



免费提供
电子教案

高等院校规划教材
软件工程系列

软件测试教程

周元哲 编著



机械工业出版社
CHINA MACHINE PRESS



高等院校规划教材·软件工程系列

软件测试教程

周元哲 编著

机械工业出版社

本书较为全面、系统地介绍了当前测试领域的理论和实践知识,反映了当前最新的软件测试理论、标准、技术和工具,展望了软件测试的发展趋势。全书共分10章,主要内容包括软件工程与软件测试、软件测试概论、软件测试基础知识、软件测试过程、黑盒测试、白盒测试、自动化测试技术及其工具、性能测试、面向对象测试和软件测试管理,并在附录中列举了软件测试面试题、Yahoo 真实测试案例,以及软件评测师考试的相关内容。

本书可作为高等院校相关专业软件测试的教材或教学参考书,也可作为从事计算机应用开发的各类技术人员的参考书,还可作为全国计算机软件评测师考试、软件技术资格与水平考试的培训资料。

图书在版编目 (CIP) 数据

软件测试教程/周元哲编著. —北京:机械工业出版社,2010.3

(高等院校规划教材·软件工程系列)

ISBN 978-7-111-28862-6

I. ① 软… II. ① 周… III. ① 软件-测试-高等学校-教材
IV. ① TP311.5

中国版本图书馆 CIP 数据核字 (2010) 第 035692 号

机械工业出版社 (北京市百万庄大街 22 号 邮政编码 100037)

责任编辑:陈 皓 常建丽

责任印制:杨 曦

唐山丰电印务有限公司印刷

2010 年 4 月第 1 版·第 1 次印刷

184mm × 260mm · 16.25 印张 · 396 千字

0001-3000 册

标准书号:ISBN 978-7-111-28862-6

定价:29.00 元

凡购本书,如有缺页、倒页、脱页,由本社发行部调换

电话服务

网络服务

社服务中心:(010) 88361066

门户网:<http://www.cmpbook.com>

销售一部:(010) 68326294

教材网:<http://www.cmpedu.com>

销售二部:(010) 88379649

读者服务部:(010) 68993821

封面无防伪标均为盗版

出版说明

计算机技术的发展极大地促进了现代科学技术的发展，明显地加快了社会发展的进程。因此，各国都非常重视计算机教育。

近年来，随着我国信息化建设的全面推进和高等教育的蓬勃发展，高等院校的计算机教育模式也在不断改革，计算机学科的课程体系和教学内容趋于更加科学和合理，计算机教材建设逐渐成熟。在“十五”期间，机械工业出版社组织出版了大量计算机教材，包括“21世纪高等院校计算机教材系列”、“21世纪重点大学规划教材”、“高等院校计算机科学与技术‘十五’规划教材”、“21世纪高等院校应用型规划教材”等，均取得了可喜成果，其中多个品种的教材被评为国家级、省部级的精品教材。

为了进一步满足计算机教育的需求，机械工业出版社策划开发了“高等院校规划教材”。这套教材是在总结我社以往计算机教材出版经验的基础上策划的，同时借鉴了其他出版社同类教材的优点，对我社已有的计算机教材资源进行整合，旨在大幅提高教材质量。我们邀请多所高校的计算机专家、教师及教务部门针对此次计算机教材建设进行了充分的研讨，达成了许多共识，并由此形成了“高等院校规划教材”的体系架构与编写原则，以保证本套教材与各高等院校的办学层次、学科设置和人才培养模式等相匹配，满足其计算机教学的需要。

本套教材包括计算机科学与技术、软件工程、网络工程、信息管理与信息系统、计算机应用技术以及计算机基础教育等系列。其中，计算机科学与技术系列、软件工程系列、网络工程系列和信息管理与信息系统系列是针对高校相应专业方向的课程设置而组织编写的，体系完整，讲解透彻；计算机应用技术系列是针对计算机应用类课程而组织编写的，着重培养学生利用计算机技术解决实际问题的能力；计算机基础教育系列是为大学公共基础课层面的计算机基础教学而设计的，采用通俗易懂的方法讲解计算机的基础理论、常用技术及应用。

本套教材的内容源自致力于教学与科研一线的骨干教师与资深专家的实践经验和研究成果，融合了先进的教学理念，涵盖了计算机领域的核心理论和最新的应用技术，真正在教材体系、内容和方法上做到了创新。同时本套教材根据实际需要配有电子教案、实验指导或多媒体光盘等教学资源，实现了教材的“立体化”配套。本套教材将随着计算机技术的进步和计算机应用领域的扩展而及时改版，并及时吸纳新兴课程和特色课程的教材。我们将努力把这套教材打造成为国家级或省部级精品教材，为高等院校的计算机教育提供更好的服务。

对于本套教材的组织出版工作，希望计算机教育界的专家和老师能提出宝贵的意见和建议。衷心感谢计算机教育工作者和广大读者的支持与帮助！

机械工业出版社

前 言

随着软件的规模和复杂性的大幅提升，如何保证软件质量的可靠性变得日益重要。软件测试是保证软件质量的关键技术之一，同时也是软件开发过程中的一个重要环节，其理论知识和技术工具都在不断更新。

软件测试从软件工程中演化而来，并且还在不断的发展之中。在学习本书之前，需要一些先行知识（如一门高级语言、数据库、数据结构，以及软件工程的基本理论知识等）作为支撑。

本书较为全面地涵盖了当前测试领域的专业知识，追溯了软件测试的发展史，反映了当前最新的软件测试理论、标准、技术和工具，展望了软件测试的发展趋势。全书共分 10 章，主要内容包括软件工程与软件测试、软件测试概论、软件测试基本知识、软件测试过程、黑盒测试、白盒测试、自动测试技术及其工具、性能测试、面向对象测试、软件测试管理，并在附录中列举了软件测试面试题、Yahoo 测试真实案例以及软件评测师考试的相关内容。

本书内容精练、文字简洁、结构合理、综合性强，并明确面向初、中级读者，由“入门”起步，侧重“提高”。

在本书的编写过程中，作者与软件公司的友人进行了大量深入的讨论，其中，神州数码融信软件有限公司西安开发中心技术总监马明全博士、Yahoo 测试负责人蔡雪尽女士和深圳市易思博信息技术有限公司西安分公司李荣生经理等软件企业人士对本书的写作大纲、写作风格提出了很多宝贵的意见。西安邮电学院计算机学院王忠民教授、王曙燕教授、陈莉君教授对本书的编写也给予了大力的支持，并提出了指导性意见。中原工学院计算机学院刘卫光副院长，西安电子科技大学计算机学院段振华博导、鱼滨教授，以及西安邮电学院实训中心的孟伟君、宋辉等审阅了部分书稿。在此对以上人士一并表示感谢！本书在写作过程中参阅了大量中外文的专著、教材、论文、报告及网上的资料，在此向各位作者表示诚挚的敬意和衷心的感谢！

由于作者水平有限，书中难免有不足之处，恳请广大读者批评指正。

作 者

目 录

出版说明

前言

第 1 章 软件工程与软件测试	1
1.1 软件	1
1.1.1 软件的发展史	1
1.1.2 软件的生命周期	2
1.2 软件过程	3
1.2.1 RUP	3
1.2.2 敏捷过程	6
1.3 软件缺陷	7
1.3.1 缺陷	7
1.3.2 缺陷后果	8
1.3.3 缺陷分析方法	10
1.3.4 与缺陷相关的几个概念	10
1.4 软件质量	12
1.4.1 简介	12
1.4.2 质量模型	12
1.4.3 质量管理	13
1.4.4 CMM/CMMI	15
1.4.5 质量与测试	21
1.5 软件测试与软件开发的关系	21
1.6 思考题	23
第 2 章 软件测试概论	24
2.1 关于测试的几种观点	24
2.2 测试发展历程	25
2.3 测试行业现状	26
2.4 软件测试职业	27
2.4.1 测试岗位的职能	27
2.4.2 测试人员的素质	27
2.5 测试用例	28
2.5.1 简介	28
2.5.2 测试用例的设计准则	30
2.5.3 测试用例的设计步骤	31
2.5.4 测试用例的内容及维护	31
2.5.5 测试用例的设计误区	32

2.6	与测试相关的几个概念	33
2.6.1	排错	33
2.6.2	验证与确认	33
2.7	测试认识的误区	34
2.8	思考题	35
第3章	软件测试基础知识	37
3.1	软件的可测试性与测试目的	37
3.2	软件测试分类	38
3.2.1	按照开发阶段划分	39
3.2.2	按照执行主体划分	39
3.2.3	按照执行状态划分	39
3.2.4	按照测试技术划分	44
3.3	软件测试模型	45
3.3.1	V模型	45
3.3.2	W模型	46
3.3.3	H模型	47
3.3.4	X模型	47
3.3.5	前置模型	48
3.3.6	测试模型总结	48
3.4	两个“3”	50
3.4.1	测试3维空间	50
3.4.2	测试驱动的3种方法	51
3.5	软件测试注意事项	52
3.6	终止软件测试的标准	53
3.7	思考题	53
第4章	软件的测试过程	54
4.1	概述	54
4.2	测试计划	55
4.2.1	测试计划的内容	56
4.2.2	一些关键问题	58
4.3	测试设计	59
4.4	测试实施过程	60
4.4.1	单元测试	61
4.4.2	集成测试	66
4.4.3	确认测试	71
4.4.4	系统测试	72
4.4.5	验收测试	73
4.4.6	回归测试	74
4.4.7	小结	76

4.5	评估测试	77
4.5.1	评估测试的方法	77
4.5.2	评估测试的内容	78
4.6	思考题	78
第5章	黑盒测试	79
5.1	概述	79
5.2	等价类划分	79
5.2.1	划分原则	79
5.2.2	应用实例	80
5.3	边界值分析法	81
5.3.1	设计原则	82
5.3.2	应用实例	82
5.4	功能图测试	83
5.5	随机数据法	84
5.6	元素分析法与错误推测法	85
5.7	决策表	85
5.7.1	应用实例	87
5.7.2	优点和缺点	88
5.8	因果图	88
5.8.1	基本术语	89
5.8.2	测试用例的生成步骤	89
5.8.3	应用实例	90
5.9	测试方法的综合使用策略	92
5.10	GUI 测试	93
5.10.1	GUI 测试的 10 条原则	94
5.10.2	GUI 测试步骤	98
5.11	测试工具 QTP	99
5.12	GUI 捕捉/回放实例	108
5.12.1	采用原理	108
5.12.2	具体实现	108
5.13	思考题	112
第6章	白盒测试	113
6.1	概述	113
6.2	白盒测试的发展历程	113
6.3	逻辑覆盖法	115
6.3.1	语句覆盖	116
6.3.2	判定覆盖	116
6.3.3	条件覆盖	117
6.3.4	判定 - 条件覆盖	117

6.3.5	条件组合覆盖	118
6.3.6	路径覆盖	119
6.3.7	逻辑覆盖法总结	120
6.4	结构性测试	120
6.4.1	数据流测试	121
6.4.2	基本路径测试	122
6.5	控制结构测试	129
6.5.1	条件测试	129
6.5.2	循环测试	130
6.6	程序插桩	131
6.7	其他测试方法	132
6.8	白盒测试综合策略	132
6.8.1	最小测试用例数的计算	133
6.8.2	覆盖准则	133
6.9	调试	135
6.9.1	调试策略	135
6.9.2	3种调试技术	136
6.10	思考题	139
第7章	自动化测试及其工具	140
7.1	自动化测试	140
7.1.1	自动化测试的作用	140
7.1.2	自动化测试的应用场合	142
7.2	自动化测试的发展历程	142
7.3	测试成熟度模型	144
7.4	自动化测试的属性	148
7.5	自动化测试的原理	149
7.5.1	代码分析	149
7.5.2	录制回放	149
7.5.3	脚本技术	150
7.5.4	虚拟用户技术	151
7.6	自动化测试生存周期方法学	151
7.7	自动化测试工具	152
7.7.1	简介	152
7.7.2	自动化测试工具的分类	152
7.7.3	测试工具的特征	159
7.7.4	测试工具的选择	160
7.8	JUnit	160
7.8.1	JUnit 的安装	161
7.8.2	JUnit 的特点	161

7.8.3 JUnit 的内容	162
7.8.4 JUnit 的设计原则	163
7.8.5 实例	164
7.9 自动化测试研究的热点	169
7.10 思考题	170
第8章 性能测试	171
8.1 基本概念	171
8.1.1 响应时间	172
8.1.2 并发用户数	172
8.1.3 吞吐量	173
8.1.4 性能计数器与资源利用率	173
8.1.5 休眠时间	174
8.1.6 点击率	175
8.2 性能测试的分类	175
8.2.1 负载测试	175
8.2.2 压力测试	176
8.2.3 可靠性测试	177
8.2.4 安全性测试	177
8.2.5 兼容性/配置测试	178
8.2.6 基准/容错性测试	178
8.2.7 可用性测试	179
8.2.8 文档测试	180
8.3 Web 测试	181
8.3.1 Web 结构模型	181
8.3.2 Web 测试策略	181
8.4 性能测试工具 JMeter	186
8.4.1 简介	186
8.4.2 操作流程	186
8.5 思考题	190
第9章 面向对象的测试	191
9.1 面向对象的基本思想	191
9.1.1 简介	191
9.1.2 基本术语	191
9.1.3 设计原则	192
9.2 面向对象的影响测试	193
9.2.1 封装性影响测试	193
9.2.2 继承性影响测试	194
9.2.3 多态性影响测试	194
9.3 面向对象的测试方法	195

9.3.1	基于故障的测试	195
9.3.2	随机测试	195
9.3.3	基于脚本的测试	196
9.3.4	分割测试	196
9.3.5	行为模型测试	196
9.4	面向对象的测试模型	197
9.5	面向对象的分析测试	197
9.5.1	对象测试	197
9.5.2	结构测试	198
9.5.3	主题测试	198
9.5.4	属性和实例关联测试	198
9.5.5	服务和消息关联测试	199
9.6	面向对象的设计测试	200
9.6.1	对认定类的测试	200
9.6.2	对类层次结构的测试	200
9.6.3	对类库支持的测试	201
9.7	面向对象编程的测试	201
9.7.1	检测数据成员是否满足数据封装的要求	201
9.7.2	检测类是否实现了要求的功能	202
9.8	面向对象的单元测试	202
9.8.1	功能性和结构性测试	202
9.8.2	测试用例设计和选择	203
9.8.3	测试充分性准则	204
9.8.4	测试继承的几个问题	204
9.9	面向对象的集成测试	206
9.9.1	简介	206
9.9.2	两种策略	207
9.9.3	面向对象的交互测试	207
9.9.4	举例	208
9.10	面向对象的系统测试	210
9.11	思考题	211
第10章	软件测试管理	212
10.1	测试项目管理概述	212
10.2	测试管理计划	213
10.3	软件测试文档	214
10.4	测试组织	215
10.4.1	测试组织构成具有阶段性	215
10.4.2	测试组织的构成人员	216
10.5	测试过程的管理	217

10.5.1	测试项目的跟踪与监控	217
10.5.2	测试环境及配置管理	218
10.5.3	测试风险管理	219
10.5.4	测试成本管理	220
10.6	测试管理工具	221
10.6.1	3代测试框架	221
10.6.2	测试用例管理工具	222
10.6.3	测试管理工具	223
10.7	测试管理工具 Bugzilla	223
10.7.1	简介	223
10.7.2	操作流程	224
10.8	思考题	227
附录		228
附录 A	软件测试工程师面试题	228
附录 B	Yahoo 真实测试案例——企业邮箱测试	231
B.1	测试项目概述	231
B.2	测试评估产品质量等级标准	231
B.3	测试内容及执行情况	232
B.4	测试 Bug 统计	234
B.5	遗留问题分析及风险评估	235
B.6	测试人员自我总结	236
附录 C	软件评测师	237
C.1	概述	237
C.2	内容介绍	238
C.3	相关资料	243
参考文献		244

第1章 软件工程与软件测试

本章介绍了软件的发展史、当前流行的软件过程模型、软件缺陷和软件故障的概念、软件质量及质量保证体系的概念，软件测试与软件开发过程的关系等理论知识，为学习本书的后续内容做了准备。

1.1 软件

软件是一系列按照特定顺序组织的计算机数据和指令的集合。它并不只是计算机上运行的程序代码，还包括数据结构和文档等。

一般地，软件包括如下内容：

- 1) 运行时，能够提供所要求功能和性能的指令或计算机程序集合。
- 2) 程序能够满意地处理信息的数据结构。
- 3) 描述程序功能需求以及程序如何操作和使用所要求的文档。

1.1.1 软件的发展史

软件的发展经历了如下几个阶段：

20 世纪 50 年代初期至 60 年代中期，是软件发展的第 1 阶段，又称为程序设计阶段。此时硬件已经通用化，而软件的生产却是个体化的。软件产品为专用软件，规模较小，功能单一，开发者即使用者，软件只有程序，无文档。软件设计在人们的头脑中完成，形成了“软件等于程序”的错误观念。

第 2 阶段从 20 世纪 60 年代中期至 70 年代末期，称为程序系统阶段。此时，随着多道程序设计技术、多用户系统、人机交互式技术、实时系统和第 1 代数据库管理系统的出现，专门从事软件开发的“软件作坊”广泛应用，但软件技术和管理水平相对落后，导致“软件危机”出现。软件危机主要表现在以下几个方面：

- 1) 软件项目无法按期完成，超出经费预算，软件质量难以控制。
- 2) 开发过程管理不规范，文档书写不完整，使得软件维护费用高。
- 3) 缺乏严密有效的质量检测手段，交付给用户的软件质量差，在运行中出现许多问题，甚至带来严重的后果。
- 4) 系统更新换代难度大。

第 3 阶段称为软件工程阶段，从 20 世纪 70 年代中期至 80 年代中期，由于微处理器的出现、分布式系统广泛应用，使得计算机真正成为大众化的东西。以软件的产品化、系列化、工程化和标准化为特征的软件产业发展起来，软件开发有了可以遵循的软件工程化的设计准则、方法和标准。1968 年，北大西洋公约组织的计算机科学家在联邦德国召开了国际会议，主要讨论软件危机问题，正式提出并使用“软件工程”概念，这标志着软件工程的诞生。

软件工程的任务是通过提供规范化的分析设计方法及工具软件，来避免或者减少软件错

误的发生，为最终根除软件危机提供强有力的技术保障。

软件工程学涉及到与生产软件相关的所有活动，包括计算机科学、管理学、经济学、心理学等。其研究的主要内容是：如何应用科学的理论和工程上的技术来指导软件的开发，从而达到较少的投资获得高质量软件的最终目标。

第4阶段是从20世纪80年代中期至今，客户端/服务器（C/S）体系结构、特别是Web技术和网络分布式对象技术的飞速发展，导致软件系统体系结构向更加灵活的多层分布式结构演变，CORBA、EJB、COM/DCOM 3大分布式的对象模型技术相继出现。

2006年，面向服务架构（Service-Oriented Architecture, SOA）出现。作为“抽象、松散耦合和粗粒度”的软件架构，其根据需求通过网络对松散耦合的粗粒度应用组件进行分布式部署、组合和使用，主要用于解决传统对象模型中无法解决的异构和耦合问题。

至此，软件发展经历了Mainframe结构、Client/Server结构、B/S多层分布式结构、SOA的演变过程，整个软件系统变得越来越分散、越来越开放、越来越强调互操作性。

1.1.2 软件的生命周期

软件生命周期是指软件从概念形成开始，经过开发、使用和维护，直到最终被废弃的整个过程。

生命周期方法是从时间上对软件开发和维护的复杂问题进行分解，把软件生命的漫长周期依次分为若干阶段，每个阶段有相对独立的任务，然后逐步完成每个阶段的任务。前阶段任务的完成是后阶段工作开始的前提与基础，后阶段任务是前阶段的具体与细化。各阶段分工协作，从而降低开发工作的难度，便于科学组织与管理，保证了产品的质量，提高了软件的可维护性。

软件生命周期一般包括如下3个部分：软件定义、软件开发以及软件使用与维护。软件生命周期也可分为制定计划、需求分析定义、软件设计、程序编码、软件测试、软件运行、软件维护和软件停用8个阶段，如图1-1所示。

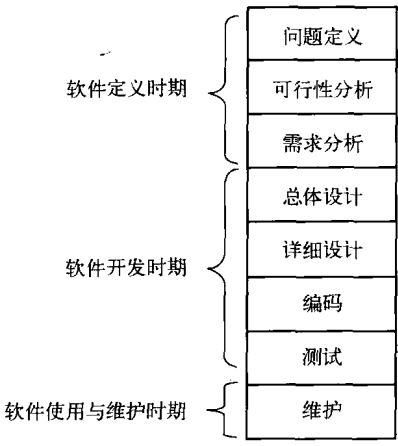


图 1-1 软件生命周期的阶段划分

软件生存周期每个阶段的任务如下所述。

1. 软件的定义

软件的定义包括问题定义、可行性分析以及需求分析。问题定义是通过对系统实际用户和使用部门负责人访问调查，明确要解决的问题的性质、工程目标和规模。可行性分析的任务是了解用户的要求及实现环境，从技术、经济和社会等多个方面研究并论证软件系统的可行性。

需求分析的任务是确定所要开发软件的功能需求、性能需求和运行环境约束，编制软件需求规格说明、确认软件系统的测试准则。软件的性能需求包括软件的适应性、安全性、可靠性和可维护性等。

2. 软件开发

软件开发是按照需求规格说明的要求，从抽象到具体，逐步生成软件的过程。软件开发

一般由设计、编码和测试等多个阶段组成。其中，总体设计又称为框架设计，主要是设计软件的结构，确定软件的模块构成及模块之间的关系。详细设计是把解决问题的方案具体化，设计出程序的详细规格说明，并给出细节信息。

编码和测试阶段，主要是将详细设计文档翻译成高级程序设计语言的源程序，通过测试与调试，使软件达到预定的要求。

3. 软件使用与维护

软件通过测试后，安装在用户确定的运行环境中，移交给用户使用，并对软件系统进行修改或对软件需求变化做出维护。

1.2 软件过程

软件工程是应用计算机科学、数学及管理科学等原理开发软件的工程。通俗地说，软件工程是实现一个大型程序的一套原则方法，即按工程化的原则和方法组织软件开发工作。

软件过程作为软件工程的核心问题，是指为了获得高质量软件所需要完成的一系列任务的框架，它规定了完成各项任务的工作步骤。由于软件项目的性质不同，因此软件过程模型也不相同。下面介绍当前较为流行的软件过程。其中，RUP 是当前企业广泛应用的软件过程模型，采用迭代增量、以架构为中心和用例驱动的软件开发方法。而敏捷过程作为轻量级开发过程，具有 5 个价值观：沟通、简单化、反馈、勇气、谦逊。

1.2.1 RUP

Rational 统一过程（Rational Unified Process, RUP），是 Rational 公司（现归属 IBM 公司）推出的一种软件过程产品，RUP 以统一建模语言（UML）描述软件开发过程，具体如下所述。

1. RUP 各个阶段

RUP 中的软件生命周期在时间上被分解为 4 个顺序的阶段，分别是初始阶段、细化阶段、构建阶段和交付阶段。每个阶段结束于一个主要的里程碑；每个阶段本质上是两个里程碑之间的时间跨度。在每个阶段的结尾执行一次评估，以确定这个阶段的目标是否已经满足，如果评估结果满意，则允许项目进入下一个阶段。其中，每个阶段又可以进一步分解迭代。一个迭代是一个完整的开发循环，产生一个可执行的产品版本，作为最终产品的一个子集，产品增量式地发展，从一个迭代过程到另一个迭代过程直到成为最终的系统。

RUP 的过程可用二维坐标来描述，如图 1-2 所示。横轴以时间来组织，是过程展开的生命周期特征，体现了开发过程的动态结构，用来描述它的术语主要包括周期、阶段、迭代和里程碑；纵轴是以内容来组织的逻辑活动，体现了开发过程的静态结构，用来描述它的术语主要包括活动、产物、工作者和工作流。

下面依次介绍 RUP 软件生命周期的 4 个阶段：

（1）初始阶段

初始阶段的目标是为系统建立商业案例，并确定项目的边界。为了达到该目的，必须识别所有与系统交互的外部实体，并在较高层次上定义交互的特性。这包括识别出所有用例并

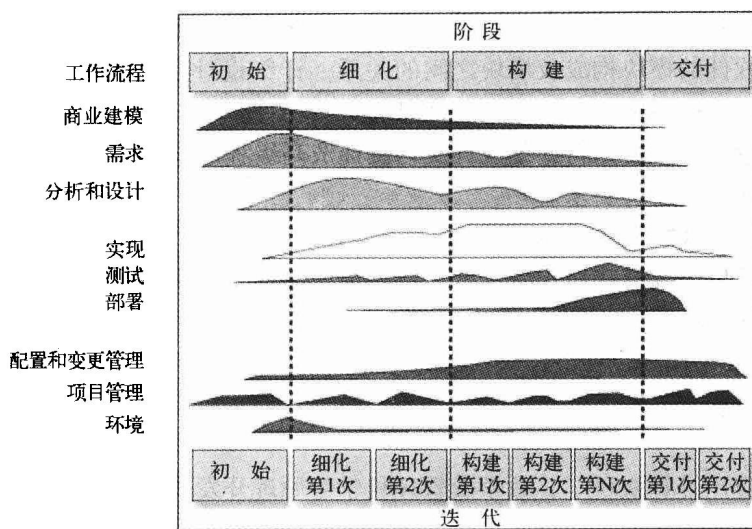


图 1-2 RUP 的过程图

描述几个重要的用例。本阶段具有非常重要的意义，在这个阶段中所关注的是整个项目进行中的业务和需求方面的主要风险。对于建立在原有系统基础上的开发项目来讲，初始阶段可能很短。

在初始阶段应该取得如下成果：

- 1) 蓝图文档，即关于项目的核心需求、关键特性和主约束的总体蓝图。
- 2) 初始的用例模型（约占总体的 10% ~ 20%）。
- 3) 初始的项目术语表。
- 4) 初始的商业案例，包括商业环境和验收标准等。
- 5) 初始的风险评估。
- 6) 项目计划。
- 7) 一个或多个原型。

(2) 细化阶段

细化阶段的目标是分析问题领域，建立坚实的体系结构基础、制定项目计划、消除项目中最高风险的因素。为了达到该目的，必须在理解整个系统的基础上，对体系结构作出决策，包括其范围、主要功能和诸如性能等非功能需求。同时为项目建立支持环境，包括创建开发案例，创建模板、工作准则并准备工具。

细化阶段的成果包括：

- 1) 用例模型，所有的用例和参与者都已被识别出，并完成大部分的用例描述。
- 2) 补充非功能性要求以及与特定用例没有关联的需求。
- 3) 软件体系结构的描述。
- 4) 可执行的软件原型。
- 5) 修订过的风险清单和商业案例。
- 6) 整个项目的开发计划，该开发计划应体现迭代过程和每次迭代的评价标准。
- 7) 更新的开发案例。

8) 初步的用户手册。

(3) 构建阶段

在构建阶段，组件和应用程序的其余功能被开发并集成成为产品，所有的功能都被彻底地测试。从某种意义上说，构建阶段是一个制造过程，其重点为管理资源及控制运作，从而降低成本、加速进度和优化质量。

构造阶段的成果是可以交付给最终用户的产品，它至少包括：

1) 集成于适当操作系统平台上的软件产品。

2) 用户手册。

3) 当前版本的描述。

(4) 交付阶段

交付阶段的重点是确保软件对最终用户是可用的。交付阶段可以跨越几次迭代，包括为发布做准备的产品测试，基于用户反馈的少量的调整。在这一阶段，用户反馈应主要集中在产品调整，设置、安装和可用性问题。

2. RUP 核心工作流

RUP 共有 9 个核心工作流，分为 6 个核心过程工作流和 3 个核心支持工作流。6 个核心过程工作流可能使人想起传统瀑布模型中的几个阶段，但应注意迭代过程中的阶段是完全不同的，这些工作流在整个生命周期中被一次又一次地执行。

(1) 商业建模

商业建模工作流描述了如何为新的目标组织开发一个构想，并基于这个构想在商业用例模型和商业对象模型中定义组织的过程、角色和责任。

(2) 需求

需求工作流的目标是描述系统应该做什么，并使开发人员和用户就这一描述达成共识。为了达到该目标，要对需要的功能和约束进行提取、组织、文档化，最重要的是理解系统所解决问题的定义和范围。

(3) 分析和设计

分析和设计工作流是将需求转化成系统的设计，为系统开发一个健壮的结构，使其与实现环境相匹配，设计活动以体系结构设计为中心，其结果是一个设计模型和一个可选的分析模型。设计模型是源代码的抽象，由设计类和一些描述组成。设计类被组织成具有良好接口的设计包和设计子系统，而描述则体现了类的对象如何协同工作实现用例的功能。

(4) 实现

实现工作流的目的包括以层次化的子系统形式定义代码的组织结构，以组件的形式（如源文件、二进制文件和可执行文件等）实现类和对象，将开发出的组件作为单元进行测试，集成单元使其成为可执行的系统。

(5) 测试

测试从可靠性、功能性和系统性的三维模型来进行。测试工作流要验证对象间的交互作用，验证软件中所有组件的正确集成，检验所有的需求已被正确地实现，识别并确认缺陷在软件部署之前被提出并处理。RUP 提出了迭代的方法是在整个项目中进行测试，从而尽早地发现缺陷，从根本上降低修改缺陷的成本。