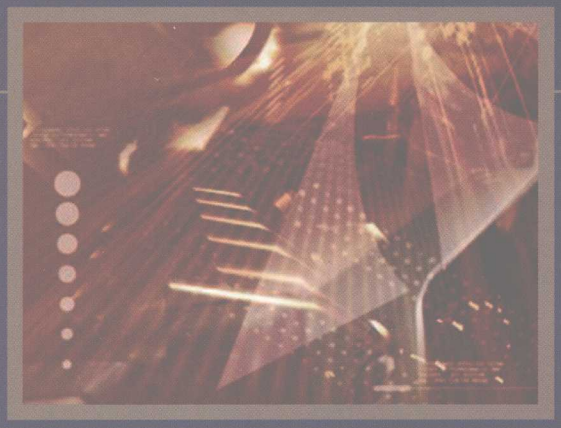
高等学校计算机科学与技术教材

综合应用软件设计

□ 曹渠江　主编



- 原理与技术的完美结合
- 教学与科研的最新成果
- 语言精炼，实例丰富
- 可操作性强，实用性突出

清华大学出版社　北京交通大学出版社

高等学校计算机科学与技术教材

综合应用软件设计

曹渠江 主 编

清华大学出版社
北京交通大学出版社

·北京·

内 容 简 介

本书是为计算机、信息管理等相关专业高年级本科生在开设相关课程时所编写的一本教学参考书，它将全面指导学生学习、熟悉和运用当今被广泛采纳的一些主流技术去进行综合应用软件设计的有效实践。

本教材的主要内容：根据课程项目的要求，运用软件工程的理论，进行需求分析及概要设计；制订软件开发进度；运用面向对象程序设计、计算机网络、数据库原理、多媒体技术等相关知识，运用系统软件代码的开发、集成测试、修改、完善与总结等相关技术，全面实现数据库的设计。

本教材是上海理工大学计算机工程学院曹渠江教授连续七届的教学实践经验的积累，也是上海市教委重点课程建设项目的教学研究成果。

本书可作为高等院校相关课程的参考教材，也可供从事计算机软件开发的科技人员、工程技术人员及相关部门人员参阅。

图书在版编目（CIP）数据

综合应用软件设计/曹渠江主编. —北京：北京交通大学出版社，2008.11
（高等学校计算机科学与技术教材）
ISBN 978－7－81123－449－7

Ⅰ.综…　Ⅱ.曹…　Ⅲ.应用软件-程序设计-高等学校-教材　Ⅳ.TP311.1

中国版本图书馆 CIP 数据核字（2008）第 177356 号

责任编辑：杨正泽
出版发行：清华大学出版社　邮编：100084　电话：010－62776969　http：//www.tup.com.cn
　　北京交通大学出版社　邮编：100044　电话：010－51686414　http：//press.bjtu.edu.cn
印 刷 者：北京交大印刷厂
经　　销：全国新华书店
开　　本：185×260　印张：19.25　字数：500 千字
版　　次：2008 年 12 月第 1 版　2008 年 12 月第 1 次印刷
书　　号：ISBN 978－7－81123－449－7/TP·448
印　　数：1～3 000 册　定价：29.00 元

前　言

自20世纪70年代人们提出“软件危机”（Software Crisis）以来，软件工程的发展已经经历了三十多个春秋。对于一门学科而言，30年不算短也并不算长。

在三十多年的时间里，软件工程获得了迅速的发展，软件工程的研究内容得到了极大的丰富，软件工程的分支如雨后春笋般不断涌现、蓬勃发展。特别是20世纪90年代以来，在管理科学领域新观念不断兴起的大背景下，软件工程作为一门学科，其新兴子学科不断涌现。全面质量管理（Total Quality Management）的浪潮推进了软件质量管理（Software Quality Management）的发展。流程观念的兴起促成了软件过程管理（Software Process Management）在当今IT业界方兴未艾的局面。受企业业务流程再造（Business process re-engineering）、结构重组（Reorganization）等管理科学中“Re”浪潮的影响，新兴的软件再造工程（Software Reengineering）也应运而生。体现于敏捷组织（Agile Organization）和敏捷制造（Agile Production），这些新生事物中的敏捷思路指引了软件工程中敏捷模式（Agile Model）的发展。当还原论逐渐退出历史舞台而系统论粉墨登场之时，学界已经逐步确立了复杂的、非线性的世界图景，受其启发，软件工程领域也逐渐开始了对其思维方法的逻辑起点进行扬弃的尝试：自适应软件开发方法（Adaptive Software Development），基于代理的计算方法（Agent Based Computing）都在呼唤着软件工程领域另一场革命性变革的到来。与此同时，软件行业也在不断吸取系统工程中一些定量的方法，越来越多的系统分析与设计方法被运用到软件工程的实践之中。柔性管理的浪潮也对软件工程的发展起到了深远的影响，软件工程由传统的重视软件质量的重载软件过程模型逐渐过渡到以人为本的轻载软件工程模型与之鼎立的状态。

传统的瀑布（Waterfall）软件过程模型中的各分支在这一时期也得到长足发展，具体表现在其分类逐渐细化，其规模不断庞大。瀑布模型中的可行性分析环节中的经济可行性分析已经发展成为一门独立学科软件经济学而备受业界重视。传统的需求分析从软件开发过程的一个主要环节发展成独立需求工程、领域分析等各类专题。传统的由概要设计、详细设计组成的软件设计环节也逐渐分类细化为架构设计、数据设计、接口设计、流程设计、功能设计等门类丰富的设计类别；设计的描述方式也从传统的简单的图表、伪码描述转变到各类严格而灵活统一的建模语言的描述，其中较为典型的是统一建模语言UML（Unified Modeling Language）。此外软件设计环节也涌现出大量新的技术或方法：构件技术为软件的可复用、可移植提供了可能；模式、框架的兴起为设计经验的复用提供了可能；正向工程的发展为设计向软件构造的转化带来了极大的方便。软件的构造也从单纯的软件编码发展到包括编码技术、调试技术、集成开发环境的研究、GUI与功能分离等各项专题研究在内的崭新阶段。测试技术的内容已经不仅仅是传统的白盒、黑盒方法，它已经发展为包括一套完善的测试理念、方法、过程、工具、人员组织等众多内容的完整系统，这一点又较为明显地体现在：最近测试先行理念的兴起，测试自动化工具的普及。这些测试工具也不仅仅只是功能性

测试工具，门类众多的非功能性测试工具在这一阶段也都获得了巨大的发展，包括并发测试工具，界面自动测试工具，压力测试工具等。软件的维护也不仅仅是传统的修正性、适应性、完善性维护等单薄的内容，它已经发展成为包括遗产系统（Legacy System）的继承话题在内的内涵丰富的完整体系。

软件工程的理论与方法在这一时期也取得了革命性的突破。面向对象的分析设计方法逐渐取代了传统的结构化分析设计方法，成为当前软件开发方法的主流。而正当人们沉浸于面向对象方法的浪潮中时，在新世纪初，基于代理（Agent）的思维方法与计算方法似乎又在酝酿计算机界的另一场变革；计算机辅助软件工程的研究，已经从理论逐渐走向实践，它与新的软件工程学科分支如需求工程（Requirements Engineering）、领域工程（Domain Engineering）、正向工程（Forward Engineering）、反向工程（Reverse Engineering）、基于构件的软件工程（Component Based Software Engineering）相得益彰，极大地推进了软件工程方法论、软件工程辅助工具的丰富；组织理论、团队理论、知识管理论、质量管理论、过程管理论、内容管理论等都为软件工程在新世纪的发展提供了新的动力。

在这三十多年的时间内，软件工程的研究也不仅仅局限于计算机技术、管理科学的研究范围，在软件工程与社会科学及人文科学的交叉领域也取得了一定的理论与实践成果，为工程实践提供了很多实际的参考价值。软件工程与人体工程学、技术美学的相结合导致了人机交互研究的兴起（对人机交互的研究在国外已经受到极大的重视，但在国内这方面作得还远远不够）。此外在软件工程领域，人们也越来越意识到软件工程中的知识产权问题，软件开发过程中开发人员的心理因素对软件开发的影响问题，软件工程的相关职业道德的建立等问题。这些问题的研究对促进软件工程健康发展起到了积极的作用。

然而30年的时间相对于有着上百年历史传统的自然学科而言，也实在太短。因此，从一定程度上说，相对于当代复杂的市场环境而言，软件工程充其量不过是羽翼尚未丰满、血气尚未刚强的少年。它的骨架已经基本形成，但血气还需充实，骨肉还需健壮。因此它还有充分发展的必要，这种发展一方面是对一些不良习性的改正，尚待完善的习性的改良；另一面是积极培养未知的良好习性的形成。同时，软件工程在新世纪也有着充分的发展空间。经济全球化是不可避免的趋势，而经济全球化先决条件是信息在全球范围内的共享与交流，形形色色的软件产品，包括企业内部各类管理信息系统、辅助决策系统、企业对外的用于各种目的的Web系统，无疑是当今世界信息的最大载体。如何分析、设计、实现这些日益复杂的软件系统，正是软件工程要解决的主要问题。因而，全球化浪潮在激励软件行业的不断发展的同时，也为其茁壮成长提供了丰富的养分。所以软件行业必将在未来一段时间内获得巨大发展。

在软件工程发展的时代背景下，国内的软件工程也有了长足的发展。这一点尤为明显地体现在我们手头可以容易地接触到大量的与软件工程相关的书籍。从数量上来说，国内软件工程相关书籍的确已经相当丰富，但质量上，总体而言还存在着一定的不足。

一方面，软件工程发展相当迅速，新的理论方法不断涌现，而国内的很多教材讲授的还是20世纪90年代之前的软件工程理论与方法。显然已经跟不上形势发展的需要。因而国内的软件工程读物很有必要推陈出新以不断向读者补给新鲜血液。

国内也有众多针对国外最新的软件工程理论方法做不同程度介绍的书籍，但这些书籍总的来说都侧重于软件工程的某些方面，读者很难通过阅读这类有限的书籍达到对软件工程宏

观的把握。很多书籍对某些方面知识点的讲解是很透彻的，比如介绍使用 UML 的书籍，读者读完后，对 UML 的细节可能有所掌握，但读者可能很难将其在整个软件工程学科中加以定位。也就是说读者在阅读了这类读物后，可能只见树木，不见森林。因而很有必要对这些分散的知识点加以整合，从宏观层次上帮助读者建立整个软件工程的知识体系结构。

现在市面上也有很多冠之以软件工程名义的读物，其中不乏精品。但总体来说，这些读物，要么精于理论阐述，要么精于实践描述，却很少有将二者有机融合的读物。对于刚刚接触软件工程的读者来说，与实践脱节的理论读物未免乏味，与理论脱节的实践描述却又难以提升理论品位。在象牙塔内，技术与理论脱节情况尤为严重：数据库理论中的很多内容在实践时学生不知如何运用，更难说得上理论结合实践；软件工程课程几乎被学生当作政治理论课，认为只要背诵就能对付考试，固然应对考试应该没什么问题，但学生却因此失去难能可贵的掌握理论、锻炼实践能力的机会，实在可惜。因而很有必要推出能够将软件工程的理论与实践有机融合的读物。

国内也翻译或影印了不少国外软件工程实践的书籍，但是这些书籍中讲解的很多理论与方法只是适应国外的特定环境，在国内的一些特定的软件开发环境下，这些方法或者需要调整或者需要创新。所以关键在于灌输一种“渔”的方法，而非简单地阐述某一开发过程或者某一特定理论。例如，讲解软件过程管理，重要的不是将哪一种软件过程改进模型当作教条加以宣扬，而是阐述软件过程管理的基本内容、基本框架、基本原则。如果将重点放在介绍某一种模型上，读者很可能先入为主，形成偏见，有碍于客观的整体知识结构的形成。

正是基于以上对国内软件工程领域读物现状的认识，我们开始了编写本书的尝试。编者的初衷是为本科阶段的学生学习软件综合课程设计编写一本辅导书籍，但本书编写时的定位，却是希望能够为上述问题的解决贡献一份微薄的力量。诚然，软件工程知识体系结构庞大繁杂，精于一隅已属不易，毋用说整理总体的知识框架并且将理论结合实际了。所以本书很大程度上是一次尝试与创新。但毕竟由于时间、编者学识等各方面的因素，疏漏之处，在所难免。因而，正如软件质量评价的一个很重要标准是可扩展性，本书在编写之初，在内容结构的安排上，也充分考虑到将来的扩展。由于各方面因素的限制没有来得及安排上的内容，均可以在将来得以有机的衔接。

本书本质上是一本针对软件工程的引导性读物，旨在帮助读者建立最新的、关于软件工程知识体系的宏观视野，为读者带来实践软件工程时最为实用的指导。因而本书比较适用于刚刚接触软件工程，希望对软件工程有一个整体性了解的读者；也比较适合于正在寻找一些软件工程实践时最为实用的一些指导的读者。这些指导可能帮读者解决以下问题：不知如何进行计划；不知如何进行数据库设计；不知怎么进行软件的需求分析；不知如何解决 .NET 编程时的某一个特定技术问题；等等。

本书基本上是按照以下行文思路来编写的。先以软件工程知识体为指引，构思全书。第 1 章全面介绍软件工程知识体，重点介绍了软件过程管理，一些最新的过程管理模型，如 CMM、XP 均有涉及，同时本书还倡导软件质量管理的理念。第 2 章对软件工程管理的几个最为主要的内容作了一定的介绍，重点介绍了一些对于学生来说较为实用的管理方法和思路。第 3 章和第 4 章分别就软件工程过程几个重要阶段中的软件需求分析、软件设计作出了详细介绍，对当前的两大主流分析设计方法：结构化分析方法、面向对象的方法均作了一定程度的介绍。第 5 章是数据库的分析与设计，将数据库的各类理论运用于分析设计的实践，

并尝试了数据库设计模式的探索。第6章对.NET框架下的软件构造过程给出了详细的描述，同时针对刚刚接触.NET时的常见问题给出了解答。第7章详细地介绍了软件测试技术，在阐述一些先进的测试理念的前提下，特别突出了自动化软件测试工具的使用。在本书的最后，第8章详细描述了一个学生团体实际的软件过程。

本书在编写过程中力求体现以下几个的特点。

1. 新颖

本书参考的绝大部分资料来自2000年后出版的关于软件工程方面最新的书籍，很多概念的引述直接来自于IEEE、ACM、SEI、PMI等权威组织的最新定义，其中包括IEEE、ACM、Rational等多家组织或单位共同推出的Guide To the Software Engineering Body of Knowledge（2001年3月版，简称SWEBOK），包括Project Management Institute推出的A Guide to the Project Management Body of Knowledge（2000年版，简称PMBOK）。

本书的很多的材料来自于互联网各类技术论坛、社区、技术宣传网站的第一手资料，其中包括很多经典的分析设计案例及实际开发过程中用到的过程规划、时间质量控制计划、软件书写规范、各类文档模板。

本书的行文思路也打破了传统的软件工程教材的思路，以SWEBOK中论及到的软件工程知识体系结构为骨架和指引，将软件工程中众多分散的概念有机地组织在一个总体的框架之下，这个框架就是SWEBOK为读者描述的软件工程知识体。

在内容方面，本书对软件工程的最新事物，都作了一定程度上的介绍。与国内的传统软件工程教材相比，本书尤为突出地介绍了软件过程管理（一些最新的过程管理模型，如CMM、XP均有涉及），同时本书强调了软件质量管理的理念。在软件工程方法的介绍上，结构化方法和现在日益成为主流的面向对象的方法并重。关于测试，本书在介绍传统测试技术的同时，更侧重于向学生灌输了一些新的实用测试理念，如测试先行、测试自动化等全新理念。

本书中分析设计实例主要针对当今主流的分析设计方法、UML的分析设计方法来介绍，对传统结构化分析设计方法也有介绍。本书中所有编码实例均是基于.NET平台，数据库服务器使用SQL Server 2000，这是当今主流的软件实现平台。

2. 全面

本书侧重于学生总体知识框架的建立，因而对软件工程的众多基本概念均作了一定程度介绍，当然由于篇幅所限，很多概念不可能一步到位，但对这些粗略提到的概念均给出了参考书籍。

本书在对概念作基本介绍的同时，更给出了很多实际的例子，这些例子涉及软件的管理、分析设计、技术实现。

本书既介绍了传统的经典的软件工程模型，也介绍了最新的软件过程发展。既讲解了传统的结构化的方法，也介绍了新的面向对象的方法。

本书按照软件工程知识的架构，在各章分别对各项知识点作了介绍，并在此基础上给出了完全由学生完成的软件工程实践的综合实例。

3. 实用

本书罗列的各类技术问题及其解决方案，均来自于第一线软件开发组织的最新的实际问题，其中，绝大部分来自于上海理工大学曹渠江实验室几年来在.NET平台下实际项目的

知识积累以及学生课程设计辅导的经验总结。应该说这些问题都是使用 .NET 框架时最有可能遇到的最新的问题，这些方案都是较为典型的实用方案。

本书的软件实现一章，在理清基本概念的同时，以学生软件开发过程中经常遇到的众多基础技术问题为核心，以解答问题的方式讲解技术，以方便学生查阅。在附录部分，对入门级的学生可能遇到的疑难，给出了详细解答。

本书的附录部分也是本书的一大特色，这部分给出了很多实用的互联网资源的连接，很多实用的简洁的文档模板，实际软件开发过程中的编码规范、设计规范等。

本书是上海理工大学计算机工程学院曹渠江教授连续七届的教学实践积累，也是上海理工大学精品课程与上海市教委重点课程建设项目的教学研究成果。本书内容的多次充实、编辑与修订得到了曹渠江教授的历届研究生的大力支持，在此向已毕业的肖仰华、叶文珺、王宇鹏、周树明、胡震宇、张溶冰、张博、张凤、陈洁、俞心禹、施佺、应脂、卢彬、王伟良、杨希亮、李进京、耿少峰、赵健和施振佺等同学表示衷心感谢。上海理工大学计算机学院 2001 届中国爱尔兰合作班学生邹怡、王庆玮、徐吟晖、杨晨、陈佳韵、曹佳樑出色地完成综合应用软件设计的教学任务，并书写了完整的项目报告，构成了本书第 8 章的主要内容，在此向他们表示感谢。曹春萍老师为本书各章编写了思考题，并参与了部分章节的审阅工作，在此也一并表示感谢！

限于水平，书中难免有欠妥之处，欢迎广大读者和专家批评指正。

编　者

于上海理工大学

目　录

第1章 软件工程概述

当明确了要去实现一个软件时，就必须知道究竟应该怎样实现这个软件，至少应该弄明白下列问题：

① 这个软件值得开发吗？现有的技术水平能实现用户需要的软件吗？

② 怎样与客户沟通从而知道客户要求什么？又如何描述已知的要求？

③ 怎样才能将用户用文字描述的系统转换成最终的软件？

④ 要用到数据库吗？若要，数据库又怎么设计呢？

⑤ 使用哪一种开发工具？使用 Windows 风格的界面，还是 Web 风格的界面？

⑥ 与合作伙伴怎样进行分工合作？

⑦ 软件整个开发过程应该是什么样的？

只有在真正地解决了诸如此类的这些问题后，才能进入开发过程，也只有这样，才有可能实现预期的软件。而这些在软件开发之前和软件开发过程中摆在设计人员面前的问题，就是软件工程需要解决的问题。

本章重点介绍一些软件工程的基本概念，帮助同学们整理出软件工程的总体知识框架，理清学习软件开发的思路。

1.1 软件工程的基本概念

当人们第一次听说软件工程（Software Engineering）时，第一个联想到的可能是建筑工程、水利工程等等其他类型的工程。再进一步推理，软件工程应该与建筑工程等有相似之处，因为都含有“工程”二字，但它们必定也有不同之处，因为软件工程突出了“软件”而非其他。所以要理解软件工程，必须先弄明白什么是“软件”（Software）。

1.1.1 软件

何谓软件？虽然人们一直把这个名词挂在嘴边，但试图给其下一个全面而严格的定义绝非易事。本书不将笔墨纠缠于概念的面面俱到的学究式阐述上，而是在博采众家之长的基础上稍加筛选，但求以点盖面，起到梳理概念，引导入门的作用。所以，对于软件的定义，只想纠正一个错误——软件就是程序。

“软件就是程序”是错误的，软件是程序和实现这些程序所需要的所有文档的集合。

程序仅仅是软件的一部分，软件更为重要的部分是人们刚开始做软件时最不愿意做的文档。这些文档包括可行性分析文档、需求分析文档、设计文档、测试报告、用户手册、培训计划、软件实施方案、项目管理文档等所有为了实现这个软件所需要的文档的集合。

1.1.2　软件工程的定义

20世纪70年代，随着计算机应用的日益普及，软件系统日益复杂，日益庞大，这时出现了软件的高成本投入与低质量回报之间的尖锐矛盾，此现象被称为“软件危机”。

为了解决这对日益尖锐的矛盾，人们尝试将传统工业工程上的一些系统分析、设计方法应用于软件开发的过程之中，以期待在预期的时间内以低成本的投入完成较高质量的软件开发。此尝试促成了软件工程的诞生。

IEEE对软件工程给出了如下定义：①The application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software; that is, the application of engineering to software. ②The study of approaches as in ①。

这一定义突出了软件工程作为工程的特点，也突出了软件工程的本质特征——软件的工程化。

1.1.3　软件工程的目标

从IEEE给出的软件工程的定义中，可以看出软件工程最终要实现的目标是：在预期的时间内，以较低成本的投入完成较高质量的软件。

在目标中，有3个最为重要的因素：时间，成本，质量。这3个主要因素之间的关系如图1-1所示。

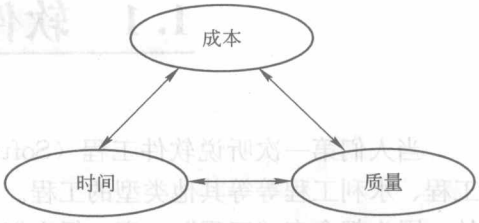


图1-1　制约软件工程的3个要素

可以从以下几个角度来衡量软件的质量。

1. 可维护性（Maintainability）

软件的维护是指在软件交付使用的过程中，对软件做出的修正、完善和适应性修改。软件的可维护性就是指软件在维护方面的性能，一般而言包括软件的可读性（Readability）；软件的可修改性（Modifiability）和软件的可测试性（Testability）。软件的可读性用来衡量软件是否容易理解。软件的可修改性用来衡量软件是否容易修改。软件的可测试性用来衡量软件是否容易测试。

软件的维护是延长软件生命周期，降低软件开发成本的关键。因此软件的可维护性是一个非常重要的指标，近年在软件投资中的比例也有日益上升的趋势。

2. 可靠性（Reliability）

软件的可靠性，通常包括软件的正确性（Correctness）和软件的健壮性（Robustness）两个方面。软件的正确性是指软件系统自身没有错误，能够在预期环境下完成预期功能。软件的健壮性则是从相反的角度进行衡量，描述软件，处理异常情况的能力。软件的正确性和

健壮性是同一个问题的两个方面，它们相互补充，共同构成了软件的可靠性。

3. 可理解性（Understandability）

软件的可理解性是指软件的简明性和清晰性。这里的简明清晰涉及程序、文档及最终实现的软件系统的界面。

4. 性能（Performance）

软件的性能包括运行该软件时所需要的时间和空间；软件的响应时间（Response Time）和软件的吞吐率（Throughoutput）。软件系统的响应时间是指从客户发起请求到软件系统对客户的请求作出响应的时间。软件系统的吞吐率是指单位时间内软件系统执行的作业数。

5. 可扩展性（Extensibility）

软件的可扩展性是指软件为了适应用户需求的变化从而不断扩展的能力。可扩展的软件系统应该尽可能的模块化，以保证新添加的功能模块对已有的软件系统不会带来负面影响。同时可扩展的软件系统必须具有较高的可适应性，以方便替换已经存在的模块。

6. 可伸缩性（Scalability）

可伸缩性是指软件系统能够适应系统不断发展的能力。可伸缩的系统能够将系统未来的发展考虑在内，可以适应企业不断发展的需要，以及数据量和系统使用量不断增大的情况。

7. 安全性（Security）

软件的安全性主要用来描述软件在受到有意或无意的攻击时，或者在出现一些技术故障的情况下，仍然能保持正常运行的能力。

随着软件工程的发展，软件质量的评价越加丰富，内容不断充实。近年来讨论较多的还包括软件的可复用性（Reusability），用以描述软件的可重用能力；软件的互操作性（Interoperability），用以描述子软件系统在大系统环境下的可集成、可嵌入的能力；软件的可移植性（Portability），用以描述软件与其环境之间的相互独立性；软件的多相性（Heterogeneity），用以描述软件在跨平台，多种开发语言，多个开发环境下的构造能力。

1.2　软件工程的知识体系结构

现代软件工程作为一门学科，有着极为丰富的内容，它包括软件开发技术和软件工程管理两大部分内容，同时为了实现这两者的自动化，还包括软件自动工具的研究。

1.2.1　软件开发技术

按照软件生命周期模型，软件开发技术的研究主要包括软件需求分析、软件设计、软件构造、软件测试与软件维护，软件生命周期模型如图 1-2 所示。

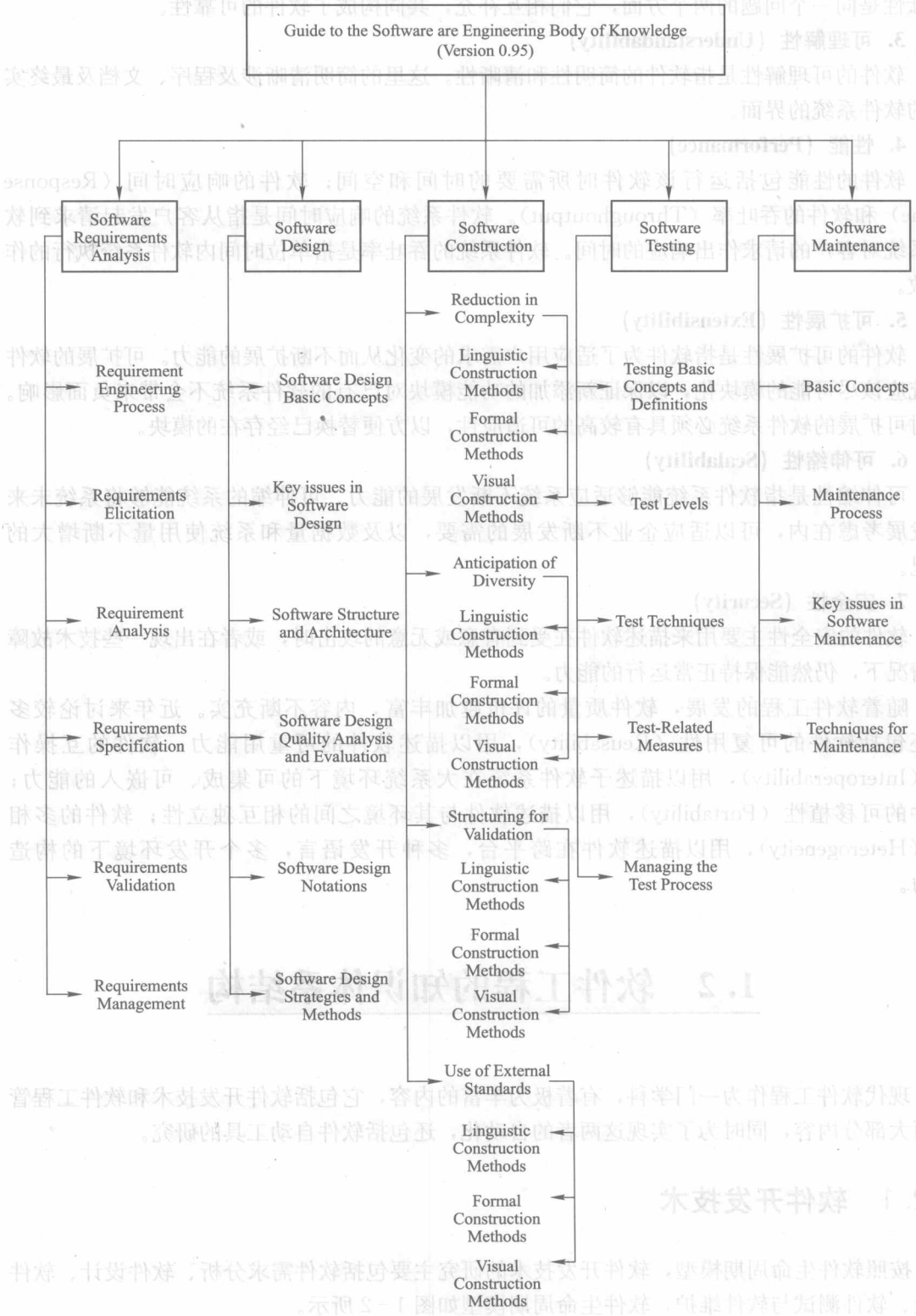


图 1-2 软件生命周期模型