



高等教育
计算机类课程规划教材

软件工程实用教程

主编 石冬凌 张应博 邹启杰

配有
“十一五”国家
重点电子出版物
出版规划
项目光盘



大连理工大学出版社
东软电子出版社

课程规划教材

软件工程实用教程

主编 石冬凌 张应博 邹启杰

配有
“十一五”国家
重点电子出版物
出版规划
项目光盘

大连理工大学出版社
东软电子出版社

图书在版编目(CIP)数据

软件工程实用教程 / 石冬凌, 张应博, 邹启杰主编.
大连: 大连理工大学出版社, 2011.4
高等教育计算机类课程规划教材
ISBN 978-7-5611-6178-4

I. ①软… II. ①石… ②张… ③邹… III. ①软件工程—高等学校—教材 IV. ①TP311.5

中国版本图书馆 CIP 数据核字(2011)第 067019 号

大连理工大学出版社出版

东软电子出版社出版

地址: 大连市软件园路 80 号 邮政编码: 116023

电话: 0411-84708842 邮购: 0411-84703636 传真: 0411-84701466

E-mail: dutp@dutp.cn URL: <http://www.dutp.cn>

大连理工印刷有限公司印刷 大连理工大学出版社发行

幅面尺寸: 185mm×260mm	印张: 19.5	字数: 446 千字
印数: 1~2500		附件: 光盘一张
2011 年 4 月第 1 版	2011 年 4 月第 1 次印刷	

责任编辑: 潘弘喆 武映峰

责任校对: 潘素君

封面设计: 张莹

ISBN 978-7-5611-6178-4

定 价: 45.00 元

软件工程是一门计算机科学与技术专业的专业基础学科,也是面向应用的工程性学科,是计算机专业九大主干课程之一。之所以说它基础,是因为这门课程总结了长久以来自软件诞生开始人们在软件开发过程中的种种经验教训和指导性原则,形成了一套相对完整的软件开发思想体系,并且在构建过程中,综合运用了许多计算机科学中的基础学科知识,是一门综合性的学科,是理论与实践相结合的综合载体。总体来讲,这门学科一方面关注软件开发过程中使用的技术方法和工具,另外一方面,强调了软件工程在实践过程中应当遵循的基本原理和指导性原则。

软件工程是一门发展迅速的新兴学科,随着软件的影响力在工作和生活中不断深入和扩大,新的理论、技术、工具和环境不断出现,推动软件工程学科的不断发展。本教材着重从理论与实践两个角度引导具有一定软件开发基础的初学者领略软件工程思想的精髓,希望能够以此帮助初学者提升软件开发的质量。

软件工程的教材风格一般分为两种。第一种是大而全,几乎将所有的理论都罗列出来,使得学生在学习和理解上感到眼花缭乱,很难将其与实际应用联系起来;另一种是专门针对其中的某一项技术进行阐述,忽略了软件工程的理论基石,特别是项目管理在软件开发过程中的作用,使得读者对软件工程的理解略显单薄。

本教材以软件工程的理论作为基础,介绍了项目管理在软件工程中的作用,并抓住目前主流的两种基本的系统分析和设计方法(结构化方法和面向对象方法),结合学生项目实践经验欠缺的特点进行设计和编写。本教材突出了理论联系实际,从实践中体会、理解理论的特点,以期让学生达到学以致用目的。



全书共分为三大部分。第一部分是软件工程引论,包括第1章软件工程概述和第2章软件工程管理。第1章整体性地介绍了软件工程的历史及现状,软件工程的作用,着重介绍了软件开发的基本过程模型。第2章集中介绍了软件工程中重点涉及的进度、质量、组织、风险等方面的管理内容和手段。第二部分是传统软件工程方法,包括从第3章需求工程到第7章软件维护的内容。在这部分内容中按照软件的生命周期顺序介绍了软件开发过程中所涉及的基础理论知识,并以传统软件工程方法为依托,介绍了如何结合理论应用具体的结构化方法进行实际项目的开发。第三部分是面向对象软件工程部分,包括从第8章面向对象技术概述到第14章系统的动态模型的内容。这部分内容承接了第二部分所讲述的软件工程基础理论和原则,针对面向对象思想的特点,以一个实际案例为背景,介绍了面向对象的系统分析和设计方法。在案例的讲述中采用了UML技术对系统进行建模,以RUP过程模型为过程指导,重点讲述了初始阶段和细化阶段中系统的推进过程。

为方便教师教学及学生学习,本教材配套光盘中提供了电子课件、程序代码等教学资源,光盘由东软电子出版社组织内容、完成制作。教学资源可到东软电子出版社网站上下载,网址是 <http://press.neusoft.edu.cn>,联系电话 0411-84835089。

本教材由石冬凌、张应博、邹启杰主编,具体编写分工为:第1章~第3章由张应博编写,第4章~第7章由邹启杰编写,第8章~第14章由石冬凌编写;石冬凌负责本教材的组织工作并审读了全书。

在本教材的策划和组稿过程中,得到了大连东软信息学院董本清、乔婧老师的大力支持和帮助,在此向他们表示衷心的感谢。在本教材编写过程中,编者参阅了国内外许多前辈和同行编写的软件工程教材及专著,这里不能一一列出,在此对这些作者表示深深的感谢。

由于编者水平有限,加之时间仓促,书中难免会存在缺点与不足,希望广大读者与教师提出宝贵意见,并将意见和建议及时反馈给我们,以便下次修订时改进。

所有意见和建议请发往:dutpbk@163.com

欢迎访问我们的网站:<http://www.dutpgz.cn>

联系电话:0411-84707492 84706104

编 者

2011年4月



第一篇 软件工程引论

第1章 软件工程概述	3
1.1 软件的概念与特点	3
1.2 软件分类	4
1.3 软件的发展和软件危机	6
1.4 软件工程的目标和原则	10
1.5 软件过程及其模型	13
1.5.1 软件过程	14
1.5.2 软件过程模型	17
1.6 软件开发工具	32
1.7 未来之路	35
小 结	38
习 题	39
第2章 软件工程管理	40
2.1 软件项目管理	40
2.1.1 软件项目产品的特点	40
2.1.2 软件项目管理的内容	41
2.2 IT项目范围管理	43
2.2.1 IT项目范围变更	43
2.2.2 IT项目范围变更原因	43
2.2.3 范围变更控制过程	44
2.2.4 实施范围变更管理原则	46
2.2.5 IT项目范围变更控制	46
2.3 IT项目估算、计划与进度管理	47
2.3.1 影响IT项目进度的因素	48
2.3.2 IT项目进度控制	49
2.4 配置管理	52
2.4.1 配置管理的意义	52
2.4.2 配置管理的实施过程	53
2.4.3 配置控制	56
2.4.4 配置管理报表	60
2.5 组织管理	63
2.5.1 民主制程序员组	64

2.5.2 主程序员组	65
2.5.3 现代程序员组	67
2.5.4 软件项目组	68
2.5.5 IT 组织管理	70
2.6 质量管理	73
2.6.1 软件质量概述	73
2.6.2 软件质量因素	75
2.6.3 质量认证	76
2.6.4 测试管理	80
2.7 风险管理	80
2.7.1 风险的分类	81
2.7.2 风险的识别	81
2.7.3 风险评估	82
2.7.4 风险的驾驭和监控	82
小 结	83
习 题	83

第二篇 传统软件工程方法

第3章 需求工程	87
3.1 概 述	87
3.1.1 需求的类型	88
3.1.2 需求开发目标	90
3.1.3 需求开发过程	91
3.2 需求获取	94
3.2.1 需求获取方法	95
3.2.2 分析人员与用户的合作关系	98
3.2.3 需求获取的重要性	101
3.3 需求分析	102
3.3.1 软件需求分析	102
3.3.2 需求和系统模型之间的关系	106
3.4 结构化分析方法	107
3.4.1 结构化分析	107
3.4.2 使用 PDL 描述需求	117
3.4.3 接口描述	117
3.5 需求描述与评审	118
3.5.1 分析建模	118
3.5.2 软件需求规约	119
3.6 需求验证与评审	122
3.6.1 需求有效性验证	122
3.6.2 需求评审	124

3.7 需求管理	125
3.7.1 需求跟踪表	126
3.7.2 需求变更管理	128
小 结	128
习 题	129
第4章 软件设计	131
4.1 软件设计概述	131
4.1.1 软件设计的目的	131
4.1.2 软件设计的原理	132
4.1.3 软件设计的原则	140
4.2 软件设计的过程	141
4.2.1 概要设计	142
4.2.2 详细设计	143
4.3 软件设计的内容	144
4.3.1 体系结构设计	144
4.3.2 数据设计	150
4.3.3 用户界面设计	152
4.4 结构化设计的方法	154
4.4.1 概要设计	154
4.4.2 详细设计	160
小 结	166
习 题	166
第5章 软件实现	168
5.1 软件实现的概述	168
5.2 软件实现的编码	169
5.2.1 编程语言的种类与选择	169
5.2.2 编程风格	170
5.2.3 编程的规范	171
5.2.4 编程的基本原则	171
5.3 软件实现的流程	173
小 结	174
习 题	174
第6章 软件测试	175
6.1 软件测试的概述	175
6.2 软件测试的目的和原则	176
6.3 软件测试与软件开发各阶段的关系	177
6.4 软件测试的过程	178
6.4.1 单元测试	179
6.4.2 集成测试	181

6.4.3 确认测试	182
6.4.4 系统测试	182
6.4.5 验收测试	183
6.5 软件测试的方法	183
6.6 白盒测试	183
6.6.1 逻辑覆盖	184
6.6.2 基本路径覆盖	186
6.7 黑盒测试	191
6.7.1 等价类划分	192
6.7.2 边界值分析法	194
6.7.3 错误推测法	195
6.7.4 因果图法	196
6.8 软件可靠性	198
小 结	199
习 题	199
第7章 软件维护	201
7.1 软件维护的概念及分类	201
7.2 软件维护的过程	203
7.3 软件维护的成本	205
7.4 影响软件维护成本的主要因素	205
7.5 再生工程	206
小 结	207
习 题	208

第三篇 面向对象软件工程

第8章 面向对象技术概述	211
8.1 常见的软件工程方法三论	211
8.2 结构化方法与面向对象方法的比较	212
8.3 面向对象的基本概念	218
8.4 面向对象方法的总结	221
8.5 面向对象建模(UML)	223
小 结	225
习 题	225
第9章 面向对象软件开发过程模型	226
9.1 迭代开发与瀑布式开发	227
9.2 迭代开发与快速原型法	227
9.3 迭代的适用范围	228
9.4 Rational 统一过程模型	229
9.4.1 RUP 软件开发过程模型	229

9.4.2 对 RUP 的错误理解	235
9.4.3 RUP 裁剪	235
小 结	236
习 题	236
第 10 章 启动项目	237
10.1 项目背景	237
10.2 过程模型	238
10.3 项目前景	238
10.4 术语表	240
10.5 开发案例	241
10.5.1 开发案例中使用的惯例	242
10.5.2 角色的映射	244
10.5.3 开发案例中的制品	244
10.5.4 为初始阶段制订计划	245
小 结	246
习 题	246
第 11 章 获取功能性需求	247
11.1 事件清单和事件表	247
11.1.1 事件的类型	248
11.1.2 示例中的事件	249
11.1.3 关注每个事件	251
11.1.4 业务规则的识别和分类	252
11.1.5 业务流程再造(BPR)	252
11.2 RUP 过程中的需求特点	253
11.3 用例模型	254
11.3.1 参与者	255
11.3.2 用例的描述形式	256
11.3.3 用例的可视化描述:用例图	258
11.3.4 用例之间的联系	259
11.4 用例产生的过程	260
11.4.1 非正式形式的样例项目用例	263
11.4.2 详述形式的样例项目用例	264
11.5 补充性规格说明	265
小 结	266
习 题	266
第 12 章 构建领域模型	267
12.1 过程模型	267

12.2 什么是领域模型	268
12.3 何时创建领域模型	269
12.4 如何创建领域模型	270
12.4.1 类的识别	270
12.4.2 应用筛选原则	273
12.4.3 关 系	274
12.4.4 识别属性	275
12.4.5 完成分析模型	276
小 结	277
习 题	277
第 13 章 体系结构设计	278
13.1 什么是软件体系结构	278
13.2 应用程序的分割	279
13.3 分离服务	280
13.4 框架模式及应用架构	282
13.4.1 框架模式	282
13.4.2 应用架构	283
13.5 体系结构设计过程	285
13.5.1 制定初步体系结构	285
13.5.2 逻辑结构的划分	287
13.5.3 执行体系结构	289
小 结	290
习 题	290
第 14 章 系统的动态模型	291
14.1 动态模型	291
14.2 健壮性分析	292
14.2.1 健壮图的表示法	292
14.2.2 健壮图的使用规则	293
14.3 序列图	296
14.4 协作图	297
14.5 状态图	297
14.6 活动图	298
14.7 选择正确的图	298
14.8 完善静态(类)模型	300
小 结	301
习 题	301
参考文献	302

第一篇

软件工程引论

近年来,随着微电子学技术的飞速发展,计算机硬件性价比平均每 10 年至少提高两个数量级,而且其质量也不断提高;但与此同时,计算机软件系统的成本却在逐年上升,规模越来越庞大,结构也越来越复杂,由于软件错误而导致灾难性后果的报道屡见不鲜,软件开发的生产率也越来越满足不了计算机应用日益普及的需求,成为制约计算机发展的关键因素。

在计算机系统发展早期,软件开发基本上沿用“软件作坊”式的个体化方法,这种方法在软件开发和维护过程中遇到了一系列严重问题:程序质量低下、错误频出、进度延误、费用剧增等,这些问题最终导致了“软件危机”。1968 年,北大西洋公约组织的计算机科学家在联邦德国召开国际会议讨论软件危机问题,正式提出并使用了“软件工程”这个名词,从此诞生了一门新兴的工程学科。

人们针对“软件危机”现象提出的多种解决方法归纳起来有两类:一是采用工程方法来组织、管理软件的开发过程;二是深入探讨程序和程序开发过程的规律,建立严密的理论,以期能够用来指导软件开发实践。前者导致“软件工程”的出现和发展,后者则推动了形式化方法的深入研究。

1.1 软件的概念与特点

1983 年,IEEE 给出了软件的如下定义:计算机程序、方法、规则和相关文档资料以及在计算机上运行时所必需的数据。目前对软件比较公认的解释是:程序、支持程序运行的数据以及与程序有关的文档资料的完整集合。其中,程序是按事先设计的功能和性能要求执行的指令序列,数据是使程序能正常操作信息的数据结构,文档是与程序开发、维护和使用有关的图文材料。

软件具有以下一些特点:

- (1) 软件是一种逻辑实体,它具有抽象性。
- (2) 由于软件是被开发或设计的(不是传统意义上被制造的),所以软件成本集中在开发上,对软件的质量控制必须从软件的开发着手。
- (3) 与机械、电子设备不同,软件在运行和使用过程中没有磨损、老化问题。前者在运行和使用中的故障率大都遵循如图 1-1(a)所示的 U 形曲线(即浴缸曲线);虽然软件不存在磨损和老化问题,但是它会退化,要不断地根据实际需求进行修改或维护,其故障率表现为如图 1-1(b)所示的锯齿形。因此,软件维护要比硬件维护复杂得多。

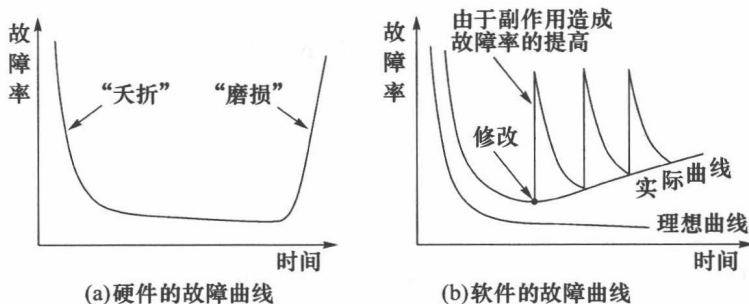


图 1-1 故障率曲线

(4) 软件一旦研制成功,其生产过程就变成复制过程,不像其他工程产品那样有明显的制造的特征,但是会出现软件产品版权保护问题。

(5) 软件的开发和运行常受到计算机硬件和运行环境的限制,这带来了软件升级和移植的问题,所产生的维护成本通常比开发成本要高许多。

(6) 虽然软件产业正在朝着构件化组装方向发展,但大多数软件仍然是定制的,尤其是软件开发尚未完全摆脱手工开发方式,尽管市场上有辅助开发工具,但最终的核心代码仍必须由程序员手工编写和组织;再加上人们对计算机的依赖程度越来越高,对软件的需求数量和规模越来越大。这同时导致软件开发人员的工作压力越来越大。

(7) 软件本身是复杂的,其复杂性可能来自它所反映的实际问题,也可能来自程序的逻辑结构,还可能会受到软件项目过程中主客观因素的影响。

(8) 涉及因素多。相当多的软件工作不但涉及社会因素,而且涉及人类社会各行各业及其他领域的专门知识。许多软件的开发和运行涉及机构、体制及管理方式等问题,某些情况下甚至涉及人的观念和人们的心理。所有这些均会直接影响到项目的成败。这对软件工程师提出了很高的要求,最终使得软件开发的分工更加明确、细致。

(9) 由于软件研制工作需要投入大量、复杂、高强度的脑力劳动,导致了软件成本昂贵。软件不单是一种在市场上推销的工业产品,往往又是与文学艺术作品相似的精神作品。与体力劳动相比,精神活动过程特点中的“不可见性”大大增加了组织管理上的困难。

1.2 软件的分类

1. 按软件的功能进行划分

(1) 系统软件:是指与计算机硬件紧密配合,使计算机系统各部件能够正常运行,使相关软件和数据协调、高效工作的软件。例如,操作系统、设备驱动程序、数据库管理系统以及通信处理程序等。

(2) 支撑软件:协助用户软件开发的工具性软件,其中包括软件开发环境、中间件、程序库、软件辅助设计工具、软件辅助测试工具等。

(3) 应用软件:为特定应用目的而开发、在特定领域内提供某些服务的软件。

2. 按软件规模进行划分

按照开发软件所投入的人力、时间等资源以及软件交付的文档和源程序行数,即软件规模,可将软件划分为6种,如表1-1所示。

一般来说,规模大、开发周期长、投入人员多的软件,其开发工作必须严格按照软件工程方法进行;而规模小的软件由于投入有限,不可能完全按照软件工程方法实施,需要开发人员根据具体情况来调整软件工程的部分实施过程。

表 1-1 软件规模的分类

分 类	参加人员	开发期限	程序规模/源程序行数	特 征
微型	1	1周~4周	500行以下	不必有严格的设计和测试文档
小型	1~2	1月~6月	1k~2k	通常没有与其他程序的接口
中型	3~5	1年~2年	5k~50k	需要有严格的文档和设计规范
大型	5~20	2年~3年	50k~100k	需要按照软件工程方法进行管理
超大型	100~1000	4年~5年	1M(=1000k)	必须按照软件工程方法开发,有严格的质量管理措施
巨型	2000~5000	5年~10年	1M~10M	同上

3. 按软件工作方式进行划分

(1)实时处理软件:是指在事件或数据产生时需要立即予以处理并及时反馈信号以控制被监控过程的软件,这类软件包括数据采集、分析、输出三部分。

(2)分时软件:允许多个联机用户同时使用计算机的软件。

(3)交互式软件:能实现人机通信的软件。

(4)批处理软件:指把一组输入作业或一批数据以批处理的方式一次运行并按顺序逐个处理的软件。

4. 按软件服务对象的范围进行划分

(1)项目软件:也称定制软件,是受某特定客户的委托,由一个或多个软件开发机构在合同约定下开发的软件。例如,气象预测分析软件、交通监控指挥系统、卫星控制系统等。

(2)产品软件:指由软件开发机构开发并直接提供给市场为众多用户服务的软件。例如,文字处理软件、图片处理软件、财务处理软件、人事管理软件等。

5. 按使用的频度进行划分

有些软件开发出来仅供一次使用(例如,用于人口普查、工业普查的软件),另外有些软件具有较高的使用频度(例如,天气预报软件等)。

6. 按软件失效的影响进行划分

有些软件在工作中出现故障而失效后,可能对整个软件系统的影响不大;而有些软件一旦失效,就可能带来灾难性后果(例如,财务金融软件、交通通信软件、航空航天软件等),这类软件称为关键软件。

1.3 软件的发展和软件危机

从 20 世纪 40 年代中期世界上第一台计算机出现以后,程序的概念就产生了,随后几十年中,计算机软件经历了 3 个发展阶段:程序设计阶段(约为 20 世纪 50 至 60 年代);程序系统阶段(约为 20 世纪 60 至 70 年代);软件工程阶段(约为 20 世纪 70 年代以后)。如表 1-2 所示。

1. 软件发展最根本变化的体现

(1)人们改变了对软件的看法。早在 20 世纪五六十年代,程序设计曾经被看做是一种自由发挥创造才能的技术领域,当时人们认为,只要能在计算机上得出正确的结果,程序的写法可以不受任何约束。随着计算机使用的日趋广泛,人们不断提出更高的要求(例如,要求程序易懂、易用、易于修改和扩充),于是程序便从按个人意图创造的“艺术品”转变为能被广大用户接受的工程化产品。

表 1-2 计算机软件发展的 3 个阶段及其特点

阶段 描述内容	程序设计	程序系统	软件工程
软件所指内容	程序	程序及说明书	程序、文档及数据
主要程序设计语言	汇编及机器语言	高级语言	软件语言 *
软件工作范围	程序编写	包括设计和测试	包括整个软件生存周期
需求者	程序设计者本人	少数用户	市场用户
开发软件的组织	个人	开发小组	开发小组及大、中型软件开发企业
软件规模	小型	中、小型	大、中、小型
决定质量的因素	个人程序设计技术	小组技术水平	管理水平
开发技术和手段	子程序、程序库	结构化程序设计	数据库、开发工具、工程化开发方法、标准和规范、网络和分布式开发、面向对象技术、软件过程与过程改进
维护责任者	程序设计者	开发小组	专职维护人员
硬件特征	价格高、存储容量小、工作可靠性差	降价幅度、速度、存储容量及工作可靠性有明显提高	向超高速、大容量、微型化及网络化方向发展
软件特征	完全不受重视	软件技术的发展不能满足需求,出现软件危机	开发技术有进步,但未获突破性进展,价格高,未完全摆脱软件危机

* 软件语言包括需求定义语言、软件功能语言、软件设计语言、程序设计语言等。

(2)需求是软件发展的动力。早期为了满足自己的需要,程序开发者不拘风格地自由创作的生产方式是软件发展初级阶段的表现。进入软件工程阶段后,软件开发的成果