

软件工程 课程设计

吕云翔 刘浩 王昕鹏 周建 等编著



TP311.5



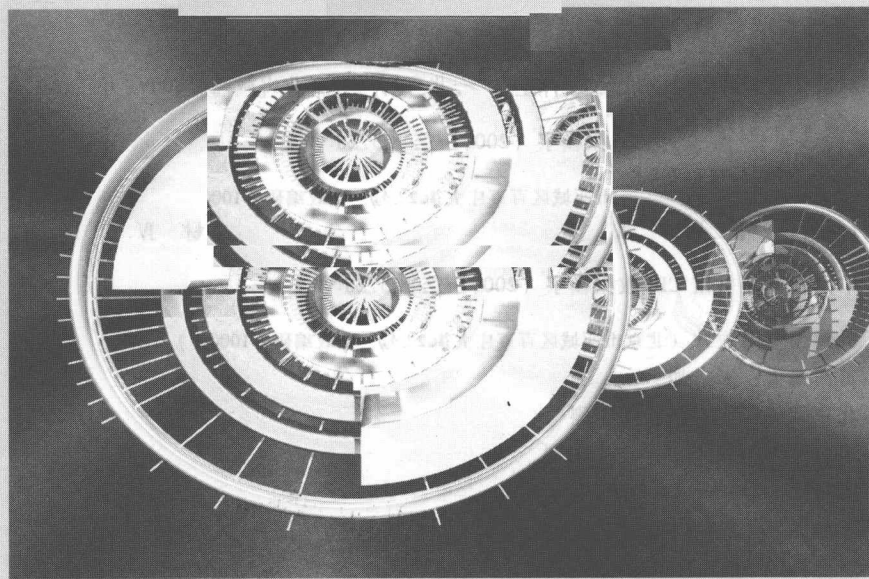
机械工业出版社
China Machine Press

高等院校计算机课程设计指导丛书

软件工程

课程设计

吕云翔 刘浩 王昕鹏 周建 等编著



机械工业出版社
China Machine Press

本书结合高等院校“软件工程”课程的相关要求，通过一个会议系统的实例，向读者介绍软件工程理论在实际项目中的应用，以达到深入理解软件工程过程和实现方法的目的。本书内容包括项目准备、项目管理、需求分析、软件设计、软件实现、软件测试，以及如何编写用户手册和如何进行配置管理等。此外，本书还给出了各个关键阶段的项目文档，具有很强的参考价值。

本书适合作为高等院校计算机及相关专业“软件工程”课程的配套课程设计教材，也可作为软件工程技术人员的参考读物。

版权所有，侵权必究。

本书法律顾问 北京市展达律师事务所

图书在版编目 (CIP) 数据

软件工程课程设计 / 吕云翔等编著. —北京: 机械工业出版社, 2009. 6
(高等院校计算机课程设计指导丛书)

ISBN 978-7-111-26829-1

I. 软… II. 吕… III. 软件工程—课程设计—高等学校—教材 IV. TP311. 5

中国版本图书馆 CIP 数据核字 (2009) 第 055728 号

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: 朱 劭

北京瑞德印刷有限公司印刷

2009 年 9 月第 1 版第 1 次印刷

186mm × 240mm · 9 印张

标准书号: ISBN 978-7-111-26829-1

定价: 19.00 元

凡购本书, 如有倒页、脱页、缺页, 由本社发行部调换
本社购书热线: (010) 68326294

高等院校 计算机课程设计指导丛书

专家指导委员会

(以姓氏拼音为序)

| | |
|-----|------------|
| 陈向群 | (北京大学) |
| 陈 鸣 | (解放军理工大学) |
| 戴 葵 | (国防科技大学) |
| 何钦铭 | (浙江大学) |
| 廖明宏 | (哈尔滨工业大学) |
| 林 闯 | (清华大学) |
| 刘振安 | (中国科技大学) |
| 马殿富 | (北京航空航天大学) |
| 齐 勇 | (西安交通大学) |
| 宋方敏 | (南京大学) |
| 汤 庸 | (中山大学) |
| 王立福 | (北京大学) |
| 吴功宜 | (南开大学) |
| 赵一鸣 | (复旦大学) |

联络人 温莉芳

丛书序言

近年来，我国在计算机应用、计算机软件和电子类相关专业的人才培养方面，取得了长足的进展，每年的毕业生都有数十万人。但是这些毕业生走进企业、公司、政府机构或研究单位之后，往往深刻地感觉到缺乏实际开发设计项目的经验，不善于综合运用所学理论，对知识的把握缺乏融会贯通的能力。

综合考察目前高等院校教学大纲、课程设置以及内容安排等方面的情况，多数学校还是比较重视训练学生的实际设计能力。但是，从安排设计实践的内容上看，基本上是围绕相关课程教学内容而展开的，不能够构成对实际问题的解决方案；从配套程序的规模上看，一般只是几十行到几百行的源代码，或者是一个单独电路的设计，远远小于一个小型项目的规模；从设计的结构上看，由于设计实践是围绕着课程教学内容而进行的，问题已经高度抽象，学生很难得到有关综合运用所学知识的整体训练机会。而且，这些内容相对简单、问题域已经高度抽象、规模较小的设计实践一人基本上就能完成，学生几乎无法通过这些设计实践，去真正获得有关项目管理和团队协作等方面的基本训练和工作经验。

由此可以看出，大多数学校对学生实际设计能力的训练与国外知名大学和国内精品课程相比较，还是存在一些差距的。为此，机械工业出版社华章分社和一批高等院校的教师，针对当前高等院校计算机硬件、软件和电子类相关课程教学中存在的问题，参考国内外知名大学相关课程成功的教学经验，设计编写了这套“高等院校计算机课程设计指导丛书”，其目的就是通过课程设计的一系列训练，把知识获取和项目实践两个方面有机地结合起来。

在这套“高等院校计算机课程设计指导丛书”中的每一门课程设计里，都安排了由多个子项目组成的一个课程设计项目。学生们可以在教师的指导下，逐步设计实现这些子项目，并最终完成一个功能相对完整，可以运行的系统，其代码可以是数千行，甚至上万行。通过这种设计课程，学生一方面可以结合课程的教学内容循序渐进地进行设计方面的实践训练，另一方面，在参与一系列子项目的实践过程中，还能提高如何综合运用所学知识解决实际问题的能力，以及获得有关项目管理和团队合作等众多方面的具体经验，增强对相关课程具体内容的理解和掌握能力，培养对整体课程知识综合运用和融会贯通能力。

参加丛书编写的各高等院校的教师都有着丰富的教学、科研，以及与企业合作开发项目等多方面的经验。每个课程设计中的子项目和整体项目，都来自教师们具体的科研和设计开发实践，所选设计项目与教学内容配合紧密，项目的难度与规模适宜。

最后，感谢机械工业出版社华章分社编辑们的大力支持，使出版有关这套丛书的计划，从单纯的构想演化成带有油墨芳香的真实。

丛书写作组

前 言

软件工程是研究软件开发和管理的一门工程学科，是计算机及相关专业的主干课程，软件从业人员必须了解该领域的相关知识。目前，软件工程的理论在我国企业和部门的软件开发中得到了广泛应用，各高等院校纷纷开设了软件工程课程，并出现了大批软件工程理论教材。与操作系统、编译原理等计算机主干课程一样，软件工程课程也有相应的课程设计，遗憾的是，介绍软件工程课程设计的实践类教辅书籍却很少。这与软件工程课程的地位是不一致的，也不符合重视实践的教学思想。本书正是为了满足这一需求而编写的。

在本书中，我们将结合软件工程课程设计项目——Kernel 会议管理系统，向读者介绍软件工程相关理论在实际项目中的应用。本书最大的特点在于可模仿性和可操作性强，读者完全可以依照本书的示例和讲解，完成自己的课程设计。

本书各章主要涵盖以下内容：

第1章 该章将介绍软件工程课程设计的选题、团队的组建等项目准备工作。面对众多选题，什么样的选题适合于软件工程的课程设计呢？读者将会在该章中找到答案。通过阅读该章，读者也将了解如何组建团队，如何组建一个成功的团队。

第2章 该章主要结合 Microsoft Project 介绍如何使用 CASE 工具来进行软件项目的计划和管理。读者将感受到 CASE 工具管理项目的方便性和高效性。

第3章 作为软件工程生命周期的起始点，我们应高度重视需求分析的复杂性和重要性。该章将结合 Kernel 会议管理系统，按照确定大体的项目方向、详细获取需求、讨论并确认需求、将需求文档化、整合需求规格说明书五个阶段向读者讲述一种比较合理的需求分析过程和方法。

第4章 该章将阐述软件设计的一般原则，“高内聚、低耦合”是我们在设计中要遵循的规范。功能模块设计和系统数据设计是软件设计的核心，该章将主要从这两个方面讲述软件设计的方法。

第5章 软件实现是软件工程过程中不可或缺的步骤。该章将从保证编码的质量和效率的角度来讲述软件实现过程中的一般方法和技巧。

第6章 软件测试是软件质量保证的关键所在。该章将结合 Kernel 会议管理系统实际采用的测试准备、单元测试、集成和系统测试三大阶段测试的工作来讲述软件测试的注意事项和方法。

第7章 用户手册是软件产品的最终用户使用软件的参考，对软件的易用性有较大影响。将思维的角度由开发人员转变为最终用户是本阶段的难点。另外，该章还将介绍用户手册应涵盖的内容以及写作的技巧。

第8章 配置管理贯穿于软件工程生命周期的全过程。好的配置管理能保证软件的完整性、一致性、可控性。该章将结合软件配置管理工具 SVN 向读者讲述配置管理的方法和技巧。

本书将采用以下的章节安排：对于软件工程过程中的核心环节（需求分析、软件设计、软件实现、软件测试），各章都分为基本概念、实施过程、项目文档三部分。“基本概念”部分对该章涉及的软件工程相关知识和理论加以简单回顾，以利于其他部分的阅读。“实施过程”部分则结合 **Kernel** 会议管理系统的实际操作过程，介绍构建系统所遇到的问题、相应的解决方案、心得体会，同时还会对读者的实际操作加以推荐。“项目文档”部分则是 **Kernel** 会议管理系统在该环节的实际文档，在文档前一般还会简要介绍文档应包括的内容、写作方法、注意事项等。该文档一方面作为“实施过程”部分的有力对照，另一方面也是为读者编写相关文档提供参考和模板。而对于其他章节，则采取的是比较灵活的方式，各章会根据其实际需要进行结构的划分。

参加本书编写的还有周俊、朱鑫、马识路。由于笔者经验有限，书中纰漏之处在所难免，还望读者批评指正。

编 者

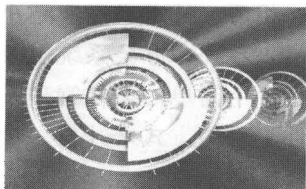
Email:yunxianglu@hotmail.com

2009 年 1 月

目 录

| | |
|------------------------------------|-----|
| 专家指导委员会 | |
| 丛书序言 | |
| 前言 | |
| 第 1 章 项目准备 | 1 |
| 1.1 项目选题 | 1 |
| 1.2 组建团队 | 2 |
| 1.3 团队工作方式 | 3 |
| 1.4 项目进度安排 | 3 |
| 第 2 章 项目管理 | 5 |
| 2.1 项目管理的范围 | 5 |
| 2.2 利用 Microsoft Project 对项目进行时间管理 | 6 |
| 第 3 章 需求分析 | 17 |
| 3.1 需求分析的基本概念 | 17 |
| 3.2 需求分析阶段的具体实施过程 | 17 |
| 3.2.1 确定项目的大体方向 | 18 |
| 3.2.2 详细获取需求 | 18 |
| 3.2.3 讨论并确认需求 | 18 |
| 3.2.4 以需求规格说明书为基点, 将需求文档化 | 19 |
| 3.2.5 整合需求规格说明书 | 26 |
| 3.3 Kernel 会议管理系统需求规格说明书 | 26 |
| 第 4 章 软件设计 | 35 |
| 4.1 软件设计的基本概念 | 35 |
| 4.2 软件设计的具体实施过程 | 36 |
| 4.2.1 功能模块设计 | 36 |
| 4.2.2 系统数据设计 | 40 |
| 4.2.3 需求迭代 | 46 |
| 4.3 Kernel 会议管理系统设计说明书 | 46 |
| 第 5 章 软件实现 | 59 |
| 5.1 软件实现的基本概念 | 59 |
| 5.2 软件实现的具体实施过程 | 59 |
| 5.2.1 程序的注释 | 59 |
| 5.2.2 规范化的源代码布局和命名规范 | 62 |
| 5.2.3 挖掘 IDE 的强大功能 | 66 |
| 5.2.4 软件的目录划分 | 68 |
| 5.3 Kernel 会议管理系统编码规范 | 69 |
| 第 6 章 软件测试 | 77 |
| 6.1 软件测试的基本概念 | 77 |
| 6.2 软件测试的具体实施过程 | 78 |
| 6.2.1 第一阶段: 测试准备阶段 | 79 |
| 6.2.2 第二阶段: 单元测试阶段 | 86 |
| 6.2.3 第三阶段: 集成和系统测试阶段 | 88 |
| 6.3 Kernel 会议管理系统测试报告 | 90 |
| 第 7 章 用户手册 | 105 |
| 7.1 一切从用户的角度出发 | 105 |
| 7.2 用户手册应该写些什么 | 106 |
| 7.3 编写用户手册的技巧 | 106 |
| 7.3.1 图文结合 | 107 |
| 7.3.2 操作截图 | 107 |
| 7.4 Kernel 会议管理系统用户手册 | 108 |
| 第 8 章 配置管理 | 121 |
| 8.1 配置管理的基本概念 | 121 |

| | | | |
|-------------------------------|-----|----------------------------------|-----|
| 8.2 为什么需要配置管理 | 121 | 8.4.2 日志和记录 | 127 |
| 8.3 配置管理的方式 | 122 | 8.4.3 上传操作文件之前一定要确保 正确性 | 128 |
| 8.3.1 一种原始的文件共享的方式 | 122 | 附录 A 软件工程课程设计题目 | 129 |
| 8.3.2 采用专业的软件配置管理 工具 | 123 | 附录 B 软件工程课程设计的 评价 | 131 |
| 8.4 配置管理需要注意的问题 | 126 | | |
| 8.4.1 一天一个版本 | 127 | | |



第1章

项目准备

对于软件工程课程的课程设计而言，项目准备阶段主要包括项目选题、组建团队、确定团队工作方式、制定项目进度等工作。

大多数院校的软件工程课程设计的的要求如下：题目自选；周期10周左右；学生划分为多个项目组，每组4~6人，确定1名项目经理，组员之间既分工又协作。这种形式的课程设计安排在各高校应该都是比较普遍的。

一般来说，课程设计开始后，我们首先要完成的是选题、组队工作，并确定团队的工作方式和开发进度。可以先选题，后组队；也可先组队，后选题。这要根据具体情况而定。如果学生有比较好的想法，可自行寻找自己认为合适的同学来组成团队。若是几位同学有过共同合作的经历，或是彼此熟悉，那么先建立一个关系融洽的团队，而后再确定选题当然也是可行的。

1.1 项目选题

软件工程有一套完整的理论和方法体系，当前绝大多数的软件都是按照这套体系来实施的，很多项目也因此取得了很好的成效。但问题是，软件工程的思想并不适用于所有软件开发项目，软件工程的具体理论和方法在实际应用中也是需要合理的变通的。在本书中，将多次强调这个观点。

那么，什么样的软件不适合采用软件工程的思想来开发呢？例如，下面几种软件都不适合采用软件工程思想来开发：

- 算法相对密集，性能要求较高，而功能需求很简单的；
- 一个裁剪图片的脚本；
- 类似于 Google 的搜索引擎；

.....

在选题时，应该避免选择这类软件。因为软件工程实际上是采用工程化的方法来开发软件，以提高团队开发的效率，保证开发软件的质量。对于非功能密集型、非流程复杂型的软件，是难以用工程化的方法和手段对其开发过程加以促进的。

一般说来，我们应该尽量选择功能需求较多、流程较复杂的软件开发选题。这样的软件项目采用软件工程的方式来开发，是非常合适的。作为学生的课程设计，一般是自主选题，因此我们有必要选择一个好的题目。选题是否满足上述的功能需求较多、流程较复杂的条件，是课程设计选题时的一个重要考虑因素。

由于软件工程课程设计的目的是让学生在实践去领会和理解软件工程，因此在课程设

计的选题问题上，应避免选择技术性较强，开发难度大的软件项目；相反，技术难度应越小越好，但项目的功能和流程应该能完整地体现实际要求。诸如小型信息系统、办公系统以及网上书店之类的软件项目一般都是符合上述需求的。本书中选择的是一个简单的会议管理系统，其主要的要求都集中在功能和流程上，技术难度也很小，用软件工程的方式来开发，可以说是麻雀虽小，五脏俱全。

选择了一个符合上述要求的题目后，我们还需要在可行性方面对选题进行一定的限定和优化。课程设计一般对团队人数、开发周期都有限制，我们需要依此来确定项目的规模和范围，以使项目能在规定的时间内顺利完成。学生的课程设计虽然不用像大型项目那样做一份专业的可行性报告，但我们也需要对软件的可行性进行一定的分析。对于在校学生，由于个人能力都有限，不可能熟悉所有的相关知识和技术，在考虑可行性的时候，主要应从团队成员能力的角度出发。对可行性考虑得不充分，通常会导致软件开发延期，从而无法在课程设计规定的开发周期内完成软件工程过程。学生团队因这个问题而导致最终开发失败的例子不在少数。在软件需求分析的过程中，还要对软件的规模和范围进行深入的定义，而在选题时，只需要一个大概的定义即可。

1.2 组建团队

在组建团队时，应注意三个问题：

- 1) 要按照软件工程要求的角色，寻找各方面我们认为优秀和适合的人选；
- 2) 要有团队融洽度的考虑；
- 3) 对团队成员的可工作时间要有充分的考虑。

第一点自然毋庸置疑了。比如美工、数据库等工作，如果之前没有相关的经验，是很难做好的。例如，如果一个团队缺乏一位擅长美工或有美工经验的成员，这很有可能直接造成项目最终在界面美化、用户舒适度方面存在一定的缺陷。因此，对于各角色，尤其是需要天分和经验的角色，一定要结合工作需要，寻找在这方面有能力、可出色完成任务的团队成员。

在选择成员时，要考虑该成员对团队融洽的影响。学生组队完成一项工作，一方面可锻炼能力，另一方面也可以结交朋友。团队成员间若能建立亲密的友谊，工作效率自然会高，同时减少很多不必要的麻烦。反之，队员间勾心斗角，最终是不欢而散，项目也以失败告终。例如，某个团队实际上并不是因为软件工程课程设计而临时组建的，在这之前队员已经合作完成了其他项目。大家在一起工作，感觉比较开心。于是，在软件工程课程设计中，就可能很自然地又走到了一起。

团队成员可工作时间的考虑对于本科学生团队来说尤其重要。本科生往往有繁重的课程和多样的社团工作和活动，同时还可能准备考研、出国或者承担实验室任务等。因此，其空闲时间并不多。而为了使自己的团队保质保量地按时完成课程设计，必须要充分考虑团队成员对于本项目的可工作时间。即使一个学生很优秀，能力完全满足甚至超出了项目需求，但如果他很忙（这种学生一般都很忙），对于本项目的可工作时间太少，那么可以毫不犹豫地放弃他。在 Kernel 团队的组队过程中，曾有非常优秀的学生主动要求加入，但考虑到他并

没有足够的时间参与到项目中，因此果断地拒绝了他的要求。

对于软件工程的课程设计，组建团队时还应包括明确的角色分工。软件工程课程设计的团队人数一般为4~6人。以Kernel会议管理系统为例，角色分为：项目经理、编码、测试、美工、数据库，其中编码角色2人，其他角色1人。读者也许会问，怎么没有需求人员和设计人员？这是不是不符合软件工程的要求？这里有必要向读者解释一下。学生团队与企业中真正意义上的团队还是有很大差距的。企业级团队成员一般是不固定的，各种角色并不存在于整个软件生命周期，这些成员完成某个项目的工作后就可能转去完成另一个项目。因此，参与某一软件项目的总人数会很多，但每个人的角色很单一，一般都只存在于软件生命周期的某一阶段。这样，自然会设置专业的需求和设计人员等。但学生课程设计的团队却不一样，由于教师出于考核方面的考虑，对团队人数有严格的限定，且每个成员都应自始至终地参与到项目中。因此，笔者认为，对于一个4~6人的团队，若单独分出人员做分析和设计等工作，会造成在整个软件生命周期中有一定的人员资源的浪费，同时成员无法参与软件工程的全过程，也就无法深入理解整个过程，这与软件工程课程教学大纲安排相应的课程设计等实践环节的主旨是相悖的。因此，秉承所有成员都能充分地参与到软件生命周期的全过程的宗旨，对于需求和设计等过程和环节，可以集体工作，在此过程和环节中忽略原本分配的角色，进行再分工。比如，当做需求分析的时候，团队成员都参与，然后每个人都有各自需求部分的分工。

1.3 团队工作方式

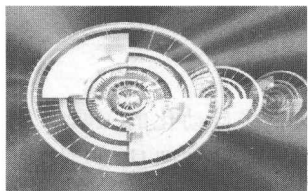
完成选题和组队后，就可以商定团队的工作方式了。对于学生，一般难以找到一个合适的独立环境来集体工作。图书馆、教室等公共场所由于不能讨论，不适合团队工作，而团队成员都在同一个实验室、教研室的概率几乎为零。因此，对于绝大多数的学生团队，采用“分布式”工作方式更为恰当。由于团队成员都是分开的，可能有的在图书馆工作，有的在宿舍工作，必然需要进行很好的沟通协调和工作同步。团队的交流可以采用QQ、MSN等即时通信工具完成；而工作的同步，则可以采用SVN等同步工具。这里只是提一下。会议是团队交流和沟通的一种重要方式。在本阶段，应该确定今后会议的地点、频率、时间等。一般说来，团队会议至少有两种：一是阶段性的会议，这应结合项目的进展情况，这种会议主要关注项目的向前推进速度；二是项目例会，项目应有周期性的固定的例会，主要是为了使团队成员能在工作上进行充分的交流。

1.4 项目进度安排

作为一个学生课程设计的团队，由于课程设计的时间一般都不会太长，所以必须严格把握时间。为了更好地控制时间，我们有必要对项目进度的安排做出详细的定义。项目进度安排是指对项目的进度、人员分工所作的项目计划。此计划主要是依据团队人员、课程设计时间、工作量估计以及成员对于本项目的可工作时间等因素而制订的。计划的生成方式建议采

用表格的形式。若采用工具（如 Microsoft Project 等）制定项目计划，则要将工具所生成的图表附于项目计划之中。例如，如果一个团队采用的是“分布式工作”和“集中式讨论”的方式，就可以采用表格的方式，且项目进度的安排可以以周为单位。因为软件产品的特殊性，在软件开发的过程中，会遇到很多预想不到的困难，这必然会影响到项目的进展，所以在制定项目进度的安排时，应该考虑到这一点。一般的处理方法是给整个项目留出一周的缓冲时间。具体到各个阶段的进度安排时，要从整体上把握，合理安排。对于需求分析阶段，一般来说，由于其重要性以及对后续阶段的持续影响性，所用的时间应该长一些，大约占整个项目的 30% 左右。但这不是必须的，我们应该根据实际情况做出合理的安排。软件设计会直接影响编码和测试的效率，对于任何一个项目来说都是十分重要的，这个过程的时间大约应占整个项目的 20%。如果软件设计足够优秀，那么编码和测试的效率和正确性将会很高，而且时间也较容易把握。一般来说，编码和测试大约占整个项目时间的 35% 左右。有了前面的工作，接下来的产品的包装和发布会变得比较容易，这个阶段大约占整个项目时间的 15%，其中应该包括一定的缓冲时间，以应对开发过程中所遇到的意外情况。

以 Kernel 会议管理系统为例，将需求分析安排在第 1 周，主要是进行需求的确认，对该项目的功能进行一些合理的裁剪和添加。对于软件设计，可安排 3 周的时间。对于编码可安排 3 周的时间。由于这个项目本身并不很大，所以测试安排 1 周时间即可。产品最后的包装与发布可用 1 周时间完成，而剩下的 1 周时间作为缓冲时间。整个项目为期 10 周。当然，这样的安排完全是根据项目本身的实际情况，可通过成员之间的讨论而最后决定，至于其合理性，将会在项目的开发过程中进行验证。我们在安排自己的项目时，要根据实际情况来安排各个阶段的进度，并不一定要完全按照某个项目的时间安排进行分配。但关键的一点是，项目进度制定以后，如没有特殊情况，我们必须严格按照该进度安排去执行。



第2章

项目管理

进行软件项目管理是一项复杂的工作，它是为了实现项目目标，运用相关的知识、技能、方法和工具，对项目的计划、进度、质量、成本、资源等进行管理、控制或协调的活动。有效的项目管理不仅可以节省项目的资金，同时也可以节省开发时间，缩短工期，提高软件产品的质量。虽然对于课程设计这种小型的软件项目而言，项目管理的地位显得不是非常重要，但是我们还是应该了解一些这方面的知识，学会使用比较简单的项目管理方法和软件工具，这样不仅有助于提高课程设计成果的质量，而且有利于我们以后的发展。

2.1 项目管理的范围

软件项目管理涉及多个领域，如下所述：

- 项目范围管理（SM）：确定并管理为完成项目所要做的全部工作。
- 项目时间管理（TM）：对项目所需要的时间进行估算，制定合理的项目进度计划，以确保项目能够及时完工。
- 项目成本管理（CM）：对项目成本的预算和管理工作。
- 项目质量管理（QM）：确定项目满足明确的需要。
- 项目人力资源管理（HM）：有效地分配参与项目的人员，并对他们的工作进行管理。
- 项目沟通管理（COM）：项目相关信息的产生、收集、发布和保存。
- 项目风险管理（RM）：识别、分析和应对项目相关风险。
- 项目采购管理（PM）：根据项目的需要，从项目执行组织外部获取或购进产品和服务。

图 2-1 是软件项目管理的框架示意图。

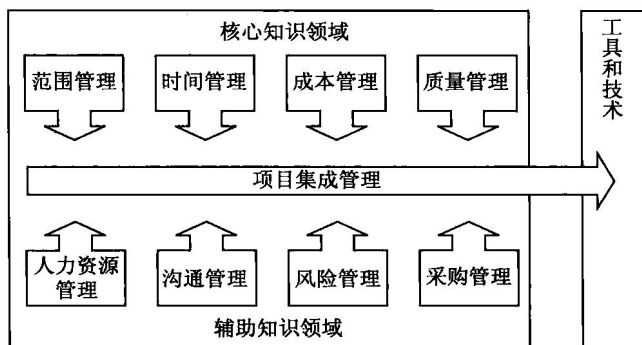


图 2-1 项目管理的框架示意图

2.2 利用 Microsoft Project 对项目进行时间管理

由于项目管理工作复杂，在实际的管理过程中，人们通常会借助一些自动化的工具，比如 Microsoft Excel、Microsoft Project 等。由于在软件项目开发过程中，项目经理是项目主要的领导者和管理者。所以，对于小型项目，最好由项目经理来完成包括相关软件操作在内的项目管理工作。对于本书中的示例项目，我们选用 Microsoft Project 进行项目管理，因为它是常用的计算机项目管理工具，可以进行项目控制和跟踪、详细的时间安排、关键路径分析等工作。具体来说，Project 实现的项目管理功能有：范围管理、时间管理、成本管理、人力资源管理、沟通管理和集成管理。考虑到软件工程课程设计的特点，这里将主要介绍时间管理的有关内容。下面以 Kernel 会议管理系统为例来介绍 Project 是如何工作的。

1) 新建项目文件，确定项目范围。

打开 Microsoft Project 后，首先应该新建一个项目。然后选择【项目】|【项目信息】，在弹出的“项目信息”对话框中填入相应的项目信息。这些项目信息就是项目的范围。如图 2-2 所示。

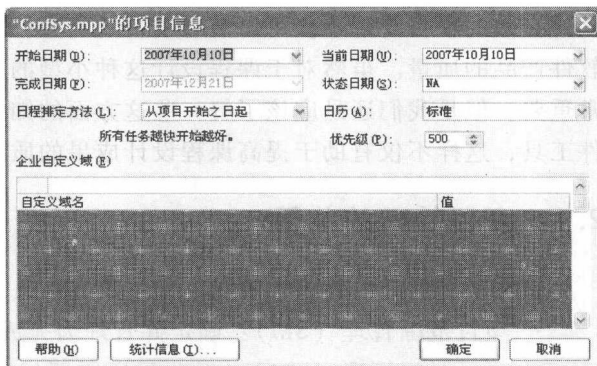


图 2-2 “项目信息”对话框

然后，可以对该项目的属性进行设定。选择【文件】|【属性】，然后在弹出的对话框里输入相应的信息，如图 2-3 所示。

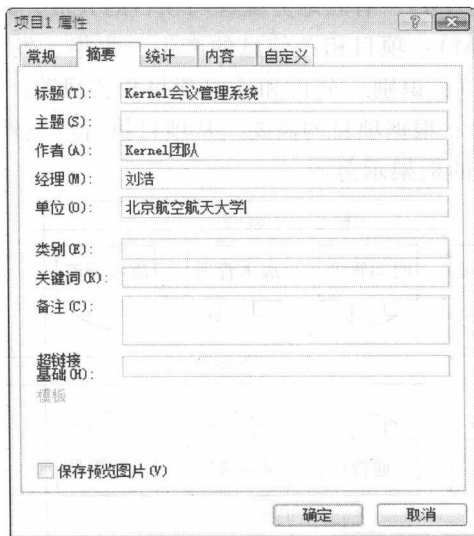


图 2-3 项目属性对话框

2) 创建任务并设定任务的属性。

新建项目后，Microsoft Project 窗口的内容区域显示的是一个任务计划表，因为目前还没有添加任务，所以是一个空表。从这个表格的第一行开始，逐一输入要完成的任务，构建任务计划表。如图 2-4 所示。在输入过程中，还可以使用“插入”功能。

| | 任务名称 | 工期 |
|----|------------------|--------|
| 1 | 项目启动 | 1 工作日? |
| 2 | 小组分工 | 1 工作日? |
| 3 | 需求分析 | 1 工作日? |
| 4 | 明确需求阶段的任务并分工 | 1 工作日? |
| 5 | 获取需求 | 1 工作日? |
| 6 | 初步确定需求 | 1 工作日? |
| 7 | 重新获取需求 | 1 工作日? |
| 8 | 最终确定需求 | 1 工作日? |
| 9 | 绘制系统的用例图 | 1 工作日? |
| 10 | 编写需求规格说明书 | 1 工作日? |
| 11 | 需求分析阶段结束 | 1 工作日? |
| 12 | 系统设计 | 1 工作日? |
| 13 | 明确设计阶段的任务并分工 | 1 工作日? |
| 14 | 设计系统的功能模块 | 1 工作日? |
| 15 | 设计系统的数据库并绘制对象关系图 | 1 工作日? |
| 16 | 编写设计文档 | 1 工作日? |
| 17 | 系统设计阶段结束 | 1 工作日? |
| 18 | 系统实现 | 1 工作日? |
| 19 | 明确实现阶段的任务并分工 | 1 工作日? |
| 20 | 编码 | 1 工作日? |
| 21 | 系统实现阶段结束 | 1 工作日? |
| 22 | 测试 | 1 工作日? |
| 23 | 明确测试的任务及分工 | 1 工作日? |
| 24 | 单元测试 | 1 工作日? |
| 25 | 集成测试 | 1 工作日? |
| 26 | 系统测试 | 1 工作日? |
| 27 | 编写测试分析报告 | 1 工作日? |
| 28 | 测试阶段结束 | 1 工作日? |
| 29 | 编写用户手册 | 1 工作日? |
| 30 | 项目结束 | 1 工作日? |

图 2-4 输入任务列表

输入的任务有多种类型。比如，图 2-4 中的任务 3“需求分析”是一项大任务，包含了任务 4 到任务 10 的全部工作，即需求分析阶段要完成“明确需求阶段的任务并分工”、“获取需求”、“初步确定需求”、“重新获取需求”、“最终确定需求”、“绘制系统的用例图”及“编写需求规格说明书”七项工作。实际上，可以把第 4 到第 10 个任务看成是“需求分析”阶段的各项活动。在 Project 里，把像“需求分析”这样的任务称为摘要任务，在图 2-4 中，摘要任务还有“系统设计”、“系统实现”、“测试”。在 Project 里，可以用缩排的方式突出摘要任务。选中任务 4 到任务 10，并在工具栏中选择【项目】|【大纲】|【降级】，如

图 2-5 所示。然后“需求分析”就自动变为黑体，在右侧的甘特图中，相应的矩形框也改变了形状和颜色。



图 2-5 设定摘要任务

利用此方法，把“系统设计”、“系统实现”、“测试”都设定为摘要任务。设定摘要任务后的视图如图 2-6 所示。

| 任务名称 | 工期 | 开始时间 | 2007年10月7日 |
|---------------------|--------|-----------|-----------------------|
| | | | 日 四 五 六 日 一 二 三 四 五 六 |
| 1 项目启动 | 1 工作日? | 2007年10月 | |
| 2 小组分工 | 1 工作日? | 2007年10月 | |
| 3 需求分析 | 1 工作日? | 2007年10月1 | |
| 4 明确需求阶段的任务并分工 | 1 工作日? | 2007年10月 | |
| 5 获取需求 | 1 工作日? | 2007年10月 | |
| 6 初步确定需求 | 1 工作日? | 2007年10月 | |
| 7 重新获取需求 | 1 工作日? | 2007年10月 | |
| 8 最终确定需求 | 1 工作日? | 2007年10月 | |
| 9 绘制系统的用例图 | 1 工作日? | 2007年10月 | |
| 10 编写需求规格说明书 | 1 工作日? | 2007年10月 | |
| 11 需求分析阶段结束 | 1 工作日? | 2007年10月 | |
| 12 系统设计 | 1 工作日? | 2007年10月1 | |
| 13 明确设计阶段的任务并分工 | 1 工作日? | 2007年10月 | |
| 14 设计系统的功能模块 | 1 工作日? | 2007年10月 | |
| 15 设计系统的数据库并绘制对象关系图 | 1 工作日? | 2007年10月 | |
| 16 编写设计文档 | 1 工作日? | 2007年10月 | |
| 17 系统设计阶段结束 | 1 工作日? | 2007年10月 | |
| 18 系统实现 | 1 工作日? | 2007年10月1 | |
| 19 明确实现阶段的任务并分工 | 1 工作日? | 2007年10月 | |
| 20 编码 | 1 工作日? | 2007年10月 | |
| 21 系统实现阶段结束 | 1 工作日? | 2007年10月 | |
| 22 测试 | 1 工作日? | 2007年10月1 | |
| 23 明确测试的任务及分工 | 1 工作日? | 2007年10月 | |
| 24 单元测试 | 1 工作日? | 2007年10月 | |
| 25 集成测试 | 1 工作日? | 2007年10月 | |
| 26 系统测试 | 1 工作日? | 2007年10月 | |
| 27 编写测试分析报告 | 1 工作日? | 2007年10月 | |
| 28 测试阶段结束 | 1 工作日? | 2007年10月 | |
| 29 编写用户手册 | 1 工作日? | 2007年10月 | |
| 30 项目结束 | 1 工作日? | 2007年10月 | |

图 2-6 创建摘要任务后的视图

3) 为每个任务输入工期。

在输入任务的过程中，Project 为每个任务设定了默认工期，即“1 工作日”。可以在“工期”栏中改变任务的工期，也可以通过设定“开始时间”和“完成时间”来自动设定任务的工期。在输入“开始时间”的时候，如果两个任务之间的“开始时间”和“完成时间”比较接近，那么会自动出现“规划向导”对话框，如图 2-7 所示。