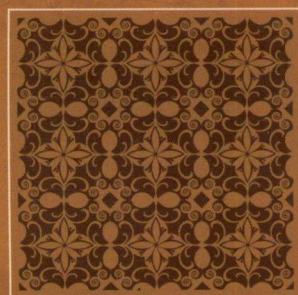


主编 窦万峰 蒋锁良 杨俊
参编 潘媛媛 汤傲

软件工程 实验教程

第3版



*S*oftware Engineering
Experiments Tutorial



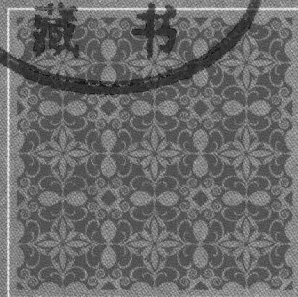
机械工业出版社
China Machine Press

高等院校计算机课程案例教程系列

主编 窦万峰 蒋锁良 杨俊
参编 潘媛媛 汤傲

软件工程 实验教程

第3版



*S*oftware Engineering
Experiments Tutorial



机械工业出版社
China Machine Press

图书在版编目 (CIP) 数据

软件工程实验教程 / 窦万峰, 蒋锁良, 杨俊主编. —3 版. —北京: 机械工业出版社, 2016.10

(高等院校计算机课程案例教程系列)

ISBN 978-7-111-55142-3

I. 软… II. ①窦… ②蒋… ③杨… 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 (2016) 第 252210 号

本书讲解软件工程中的典型 CASE 工具, 将概念和理论融入实践, 引导读者深入理解软件开发各个阶段的技术, 掌握工具的使用方法。本书共 10 章, 设计了 47 个实验, 覆盖了软件工程文档、软件分析与设计建模工具 Microsoft Visio、面向对象建模工具 Rational Rose、软件配置管理工具 Microsoft Visual SourceSafe、功能测试工具 WinRunner、压力测试工具 LoadRunner、单元测试工具 JUnit、软件代码测试工具 PurifyPlus、测试管理工具 TestDirector 和软件项目管理工具 Microsoft Project 等。每章既包含对工具功能的介绍, 也安排了针对不同侧重点的实验, 以便教师根据课时灵活选取。

本书适合作为高等院校软件工程课程的实验教材, 也可供工程技术人员阅读参考。

出版发行: 机械工业出版社 (北京市西城区百万庄大街 22 号 邮政编码: 100037)

责任编辑: 曲 熠

责任校对: 殷 虹

印 刷: 北京诚信伟业印刷有限公司

版 次: 2016 年 11 月第 3 版第 1 次印刷

开 本: 185mm×260mm 1/16

印 张: 16.75

书 号: ISBN 978-7-111-55142-3

定 价: 39.00 元

凡购本书, 如有缺页、倒页、脱页, 由本社发行部调换

客服热线: (010) 88378991 88361066

投稿热线: (010) 88379604

购书热线: (010) 68326294 88379649 68995259

读者信箱: hzjsj@hzbook.com

版权所有·侵权必究

封底无防伪标均为盗版

本书法律顾问: 北京大成律师事务所 韩光 / 邹晓东

软件工程学是一门综合性应用科学,它将计算机科学理论与现代工程方法论相结合,着重研究软件过程模型、设计方法以及工程开发技术和工具,以指导软件的生产和管理。随着计算机科学和软件产业的迅猛发展,软件工程学已成为一个重要的计算机分支学科,也是一个异常活跃的研究领域,新方法、新技术不断涌现。软件工程是计算机科学与技术专业学生必修的一门专业课程,也是工科各专业学生在计算机应用方面的一门重要选修课程。

软件工程理论与技术的发展和多种多样的 CASE (计算机辅助软件工程) 工具的不断涌现,加快了软件开发效率并节约了成本,同时也对软件及其相关行业的从业人员提出了更高的要求。为培养高素质的专业人才,应充分重视软件工程课程的实验教学,因此迫切需要优秀的软件工程课程实验教材。

为编写本书,我们在实验内容的选择、实验步骤的设计以及实验方法和文档的组织等方面做了精心的考虑和安排,结合长期的教学经验、工程项目实践经验以及各种 CASE 工具的使用经验,并与实验教学人员和专业老师反复讨论,最终才有了本书的诞生。我们参考了国内外大量的实验教材,并结合软件课程教学的特点,注重基础性、系统性、实用性和新颖性,充分体现实验的可操作性。这对推动软件工程课程的教学发展、帮助学生切实掌握软件工程课程的知识、理论与实践具有重要意义。

通过对 CASE 工具的学习和使用,软件工程实验将概念和理论知识融入实践,帮助学生深入理解软件开发中各个阶段的技术、方法和工具的基本使用方法。实验内容几乎包含了软件生存周期的各个阶段,包括软件分析与设计、软件测试和软件项目管理。全书共设计了 47 个实验,涵盖软件工程标准文档、软件分析与设计建模工具 Microsoft Visio、面向对象建模工具 Rational Rose、软件配置管理工具 Microsoft Visual SourceSafe、功能测试工具 WinRunner、压力测试工具 LoadRunner、单元测试工具 JUnit、软件代码测试工具 PurifyPlus、测试管理工具 TestDirector 和软件项目管理工具 Microsoft Project 等。部分章节安排了多个实验,目的是便于教师合理安排实验时间,也便于学生练习和操作。为了帮助学生充分理解每个实验,掌握主流软件工程工具的基本使用方法,我们给出了相关背景知识,包括实验工具的特点、功能、安装方法等。

实验内容和教学目标

第 1 章介绍软件工程 CASE 工具与标准化文档。通过学习本章,学生应熟悉软件生存周期模型及各个阶段的过程;熟悉软件工程的技术支持环境、软件工程工具以及支持环境对于软件开发和开展软件工程实践的意义;了解软件开发过程对软件工程工具和支持环境的要求;了解现有的主流软件工具及其基本功能、用途、发展与应用状况;掌握软件过程的阶段划分和各个阶段的任务;了解软件工程国家标准和其他相关技术标准的专业网站;熟悉软件工程标准化的概念、内容及意义;重点熟悉国家标准 GB/T 8567—2006 及其相关软件文档的目的、基本内容、编写要求、管理要求和适用阶段。本章安排了 2 个实验,内容涵盖软件工程

CASE 工具和软件工程标准化文档。

第2章介绍软件分析与设计工具 Microsoft Visio。通过学习本章，学生应了解 Visio 工具的功能特色和工作环境；掌握 Visio 工具的基本使用方法和基本绘图操作；了解 Visio 图形应用的基本概念；重点掌握 Visio 提供的网络图、流程图、数据流图和 UML 模型图的绘制方法；熟悉 Visio 工具的绘图操作以及如何将其与 Microsoft Office 文档结合使用。本章安排了4个实验，内容涵盖数据流图、状态模型、程序流程图和实体关系模型。

第3章介绍面向对象建模工具 Rational Rose。通过学习本章，学生应熟悉面向对象技术和面向对象软件工程模型与建模技术；掌握面向对象的统一过程模型和基本模型视图；熟悉 UML 的各种模型视图及其表示；了解面向对象建模工具 Rational Rose 的基本使用方法；了解 Rational Rose 的双向工程，包括正向工程和逆向工程，正向工程是从模型到代码，逆向工程是从代码到模型。本章安排了10个实验，内容涵盖创建用例模型、逻辑模型、动态行为模型、组件模型、部署模型、数据库模型以及正向工程和逆向工程等。

第4章介绍软件配置管理工具 Microsoft Visual SourceSafe。通过学习本章，学生应了解软件配置管理的基本概念、分类、工具集成和相关技术；了解配置管理工具 VSS 的功能和基本使用方法。本章安排了1个实验，即使用 VSS 构建项目配置环境。

第5章介绍功能测试工具 WinRunner。通过学习本章，学生应熟悉软件测试的测试任务、测试原理、常用测试策略、测试方法和测试技术；熟悉一些特定应用系统软件（面向对象软件、人机界面软件、分布式软件、实时系统软件）的测试要点；了解软件自动化测试的原理和方法；了解自动测试工具的类型、测试步骤和自动测试用例设计基础；了解测试自动化的优点和限制；掌握 WinRunner 功能测试工具的基本使用方法；使用 WinRunner 测试一个小软件并学习同步点测试；使用 WinRunner 进行多项数据驱动测试；了解 WinRunner 检查点测试；了解手工和自动合并脚本文件。本章安排了9个实验，内容涵盖录制脚本、学习 GUI 对象、同步点测试、数据驱动测试、GUI 对象检查点、图像检查点、文字检查点和批处理测试等。

第6章介绍性能测试工具 LoadRunner。通过学习本章，学生应了解如何定义性能测试要求，例如并发用户的数量、典型业务流程和所需响应时间；学会创建 Vuser 脚本，将最终用户活动捕获到自动脚本中；使用 LoadRunner Controller 设置测试环境和定义场景；通过 LoadRunner Controller 驱动和管理测试场景；通过 LoadRunner Controller 监控测试场景；使用 LoadRunner Analysis 创建图和报告并评估性能；使用 LoadRunner 联机图，指定场景执行期间 Controller 将监控的计算机，并查看监控器收集的数据；了解在场景执行期间如何监控资源，确定特定计算机上出现瓶颈的原因；学会使用 LoadRunner 的服务器资源监控器，跟踪场景执行期间使用的资源等。本章安排了8个实验，包括录制脚本、脚本回放、增强脚本、内容检查、场景设置、运行场景和结果分析等。

第7章介绍单元测试工具 JUnit。通过学习本章，学生应理解 JUnit 的目的、概念和设计模式；学会在 Eclipse 环境中加载 Junit；了解 Junit 的测试原理和测试框架；掌握 Junit 的测试方法和过程。本章安排了1个实验，即类的测试。

第8章介绍代码测试工具 PurifyPlus。通过学习本章，学生应学会使用 PureCoverage 检测代码覆盖程度，自动检测测试完整性和那些无法达到的部分；学会使用 Purify 检测内存错误和内存泄漏，以确保整个应用程序的质量和可靠性；学会使用 Quantify 检测代码性能瓶颈，自动检测出影响程序段执行速度的瓶颈，获取参数分析表；学会利用 PurifyPlus 强有力的数据收集和分析能力，最大化地利用测试生成的数据，更好地辅助测试人员进行决策。本章安排了6个实验，内容涵盖 PureCoverage 单元测试、Purify 单元测试、Quantify 单元测试、

精确粒度数据采集、可定制过滤器的生成和数据的合并与比较等。

第9章介绍测试管理工具 TestDirector。通过学习本章，学生应了解测试管理的概念和目的；了解测试项目环境的构建、用户添加和授权；掌握测试需求定义、计划测试、定义测试和执行测试。本章安排了5个实验，内容涵盖创建测试项目、定制测试项目、创建项目需求大纲、创建测试集合和执行测试。

第10章介绍软件项目管理工具 Microsoft Project。通过学习本章，学生应了解 IT 项目管理的基本概念、意义和作用；熟悉项目的核心思想和基本知识；掌握项目管理软件 Microsoft Project 的功能、用途和基本操作。本章安排了1个实验，即构建项目计划。

实验安排

本书适合作为高等院校软件工程课程的实验辅助教材，也可以作为独立开设的软件工程实验课程的教材。本书实验内容的基本概念来自软件工程课程，应与之结合学习。本书内容结构合理，章节组织有特色，应用指导性强，在实施过程中可以结合一些应用实例，以达到更好的教学效果。

本实验教程的47个实验可以根据课时需要任意组合，建议总课时为32课时。对于初级教学目标，可选每个工具的基本实验，共计36个实验，总计18课时。对于课时紧张的情况，我们在每个实验中安排了基本实例和综合应用，教师可以灵活选用。

本书第1~5章由窦万峰编写，第6章由蒋锁良编写，第7~8章由潘媛媛和窦万峰编写，第8章由杨俊编写，第9~10章由汤傲和窦万峰编写。全书由窦万峰统稿，由潘媛媛和汤傲校对。

限于编者水平，书中难免有疏漏和不当之处，敬请广大读者不吝赐教。

编者

2016年10月

目 录

前言

第 1 章 软件工程 CASE 工具与

标准化文档.....1

- 1.1 软件工程 CASE 工具.....1
- 1.2 软件工程国家标准.....6
- 1.3 软件工程文档的分类.....7
- 1.4 实验内容与方法.....9
- 1.5 实验安排说明.....17
- 1.6 小结.....18
- 1.7 习题.....18

第 2 章 软件分析与设计工具.....19

- 2.1 引言.....19
- 2.2 Visio 的基本使用方法.....20
 - 2.2.1 初识 Visio 绘图.....20
 - 2.2.2 Visio 提供的模型.....21
- 2.3 数据流图.....22
- 2.4 状态模型.....24
- 2.5 程序流程图.....27
- 2.6 实体关系模型.....28
- 2.7 实验安排说明.....30
- 2.8 小结.....30
- 2.9 习题.....30

第 3 章 面向对象建模工具.....31

- 3.1 引言.....31
- 3.2 Rational Rose 的基本使用方法.....31
 - 3.2.1 Rational Rose 的主界面.....31
 - 3.2.2 Rational Rose 基本用法简介.....33
- 3.3 创建用例模型视图.....37

3.3.1 创建用例视图.....37

3.3.2 创建用例的活动图或泳道图.....41

- 3.4 创建逻辑模型.....42
- 3.5 创建动态行为模型.....46
- 3.6 创建逻辑结构——包依赖模型.....50
- 3.7 创建组件模型视图.....51
- 3.8 创建部署模型视图.....53
- 3.9 数据库建模.....53
- 3.10 双向工程.....63
 - 3.10.1 ANSI C++ 的 Rose 双向工程.....64
 - 3.10.2 VC++ 的 Rose 双向工程.....67
 - 3.10.3 Java 的 Rose 双向工程.....79
- 3.11 实验安排说明.....81
- 3.12 小结.....82
- 3.13 习题.....82

第 4 章 软件配置管理工具.....84

- 4.1 引言.....84
- 4.2 VSS 简介.....84
- 4.3 VSS 的基本使用方法.....87
- 4.4 实验安排说明.....95
- 4.5 小结.....95
- 4.6 习题.....96

第 5 章 功能测试工具.....97

- 5.1 引言.....97
- 5.2 WinRunner 的测试过程.....97
- 5.3 WinRunner 的基本使用方法.....98
 - 5.3.1 WinRunner 的基本操作.....98
 - 5.3.2 WinRunner 的测试方法.....100

5.4 识别 GUI Map 对象	104	6.4.4 结果分析	166
5.5 功能测试方法	107	6.5 实验安排说明	173
5.5.1 使用向导识别对象	108	6.6 小结	173
5.5.2 插入同步点	111	6.7 习题	173
5.6 学习 GUI 对象	115		
5.6.1 GUI Map file per test 模式	115	第 7 章 单元测试工具	174
5.6.2 更新 GUI Map 文件	115	7.1 JUnit 简介	174
5.6.3 合并 GUI Map 文件	118	7.2 JUnit 设计模式	174
5.6.4 配置 GUI Map	120	7.2.1 命令模式	174
5.7 数据驱动测试	122	7.2.2 组合模式	175
5.8 WinRunner 检查点测试	126	7.2.3 模板方法模式	177
5.8.1 GUI 对象检查点	126	7.2.4 适配器模式	179
5.8.2 图像检查点	129	7.2.5 观察者模式	180
5.8.3 文字检查点	130	7.2.6 装饰模式	182
5.9 脚本修改	132	7.3 在 Eclipse 下为项目添加	
5.10 WinRunner 批测试	135	JUnit4 库	184
5.11 实验安排说明	138	7.4 JUnit4 单元测试框架	188
5.12 小结	138	7.5 Eclipse 单元测试	189
5.13 习题	138	7.5.1 Eclipse 单元测试的基本	
		过程	189
第 6 章 软件性能测试工具	139	7.5.2 JUnit4 测试原理	193
6.1 引言	139	7.6 JUnit 高级功能	195
6.2 LoadRunner 的基本使用方法	139	7.6.1 高级 Fixture	195
6.2.1 LoadRunner 的基本流程	141	7.6.2 限时测试	195
6.2.2 VuGen 简介	141	7.6.3 测试异常	196
6.2.3 协议选择	142	7.6.4 Runner	196
6.3 LoadRunner 的测试过程与方法	143	7.6.5 参数化测试	196
6.3.1 录制脚本	143	7.6.6 打包测试	198
6.3.2 脚本回放	146	7.7 实验安排说明	199
6.3.3 增强脚本	148	7.8 小结	199
6.3.4 关联	153	7.9 习题	199
6.3.5 内容检查	155		
6.3.6 集合点	158	第 8 章 代码测试工具	200
6.3.7 设置运行时行为	158	8.1 引言	200
6.4 场景设计	159	8.1.1 PureCoverage 工具的特性	200
6.4.1 Controller 简介	160	8.1.2 Purify 工具的特性	202
6.4.2 场景设置	160	8.1.3 Quantify 工具的特性	205
6.4.3 运行场景	164	8.2 PurifyPlus 单元测试	206

8.3 PurifyPlus 高级测试.....	213	9.7.1 建立测试集合	232
8.3.1 PurifyPlus 的高级特性.....	213	9.7.2 测试执行表	233
8.3.2 精确粒度数据的采集	213	9.7.3 执行测试过程	234
8.3.3 可定制过滤器的生成	214	9.7.4 分析测试结果	236
8.3.4 采集数据的合并与比较	214	9.8 管理软件缺陷	238
8.4 PurifyPlus 的高级功能	215	9.8.1 记录缺陷	238
8.5 实验安排说明	216	9.8.2 检查新缺陷	239
8.6 小结	217	9.9 实验安排说明	240
8.7 习题	217	9.10 小结	240
		9.11 习题	240
第 9 章 测试管理工具.....	218		
9.1 引言	218	第 10 章 软件项目管理工具.....	241
9.2 测试管理过程	218	10.1 引言	241
9.3 TestDirector 的基本使用方法.....	220	10.2 Microsoft Project 简介	241
9.4 建立测试项目	223	10.3 Microsoft Project 视图	242
9.4.1 创建项目	223	10.4 Microsoft Project 操作实验	248
9.4.2 定制项目	224	10.5 实验安排说明	253
9.5 定义测试需求	226	10.6 小结	253
9.5.1 定义测试范围	226	10.7 习题	254
9.5.2 建立需求大纲	227		
9.6 计划测试	230		
9.7 执行测试	232	参考文献.....	255

实验目录

【实验 1-1】软件工程 CASE 工具	9
【实验 1-2】软件工程标准化文档	10
【实验 2-1】订货系统的数据流图	22
【实验 2-2】电子表状态模型	25
【实验 2-3】程序流程图	28
【实验 2-4】实体关系模型	29
【实验 3-1】电梯系统用例模型	38
【实验 3-2】POS 机系统用例模型	39
【实验 3-3】电梯系统类模型	43
【实验 3-4】ATM 机取款用例的逻辑模型	45
【实验 3-5】电梯系统动态模型	48
【实验 3-6】ATM 机系统组件模型	52
【实验 3-7】ATM 机系统部署模型	53
【实验 3-8】图书馆系统的数据库模型	59
【实验 3-9】图书馆系统正向工程	71
【实验 3-10】交互绘图系统逆向工程	78
【实验 4-1】构建项目配置环境	88
【实验 5-1】机票预订系统——录制脚本	108
【实验 5-2】机票预订系统——学习 GUI 对象	108
【实验 5-3】机票预订系统——同步点测试	112
【实验 5-4】飞机订票系统数据驱动测试	123
【实验 5-5】检查订单	127
【实验 5-6】检查屏幕区域图像	129
【实验 5-7】检查卖出的票数	130
【实验 5-8】使用 TSL 检查机票总金额	133
【实验 5-9】批执行多个脚本	135
【实验 6-1】录制航班订票过程脚本	143
【实验 6-2】确认订票流程	146

【实验 6-3】插入事务 148

【实验 6-4】用户名的参数化 151

【实验 6-5】添加检查点 155

【实验 6-6】设置手动场景 160

【实验 6-7】运行场景 164

【实验 6-8】测试网上订票系统 169

【实验 7-1】计算器类的测试 189

【实验 8-1】PureCoverage 单元测试 206

【实验 8-2】Purify 单元测试 209

【实验 8-3】Quantify 单元测试 210

【实验 8-4】采集精确粒度数据 215

【实验 8-5】生成 Purify 可定制过滤器 216

【实验 8-6】采集数据的合并与比较 216

【实验 9-1】创建测试项目 223

【实验 9-2】定制测试项目 224

【实验 9-3】建立项目需求大纲 227

【实验 9-4】建立测试集合 232

【实验 9-5】执行测试 234

【实验 10-1】面对面结对编程系统进度计划安排 248

软件工程 CASE 工具与标准化文档

1.1 软件工程 CASE 工具

从功能的角度看,软件是一种产品,表达了由计算机硬件体现的计算机潜能。软件是一个信息转换器——产生、管理、获取、修改、显示或转换信息。从软件的类型看,软件是开发和运行产品的载体。它是计算机控制(如操作系统)、信息通信(如网络)以及创建和控制其他程序(如软件工具和环境)的基础。

软件的定义为:软件=程序+数据+文档。程序是按事先设计的功能和性能需求执行的指令序列。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、维护和使用有关的图文材料。软件是逻辑的而不是物理的,本质上存在着复杂性、一致性、易变性和不可见性等固有特性。

软件开发环境

软件开发环境是面向软件整个生存周期,为支持各个阶段的需要而在基本硬件和宿主软件的基础上使用的一组软件系统,也称作软件工程环境(Software Engineering Environment, SEE)。SEE是实现软件生产工程化的重要基础。它建立在先进软件开发方法的基础上,正影响和改变着软件生产方式,反过来又进一步促进了软件开发方法的推广与流行。SEE包括生产软件系统所需要的过程、方法和自动化的集合。建立开发环境首先要确定一种开发过程模型,提出成套的、有效的开发方法,然后在这一基础上利用各种软件工具实现开发活动的自动化。SEE有一套包括数据集成、控制集成和界面集成的集成机制,让各个工具使用统一的规范存取环境信息库,采用统一的用户界面,同时为各个工具或开发活动之间的通信、切换、调度和协同工作提供支持。SEE用于辅助软件开发、运行、维护和管理等各种活动,是一个软件工具集(或工具包)。这不仅意味着SEE支持开发功能的扩大,也反映了工具集成化程度的提高。软件工具是指为支持软件生存周期中某一阶段(如需求分析、系统定义、设计、编码、测试或维护等)的需要而使用的软件系统。软件设计的理论、模型、方法论、表示法上的研究成果构成软件工具的重要基础,因此,软件工具的研制应该与整个软件工程的理论方法紧密结合起来。软件工具的另一个基础是使用计算机的许多先进技术,包括编译技术、数据库技术、人工智能技术、交互图形技术和VLSI技术等。

软件工具应具有较强的通用性,不依赖于某一实现环境、某一高级语言和某种设计方法。一般来说,越是基础的、成熟的工具,往往通用性越好;然而对于一些与软件开发方法有关的软件工具,则往往专用程度较高。软件工具通用性的要求应该根据工具的特点和用户的情况全面考虑。

软件 CASE 工具

计算机辅助软件工程(Computer Aided Software Engineering, CASE)是通过一组集成

化的工具，辅助软件开发者实现各项活动的全部自动化，使软件产品在整个生存周期中的开发和维护生产率得到提高、质量得到保证。CASE 环境、CASE 工具、集成化 CASE (I-CASE) 等，实际是一切现代化软件工程环境 SEE 的代名词。

CASE 环境的组成构件如图 1-1 所示。CASE 环境应具有以下功能：

- 提供一种机制，使环境中的所有工具可以共享软件工程信息。
- 每一个信息项的改变，可以追踪到其他相关信息项。
- 对所有软件工程信息提供版本控制和配置管理。
- 对环境中的任何工具可进行直接的、非顺序的访问。
- 在标准的分解结构中提供工具和数据的自动支持。
- 使每个工具的用户共享人机界面的所有功能。
- 收集能够改善过程和产品的各项度量指标。
- 支持软件工程师们之间的通信。

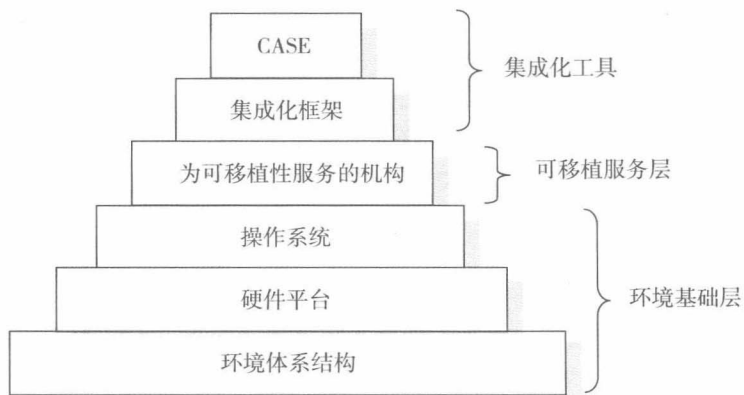


图 1-1 CASE 环境的组成构件

CASE 工具的分类标准及特点

CASE 工具的分类标准可分为 3 种：

- 功能。功能是对软件进行分类的最常用标准。
- 支持的过程。根据支持的过程，工具可分为设计工具、编程工具、维护工具等。
- 支持的范围。根据支持的范围，工具可分为窄支持工具、较宽支持工具和一般支持工具。窄支持指支持过程中特定的任务，较宽支持指支持特定过程阶段，一般支持指支持覆盖软件过程的全部阶段或大多数阶段。

根据 CASE 系统对软件过程的支持范围，CASE 系统可分为 3 类：

- 支持单个过程任务的工具。工具可能是通用的，或者也可能归组到工作台。
- 工作台支持某一过程所有活动或某些活动。它们一般以或多或少的集成度组成工具集。
- 环境支持软件过程所有活动或至少大部分活动。它们一般包括几个不同的工作台，将这些工作台以某种方式集成起来。

CASE 方法与其他方法相比有如下几方面的应用特点：

- 解决了从客观世界对象到软件系统的直接映射问题，强有力地支持软件、信息系统开发的全过程。
- 使结构化方法更加实用。

- 自动检测的方法提高了软件的质量。
- 使原型化方法和面向对象（Object Oriented，OO）方法付诸实施。
- 简化了软件的管理和维护。
- 加速了系统的开发过程。
- 使开发者从大量的分析设计图表和程序编写工作中解放出来。
- 使软件的各部分能重复使用。
- 产生统一的、标准化的系统文档。

主流 CASE 工具

软件工程 CASE 工具多种多样，可以支持软件工程不同阶段的工作。表 1-1 给出了软件工程 CASE 工具的分类和支持的工作阶段。可以看出，现有的软件工程 CASE 工具几乎涵盖了所有开发阶段的过程活动。

表 1-1 CASE 工具及其分类

工具类型	工具示例	支持的开发阶段
编辑工具	字处理器、文本编辑器、图表编辑器	软件开发全过程
编写文档工具	页面输出程序、图像编辑器	软件开发全过程
规划与估算工具	PERT 工具、估算工具、电子表格工具	软件开发全过程
变更管理工具	需求跟踪工具、变更控制系统	软件开发全过程
方法支持工具	设计编辑器、数据字典、代码生成器	描述与设计
原型建立工具	高端语言、用户界面生成器	描述、测试、有效性验证
语言处理工具	编译器、解释器	设计、实现
配置管理工具	版本管理系统、细节建立工具	设计、实现
程序分析工具	交叉索引生成器、静态 / 动态分析器	实现、测试、有效性验证
测试工具	测试数据生成器、文件比较器	实现、测试、有效性验证
调试工具	交互式调试系统	实现、测试、有效性验证
再工程工具	交叉索引系统、程序重构系统	实现

下面简要介绍一些常用的软件工程 CASE 工具。

Microsoft Visio。Microsoft Visio 提供了日常使用中的绝大多数框图的绘画功能，包括信息领域的各种原理图、设计图等，同时提供了部分信息领域的实物图。Visio 的引入之处在于其使用方便，安装后的 Visio 2003 既可以单独运行，也可以在 Word 中作为对象插入，与 Microsoft Office 2003 集成良好，其图生成后在没有安装 Microsoft Visio 的 Office 工具中仍然能够查看。Visio 在处理框和文字上流畅，同时在文件管理上提供了分页、分组的管理方式。Visio 支持 UML 静态建模和动态建模，对 UML 的建模提供了单独的组织管理。作为 Visio Office 大家庭的一员，它从 2000 版本后在各种器件模板上有许多增进。它是最通用的图表设计软件，易用性高，特别是对于不善于自己构造图的软件人员。

SmartDraw。SmartDraw 提供了大量的模版，以目录树的形式放在左边。图形设计都可以纳入模板，并且可以在某个目录里组织。SmartDraw 有许多 Visio 没有的方便功能，如插入表格、专业图、表设计、制作、管理、转换软件，可以轻松设计、制作、管理、转换各种图表、剪辑画、实验公式、流程图等。随带的图库中包含数万个示例、符号和形状，可以直接套用。SmartDraw 的独特之处是连接器，它具有随机移动功能、不易断线，内含多种模型，

可直接套用或修改。

SourceInsight. SourceInsight 以工程的方式管理原码, 提供非常适合再工程的浏览手段。SourceInsight 整个面板分成 3 个部分: 左边的树结构提供工程内的所有变量、函数、宏定义, 右边的视图区提供程序的阅读和编辑, 下边的显示栏显示鼠标在原码触及的函数或者变量定义。最让人感兴趣的是, SourceInsight 不仅高亮显示系统的关键字, 而且以近乎完美的版面编排使用户看程序如同看报纸, 非常美观。SourceInsight 提供函数交叉调用的分析, 并以树状的形式显示调用关系。

PowerDesigner. PowerDesigner 致力于采用基于实体关系 (entiry-relation) 的数据模型, 分别从概念数据模型 (conceptual data model) 和物理数据模型 (physical data model) 两个层次对数据库进行设计。概念数据模型描述的是独立于数据库管理系统 (Database Managemen System, DBMS) 的实体定义和实体关系定义。物理数据模型是在概念数据模型的基础上针对目标数据库管理系统的具体化。PowerDesigner 功能强大, 使用非常方便。首先它提供了概念模型和物理模型的分组, 呈现在使用区左边的是树状的概念模型和物理模型导航, 软件人员可以建立多个概念模型和物理模型, 并且以包的形式任意组织; 它提供增量的数据库开发功能, 支持局部更新, 软件人员可以在概念模型、物理模型、实际数据库三者间完成设计的同步。另外, PowerDesigner 还支持逆向工程、再工程和 UML 建模。

Telelogic System Architect. Telelogic System Architect 是一种综合性的建模解决方案, 用于提供成功地开发企业系统所必需的工具。它是唯一能够将各方面的建模, 包括业务流程建模 (BPM)、使用 UML 的组件和对对象建模、数据建模以及结构化分析和设计, 集成在一个多用户产品中的工具。

ModelMaker. ModelMaker 一个非常强大的软件工具。与所有强大而具有多面性的产品一样, ModelMaker 的复杂性也会让一个新手望而却步。ModelMaker 常被认为是一个 UML 图形工具或是 Delphi Case 工具。然而, 它比一般的图形工具和 Case 工具要快得多, 有时, 它可为使用者写一些人工智能式的代码。它是可扩展的, 支持 UML 图、设计模式、逆向生成与分解的双向代码管理工具等。

ModelMaker 的核心是: 它支持本地代码模型, 所有的类及其关联元素包括单元、图、文档及事件类型等都是模型内部的对象。ModelMaker 为活动模型提供了多种视图, 允许用户在类列表、元素列表或图集中进行操作。如果已有准备, 软件人员即可从模型中生成源代码单元, 并可由 Delphi 来进行编译, 以后生成的单元每次也可重新生成。软件人员可对各种不同的设置进行修改, 如代码注释选项、代码次序、方法使用等, 并且可为多种需求重新生成单元, 如调试代码、自动生成的大量注释代码等。

ArgoUML. ArgoUML 是一个交互式的、具有图形界面的软件设计环境, 支持面向对象软件的设计、开发和编写文档等工作。ArgoUML 是一个运行在 Java 1.2 及以上版本虚拟机之上, 且是免费和开源的 (遵循 BSD 协议) UML 制作工具。与其他 CASE 工具如 Viso 等类似。ArgoUML 的主要特征包括开放的标准 (XMI、SVG 和 PGML)、100% 的平台独立性、开发源代码、可扩展和可定制, 具有认知 (cognitive) 性等。

Rational Rose. Rational Rose 是一个完全的、具有能满足所有建模环境 (如 Web 开发、数据建模、Visual Studio 和 C++) 需求能力和灵活性的一套解决方案。Rose 允许开发人员、项目经理、系统工程师和分析人员在软件开发周期内将需求和系统的体系架构转换成代码, 消除浪费的消耗、对需求和系统的体系架构进行可视化、理解和精练。通过在软件开发周期内使用同一种建模工具可以确保更快、更好地创建满足客户需求的可扩展的、灵活的并且可靠的应用系统。Rose 2007 功能上可以完成 UML 的 9 种标准建模, 即静态建模 (包括用例

图、类图、对象图、组件图、配置图等)和动态建模(包括协作图、序列图、状态图、活动图等)。为了使静态建模可以直接作用于代码, Rose 提供了类设计到多种程序语言代码自动产生的插件。同时, 作为一款优秀的分析和设计工具, Rose 具有强大的正向工程和逆向工程能力。正向工程指由设计产生代码, 逆向工程指由代码归纳出设计。通过逆向工程, Rose 可以对历史系统分析, 然后进行改进, 再通过正向工程产生新系统的代码, 这样的设计方式称为再工程。

VSS. 微软的 Studio 企业版包含 VSS (Visual Source Safe) 版本管理工具。VSS 工具包括服务器和通过网络可以连接服务器的客户端。VSS 提供了基本的认证安全和版本控制机制, 包括 CheckIn (入库)、CheckOut (出库)、Branch (分支)、Label (标定) 等功能, 能够对文本、二进制、图形图像等几乎任何类型的文件进行控制, 提供历史版本对比, 并可以与 Studio 集成。VSS 的客户端既可以连接服务器运行, 也可以在本机运行, 非常适合于个人程序开发的版本管理。

PVCS. PVCS 是世界领先的软件开发管理工具, 市场占有率达 70% 以上, 是公认的事实上的工业标准。IDC 在 1996 年 9 月的报告中评述: “PVCS 是软件开发管理工业领域遥遥领先的领导者。” 全球的著名企业、软件机构、银行等诸多行业及政府机构几乎无一例外地应用了 PVCS。PVCS 包含多种工具, 其中 PVCSVersionManager 会完整、详细地记录开发过程中出现的变更和修改, 并使修订版本自动升级, 而 PVCSTracker、PVCS Notify 会自动地对上述变更和修改进行追踪。另外, PVCSRequisitePro 提供了一个独特的 Microsoft Word 界面和需求数据库, 从而可以使开发机构实时、直观地对来自于最终用户的项目需求及需求变更进行追踪和管理, 可有效地避免重复开发, 保证开发项目按期、按质、按原有的资金预算交付用户。PVCS 基于 Web 的应用比较方便, 只要设定用户和权限, 规划好目录结构, 项目组成员就可以通过浏览器完成操作。

ClearCase. ClearCase 是 Rational 公司的主要配置管理工具, 现在绝大多数企业已经从 PVCS 过渡到 ClearCase 来了, 其原因在于 ClearCase 是整个 Rational 产品系列中的中枢。

ClearCase 提供了 VOB 的概念来进行配置管理, 功能极其强大, 同时 ClearCase 使用起来也非常复杂。目前国内 ClearCase 流行版本在 NT 主域控制器上安装。ClearCase 的解密和安装比较复杂。ClearCase 是世界上目前较强大的配置管理工具之一, 由于它采用许多新的配置管理思想, 使得相对于传统的 CVS、VSS、PVCS 等版本管理工具, ClearCase 具有许多闪光点, 目前正在为世界上各大软件企业所使用。ClearCase 中有大量的新的术语, 其中比较重要的术语有 UCM (统一配置管理)、VOB (版本对象基础)、View (版本视图)、Activity (更新活动)。ClearCase 实现版本管理的基础是 VOB, 成员要更改受控资料, 需要先设置一个自己的 View, 这个 View 是用户感兴趣的受控资料范围, 然后可以 CheckOut 资料到本地资料区, 进行修改后再 CheckIn 提交。ClearCase 极为有力地支持多版本、并行开发。ClearCase 不仅可以提供基于文件的版本历史, 甚至可以对整个目录系统的演化进行跟踪记录。

WinRunner. WinRunner 是一种企业级的功能测试工具, 用于检测应用程序是否能够达到预期的功能及正常运行。通过自动录制、检测和回放用户的应用操作, WinRunner 能够有效地帮助测试人员对复杂的企业级应用的不同发布版进行测试, 提高测试人员的工作效率和质量, 确保跨平台的、复杂的企业级应用无故障发布及长期稳定运行。

LoadRunner. LoadRunner 是一种性能的负载测试工具, 通过模拟成千上万的用户进行并发负载及实时的性能检测来查找发现问题, 并能对整个企业架构进行测试。通过使用 LoadRunner 自动化性能测试工具, 能够最大限度地缩短测试时间, 优化性能和加速应用系

统的发布周期。LoadRunner 能支持广泛的协议和技术，功能比较强大，可以为特殊环境提供特殊的解决方案。

1.2 软件工程国家标准

软件工程的范围涉及软件的整个生存周期，包括软件可行性研究、计划制订、需求分析、设计、程序编写、测试、维护以及与其相应的组织管理工作等，这些过程都需要按一定的规范进行，才能有效控制软件进度和保证软件质量。

我国制定和推行标准化工作的总原则是向国际标准先进技靠拢，以促进国际交流。从 1983 年起，我国已陆续制定和发布了 20 余项软件工程国家标准。

这些标准可分为 4 类：基础标准、开发标准、文档标准、管理标准，如表 1-2 所示。

表 1-2 我国软件工程标准

类 型	标 准 名 称	标 准 号
基础标准	软件工程技术语	GB/T 11457—2006
	信息处理 数据流程图、程序流程图、系统流程图、程序网络图和系统资源图的文档编制符号及约定	GB 1526—1989
	软件工程标准分类法	GB/T 15538—1995
	信息处理 程序构造及其表示的约定	GB/T 13502—1992
	信息处理 单命中判定表规范	GB/T 15535—1995 (ISO 5806)
	信息处理系统计算机系统配置图符号及其约定	GB/T 14085—1993 (ISO 8790)
开发标准	信息技术软件生存期过程	GB/T 8566—2007
	软件支持环境	GB/T 15853—1995
	信息处理 按记录组处理顺序文卷的程序流程	GB/T 15697—1995 (ISO 6593: 1985)
	软件维护指南	GB/T 14079—1993
文档标准	计算机软件产品开发文档编制指南	GB/T 8567—2006
	计算机软件需求说明编制指南	GB/T 9385—2008
	计算机软件测试文档编制规范	GB/T 9386—2008
	软件文档管理指南	GB/T 16680—1996
管理标准	计算机软件配置管理计划规范	GB/T 12505—1990
	信息技术 软件产品评价 质量特性及其使用指南	GB/T 16260—2006 (ISO 14598)
	计算机软件质量保证计划规范	GB/T 12504—2008
	计算机软件可靠性和可维护性	GB/T 14394—2008

软件工程标准化

软件工程标准化涉及软件设计的标准化、文件编写的标准化和项目管理的标准化 3 个方面。软件设计的标准化包括设计方法、设计表达方法、程序结构、程序设计语言、程序设计风格、用户接口设计、数据结构设计、算法设计等。文件编写的标准化包括管理文件、项目实施计划、质量保证计划、开发进度月报、可行性分析报告、软件需求说明书、概要设计说明书、详细设计说明书、测试计划说明书、用户手册、维护手册、操作手册、源代码、生成