第一图书网, tushu007.com

<<软件工程教程>>

图书基本信息

书名：<<软件工程教程>>

13位ISBN编号：9787111300021

10位ISBN编号：7111300025

出版时间：2010-4

出版时间：陈建明、王辉、 孙涌 机械工业出版社 (2010-04出版)

作者：孙涌，等 编

页数：332

版权说明：本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：http://www.tushu007.com

前言

自20世纪60年代末期创立以来，软件工程伴随着计算机软、硬件的快速发展，经历了从结构化到面向对象的一系列发展过程，并且已经形成了若干工具、分支学科，目前软件工程仍然是一个异常活跃的研究领域。
人们已经认识到，如果有哪个项目不遵循软件工程原则，必定会受到实践的惩罚。
当然，软件工程学的研究范围非常广泛，包括技术方法、工具和管理等许多方面，新的技术方法和工具还在不断涌现。
本书集合作者多年从事本科生和研究生软件工程课程教学经验，在参考了大量国内外教材与专著的基础上，结合当前大学软件工程课程教学的实际要求和将来从事实际软件开发的相关情况而编写。
本教程作为苏州大学首批精品课程建设计划之一，是计算机专业的一门专业课。
本教材虽然不可能包含软件工程的全部内容，但却是本着易懂、实用的原则并结合多年从事软件工程教学、科研实践编写而成。
另外考虑到软件工程的发展，编入了一定量的现代软件工程的概念、方法及技术。
在写作过程中，本书重点关注了以下几个方面：?以实用为主的同时，适当加强理论的叙述，在相关章节中增加了一些形式化的内容，便于更深入地学习软件工程相关理论或指导实际的软件项目开发。
?按照一般软件产品（项目）开发过程的顺序进行软件工程内容叙述，便于进行相关实验或实践。
?面向对象技术在突出统一建模基本概念、方法的同时，强调建模与高级语言的结合与融合，避免了过程建模与具体实现脱节的问题。
这一点对于我国的学生尤为重要，其理由是我国当前绝大部分学生认为只有编写代码才能进行软件开发。
?对一些比较成熟的最新技术进行了介绍，如设计模式、Web工程等。
通过这部分内容的学习，既能够与当前实际开发紧密结合，也可为将来继续深化、研究起到一个入门的作用。
本书共15章。
第1章概括介绍了软件工程学的基本原理、概念和方法。
第2章到第6章主要介绍了面向过程软件工程生命周期顺序的前几个阶段的任务、过程、方法和工具。
第7章到第9章比较完整地介绍了面向对象的开发方法。
第10章到第13章介绍了软件工程中面向过程及面向对象的软件测试、维护、质量保证及相关的过程、方法和工具。
第15章较详细地介绍了目前软件工程的一些新技术，包括敏捷软件开发、设计模式、软件复用、Web工程等。
由于软件工程这门课程涉及面广，内容丰富，发展迅速，所以我们在取材方面，既考虑保持传统的内容，又充分将软件工程中的新技术、新发展融入其中。
同时，我们也兼顾到目前高校学生的实际情况，力争做到取材合理、内容新颖、理论为主、结合实际、重点突出、实用性强。
根据多年从事软件开发和软件工程课程教学的经验和体会，作者认为：软件工程这门课程的特点在于：它看似简单，大部分内容是叙述性的，但要真正掌握好并运用好，绝非易事！
特别是软件工程的思想、方法、理论、技术等贯穿整个软件产品（项目）开发的始终，这是任何一门课程所无法替代的。
它既有宏观的一面，又有微观、具体的一面，同时还与诸多学科相关联。
在此，希望广大学生与读者，能够在学习本书内容的同时，将相关知识与一个实际项目结合，哪怕是非常小的项目，只有这样，才有可能真正学好软件工程。
本书由孙涌主编，陈建明、王辉参编。
全书完稿后，由孙涌进行统稿和整理工作。
在本书的编写过程中，感谢王璁伟、房鹏、王晋、姜晓猛、沈文超、葛小培、陈祥荣、耿胜恩等同学为之付出的辛勤劳动，同时还要感谢机械工业出版社的编辑们对本书出版给予的支持。

由于时间仓促，加之作者水平有限，书中难免存在不足和疏漏之处，敬请广大读者不吝赐教。

内容概要

《软件工程教程》全面系统地介绍了软件工程的有关概念、原则、方法和工具。
全书共15章，内容包括：软件工程中面向过程、面向对象的开发方法，技术度量，质量保证，软件项目计划与管理，用统一建模语言UML开发软件的方法等。
另外，还对设计模式、敏捷软件开发、Web工程等软件工程相关领域进行了介绍和讨论。
《软件工程教程》既注重科学性和系统性，又注重实用性和新颖性。
《软件工程教程》可作为大学计算机及相关专业本（专）科学生的教材或教学参考书，也可作为研究生的参考教材。

书籍目录

出版者的话 前言 教学建议 第1章 软件工程概述 1 1.1 软件发展和软件危机 1 1.1.1 软件的定义和发展 1 1.1.2 软件危机过程 2 1.2 软件工程学的范畴 5 1.3 软件开发的生命周期 5 1.4 传统软件工程和面向对象软件工程 7 1.5 软件的特点 10 1.6 软件工程的基本目标 11 小结 12 习题 12 第2章 软件生命周期过程模型 13 2.1 过程及软件生命周期 13 2.2 软件过程模型 14 2.2.1 瀑布模型 15 2.2.2 具有原型化的瀑布模型 17 2.3 演化过程模型 18 2.3.1 原型化模型 18 2.3.2 螺旋模型 19 2.3.3 操作说明模型 20 2.4 增量过程模型 21 2.4.1 RAD模型 21 2.4.2 增量和迭代模型 22 2.5 其他类型的过程模型 23 2.5.1 喷泉模型 23 2.5.2 智能模型 24 2.5.3 V模型 25 2.5.4 变换模型 25 小结 26 习题 26 第3章 需求分析 27 3.1 需求分析的任务 27 3.1.1 需求定义 27 3.1.2 需求的层次 28 3.1.3 需求的开发与管理 28 3.2 需求获取技术 30 3.2.1 需求分析人员的组成 30 3.2.2 需求的类型 30 3.2.3 获取需求的途径 31 3.3 需求规格说明书 36 3.3.1 需求说明的目的 36 3.3.2 需求说明的方法 36 3.4 需求描述技术 36 3.4.1 结构化技术 36 3.4.2 形式化技术 40 3.5 需求验证 48 小结 49 习题 49 第4章 概要设计 51 4.1 概要设计的概念 51 4.1.1 概要设计的目标和任务 51 4.1.2 概要设计的过程 53 4.1.3 概要设计的工具 53 4.2 模块独立性 55 4.2.1 模块化 55 4.2.2 模块的耦合性 56 4.2.3 模块的内聚性 60 4.3 结构化设计方法 63 4.3.1 概念 63 4.3.2 变换分析 66 4.3.3 事务分析 68 4.3.4 设计的后处理 69 4.4 数据设计 70 4.4.1 数据设计的原则 70 4.4.2 数据结构设计 71 4.4.3 文件设计 71 4.4.4 数据库设计 72 小结 73 习题 73 第5章 详细设计 74 5.1 详细设计的任务 74 5.2 详细设计的方法 75 5.2.1 设计表示法 75 5.2.2 结构化程序设计 77 5.2.3 面向数据结构的设计 80 5.2.4 详细设计文档与复审 85 小结 90 习题 91 第6章 编码与语言选择 92 6.1 编码的目的和任务 92 6.2 编码所使用的语言 93 6.2.1 程序设计语言范型 93 6.2.2 程序设计语言的发展 94 6.2.3 常用的编码语言 96 6.2.4 编码语言的选择 98 6.3 编码的风格 99 小结 103 习题 103 第7章 面向对象方法 104 7.1 面向对象的基本概念 104 7.1.1 对象 104 7.1.2 类与消息 105 7.1.3 类的基本特征 106 7.2 面向对象的开发方法 106 7.2.1 概述 107 7.2.2 面向对象方法的发展历程 107 7.2.3 常用的面向对象分析的方法 108 7.3 面向对象的设计 111 7.3.1 面向对象设计概述 111 7.3.2 底层设计—类的设计 118 7.3.3 OOD的Yourdon的方法 120 7.3.4 Booch的方法 125 7.3.5 系统的设计过程 126 小结 129 习题 129 第8章 统一建模语言 130 8.1 统一建模语言简介 130 8.1.1 发展历史 130 8.1.2 UML简介 131 8.1.3 UML视图简介 132 8.1.4 视图 132 8.1.5 UML类、构件、部署和协作图 中的图标 133 8.1.6 扩展组件 134 8.1.7 各种视图间的关系 134 8.2 概念与视图 135 8.2.1 静态视图 135 8.2.2 用例图 136 8.2.3 交互视图 136 8.2.4 状态图 138 8.2.5 活动视图 139 8.2.6 物理视图 140 8.2.7 模型管理视图 142 8.3 UML与Java的对应关系 143 8.3.1 表示结构 143 8.3.2 表示关系 145 8.4 统一建模语言的综合应用 149 8.4.1 项目概述 149 8.4.2 静态分析和设计 150 8.4.3 持久对象设计 151 8.4.4 动态对象设计 152 8.4.5 通用接口设计 154 8.4.6 体系结构设计 157 小结 159 习题 160 第9章 统一软件过程 161 9.1 软件开发过程 161 9.2 迭代和递增 162 9.3 核心工作流 162 9.3.1 需求流 162 9.3.2 分析流 164 9.3.3 设计流 166 9.3.4 实现流 167 9.3.5 测试流 168 9.3.6 交付后维护 170 9.3.7 退役 170 9.4 统一过程的各阶段 171 9.4.1 开始阶段 171 9.4.2 细化阶段 173 9.4.3 构建阶段 173 9.4.4 转换阶段 174 9.5 二维生命周期模型 174 小结 174 习题 174 第10章 软件测试 176 10.1 软件测试概述 176 10.1.1 软件测试的目标 176 10.1.2 软件测试的原则 177 10.1.3 软件测试的方法 178 10.1.4 软件测试与软件开发各阶段的 关系 178 10.1.5 测试信息流 179 10.1.6 错误分类 179 10.2 软件测试过程与策略 182 10.2.1 单元测试 182 10.2.2 集成测试 183 10.2.3 确认测试 186 10.2.4 平行运行 187 10.3 设计测试方案 187 10.3.1 逻辑覆盖 188 10.3.2 等价划分 191 10.3.3 边界值分析 194 10.3.4 错误推测 194 10.3.5 实用测试策略 195 10.4 纠错 198 10.5 对OOA和OOD模型的测试 200 10.5.1 扩大测试的视角 201 10.5.2 测试OOA和OOD模型 201 10.6 面向对象的测试策略 203 10.6.1 在OO语境中的单元测试 203 10.6.2 在OO语境中的集成测试 203 10.6.3 在OO语境中的有效性测试 204 10.7 OO软件的测试用例设计 204 10.7.1 OO概念的测试用例设计的 含义 204 10.7.2 传统测试用例设计方法的 可用性 204 10.7.3 基于故障的测试 205 10.7.4 OO编程对测试的影响 205 10.7.5 测试用例和类层次 206 10.7.6 基于场景的测试设计 206 10.7.7 测试表层结构和深层结构 207 10.8 其他专门环境要求的测试 208 10.8.1 GUI测试 208 10.8.2 测试文档和帮助设施 209 10.8.3 实时系统测试 210 小结 211 习题 211 第11章 软件维护 213 11.1 系统的变化 213 11.1.1 系统的类型 214 11.1.2 系统生命周期中的变化 215 11.1.3 系统的生命范围 216 11.2 软件维护的基本内容和特点 218 11.2.1 软件维护概述 218 11.2.2 软件维护的特点 219 11.2.3 维护中的问题 220 11.3 软件维护的实施 221 11.3.1 软件维护的过程 221 11.3.2 软件维护的技术 224 11.4 软件的可维护性 224

11.4.1 软件可维护性概述 225 11.4.2 软件可维护性度量 225 11.4.3 提高可维护性的方法 225 11.5 软件维护的副作用 226 11.5.1 代码副作用 226 11.5.2 数据副作用 227 11.5.3 文档副作用 227 11.6 软件再工程 227 11.6.1 软件再工程的过程 227 11.6.2 软件再工程的方法 228 小结 228 习题 229 第12章 软件质量及其管理 230 12.1 软件质量的概念及属性 230 12.1.1 软件质量概述 230 12.1.2 软件质量的属性 230 12.2 软件质量保证与控制 231 12.2.1 软件质量保证概述 232 12.2.2 软件质量保证计划 232 12.2.3 软件质量成本 234 12.2.4 软件质量控制 234 12.3 软件质量度量 235 12.3.1 软件质量度量概述 235 12.3.2 质量度量模型 235 12.3.3 三种度量模型的比较 236 12.3.4 软件质量评价 238 12.4 软件可靠性 238 12.4.1 基本概念 238 12.4.2 影响软件可靠性的因素 239 12.4.3 软件可靠性模型 240 12.4.4 软件可靠性工程 242 12.5 CMM：软件能力成熟度模型 242 12.5.1 CMM的发展 242 12.5.2 基本概念 243 12.5.3 SW-CMM的用途 243 12.5.4 CMM的五个等级 244 12.5.5 CMM的内部结构 246 12.5.6 采用CMM的意义 248 小结 249 习题 249 第13章 软件项目管理 250 13.1 项目管理的概念 250 13.1.1 项目管理过程 250 13.1.2 项目管理的范围 251 13.1.3 项目管理中的资源 251 13.2 可行性研究 252 13.2.1 可行性研究的任务和过程 252 13.2.2 技术可行性研究 253 13.2.3 经济可行性研究 254 13.2.4 运行可行性研究 256 13.3 软件项目估算 257 13.3.1 代码行技术 257 13.3.2 功能点技术 257 13.4 软件开发成本估算 260 13.4.1 软件开发成本估算方法 260 13.4.2 专家判定技术 260 13.4.3 软件开发成本估算的早期经验 模型 261 13.5 进度安排 264 13.5.1 软件开发小组人数与软件 生产率 264 13.5.2 任务的确定与并行性 264 13.5.3 制定开发进度计划 265 13.5.4 项目的追踪和控制 265 13.6 人员组织 266 13.6.1 民主制程序员组 266 13.6.2 主程序员组 267 13.6.3 现代程序员组 268 13.7 软件风险管理 269 13.7.1 风险识别 269 13.7.2 风险估计 271 13.7.3 风险评价 272 13.7.4 风险驾驭和监控 273 13.8 软件配置管理 274 13.8.1 软件配置 274 13.8.2 软件配置管理过程 275 小结 276 习题 276 第14章 CASE环境与工具 278 14.1 工程环境 278 14.1.1 软件开发环境的特点 278 14.1.2 理想环境的模型 280 14.1.3 CASE环境简介 280 14.2 CASE环境的组成与结构 281 14.2.1 CASE的组成构件 281 14.2.2 CASE的一般结构 283 14.3 CASE环境工具与实践 284 14.3.1 CASE软件工程实践 284 14.3.2 常用CASE工具介绍 285 14.4 逐步求精 287 小结 290 习题 290 第15章 软件工程新技术概述 292 15.1 敏捷软件开发过程 292 15.1.1 敏捷的概念 293 15.1.2 敏捷过程的含义 293 15.1.3 敏捷过程模型 295 15.2 设计模式 302 15.2.1 设计模式的基本概念 302 15.2.2 关系环与组合模式 303 15.2.3 工厂模式 306 15.2.4 观察者模式与拉推数据 315 15.3 Web工程简介 320 15.3.1 Web系统和应用特点 320 15.3.2 Web工程的层次 321 15.3.3 Web分析 322 15.3.4 Web设计 323 15.3.5 Web测试 324 15.3.6 Web的项目管理 328 小结 330 习题 330 参考文献 331

章节摘录

插图：做好软件定义阶段的工作，是降低软件开发成本并提高软件质量的关键。
如果软件开发人员在定义阶段没有正确全面地理解用户需求，直到测试阶段或软件交付使用后才发现“已完成的”软件不完全符合用户的需要，这时再修改就为时已晚了。
严重的问题是，在软件开发的不同阶段进行修改需要付出的代价是很不相同的，在早期引入变动，涉及的面较少，因而代价也比较低。
而在开发的中期软件配置的许多部分已经完成，引入一个变动要对所有已完成的配置部分都做相应的修改，不仅工作量大，而且逻辑上也更复杂，因此付出的代价剧增。
在软件“已经完成”时再引入变动，当然需要付出更高的代价。
根据美国一些软件公司的统计资料，在后期引入一个变动比在早期引入相同变动所需付出的代价高2～3个数量级。
通过上面的论述不难认识到，轻视维护是一个最大的错误.许多软件产品的使用寿命长达10年甚至20年，在这样漫长的时期中不仅必须改正使用过程中发现的每一个潜伏的错误，而且当环境变化时（如硬件或系统软件更新换代）还必须相应地修改软件以适应新的环境。
特别是必须经常改进或扩充原来的软件以满足用户不断变化的需要。
所有这些改动都属于维护工作，而且是在软件已经完成之后进行的，因此维护是极端艰巨和复杂的工作，需要花费很大代价。
统计数据表明，实际上用于软件维护的费用占软件总费用的55%一70%。
软件工程学的：一个重要目标就是提高软件的可维护性，减少软件维护的代价。
了解产生软件危机的原因，澄清错误认识，建立起关于软件开发和维护的正确概念，还仅仅是解决软件危机的开始，全面解决软件危机需要一系列综合措施。
3.缓解软件危机的途径 软件开发不是某种个体劳动的神秘技巧，而应该是一种组织良好，管理严密，各类人员协同配合，共同完成的工程项目。
必须充分吸取和借鉴人类长期以来从事各种工程项目所积累的行之有效的原理、概念、技术和方法，特别要吸取几十年来人类从事计算机硬件研究和开发的经验教训。
应该推广使用在实践中总结出来的开发软件的成功的技术和方法，并且研究探索更好、更有效的技术和方法，尽快消除在计算机系统早期发展阶段形成的一些错误概念和做法。
应该开发和使用更好的软件工具。
正如机械工具可以“放大”人类的体力一样，软件工具可以“放大”人类的智力。
在软件开发的每个阶段都有许多烦琐重复的工作需要做，在适当的软件工具辅助下，开发人员可以把这类工作做得既快又好。
如果把各个阶段使用的软件工具有机地集合成一个整体，支持软件开发的全过程，则称为软件工程支撑环境。
总之，为了解决软件危机，既要有技术措施（方法和工具），又要有必要的组织管理措施。
软件工程正是从管理和技术两方面研究如何更好地开发和维护计算机软件的一门新兴学科。

编辑推荐

《软件工程教程》：高等院校精品课程系列教材

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:http://www.tushu007.com