现代软件工程专业系列教材

软件开发基础教程

RUANJIAN KAIFA JICHU JIAOCHENG

下 册

程国英 钱晓平 编著

清华大学出版社
http://www.tup.tsinghua.edu.cn
北京交通大学出版社
http://press.bjtu.edu.cn

软件开发基础教程

下册

现代软件工程专业系列教材

软件开发基础教程

（下册）

程国英　钱晓平　**编著**

清华大学出版社
北京交通大学出版社
·北京·

内 容 简 介

本书分上、下两册，共五部分内容。上册包含前两部分内容，主要介绍面向对象（包括面向过程）的程序设计。下册包含后三部分内容，以基于面向对象和消息发送机制的可视化软件分析、设计和开发为主，并选择 C ++ 语言，使用 Borland C ++ Builder 开发环境进行实例开发。

下册的第 3 部分是软件工程基础，主要介绍软件工程的基本概念、软件开发过程、软件项目管理、软件建模语言 UML 和软件建模，以及如何进行软件分析、软件设计、软件测试。第 4 部分是 C ++ Builder 集成开发环境，主要介绍 VCL 组件、组件事件、事件响应、组件之间的消息传递，基于数据库的软件开发，图形与多媒体的应用，多线程及多线程的同步控制，动态链接库、软件异常处理和发布应用。第 5 部分是 HIS 实例开发，以 HIS 为例讲述基于消息发送机制的面向对象软件分析、设计和实现的全过程，并提供详细的实现过程和全部源代码。

本书的特点是先提出“问题”，直接面对“问题”，然后抽象分析“问题”，再设计、解决“问题”，体会面向对象和面向过程的区别与联系，展现一个“生产”软件的全过程，加强系统性和抽象问题、分析问题的训练。通过下册的学习可使读者具备开发可视化应用软件的基本能力。

本书的上册适合作为计算机、软件工程，以及其他相关专业的 C ++ 程序设计课程的教材；本书的上、下两册适合作为计算机、软件工程专业的软件开发基础课程的教材。由于整套书贯穿实例进行，并加以实现，也非常适合自学者使用。

图书在版编目（CIP）数据

软件开发基础教程：下册 / 程国英，钱晓平编著. —北京：清华大学出版社；北京交通大学出版社，2007.3
（现代软件工程专业系列教材）
ISBN 978 - 7 - 81082 - 950 - 2

Ⅰ. 软…　Ⅱ. ① 程…　② 钱…　Ⅲ. 软件开发 - 教材　Ⅳ. TP311.52

中国版本图书馆 CIP 数据核字（2007）第 004667 号

责任编辑：招富刚　　特邀编辑：刘　云
出版发行：清 华 大 学 出 版 社　　邮编：100084　　电话：010 - 62776969
　　北京交通大学出版社　　邮编：100044　　电话：010 - 51686414
印 刷 者：北京鑫海金澳胶印有限公司
经　　销：全国新华书店
开　　本：203 × 280　　印张：18.25　　字数：578 千字
版　　次：2007 年 3 月第 1 版　　2007 年 3 月第 1 次印刷
书　　号：ISBN 978 - 7 - 81082 - 950 - 2/TP · 328
印　　数：1 ～ 5 000 册　　定价：29.00 元

本书如有质量问题，请向北京交通大学出版社质监组反映。对您的意见和批评，我们表示欢迎和感谢。
投诉电话：010 - 51686043，51686008；传真：010 - 62225406；E-mail：press@bjtu.edu.cn。

前言

软件神话曾严重影响并阻碍了软件的发展。我国人口众多，物质并不丰富，但我国具有大量的智慧型人才，而软件产业不需要损耗多少资源就可以增加大量财富，可是我们却失去了成为世界最大软件出口国的最佳机会。

至今，在不少学生的认识中还存留着许多软件神话的糊涂观念，影响了他们的学习与工作。作为一个软件领域的工作者，应该努力消除软件神话的破坏性影响，积极注重科学的工程化软件生产。尤其是作为高校的教育工作者，我们感到更有责任注重培养自己和学生的软件工程思想，掌握软件工程技术，以利于我国软件产业的健康发展。

本教材由钱晓平教授与程国英合作编著，从章节、具体内容、实例开发均不断进行讨论和斟酌，最后由钱晓平教授统稿、审校。

再次感谢我们的学生，感谢我校生物医学工程专业、联读班、MIT 试点班、软件学院和 ACM 试点班的学生，尤其是 2006 级联读班的同学，他们是第一批使用本教材进行教学活动的学生，感谢他们对本教材的指正，以及在教学过程中给予我们的建议和激励，感谢已毕业的学生反馈给我们的宝贵信息和建议。

由于我们的学术理论水平有限，纵然努力地全身心地投入此教材的编写之中，仍不免存在不周和不足之处，真诚地期盼得到读者们的批评和建议。

作者的联系邮箱是：gycheng@sjtu.edu.cn, xpqian@sjtu.edu.cn。

程国英　钱晓平　于上海交通大学

2007 年 2 月 22 日

目 录

第 3 部分　软件工程基础

第 4 部分 C++ Builder 开发环境

第 5 部分 HIS 实例开发

第 3 部分　软件工程基础

第 3 部分将结合上册的内容介绍软件工程基础。我们已经对 HIS 实例有了较深的印象，在讲解软件分析、设计、测试及项目管理时，所举的例子均针对上册实现的 HIS 实例进行，使得易于理解并易于从软件系统的角度得到认识和提高。

第 32 章　软件神话与软件工程

软件具有与硬件截然不同的特征，硬件的创造过程可以完全转换成物理形式，而软件不能，软件是逻辑产品、不是物理产品，软件产业不是传统意义上的制造业，软件是“头脑”产业。生产软件不需要像传统制造业那样为建造厂房、建造原材料仓库和成品仓库、购置生产设备、采购各种各样的原材料等一系列的基础设施建设而投入大量的前期资金。软件的这些“与众不同”的非凡特性，使得在过去的许多年里，在软件行业内和软件行业外均造就了对软件的种种“神话”。

32.1　软件神话

“The Mythical Man-Month”（人月神话）一书的作者 Frederick P. Brooks Jr.博士早在 20 多年前就提出：我认为用人月作为衡量一项工作的规模是一个危险并带有欺骗性的神话。因为，作者认为“人月”带有人员数量和时间之间是可以相互替换的暗示。“人月”通常用于软件开发时对工作量进行的估算，例如，开发某个软件项目需要 5 个人、10 个月的时间，就认为需要 50 个人月的工作量。但是，人数与时间之间是绝对不可以互换的，绝对不是“人手不够、增加时间，时间不够就增加人员”的单纯关系，如果这样认为，那就是“神话”。

在软件行业中，诸如此类的神话并不少：

- 软件就是程序、程序就是软件，开发软件就是编程，因此一群聪明的程序员就可以开发出高质量的软件；
- 时间紧迫，增加人员就可以使得任务按时完成；
- 软件是不会“磨损”的，所以软件的维护相对硬件要容易；
- 软件需求总是不断变化的，但是相对整个软件项目，变化总是微小的，是很容易实现的，何况软件是灵活的；
- 设计可以粗糙点，因为在编程时会有突发异想的灵感，可以在以后补充、细化；
- 一个成功的软件项目就是提交可执行的程序；
- 一旦完成程序编写工作，并且能够正常运行，软件开发小组的工作就结束了；
- 在程序运行之前，谁也无法评估其质量，因此预先不必多考虑质量问题，而且总会有办法解决的；
- 软件工程要求编写大量的不能运行的文档，却占用了大量的时间和精力，而用户关注的是可以运行的软件，并且希望提前交货，因此文档可以留待以后有空时补写（甚至不写）。

32.2 软件工程与软件创新

软件神话曾严重影响并阻挠了软件的发展。自 1968 年在联邦德国召开的国际会议上正式提出“软件工程”后，软件的制作开始进入工程化的生产模式，逐渐形成新兴的软件产业。如今，软件作为一种驱动力，被嵌入在各领域的应用、决策系统中，直接影响并驱动现代社会的发展，软件已成为区别现代产品和现代服务的标志性因素。

工程是对被生产的实体所进行的分析、设计、制造及其管理。

软件工程是为生产符合需求的具有质量保证的安全可靠的软件而进行的分析、设计、制造及其管理。

软件工程分支学科是研究系统化、规范化、可度量化地进行软件生产的方法、技术和工程原则，包括生产过程管理的方法、技术和原则。

软件创新主要指软件工程研究领域的创新，软件工程研究领域需要不断创新，不断向软件工程提供科学的安全可靠的新方法、新技术、新工具。一个软件工程项目必须主要采用成熟的安全可靠的方法和技术，绝对不能使用没有经过实验证明是科学的、安全的新方法和新技术。如果使用没经过实验证明是科学的、安全的新方法和新技术，必须考虑可能出现的问题及其解决方案，并在使用过程中进行“跟踪”，以便及时纠正错误和收集信息，积累经验。

科学研究强调发现和创新，强调科学实验。新的发现、新的理论、新的技术等所有的“创新”必须经过严谨的科学理论研究和严格、反复的科学实验，取得安全可靠性保证后才可以推广到工程中。

工程与科研不同，工程强调安全可靠，工程力推采用成熟的方法和技术，工程不能使用没有经过实验和证实是安全可靠的方法、技术、工具、工艺标准和工艺过程。传统领域的产业工程对这方面有非常清晰的认识和严格的控制。“创新”的实验室产品在投入生产、作为商品进入市场之前，需要经过一系列专门机构的检验和认证。例如，新研发出来的药品必须通过动物试验后才能非常谨慎地进行临床试验，绝对不能直接投入生产。传统领域产业的产品都有严格的测试、检验标准和测试、检验过程，并由专门的检验测试部门进行测试和检验。例如，汽车的里程、碰撞测试，医疗仪器设备的破坏性测试都有相当严格的国家、国际等级标准。

然而，在软件工程领域中，由于软件几乎是日新月异地飞速发展，新的软件、新的技术层出不穷。新的软件产品一定是创新产品，因为相同的软件只要复制即可，不需要重复“制造”。软件一旦开发出来就可以投入使用，不需要经过一系列专门机构的检验和认证。软件产业就像一面迎着朝阳高高飘扬的“创新”大旗，一批具有聪明才智的年轻人很快就揭去了软件的神秘面纱，欢快地聚集在这面高高飘扬的大旗下，以最快的速度学会了使用软件和制作软件的某种语言，兴奋地投入到软件这个朝阳产业之中。一时间软件工程领域中充满了具有充沛活力和极富想像力的编码狂人，这些人以高涨的热情努力工作，开发了一个又一个的软件，在短时间内填补了一个又一个领域的软件空白。

很快地，软件生产就处于重复开发、无法积累、难以在前期生产的基础上可持续性地健康发展的状况，开发符合预算和进度需求的高可靠和高性能的软件简直难以实现。这时，软件领域才开始反思并意识到软件工程、软件过程和软件组织的重要性。不幸的是，在软件产业起步较晚的中国，仍然重复了这段“发展”路程。我国人口众多，物质并不丰富，但我们具有大量的智慧型人才，而软件产业不需要损耗多少资源就可以增加大量财富，可惜我们却失去了成为世界最大软件出口国的最佳机会。

为什么学会了使用软件和制作软件的某种语言就会认为具有开发软件的能力，聚集了一些这样的人就可以办起软件公司？而学会开车的人都理智地认为自己仅仅具有使用汽车的能力，绝不具有开发、制造汽车的能力，聚集一群驾驶员是绝对不可能制造出汽车的。其中一个关键的原因就是汽车是实体，人们看得见、摸得着，能够直接体会其复杂的设计和制造过程，而软件是看不见、摸不着的，很难体会“学会使用软件和制作软件的某种语言仅仅是具备了使用计算机的能力（犹如学会开车仅仅具有使用汽车的能力而已）”与“学会使用软件和制作软件的某种语言、但并不具有设计和生产计算机系统的主要组成部分软件的能力”之间的区别，更难体会其复杂的设计和生产过程。

在其他许多科学技术领域中，工作人员上岗前普遍都要通过专业培训和实践，并且规定必须经历足够的专业培训和实践时间，在取得基本能力和技能并获得能力和技能的“持续性”证实后才能从业或上岗。例如，驾车者必须持有驾驶证才能驾驶汽车，不得在驾驶工作岗位上学会驾驶。持有专业毕业文凭仅仅证明具有该专业的学习能力和学历，而不证明具有从事该专业的工作能力，例如，医科大学的外科专业毕业生必须先任“实习医生”，不得在为病人进行的外科手术中学会开刀而成为外科医师；法律专业毕业的学生必须先持有律师证才能从事律师职业。

但是，在软件行业，目前对从业人员还没有任何规定，没有不得在工作岗位上学习和掌握软件开发技能、软件工程方法和技术的规定。在软件行业的普遍现象是，许多刚从学校毕业的学生没有经过任何严格的职业规范训练就上岗，他们只能在工作中逐渐掌握专业技能，这势必给软件行业带来了莫大的“风险”。

软件与其他领域的产品有太多的不一样，软件在生产过程中不会出现任何“工伤事故”，但是软件的潜在错误在使用中却会造成致命的难以挽回的损失。因此，只有在软件设计、开发时尽量多地发现并纠正错误，以避免或最大限度地降低“风险”所造成的损失。而这点对于无长期从事软件开发而积累起来的经验的人来讲，是根本不可能做到的。软件生产者和生产组织需要积累，需要对软件生产过程加以管理和控制，需要走向成熟，而成熟是安全可靠的最大前提和保障。

32.3　软件组织与 CMM

软件是一种团队性的脑力活动的产物，软件组织的有效性已显得越来越重要。软件作为一种新兴产业已逐渐走向成熟，如果要使得自己的组织具有可信度和行业竞争力，除对软件设计和开发过程进行严格管理外，还必须对软件组织本身进行标准化管理。犹如一个工厂，如果该工厂的某个类型的产品能够达到国际标准，但其他类型的产品却不一定能达到国际标准，这说明该工厂不成熟，缺少持久、平衡的技术和管理上的积累，缺乏全面的标准化的生产能力，这样的工厂明显不具有企业可信度和行业竞争力。

CMM（Capability Maturity Model）是软件能力成熟度模型（Capability Maturity Model for Software）的简称。CMM 代表软件领域对良好的工程和管理实践的一种广泛一致的认同，已成为衡量软件过程能力的国际工业标准。

CMM 最初是由美国卡内基梅隆大学的软件工程研究所（Software Engineering Institute，SEI）为美国联邦政府评估软件供应商能力，于 1986 年开始研究的模型，于 1993 年推出 CMM 1.1 版，之后 CMM 不断发展并于 2000 年 8 月发布了最初的集成版 CMMI 1.0 版。CMM 仍在发展之中，版本不断地在更新。

CMM 描述了有效的软件过程的基本要素，为软件机构提供并描述了从混乱的、不成熟的软件过程向成熟的、有纪律的软件过程改进的一条途径。CMM 强调对过程进行控制和管理，强调过程监控和文档必须先行，力求软件生产的每个步骤可控、可预见，使得软件组织真正有效地控制软件生产，软件产品的质量得以真正的保证。

CMM 的主要作用是：

- 软件过程的改进（Software Process Improvement）；
- 软件过程评估（Software Process Assessment）；
- 软件能力评估（Software Capability Evaluation）。

CMM 评估软件组织（或软件公司）有 5 个等级标准。

（1）初始级：处于初始级的软件组织，其软件过程的特点是无序的，甚至是混乱的。几乎没有什么过程是经过妥善定义的，成功往往依赖于个人或小组的能力。

（2）可重复级：处于可重复级的软件组织，具有建立了基本的项目管理过程来跟踪成本、进度和功能的特性，制定了必要的过程纪律，有一定的过程规范，积累并重复早先开发的类似的应用项目所取得的成功。

（3）已定义级：处于已定义级的软件组织包含可重复级的所有特征，并且已将管理和工程活动两方

面的软件过程文档化、标准化、规范化，并与整个组织的软件过程相集成，所有项目都使用软件组织经批准的标准软件过程的剪裁版本。

（4）已管理级：处于已管理级的软件组织包含已定义级的所有特征，并且收集对软件过程和成品质量的详细度量数据，通过这些数据定量地理解、控制和预测软件过程和产品的生产。

（5）优化级：处于优化级的软件组织包含已管理级的所有特征，并且通过来自于过程的量化反馈和先进的新思想、新技术促使过程不断改进。该等级组织的核心能力就是持续改进。

32.4 软件工程师的任务

软件工程师的任务就是在规定的时间和预算费用内、按计划和进度完成并交付高质量的软件产品。每个软件工程师必须对自己的工作制订工作计划和工作目标，然后按预定的计划和目标努力工作，尽力生产出高质量的产品。

在上册的 2.6 节中讲过：对于软件产品我们只能尽最大努力力求正确，而不可能完全杜绝错误隐患。软件难免有缺陷，但这并不表示软件允许有缺陷。对于软件的缺陷我们不可能像机械、电器系统那样增设物理的安全隔离防护系统，事实上软件的安全防护系统通常也是由软件来实现的。

软件的质量依赖于每个参与软件生产的工作人员对努力生产无缺陷的高质量的软件的承诺和兑现。因此，始终以高质量的产品满足契约要求的能力已成为软件工程师最重要的个人资产。

作为一个合格的软件工程师，必须强调自己工作行为的规范，积极投入掌握和提高软件开发技能的活动或训练之中，不断学习与自我提高。在生产软件产品的过程中，对每个软件工作人员来讲是一个积累经验、趋向更加成熟的过程，绝不能视为一个学习、训练的过程，因为“工作”与“学习”所承担的责任和义务是不同的，“学习”所“练习”出来的软件中包含的错误极有可能会导致非常严重的后果，软件付不起这个“学费”。

◇ 思考与练习

1. 你对“软件神话”有何体会？请列举其他关于软件和软件开发方面的类似“神话”的说法或想法或理念。

2. 工程与科学研究之间有什么不同？有什么内在联系？通过上册的学习、练习和实践，你对软件工程是否有更进一步的体会？请简要描述软件工程领域的方法、技术、工具、工艺标准和工艺过程。

3. “软件能力成熟度模型 CMM 已成为衡量软件组织的软件过程能力的国际工业标准”，其中“能力”和“过程”指的是什么？

4. 为什么说“始终以高质量的产品满足契约要求的能力已成为软件工程师最重要的个人资产”？你怎么理解这句话？

5. 软件工程师的根本任务是什么？为什么？

第 33 章　软件过程与项目管理

过程（Process）是为了导致某一预期结果而进行的一系列预定的各项步骤（行动、变化或作用处理），这些步骤可以是系统的或有次序的、并且能够使得各步骤无阻碍地进展，直至最终步骤并获得预期结果。

例如，生产过程是制作或处理某一产品的一系列的操作，经历了这一系列的操作就可以获取预期的产品，或者说产品在这一系列的操作中逐渐形成。又例如，国家培养大学生，必须制定从小学、中学到大学这个漫长过程中各个阶段的学习课程、课程内容、阶段目标、最终目标，学生经历了规定的整个学习过程并达到各学习阶段的目标要求后，就被国家认为可以大学毕业并取得毕业证书；对这整个学习过程，为保证最终的学生质量及各种可能的变化（包括学生个体的变化），又规定了许多规章制度进行过程管理，例如，招生、学分、奖惩等规定。

33.1　软件过程与成熟度等级

软件工程的最终目标是生产符合需求的软件产品。软件的开发是由具有开发软件能力的、有创造力的人组织成的团队（软件开发组织），在预先制定好的一系列软件过程中进行活动，这些过程的活动帮助团队生产符合市场需求的软件产品。

CMM 指定了软件过程的 18 个关键过程域，每个关键过程域都确定了一套相应的活动，这些关键过程域的实施（即进行过程域的活动）并制度化（即把进行过程域的活动作为制度），体现了软件组织的软件过程的成熟度。软件过程的成熟度标志了软件组织开发软件的能力及能力的增长潜力。18 个关键过程域被映射到不同的 CMM 成熟度等级，CMM 的第 5 级（优化级）应实现全部的 18 个关键过程域。如果不考虑项目的规模和复杂性，这些过程域适用于所有软件项目，整个项目的安全保护性活动应该贯穿整个软件过程，并且独立于任何一个过程。每个关键过程域都可通过“目标”（描述该关键过程域要达到的总体目标）及下列“共同特征”（指各关键活动都具有的属性）进行描述。

（1）执行约定：指执行某一关键过程所需要的约定。即描述为保证该关键过程域活动持续进行而采取的必要措施，通常包括制定组织的领导体制（例如，各领导的分工及之间的关系）和策略。

（2）执行能力：指执行某一关键过程所需要具备的能力。即描述组织执行该关键过程、或为了项目成功而必须具备的条件和能力，通常包括资源、培训、组织结构方面的条件或能力。

（3）执行活动：指执行某一关键过程所需要进行的活动。即描述为了完成该关键过程域的目标所需要进行的具体任务，通常包括制定计划、规则、跟踪及纠正措施。

（4）测量、分析：描述某一关键过程域在进行过程中控制和改进的方式，通常描述为确定过程的进展情况所必须进行的基本测量实践。

（5）验证执行：描述某一关键过程域能够被验证的方式。通常描述为验证关键过程“是否被正确进行”而进行的验证活动，包括主管部门和软件质量保证部门所进行的评审和审核。

33.1.1　CMM 1 级（初始级）

CMM 1 级是处于无序甚至混乱状态的初始级，几乎无任何预定的软件过程，软件组织通常不能提供开发和维护软件的稳定环境，软件开发的成功完全依赖于个人。软件组织也许具有相当优秀的项目经理和一群具有丰富经验和实战能力的技术高超的开发人员，他们的存在能够使得软件开发组织运转正常、甚至高效，但是一旦他们中的某个人或部分人“撤离”该组织，将立即产生问题，甚至软件组织将无法继续运转。

CMM 1 级的软件组织的能力具有个人特征而不具有组织特征。