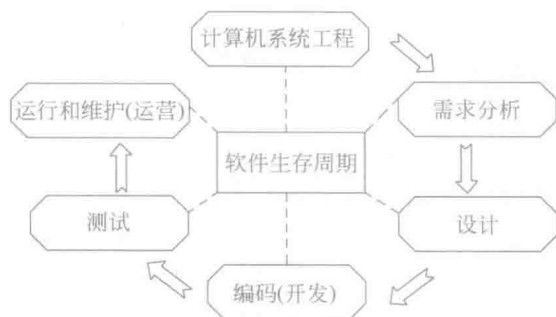


图 1-2 软件生存周期

但随着新的面向对象的设计方法和技术的成熟,以及敏捷思想的出现,早期软件生存周期设计方法和理念需要进行一些调整。

1.1.4 软件生存周期模型

根据软件生存周期的概念,软件从产生到消亡都要经历需求、设计、编码、测试和维护等过程。这样的完整过程称为“生存周期模型”。常见的软件生存周期模型包括瀑布模型、增量模型、螺旋模型、快速原型模型、喷泉模型、统一软件开发过程模型、敏捷过程与极限编程模型等。各个模型的特点如表 1-1 所示。

表 1-1 软件生存周期模型及特点

生存周期模型	主要特点
瀑布模型	瀑布模型将软件生存周期中各活动(制定计划、需求分析、系统设计、编码、测试和维护)规定为依线性顺序的若干阶段,各项活动自上而下、相互衔接的固定次序,如同瀑布流水,逐级下落
增量模型	增量模型是把整个软件系统分解为若干个软件构件,在开发过程中,逐个实现每个构件。如果发现问题可以及早进行修正,逐步进行完善,最终获得满意的软件产品
螺旋模型	螺旋模型将瀑布模型和快速原型模型集合起来,强调其他模型所忽视的风险,特别适合于大型复杂系统。该模型分为 4 个环节:制定计划、风险分析、开发实施和用户评价。4 个活动螺旋式地重复执行,直到最终得到用户认可的产品
快速原型模型	通过快速建立一个能反映用户主要需求的原型系统,让用户在使用和实践中提出改进意见,完善目标系统的概貌。开发人员根据意见快速修改原型系统,然后再次请用户试用直到用户认可原型的功能。至此,根据原型系统获取了用户的所有需求,可以形成需求规格说明书并进行设计与开发
喷泉模型	喷泉模型是典型的面向对象生存周期模型,强调面向对象软件开发过程中迭代和无缝的特性
统一软件开发过程模型	统一软件开发过程模型(RUP)是一种面向对象的、基于迭代思想的软件生存周期模型。按时间划分为 4 个阶段:初始、细化、构造和交付。这 4 个阶段按顺序依次进行,在每个阶段都可能若有若干次迭代
敏捷过程和极限编程模型	敏捷过程强调“个体和交互”胜过“过程和工具”、“可以使用的软件”胜过“面面俱到的文档”、“与客户协作”胜过“合同”、“响应变化”胜过“遵循计划” 极限编程是一种典型的敏捷过程模型,“极限”指把好的开发实践运用到极致,包括结对编程、短周期交付、测试驱动开发、集体所有等

1.1.5 软件工程实用工具

随着软件工程理论的发展,用于辅助软件开发的应用工具也大量涌现,统称计算机辅助软件工程工具(CASE 工具)。这些工具通过自动化软件开发过程中的需求分析、系统设计、编码、测试及管理工 作,大大降低了软件开发的成本,加快了软件开发的速度。下面将结合软件生存周期的各个阶段简要介绍相关工具实例。

在需求分析与系统设计阶段,软件开发团队通过多种手段收集原始业务需求,通过需求分析予以抽象提炼,转化为可用的需求规格说明,并基于需求规格说明进行系统设计与详细设计。在这一过程中常用的 CASE 工具有面向通用软件设计的 Microsoft Visio,用于面向对象软件设计的 Rational Rose,用于数据库设计的 PowerDesigner,除此之外近年还出现了更加集成化的工具 Enterprise Architect 等。这些工具通过简化 UML 图的绘制工作,以及强大的模型转换功能(诸如正向工程、反向工程、数据库模型转化等),大大简化了设计以及从设计向编码转化的工作。

在编码阶段,软件开发团队将设计阶段的成果付诸实践,通过编写代码将设计转化为可用的软件实例。早期的编码工作使用文本编辑器进行,然而随着系统规模的扩张,其编码、调试的不便也大大提升。集成开发环境(IDE)通过提供代码高亮、补全,内置调试工具等功能,大大降低了这种不便。主流的实例如表 1-2 所示。

表 1-2 主流的实例

名 称	编 程 语 言
Turbo Pascal	Pascal
Dev C++	C/C++
Codeblocks	C/C++
Visual Studio	C++/VB/C# /JavaScript 等
Rubymine	Ruby
Webstorm	JavaScript
PHPstorm	PHP
PyCharm	Python
Eclipse	Java
IntelliJ Idea	Java
XCode	Objective-C/Swift

软件的测试包括单元测试、集成测试等。若进一步细化,则单元测试可分为白盒测试与黑盒测试,集成测试可分为功能测试、负载测试、平台测试、文档测试、可用性测试等。若由开发人员自行编写测试代码,不仅十分烦琐,而且测试编码本身便可能存在错误。因此通常会使用自动化测试工具进行测试。除单元测试工具外,较为流行的自动化测试工具包括性能测试工具 LoadRunner、测试管理工具 TestManager、Web 服务测试工具 QTester(简称 QT)等。单元测试工具通常与语言及开发框架关联密切,部分实例如表 1-3 所示。

表 1-3 部分实例

名 称	编程语言
CUnit	C
CppUnit	C++
JUnit	Java
NUnit	.NET
Perl Testing	Perl
Mocha/Should.js	Node.js
内置 unittest 模块	Python
PHPUnit	PHP
内置 Test::Unit 模块	Ruby

如前文所述,除主过程外,软件开发过程还包括诸多其他活动,而其中最重要的便是配置管理与项目管理。配置管理通常分为不同模式,每一种模式均有对应工具,较为著名的有 Microsoft VSS、CVS、SVN 等,近年来最常用的为 Git。而项目管理领域最普遍使用的为微软公司开发的 Microsoft Project,该软件提供了强大的项目管理功能,基本能够满足企业级项目管理的全部需要。此外,近年来随着敏捷开发的兴起,诸如 Teambition 等基于看板的轻量级开发平台也拥有了广大的用户群体。

除此之外,在软件过程的其他活动中同样存在众多 CASE 工具。在原型设计方面有快速原型构建系统 Dreamweaver,在协作文档管理方面有在线协作办公系统 Microsoft Office Online,除此之外,还有在线协作软件设计平台 Processon 等,由于篇幅有限,这里不再赘述。

1.2 “小型二手货交易平台”案例介绍

本书所有章节实验均以同一案例——“小型二手货交易平台”为基础,以保障各阶段实验的连贯、一致性。下面对该案例做一简要介绍。

1.2.1 项目背景

大学生作为社会中不可忽视的团体,每学期对各类商品如图书、电子产品等都有不同程度的需求。考虑到大学生没有经济来源,及部分商品在各年级同学间重复使用的特殊情况,每学期购买新商品的行为并不划算。但学生在校园内随意交易的行为不易管理,因此这里考虑建立校园二手货交易平台,使校园二手货交易方便并易于管理。

1.2.2 需求说明

根据需求调研的结果,平台的使用者主要包括希望购买二手货的用户与希望将手中积存的二手货予以售卖的用户。前者应能够通过平台发布并管理需求,能够通过平台找到所需二手货并联系卖方以达成交易。后者应能够通过平台发布并管理二手货,查找感兴趣的需求并联系买方以达成交易。沟通过程可以在站内进行,也可通过线下手段(诸如手机、邮件等)联系。除此之外,平台应提供评价系统,买卖双方可以针对特定交易对对方进行评分,该评分将参与用户信用额度的计算。信用额度能够被其他用户访问,以便在使用系统服务时提供参考。

除此之外,为保证平台交易质量,二手货交易平台应设置若干管理人员。管理人员负责监控交易记录,接受买卖双方对不良交易现象的检举并予以处理,能够撤除不良商品并给予相关用户一定惩罚。系统管理人员还应能够发布关于系统维护、更新、新闻等信息的公告。

1.2.3 平台要求

“二手货交易平台”所面向的用户群体主要为 PC 端用户(后续将开放针对 Android 与 iOS 平台的服务),为保障便捷性,平台应为 Web 服务。由于客户所使用设备分辨率各有不同,平台应保障在不同分辨率下均能提供友好的交互服务。平台使用范围局限于单所大学校园内。在此基础上应保证即便出现可能发生的最大同时访问量,系统仍然能够正常运转,响应时间应在可接受范围内。除此之外,系统还应在安全性、可靠性上予以保障,不应出现数据丢失、隐私泄漏等事故。

1.2.4 工期与开发团队说明

工期:平台第一版工期约为四五个月,除特殊原因,不应久于 5 个月。

开发团队:包含必要的项目经理、架构师、UI 设计师、前端工程师、后端工程师、数据库管理员。本书中假设团队成员组成如表 1-4 所示。

表 1-4 假设团队成员

姓名	职 位	描 述
小赵	项目经理	负责制定项目计划与预算,监控项目进度与成本,管理人员,协调团队氛围等
小钱	架构师	作为技术主管加入团队,精通完成项目所需的全部技术,负责规范项目开发流程,在技术上指导团队成员,解决技术难点等工作,主导系统体系架构设计
小孙	UI 设计师	负责设计系统与用户的交互界面,主要包括功能设计与样式设计,通常需要提供概念设计图与相关资源素材
小李	UI 设计师	
小周	前端工程师	负责开发 C/S 模型系统的 Client 端功能
小吴	前端工程师	
小郑	后端工程师	负责 C/S 模型系统的 Server 端功能
小王	后端工程师	
小冯	数 据 库 管 理员	主导数据库概念模型、逻辑模型、物理模型设计,负责数据库部署、数据管理、数据灾备、数据迁移等工作

上述团队成员构成与工期将用于第 9 章中的项目管理实验中。此外,本书从第 2 章开始,将结合以上项目需求,利用不同 CASE 工具从该项目的需求分析、软件设计、编码、测试、项目管理、配置管理共 6 个方面进行实践。

上述 8 人团队为本书所定义的虚拟团队。在实际工作中的团队人数可能多或少于这个数字。但是表 1-4 中所定义的职能划分基本不变。

小结

软件工程将工程化应用于软件开发,旨在降低软件开发的风险与成本,自从软件危机开始,已经发展出诸多理论成果。基于软件工程理论,可将软件生存周期分为需求分析、设计、

编码、测试等多个阶段,并由此演化出诸多生存周期模型以指导软件开发。而基于这些阶段与活动,结合软件工程理论而产生的诸多 CASE 工具大大降低了软件开发的难度。

本章对软件工程的基本概念与发展、软件生存周期各个阶段的活动以及软件生存周期模型予以讲解,介绍了软件生存周期各阶段常用的 CASE 工具。并对本书的排版约定、章节组织、实践内容等做简要介绍,以便读者根据自身需要裁剪学习内容。

思考题

1. 根据软件工程的发展历史,思考现代软件工程的特点和未来趋势。
2. 软件生存周期的各个阶段之间的联系是怎样的,在具体的软件工程过程中是如何体现的?
3. 结合有关文献对比各个软件工程生存周期模型,思考其优劣之处和适用的软件系统类型。
4. 为什么会出现敏捷模型?敏捷过程模型与传统软件生存周期模型的联系与区别有哪些?
5. 选择一款 CASE 工具,对其功能进行调研,思考:它是在软件生存周期的什么阶段(或什么活动)中,通过什么方式,起到何种作用的?

参考文献

软件工程自诞生以来已有数十年历史,Barry Boehm 在参考文献[1]中详细分析了软件工程自诞生以来的发展历程,并对其未来发展做一简要展望。对于相关发展的反思还可参见 Frederic P. Brooks 所著的《人月神话》^[2]与《设计原本》^[3]。

在软件工程理论领域存在众多标准,其中最重要的为 ISO 12207 标准。该标准为软件从诞生到作废的生存周期过程提供了标准化的框架,并按照级别不同定义了过程、活动与任务,详见参考文献[4]。

软件生存周期模型至今已拥有众多家庭成员,从最早的瀑布模型^[5]开始,先后演化出增量模型^[6]、演化模型^[7]、螺旋模型^[8]、统一过程模型 RUP^[9]等,参考文献[10]对上述模型做一对比,详细分析了各个模型的优势与不足。此外,近年来以 XP、Scrum 等为首的敏捷模型正广受推崇,关于 XP 模型与 Scrum 模型详见参考文献[11,12],Kenneth S. Rubin 在文献中详细介绍了两个模型的指导原则与实践方法。

- [1] B Boehm. A view of 20th and 21st century software engineering, Proceedings of the 28th international conference on Software engineering. ACM, pp. 12-29, 2006.
- [2] F Brooks. The Mythical Man-Month: Essays on Software Engineering. Addison-Wesley Professional, 1995.
- [3] F Brooks. The Design of Design: Essays from a Computer Scientist. Addison-Wesley Professional, 2010.
- [4] ISO/IEC JTC 1/SC 7. ISO/IEC 12207:2008 Systems and software engineering-Software life cycle processes. Geneva Switzerland, ISO, 2008.