普通高等教育“十一五”国家级规划教材

高等学校软件工程系列教材

软件工程基础

胡飞　武君胜　杜承烈　马春燕　编著

高等教育出版社
Higher Education Press

普通高等教育“十一五”国家级规划教材
高等学校软件工程系列教材

软件工程基础

胡　飞　武君胜　杜承烈　马春燕　编著

高等教育出版社

内容提要

本书比较全面地反映了软件工程技术的全貌，不仅介绍了传统的结构化程序软件工程方法，而且以面向对象的软件工程技术为主，重点讲解了软件分析、设计、测试的方法和技术，并以实际案例分析贯穿始终。本书还介绍了软件的形式化方法、软件能力成熟度模型（CMM）、软件文档与标准、团队组织等内容。强调实例分析和应用训练是本书的主要特色。

本书可作为高等学校计算机及相关专业软件工程课程的教材，也可供有关技术人员参考使用。

图书在版编目（CIP）数据

软件工程基础/胡飞等编著. —北京：高等教育出版社，2008.1

ISBN 978-7-04-022077-3

Ⅰ.软… Ⅱ.胡… 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（2007）第 195951 号

策划编辑 倪文慧　**责任编辑** 萧　潇　**封面设计** 于文燕　**版式设计** 王艳红

责任校对 王　超　**责任印制** 毛斯璐

出版发行	高等教育出版社	**购书热线**	010-58581118
社　　址	北京市西城区德外大街 4 号	**免费咨询**	800-810-0598
邮政编码	100011	**网　　址**	http://www.hep.edu.cn
总　　机	010-58581000		http://www.hep.com.cn
		网上订购	http://www.landraco.com
经　　销	蓝色畅想图书发行有限公司		http://www.landraco.com.cn
印　　刷	唐山市润丰印务有限公司	**畅想教育**	http://www.widedu.com
开　　本	787×1092　1/16	**版　　次**	2008 年 1 月第 1 版
印　　张	20.5	**印　　次**	2008 年 1 月第 1 次印刷
字　　数	460 000	**定　　价**	28.80 元

本书如有缺页、倒页、脱页等质量问题，请到所购图书销售部门联系调换。

物料号　22077-00

高等学校计算机科学与技术系列教材编审委员会

序

计算机和通信技术的迅猛发展，不仅形成了融合度最高、潜力最大、增长最快的信息产业，而且成为推动全球经济快速增长和全面变革的关键因素。进入21世纪，我国的信息产业虽然已取得了长足的发展，但与发达国家相比，还有不小的差距。国家信息化的发展和信息产业国际竞争能力的提高，迫切需要高素质、创新型的计算机专业人才。

高素质计算机专业人才的培养离不开高质量的计算机教育。我们的专业虽然机会多，处于非常有利的条件，但是我们同样面临着一件事，就是从规模发展向质量提高的转变。怎么提高质量？专业素质的教育和应用素质的训练非常重要。尤其是我国高等教育进入大众化发展阶段，社会对计算机专业人才呈现出了多样化的需求。而与此同时，计算机学科的发展已极大地突破了原有的学科体系框架，形成了在“计算机科学与技术”之下向多个专业方向发展的新格局。在这种背景下，教育部高等学校计算机科学与技术教学指导委员会编制了《高等学校计算机科学与技术专业发展战略研究报告暨专业规范(试行)》(以下简称“专业规范”)。专业规范按照“培养规格分类”的指导思想，提出了三种类型、四个方向，即科学型(计算机科学方向)，工程型(计算机工程方向、软件工程方向)，应用型(信息技术方向)的计算机专业发展建议，体现了社会对不同人才类型的需求，对于指导我国计算机教学改革与建设，规范计算机教学工作，促进计算机教学质量的提高都具有重要的意义。

高水平的教材是一流教育质量的重要保证。为了配合专业规范的试行，便于广大高等学校教师按照新的专业规范组织实施教学，高等教育出版社在大力支持专业规范研究与起草工作的同时，还邀请规范起草小组的有关专家成立“高等学校计算机科学与技术系列教材编审委员会”，组织规划了结合计算机专业规范、面向全国高等学校计算机专业本科生的“高等学校计算机科学与技术系列教材”。令人高兴的是，一批有创新、改革精神，且有丰富教学经验的高等学校教师投身到新体系计算机专业教材的编写中来，他们用自己创造性的思维、辛勤的汗水诠释专业规范的思想，把新的课程体系和教学内容生动地传达给师生，并进行着有意义的教学实践。

“高等学校计算机科学与技术系列教材”以专业规范和CC2001－CC2005有关教程为依据，以强化基础、突出实践、注重创新为原则，体现了学科课程体系和教学内容改革的新成果。此外，这一系列教材还配有丰富的教学辅助资源，并与现代教育技术手段相结合，充分发挥网络平台的作用，使教材更有利于广大教师和学生使用。目前，这一系列教材有不少选题已列入普通高等教育“十一五”国家级规划教材，希望这些教材的出版能够对新形势下我国高等学

校计算机专业课程改革与建设起到积极的推动作用，使我国高等学校的计算机专业教学质量再上一个台阶。

中国科学院院士

2006—2010年教育部高等学校计算机科学与技术教学指导委员会主任

二〇〇七年十一月

前　言

软件工程是信息化技术和计算机科学中一个重要而充满活力的研究领域。自20世纪60年代提出软件工程概念以来，为克服“软件危机”和提高软件质量，人们在软件需求分析、程序设计、软件测试、项目管理、过程改进、软件架构以及客户服务模型等方面进行了大量的研究工作，并逐步形成了软件工程技术领域的系统知识。随着计算机软件应用的日益普及，软件工程已成为信息技术发展的关键技术领域之一。

软件工程是高等学校计算机专业、软件专业必修的核心课程，是信息类专业的主要推荐课程，也是每一个从事软件设计、开发、管理、维护人员的必备知识。

本书比较全面地反映了软件工程技术的全貌，不仅介绍了传统的结构化程序软件工程方法，而且以面向对象的软件工程技术为主，重点讲解了软件分析、设计、测试的方法和技术，并以实际案例分析贯穿始终。考虑到内容的完整性，本书对软件的形式化方法、软件能力成熟度模型(CMM)、软件文档与标准、团队组织等内容也进行了介绍。强调实例分析和应用训练是本教材的主要特色，因此，本书需要学生和读者已经具备以下条件：至少掌握一种编程语言的使用，基本掌握面向对象编程技术，有一定编写和开发软件的经验，将从事与计算机软件开发、信息管理等相关的工作。本书是一本实用性很强的教材。

全书共由13章组成，可以分为4个部分。

第一部分介绍软件工程技术，内容包括软件工程概述(第一章)、软件生命周期模型(第二章)、传统软件工程技术简介(第三章)，阐述软件工程的作用、意义、涉及领域、技术和传统软件工程的基本方法(课时2周，8学时)。

第二部分介绍面向对象技术与UML，内容包括面向对象技术(第四章)和UML语言(第五章)，介绍和学习面向对象技术的特点和UML的描述方法(课时2周，8学时)。

第三部分介绍软件设计与实现，内容包括需求分析与描述(第六章)、面向对象分析(第七章)、面向对象设计(第八章)、实现与测试(第九章)、软件维护(第十章)。以开发小组为单位，每个小组分为设计和测试两部分，以实际应用软件开发为目标(如图书馆用户管理、网络销售服务、个人桌面工具、网络教学平台、学生信息数据库等)，完成一个软件从需求分析、设计、实现、测试到维护方案的全部过程(课时8～10周，32学时)。

第四部分介绍软件管理技术，内容包括软件的标准与软件文档(第十一章)、软件开发团队(第十二章)和软件工程技术发展(第十三章)。介绍软件管理的相关技术，并对软件工程技术发展进行更深入的介绍(课时4周，16学时)。

全部课程学习建议安排40～60学时。若学习全部内容，建议学习时间为16～18周，64学

时。建议的授课章节与次序为：第一至三章→第十一、十二章→第四至十章→第十三章。如果读者已经具备面向对象软件开发的知识基础，可安排40学时，可省略第四、五章，自学第十一、十二章。

胡飞教授组织了本书的编写工作，并编写了第一章至第三章、第九章、第十章、第十二章以及第十三章的13.1节与13.5节；武君胜教授编写了第六章、第十一章和第十三章的13.3节；杜承烈教授编写了第七章和第八章；马春燕博士编写了第四、五章和第十三章的13.2节；郑炜博士编写了第三章的部分内容和第十三章的13.4节。

在本书的编写过程中，美国马里兰大学计算机系的Atif. Memon教授提出了许多有益的建议，西北工业大学软件学院的领导和教师给予了大力支持。华东理工大学的顾春华教授对全书进行了仔细的审阅和推敲，对书稿修改提出了许多宝贵的意见，使书稿在内容安排、知识点补充和实践性方面有了明显提高。在此，作者向所有对本书编写工作给予支持和帮助的人们表示衷心的感谢。

由于作者学识有限，加之时间仓促，书中难免存在不妥之处，恳请广大读者批评指正。

编者

2007年10月

目　录

第一章　软件工程技术概述 …… 1
1.1　软件的神话 …… 1
1.2　软件工程的产生背景 …… 4
1.3　软件的特点与软件工程的定义 …… 6
1.3.1　软件的特点 …… 6
1.3.2　软件工程的定义 …… 7
1.4　软件的生命周期与软件工程的内容 …… 8
1.5　计算机辅助软件工程 …… 12
1.5.1　基本概念 …… 12
1.5.2　常用工具 …… 13
1.5.3　集成环境 …… 14
1.6　本章小结 …… 14
作业与练习 …… 15
第二章　软件生命周期模型 …… 16
2.1　软件过程与软件模型 …… 16
2.1.1　问题的确认和范围 …… 16
2.1.2　需求分析与描述 …… 17
2.1.3　系统设计 …… 21
2.1.4　实现 …… 22
2.1.5　测试与交付 …… 24
2.1.6　软件维护 …… 25
2.1.7　软件淘汰 …… 26
2.2　软件开发的困难与问题 …… 26
2.3　软件过程模型 …… 28
2.3.1　构造-修复模型 …… 28
2.3.2　瀑布模型 …… 29
2.3.3　快速原型模型 …… 30
2.3.4　增量模型 …… 32
2.3.5　同步-稳定模型 …… 33
2.3.6　螺旋模型 …… 34
2.3.7　面向对象模型 …… 36
2.4　各种模型的比较 …… 37
2.5　本章小结 …… 38
作业与练习 …… 38
实习题一 …… 39
第三章　传统软件工程技术简介 … 43
3.1　结构化程序的发展 …… 43
3.2　结构化程序的开发流程与特点 …… 44
3.2.1　结构化程序设计的分析与建模 …… 44
3.2.2　结构化程序设计的原则与方法 …… 52
3.2.3　测试 …… 56
3.2.4　软件维护 …… 57
3.3　结构化程序设计与面向对象程序设计的比较 …… 57
3.3.1　结构化程序设计 …… 58
3.3.2　面向对象程序设计 …… 60
3.4　结构化程序的应用 …… 63
3.5　本章小结 …… 64
作业与练习 …… 64
第四章　面向对象技术 …… 65
4.1　对象的概念 …… 65
4.2　面向对象的概念 …… 65
4.2.1　类与对象 …… 66
4.2.2　属性 …… 68
4.2.3　方法、操作、服务与行为 …… 68
4.2.4　消息机制 …… 69

4.3 面向对象程序的特点 …… 69
4.3.1 封装性 …… 70
4.3.2 继承性 …… 70
4.3.3 多态性 …… 72
4.3.4 重用性 …… 75
4.4 面向对象开发的方法 …… 76
4.4.1 Booch 方法 …… 77
4.4.2 Coad 方法 …… 77
4.4.3 OMT 方法 …… 78
4.4.4 VMT 方法 …… 78
4.4.5 四种方法的比较 …… 79
4.5 本章小结 …… 79
作业与练习 …… 80
实习题二 …… 81
第五章 UML 语言 …… 83
5.1 UML 语言 …… 83
5.1.1 UML 语言 …… 83
5.1.2 UML 的应用领域 …… 84
5.2 UML 与面向对象软件的分析与设计 …… 84
5.2.1 建立模型的必要性 …… 84
5.2.2 UML 在面向对象软件开发不同阶段的应用 …… 85
5.3 面向对象软件开发中的 UML 基础 …… 86
5.3.1 用例图和用例描述 …… 86
5.3.2 类图和对象图 …… 90
5.3.3 包图 …… 94
5.3.4 状态图 …… 95
5.3.5 序列图 …… 97
5.3.6 协作图 …… 100
5.3.7 活动图 …… 101
5.4 UML 建模工具介绍 …… 103
5.5 本章小结 …… 104
作业与练习 …… 104
第六章 需求分析与描述 …… 106
6.1 软件需求 …… 106
6.1.1 业务需求 …… 107
6.1.2 用户需求 …… 108
6.1.3 功能需求和非功能需求 …… 109
6.1.4 系统需求 …… 110
6.2 需求工程过程 …… 110
6.2.1 需求获取 …… 111
6.2.2 需求分析 …… 111
6.2.3 需求规格说明 …… 112
6.2.4 需求验证 …… 116
6.2.5 需求管理 …… 119
6.3 需求获取技术 …… 121
6.3.1 面谈 …… 122
6.3.2 需求专题讨论会 …… 124
6.3.3 观察用户工作流程 …… 125
6.3.4 原型化方法 …… 125
6.3.5 基于用例的方法 …… 126
6.4 案例分析 …… 126
6.4.1 确定系统参与者 …… 127
6.4.2 确定场景 …… 128
6.4.3 确定用例 …… 128
6.4.4 编写用例描述 …… 129
6.5 本章小结 …… 130
作业与练习 …… 131
实习题三 …… 131
第七章 面向对象分析 …… 132
7.1 基本原理与概念 …… 132
7.2 基本方法与过程 …… 133
7.2.1 OOA 原则 …… 133
7.2.2 OOA 建模的基本过程 …… 134
7.3 OOA 实践 1:ATM 系统 …… 139
7.4 OOA 实践 2:电梯控制系统 …… 149
7.5 本章小结 …… 151
作业与练习 …… 152
实习题四 …… 153

第八章 面向对象设计 …… 154
8.1 OOD的基本概念与原理 …… 154
8.2 OOD的方法和基本过程 …… 156
8.2.1 OOD的设计原则 …… 157
8.2.2 系统设计过程 …… 159
8.2.3 对象设计过程 …… 163
8.2.4 设计测试 …… 165
8.2.5 正式设计审查 …… 165
8.3 OOD实践1:ATM系统 …… 166
8.3.1 系统设计 …… 166
8.3.2 对象设计 …… 169
8.4 OOD实践2:电梯控制系统 …… 170
8.4.1 系统设计 …… 170
8.4.2 对象设计 …… 172
8.5 本章小结 …… 176
作业与练习 …… 176
实习题五 …… 177
实习题六 …… 178
第九章 实现与测试 …… 179
9.1 重用性 …… 179
9.1.1 重用的概念 …… 179
9.1.2 对象与重用 …… 180
9.1.3 软件开发各个阶段中的重用 …… 180
9.2 选择编程语言 …… 182
9.2.1 编程语言的类型 …… 182
9.2.2 快速原型语言 …… 183
9.2.3 最终实现语言 …… 183
9.3 好的编程风格与原则 …… 185
9.4 单元测试 …… 188
9.4.1 黑盒测试 …… 189
9.4.2 白盒测试 …… 190
9.4.3 其他测试 …… 192
9.4.4 测试方法的评价与选择 …… 197
9.5 集成测试 …… 198
9.5.1 自顶向下测试 …… 199
9.5.2 自底向上测试 …… 200
9.5.3 三明治测试 …… 200
9.6 产品测试与验收测试 …… 201
9.6.1 产品测试 …… 201
9.6.2 验收测试 …… 201
9.7 面向对象的软件测试 …… 202
9.7.1 面向对象软件测试的单元 …… 202
9.7.2 类测试 …… 203
9.8 软件测试文档 …… 204
9.8.1 软件测试计划 …… 204
9.8.2 软件测试记录 …… 206
9.8.3 软件测试评估分析报告 …… 207
9.9 软件实现与测试的CASE工具 …… 209
9.9.1 关于JUnit …… 209
9.9.2 关于EMMA …… 211
9.10 本章小结 …… 212
作业与练习 …… 212
实习题七 …… 213
第十章 软件维护 …… 214
10.1 软件维护的重要性 …… 214
10.1.1 软件维护的种类 …… 214
10.1.2 对软件维护工程师的要求 …… 215
10.2 软件维护的管理 …… 216
10.2.1 错误与缺陷的报告 …… 216
10.2.2 错误与缺陷的划分 …… 217
10.2.3 维护内容的明确 …… 218
10.2.4 维护与终止的选择 …… 218
10.3 面向对象软件的维护 …… 219
10.3.1 维护面向对象软件的优势 …… 219
10.3.2 维护面向对象软件的困难 …… 219
10.4 逆向工程 …… 221
10.5 软件维护的CASE工具 …… 221
10.6 软件维护的发展 …… 222
10.7 本章小结 …… 222
作业与练习 …… 223

实习题八 …… 224
第十一章 软件的标准与软件文档 …… 225
11.1 软件工程标准化 …… 225
11.1.1 软件工程标准化的概念 …… 225
11.1.2 软件工程标准化的意义 …… 226
11.1.3 软件工程标准的分类 …… 226
11.1.4 软件工程标准的制定与推行 … 227
11.1.5 软件工程标准的层次 …… 227
11.1.6 国外的标准化组织 …… 228
11.1.7 我国的软件工程标准化工作 … 232
11.2 软件文档 …… 234
11.2.1 软件文档的含义 …… 234
11.2.2 软件文档的作用 …… 235
11.2.3 软件文档的分类 …… 235
11.2.4 常用软件文档 …… 235
11.2.5 软件文档的编写 …… 237
11.2.6 编写的文档数量与其主要内容 …… 244
11.2.7 各级软件应该编写的文档 …… 245
11.2.8 几种常用标准中文档的名称 … 245
11.2.9 文档的管理与维护 …… 246
11.3 软件质量认证 …… 247
11.3.1 ISO 9000 标准概述 …… 248
11.3.2 ISO 9000 标准的特点 …… 249
11.3.3 ISO 9000 质量认证的一般程序 …… 250
11.3.4 ISO 9000 标准的构成 …… 251
11.3.5 2000 版 ISO 9000 标准的质量管理原则 …… 253
11.3.6 2000 版 ISO 9000 系列标准与 1994 版的主要变化 …… 253
11.4 注释文档工具 javadoc …… 254
11.4.1 javadoc 简介 …… 254
11.4.2 javadoc 的命令行语法 …… 255
11.4.3 javadoc 文档注释方法 …… 258
11.5 本章小结 …… 261
作业与练习 …… 261
实习题九 …… 261
第十二章 软件开发团队 …… 262
12.1 团队的作用与组织 …… 262
12.2 民主团队 …… 263
12.3 首领程序员团队 …… 264
12.4 其他形式的团队 …… 266
12.5 角色与责任的划分 …… 268
12.6 本章小结 …… 270
作业与练习 …… 271
第十三章 软件工程技术发展 …… 272
13.1 软件开发中的形式化方法 …… 272
13.1.1 基本概念 …… 272
13.1.2 有穷状态机 …… 274
13.1.3 Petri 网 …… 278
13.1.4 Z 语言 …… 281
13.1.5 小结 …… 283
13.2 基于组件的软件工程 …… 283
13.2.1 基于组件的软件开发过程 …… 283
13.2.2 组件的开发流程 …… 284
13.2.3 基于组件的软件工程及其研究 …… 286
13.3 软件能力成熟度模型 …… 286
13.3.1 CMM 概述 …… 287
13.3.2 CMM 的基本概念 …… 288
13.3.3 CMM 的五个级别 …… 288
13.3.4 成熟度提问单 …… 290
13.3.5 利用 CMM 对软件机构进行成熟度评估 …… 292
13.3.6 关键过程域 …… 292
13.3.7 关键实践 …… 293
13.3.8 基于模型改进的优点与风险 …… 294

13.3.9 CMMI …… 295
13.4 软件测试技术 …… 296
13.4.1 GUI测试 …… 296
13.4.2 Web测试 …… 299
13.4.3 回归测试 …… 301
13.4.4 软件测试技术展望 …… 304
13.5 面向代理的软件工程技术 …… 304
13.5.1 面向代理的软件技术 …… 304
13.5.2 面向代理的软件工程技术 …… 305
13.6 结束语 …… 307
作业与练习 …… 308
参考文献 …… 309

第一章　软件工程技术概述

什么是计算机软件？这似乎是一个非常简单的问题。几乎所有行业的人员都或多或少地接触过计算机软件。“软件就是可以运行的一段程序代码”，这可能是大部分人的答案。

一段可以运行的程序代码就是软件吗？随手写的一段练习程序（如输出“hello world”，大部分程序语言的第一个练习）就是软件了吗？

“软件应该完成一定的功能”。那么，“能够完成特定功能的一段可以运行的程序”就是软件了吗？人们想到的往往是软件的功能，而忽略了许多重要的指标。

计算机软件是软件工程师设计和生产的产品，它包含在计算机上执行的任何大小和结构的程序、电子文档以及由数字和文本组成的数据（包括图示、视频和音频信息）。[5]

软件是产品吗？当然是。现在的人们可以买到大量的软件产品，从操作系统、办公软件、数据库、网络服务到计算机游戏等。既然软件是产品，它就应该具有产品的特点：有价值（能完成特定的功能）、有标准（符合产品的规范）、可检测（衡量与评价的标准）、有质量（保证功能的正确与可靠）。

如何生产一个高质量的软件产品？如何成为一名合格的软件工程师？这正是本书要阐述的内容。

本章首先从产品的角度说明软件工程的意义和作用。

1.1　软件的神话

关于软件的意义与作用，不需要再花费更多的笔墨去阐述了。我们知道它已经成为当今人类学习、工作、生活不可或缺的组成部分。几乎所有看到本书的人，都多多少少接触过软件，还有相当多的人学习、设计和开发过软件。那么，先看看以下对软件的一些认识和说法。

1. 乐观的看法

① 软件是艺术创作，过多的规矩限制了软件开发人员的创造性。

这往往是一些所谓软件高手们面对软件质量人员和管理者关于文档、测试、报告的询问时最好的托辞。

事实上，依靠个人能力编制精巧软件的时代早已过去，人们需要的是规范、可靠、实用的软件产品，而不是独创性的艺术品。如果还有哪位软件工程师强调他/她的软件是艺术品，不能有规范约束，只能说明他/她并不适合做一名软件工程师。

② 我们已经有了规范的软件标准，我们就会有高质量的软件。

这是许多软件企业管理者的想法：有了标准和规范，软件工程师就会去按标准做的。

事实上，我去过一些软件企业，当谈到软件质量时，他们都能搬出足够我看上 3 天的公司软件开发流程与规范，来说明他们公司软件开发的规范性。可当我阅读他们的软件工程师提交的各类文档、数据和报告时，惊讶地发现：大部分文档只是软件工程师的个人杰作，与企业的标准相去甚远。问题出在哪里？我询问提供文档材料的软件工程师。他们说，既然技术管理者们并不知道如何检测报告、文档和数据的质量和标准，那还有谁会在意这些标准呢？

③ 因为我们有许多软件高手，我们的软件就是先进的软件。

这是许多软件企业管理者的想法：有了软件高手，就可以有领先的技术，就可以有领先的软件产品。

事实上，高水平的技术未必能制造出高水平的产品。就如同有了最好的厨师，但未必就是最好的饭店。软件产品是一个团队整体水平的体现，仅仅依靠几名高水平的技术人员，并不能保证开发出高水平的软件产品。

④ 软件高手是软件成功的关键。

这是许多软件项目经理的想法：我的组里有两名软件高手，所以我的软件项目一定能成功。

事实上，软件高手并不能包打天下，缺少高效的管理和团队的合作，高手只能充当救火队长，而软件成功就会成为难以预料的事情。

⑤ 软件开发进展缓慢，没关系，我们赶快增加人手吧。

这是许多软件项目经理和软件工程师面对进度落后的反应。24 个人月的任务，3 个人需要工作 8 个月。当 4 个月后，发现需要加快进度，认为再增加 3 个人，2 个月就能完成。

事实上，新增加的 3 名软件工程师，仅是花费在熟悉项目工作上的时间可能就需要 1 个月，因为人数的增加，交流的工作更多，非但 2 个月不能完成，可能速度反而更慢。[15]

⑥ 我们的软件太大了，外包出去一块吧，那部分我们就不用操心了。

这是许多软件项目经理面对大型软件开发时的想法：软件外包解决了大部分的软件模块设计与实现，我们只需要进行模块总装就可以了。

事实上，软件项目经理需要完成所有模块的详细设计与定义，不断地沟通与调整、验收与测试，唯一可以省略的是软件实现与单元测试。就如同汽车零件外包一样，需要准确地定义该零件的形状、尺寸、材料、强度、质量以及测试指标，然后才能外包出去。

⑦ 我们的功能需求已经全部交给了软件开发者，等他们的软件交付了，就能提高我们的工作效率了。

这是大部分软件客户对软件产品开发的理想与愿望：当软件研制完成时，我们所有的要求就都实现了。

事实上，软件需求者（客户）与软件开发者（软件工程师）在对软件产品的功能定义、实现方式、人机界面、工作流程等方面一直存在极大的认识差异，甚至出现功能的偏差。当客户需求说明“自动审核”时，没有哪个软件工程师能仅仅从这 4 个字理解程序要干什么。所以当客户拿到

软件时，可能与他想象中的软件产品相去甚远。

⑧ 软件修改是容易的事，让软件工程师改几行程序就好了。

这是大部分客户对软件维护的看法：这个软件功能如果再加上自动统计就更好了，如果能够增加一个用户类型就更方便了，如果增加一个快捷方式就更好了。你们修改一下，一周时间足够了，就增加几行程序嘛。

事实是，为了软件的维护与更新，相对于在设计阶段的同样功能，软件工程师往往要花费数倍甚至数十倍的工作量来修改。有时甚至会导致整个软件结构的更改。所以，软件功能的增加与维护并不是任何条件下都可以完成的。

⑨ 我们终于解脱了，软件已经交付用户了。

软件工程师在一个软件项交付后，都会有这样的想法：任务终于完成了！

事实上，软件开发与实现的工作量在整个软件生命周期中不到40%，艰难的维护和长期的更新任务才刚刚开始，软件工程师还要为它付出更长期和更大量的工作。

⑩ 软件文档没什么用，加些注释就行了，剩下的我全记在脑子里了。

许多软件工程师即使嘴上不说，心里也是这么想的，而且也是这么做的。

事实上，软件本身就是复杂而难以理解的产品，如果是缺少了文档的软件产品，那就更不会有人愿意去维护它，软件就会逐渐成为废品。我曾经让一名研究生分析一个控制系统程序，3周后，他告诉我无法分析。为什么？这个程序并不十分艰涩呀？经了解，他是没有软件文档。于是我告诉他拿到软件文档后再看，否则，再分析3周也不会有什么结果。

2. 悲观的看法

也有许多软件工程师在工作了一段时间之后，生出许多对软件的恐惧，不妨列举常见的一些。

① 软件质量无法控制，它是人的智力表现。

这是相当数量软件工程师的看法。一个存在于人的大脑中的思维与活动，实在是难以控制的。

事实上，哪一种工作不需要大脑的思维呢？哪怕只是完成搬运这样的简单劳动，大脑一样要处理路线、重量、方式等一系列问题。我们最终检测的不是你的大脑过程，而是你搬运的物品多少，质量如何。同样，我们不可能也不需要控制软件工程师的大脑活动，而是关心完成了多少程序的代码以及质量如何。

② 软件总是有错的，所以测试意义不大。

这正是许多软件开发者轻视软件测试的核心问题，因为无论怎样的软件测试，都无法保证软件完全正确。

事实上，软件测试可以将软件错误控制在一定的范围之内，可以避免灾难性错误产生的几率。任何一个产品都不能保证100%可靠，但一个没有经过任何检测的产品，又有哪个用户敢去使用呢？

③ 软件是无法度量的，所以我无法评估我的软件工程师的工作量。

这是软件工程师抱怨最多的问题，总觉得自己的软件模块是最复杂和最困难的，而待遇却与别人一样。

事实上，软件是可以度量的，不论从难度（复杂度）还是数量（代码行数）上都可以实现。

④ 软件项目进度无法预测，因为人的智力与情绪都会影响软件项目进度。

在项目进度落后或无法按时完成时，这往往成为推脱责任的借口。

事实上，即使人的智力与情绪会影响软件项目的质量与进度，但作为软件项目经理，就应该了解每一位软件工程师的能力，了解每一个困难与问题，了解资源的分配与进度的安排，并不断督促大家按计划工作，而不是仅仅告诉大家进度的要求。

⑤ 我不是软件高手，所以我不会成为出色的软件工程师。

常常听到软件工程师这样说。软件高手是出色软件工程师的标准吗？

事实上，只要你细心、认真、努力，你就已经是一个优秀的软件工程师了。如果你还有良好的沟通能力、对工作的积极态度和勇于不断学习的勇气，你就已经具备了出色软件工程的所有条件。

⑥ 软件技术更新太快了，我实在无法跟上技术发展的步伐。

这是软件工程师最头痛的一件事，似乎年过 35 岁，他们就成为落伍的同义词，就难以追赶技术的进步了。

事实上，软件工程师会发现，尽管各种技术不断出现，软件设计核心技术的变化并不大，而软件开发经验往往成为比开发技术更宝贵的财富。许多的软件工作依赖于经验和成功的模式，更不要说还有大量管理性的工作了。事实上，具有丰富开发经验的软件工程师正是每一个软件企业最宝贵的资源。

⑦ 我的软件工程师被挖走了 2 名，整个项目无法按时完成了。

的确，这是一件很棘手的事情。要想使项目不受影响是不可能的。还是想办法补救吧。

如果你安排计划时就预计到类似情况的发生，也许你就早有补救的安排，预备队也许可以派上用处了。也许你没有预计到这种情况，那么，他们的工作有详细的说明文档和进度计划吗？是的，很完善。那我们就重新安排新的软件工程师接手吧，尽管要耽误一些时间，可对整个项目而言，影响并不太大。如果所有的设计与实现全在他们个人的脑子里，那他们愿意全部写出来吗？如果不愿意，那你真的要面临重新开发了。

1.2 软件工程的产生背景

计算机软件从进入人们的生活到承担日益重要的角色，经过了数十年的时间。1979 年，Osborne 将计算机与软件定义为一场新的“工业革命”。1980 年，Toffler 在《第三次浪潮》中将微电子技术称为人类有史以来的第三次革命。1982 年，Naisbitt 预言了从“工业社会”到“信息社会”的转变。1983 年，Feigenbaum 和 McCorduck 阐明由计算机控制的信息和知识将成为 21 世纪的核心力量。