HZ BOOKS
华章教育

国家示范性软件学院系列教材

“十二五”普通高等教育本科国家级规划教材

韩万江 姜立新 编著

软件工程案例教程

软件项目开发实践

第3版

Software Engineering
A Case Study Approach

机械工业出版社
China Machine Press

国家示范性软件学院系列教材

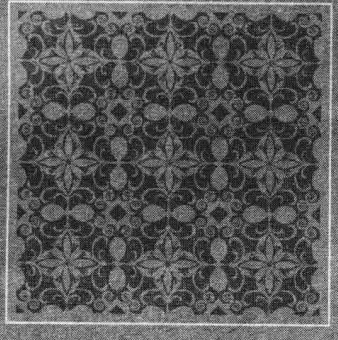
本科国家级规划教材

韩万江 姜立新 编著

软件工程案例教程

软件项目开发实践

第3版



Software Engineering
A Case Study Approach

RFID

机械工业出版社
China Machine Press

图书在版编目（CIP）数据

软件工程案例教程：软件项目开发实践 / 韩万江，姜立新编著．—3 版．—北京：机械工业出版社，2017.3
（国家示范性软件学院系列教材）

ISBN 978-7-111-55984-9

I. 软… II. ①韩… ②姜… III. 软件工程－案例－高等学校－教材 IV. TP311.5

中国版本图书馆 CIP 数据核字（2017）第 030305 号

本书以一个贯穿始终的软件项目案例为基础，讲解软件项目开发中需求分析、概要设计、详细设计、编码、测试、产品交付以及维护等各个过程中涉及的理论、方法、技术、交付的产品和文档等。本书系统、全面、注重实效，可以帮助读者在短时间内掌握软件项目开发的基本知识和基本过程，并有效提高实践能力。

本书既适合作为高等院校计算机及相关专业软件工程、软件测试课程的教材，也适合作为广大软件技术人员的培训教程或参考书。

出版发行：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：曲　熠　　责任校对：董纪丽
印　　刷：北京文昌阁彩色印刷有限责任公司　　版　　次：2017 年 3 月第 3 版第 1 次印刷
开　　本：185mm×260mm　1/16　　印　　张：21.25
书　　号：ISBN 978-7-111-55984-9　　定　　价：45.00 元

凡购本书，如有缺页、倒页、脱页，由本社发行部调换
客服热线：（010）88378991　88361066　　投稿热线：（010）88379604
购书热线：（010）68326294　88379649　68995259　　读者信箱：hzjsj@hzbook.com

前 言

本书第1、2版出版后得到了广大读者的好评，被众多高校选为教材，也收获了很多反馈，其中既有热情的赞扬，也有很多中肯的建议，在此表示深深的感谢。参考这些建议，同时结合近年对软件工程理论新发展的研究，以及多年的教学经验和项目实践，我们对第2版进行了全面修订。第3版的主要更新之处包括：面向软件工程新技术，总结了软件开发实践的过程、经验和方法；重新甄选项目案例，并对这些案例进行了精心整理。本书是理论与实践相结合的典范，每章都有对应的项目案例展示和分析，并且提供案例文档。通过对软件工程中的需求分析、概要设计、详细设计、编码、测试、产品交付、维护等过程的学习，学生可以掌握软件开发的基本流程；同时结合每章的案例分析，学生可以更加深入地理解软件开发实践过程，在短时间内提高软件开发技能。

本书是一本系统的、有针对性且有实效性的书籍，对从事软件项目开发以及希望学习软件开发的人员都有非常好的借鉴作用。

本书由韩万江、姜立新编著，感谢陆天波、杨金翠、孙艺、孙泉、杨元民、岳鹏、郭士榕等的参与，同时对薛忆非、陈甜、韩新雨、郭捷、钱蕴哲、王镱臻等的贡献也一并表示感谢！

由于作者水平有限，书中难免有疏漏之处，诚请各位读者批评指正，并希望你们将使用本书的体会和遇到的问题告诉我们，以便我们在下一版中进行完善。

韩万江

casey_han@263.net

2016年12月于北京

目 录

第1章

软件工程概述

软件（software）是计算机系统中与硬件（hardware）相互依存的另一部分，它包括程序（program）、相关数据（data）及其说明文档（document）。其中，程序是按照事先设计的功能和性能要求执行的指令序列，数据是程序能正常操纵信息的数据结构，文档是与程序开发、维护和使用有关的各种图文资料。

软件工程（Software Engineering，SE）是针对软件这一具有特殊性质的产品的工程化方法，它涵盖了软件生存周期的所有阶段，并提供了一整套工程化的方法来指导软件人员的工作。软件工程是用工程科学的知识和技术原理来定义、开发和维护软件的一门学科。

1.1 软件工程的背景

计算机软件已经成为现代科学研究和解决工程问题的基础，是管理部门、生产部门、服务行业中的关键因素，它已渗透到了各个领域，成为当今世界不可缺少的一部分。展望未来，软件仍将是驱动各行各业取得新进展的动力。因此，人们需要学习并研究工程化的软件开发方法，使开发过程更加规范，这已变得越来越重要。

20世纪中期软件产业从零开始起步，并迅速发展成为推动人类社会发展的龙头产业。随着信息产业的发展，软件对人类社会越来越重要，人们对软件的认识也经历了一个由浅到深的过程。

第一个写软件的人是Ada（Augusta Ada Lovelace），在19世纪60年代她尝试为Babbage（Charles Babbage）的机械式计算机写软件，尽管失败了，但她永远载入了计算机发展的史册。20世纪50年代，软件伴随着第一台电子计算机的问世诞生了，以写软件为职业的人也开始出现，他们多是经过训练的数学家和电子工程师。20世纪60年代，美国大学里开始出现计算机专业，教人们编写软件。

软件发展的历史大致可以分为如下的三个阶段：

第一个阶段是20世纪50 ~ 60年代，即程序设计阶段，基本采用个体手工劳动的生产方式。这个时期的程序是为特定目的而编制的，软件的通用性很有限，往往带有强烈的个人色彩。早期的软件开发也没有什么系统的方法可以遵循，软件设计是在某个人的头脑中完成的一个隐藏的过程，而且，除了源代码往往没有软件说明书等文档。因此这个时期尚无软件

的概念，基本上只有程序、程序设计概念；不重视程序设计方法；而且设计的程序主要用于科学计算，规模很小，采用简单的工具，基本上采用低级语言；硬件的存储容量小，运行可靠性差。

第二阶段是20世纪60 ~ 70年代，即软件设计阶段，采用小组合作生产方式。这个时期软件开始作为一种产品被广泛使用，出现了“软件作坊”，专门应别人的要求写软件。这个阶段的程序设计基本采用高级语言开发工具，人们开始提出结构化方法；硬件的速度、容量、工作可靠性有明显提高，而且硬件的价格降低；人们开始使用产品软件（可购买），从而建立了软件的概念。但是开发技术没有新的突破，软件开发的方法基本上仍然沿用早期的个体化软件开发方式。随着软件数量的急剧膨胀，软件需求日趋复杂，维护的难度越来越大，开发成本日益高涨，此时的开发技术已不适应规模大、结构复杂的软件开发，失败的项目越来越多。

第三个阶段从20世纪70年代开始，即软件工程时代，采用工程化的生产方式。这个阶段的硬件向超高速、大容量、微型化以及网络化方向发展；第三、四代语言出现；数据库、开发工具、开发环境、网络、分布式、面向对象技术等工具方法都得到应用；软件开发技术有很大进步，但未能获得突破性进展，软件开发技术的进步一直未能满足发展的要求。这个时期很多的软件项目开发时间大大超出了规划的时间表，一些项目导致了财产的流失，甚至导致了人员伤亡。同时，一些复杂的、大型的软件开发项目提出来了，软件开发的难度越来越大，在软件开发中遇到的问题找不到解决的办法，使问题积累起来，形成了尖锐的矛盾，失败的软件开发项目屡见不鲜，因而导致了软件危机。

软件危机指的是计算机软件开发和维护过程中所遇到的一系列严重问题。概括来说，软件危机包含两方面的问题：一是如何开发软件，以满足不断增长、日趋复杂的需求；二是如何维护数量不断膨胀的软件产品。落后的软件生产方式无法满足迅速增长的计算机软件需求，从而导致软件开发与维护过程中出现一系列严重问题。

最为突出的例子是美国IBM公司于1963 ~ 1966年开发的IBM360系列机的操作系统。该项目的负责人Fred Brooks在总结该项目时无比沉痛地说：“……正像一只逃亡的野兽落到泥潭中做垂死挣扎，越是挣扎，陷得越深，最后无法逃脱灭顶的灾难……程序设计工作正像这样一个泥潭……一批批程序员被迫在泥潭中拼命挣扎……谁也没有料到问题竟会陷入这样的困境……”IBM360操作系统的历史教训已成为软件开发项目中的典型事例被记入史册。软件开发中的最大的问题不是技术问题，而是管理问题。

具体地说，软件危机主要有以下表现：

- 对软件开发成本和进度的估计常常不准确，开发成本超出预算，项目经常延期，无法按时完成任务。
- 开发的软件不能满足用户要求。
- 软件产品的质量低。
- 开发的软件可维护性差。
- 软件通常没有适当的文档资料。
- 软件的成本不断提高。
- 软件开发生产率的提高赶不上硬件的发展和人们需求的增长。

软件危机的原因，一方面与软件本身的特点有关，另一方面与软件开发和维护的方法不正确有关。软件危机的产生，迫使人们不得不研究、改变软件开发的技术手段和管理方法。

从此软件生产进入软件工程时代。

1968年北大西洋公约组织的计算机科学家在联邦德国召开的国际学术会议上第一次提出了“软件危机”这个名词，同时讨论和制定了摆脱“软件危机”的对策。在那次会议上第一次提出了软件工程这个概念，从此一门新兴的工程学科——软件工程学——应运而生。

“软件工程”的概念是为了有效地控制软件危机的发生而提出来的，它的中心目标就是把软件作为一种物理的工业产品来开发，要求“采用工程化的原理与方法对软件进行计划、开发和维护”。软件工程是一门旨在开发满足用户需求、及时交付、不超过预算和无故障的软件的学科，它的主要对象是大型软件，最终目的是摆脱手工生产软件的状况，逐步实现软件开发和维护的自动化。

从微观上看，软件危机的特征表现在完工日期一再拖后、经费一再超支，甚至工程最终宣告失败等方面；而从宏观上看，软件危机的实质是软件产品的供应赶不上需求的增长。

虽然“软件危机”还没得到彻底解决，但自从软件工程概念提出以来，经过几十年的研究与实践，在软件开发方法和技术方面已经有了很大的进步。尤其应该指出的是，人们逐渐认识到，在软件开发中最关键的问题是软件开发组织不能很好地定义和管理其软件过程，从而使一些好的开发方法和技术都起不到应有的作用。也就是说，在没有很好定义和管理软件过程的软件开发中，开发组织不可能在好的软件方法和工具中获益。

1.2 软件工程知识体系

“工程”是科学和数学的某种应用，通过这一应用，使自然界的物质和能源的特性能够通过各种结构、机器、产品、系统和过程，成为对人类有用的东西。因而，“软件工程”就是科学和数学的某种应用，通过这一应用，使计算机设备的能力借助于计算机程序、过程和有关文档成为对人类有用的东西。它涉及计算机学科、工程学科、管理学科和数学学科。软件工程包括运用现代科学技术知识来设计并构造计算机程序的过程以及开发、运行和维护这些程序所必需的相关文件资料。软件工程研究的主要内容是方法、过程和工具。

软件工程的成果是为软件设计和开发人员提供思想方法和工具，而软件开发是一项需要良好组织、严密管理且各方面人员配合协作的复杂工作。软件工程正是指导这项工作的一门科学。软件工程学一般包含软件开发技术和软件工程管理两方面的内容，其中软件开发方法学和软件工程环境属于软件开发技术的内容，软件工程经济学属于软件工程管理的内容。软件工程在过去一段时间内已经取得了长足的进展，在软件的开发和应用中起到了积极的作用。随着软件开发的深入以及各种技术的不断创新和软件产业的形成，人们越来越意识到软件过程管理的重要性，并且传统的软件工程理论也随着人们的开发实践不断完善发展。

高质量的软件工程可以保证生产出高质量的、用户满意的软件产品。但是，人们对软件工程的界定，总是存在一定的差异。软件工程应该包括哪些知识？这里我们引用IEEE在软件工程知识体系指南（Guide to the Software Engineering Body of Knowledge，SWEBOK）中对软件工程的定义：软件开发、实施、维护的系统化、规范化、质量化方法的应用，也就是软件的应用工程；对上述方法的研究。

1998年，美国联邦航空管理局在启动一个旨在提高技术和管理人员的软件工程能力的项目时发现，他们找不到软件工程师应该具备的公认的知识结构，于是他们向美国联邦政府提出了关于开发“软件工程知识体系指南”的项目建议。美国Embry-Riddle航空大学计算与数学系的Thomas B. Hilburn教授接手了该研究项目，并于1999年4月完成了《软件工程知

识本体结构》的报告。该报告发布后迅速引起软件工程界、教育界和一些政府对建立软件工程本体知识结构的兴趣。很快人们普遍接受了这样的认识：建立软件工程知识体系的结构是确立软件工程专业至关重要的一步，如果没有一个得到共识的软件工程知识体系结构，将无法验证软件工程师的资格，无法设置相应的课程，或者无法建立对相应课程进行认可的判断准则。对建立权威的软件工程知识体系结构的需求迅速在世界各地反映出来。1999 年 5 月，ISO 和 IEC 的第一联合技术委员会（ISO/IEC JTC1）为顺应这种需求，立即启动了标准化项目——“软件工程知识体系指南”。美国电子电气工程师学会与美国计算机联合会联合建立的软件工程协调委员会（SECC）、加拿大魁北克大学以及美国 MITRE 公司（与美国 SEI 共同开发 SW-CMM 的软件工程咨询公司）等共同承担了 ISO/IEC JTC1“SWEBOK（Software Engineering Body of Knowledge）指南”项目任务。

2014 年 2 月 20 日，IEEE 计算机协会发布了软件工程知识体系 SWEBOK 指南第 3 版。SWEBOK 指南第 3 版标志着 SWEBOK 项目达到了一个新的里程碑。

SWEBOK V2 界定了软件工程的 10 个知识领域（Knowledge Area，KS）：软件需求（software requirements）、软件设计（software design）、软件构建（software construction）、软件测试（software testing）、软件维护（software maintenance）、软件配置管理（software configuration management）、软件工程管理（software engineering management）、软件工程过程（software engineering process）、软件工程工具和方法（software engineering tools and methods）、软件质量（software quality）。

在 SWEBOK V3 中，软件工程知识体被补充、细分为软件工程教育需求（the educational requirements of software engineering）和软件工程实践（the practice of software engineering）两大类，共 15 个知识域。其中软件工程教育需求包含 4 个知识域，软件工程实践包含 11 个知识域。软件工程教育需求包含的 4 个知识域分别是工程经济基础（engineering economy foundations）、计算基础（computing foundations）、数学基础（mathematical foundations）、工程基础（engineering foundations）。“软件工程实践”包含的 11 个知识域分别是软件需求、软件设计、软件构建、软件测试、软件维护、软件配置管理、软件工程管理、软件工程过程、软件工程模型与方法（software engineering models and methods）、软件质量、软件工程专业实践（software engineering professional practice）。

与 SWEBOK V2 相比，SWEBOK V3 的主要内容有以下几个方面变化：

- 更新了所有知识域的内容，反映出软件工程近 10 年的新成果，并与 CSDA、CSDP、SE2004、GSwE2009 和 SEVOCAB 等标准进行了知识体系的统一。
- 新增了 4 个基础知识域（软件工程经济基础、计算基础、数学基础和工程基础）和一个软件工程专业实践知识域。
- 在软件设计和软件测试中新增了人机界面的内容；把软件工具的内容从原先的“软件工程工具和方法”中移到其他各知识域中，并将该知识域重命名为“软件工程模型和方法”，使其更关注方法。
- 突出了架构设计和详细设计的不同，同时在软件设计中增加了硬件问题的新主题和面向方面（aspect-oriented）设计的讨论。
- 新增了软件重构、迁移和退役的新主题，更多地讨论了建模和敏捷方法。
- 在多个知识域中都增加了对保密安全性（security）的考虑。
- 合并了多个标准中的参考文献，并进行更新和遴选，减少了参考文献数量。

1.3 软件工程的三段论

软件工程是用工程科学的知识和技术原理来定义、开发和维护软件的一门学科。在不断探索软件工程的原理、技术和方法的过程中，人们研究和借鉴了工程学的某些原理和方法，并形成了软件工程学。软件工程的目标是提高软件的质量与生产率，最终实现软件的工业化生产。既然软件工程是“工程”，那么我们从工程的角度看一下软件项目的实施过程，如图 1-1 所示。

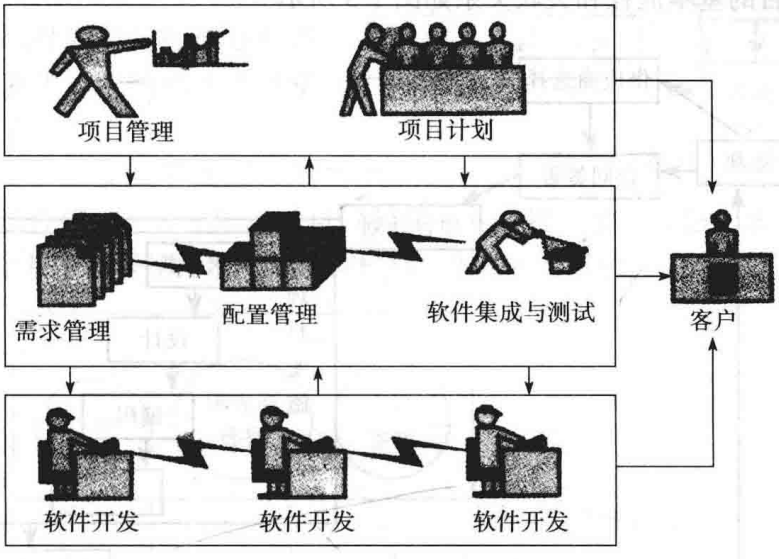


图 1-1 工程化软件开发

客户的需求启动了一个软件项目，为此我们需要先规划这个项目，即完成项目计划，然后根据这个项目计划实施项目。项目实施的依据是需求，这个需求类似工程项目的图纸，开发人员按照这个图纸生产软件，即设计、编码。在开发生产线上，将开发过程的半成品通过配置管理来存储和管理，然后进行必要的集成和测试，直到最后提交给客户。在整个开发过程中需要进行项目跟踪管理。软件工程活动是“生产一个最终满足需求且达到工程目标的软件产品所需要的步骤”。这些活动主要包括开发类活动、管理类活动和过程改进类活动，这里将它定义为“软件工程的三段论”，或者“软件工程的三线索”。一段论是“软件项目管理”，二段论是“软件项目开发”，三段论是“软件过程改进”。这个三段论可以用一个三角形表示，如图 1-2 所示，它们类似于相互支撑的三角形的三条边。我们知道三角形是最稳定的，要保证三角形的稳定性，三角形的三条边必不可少，而且要保持一定的相互关系。

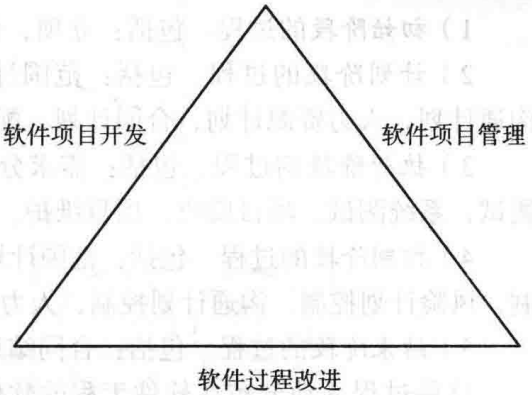


图 1-2 软件工程的三个线索

为了保证软件管理、软件开发过程的有效性，应该保证上述过程的高质量和过程的持续改进。

让软件工程成为真正的工程，就需要软件项目的开发、管理、过程改进等方面规范化、工程化、工艺化、机械化。

软件开发过程中脑力活动的“不可见性”大

大增加了过程管理的困难。因此软件工程管理中的一项指导思想就是千方百计地使这些过程变为“可见的”、事后可查的记录。只有从一开始就在开发过程中严格贯彻质量管理，软件产品的质量才会有保证。否则，开发工作一旦进行到后期，无论怎样测试和补漏洞，都无济于事。

1.4 软件工程模型

一个软件项目的基本流程和关联关系如图 1-3 所示。

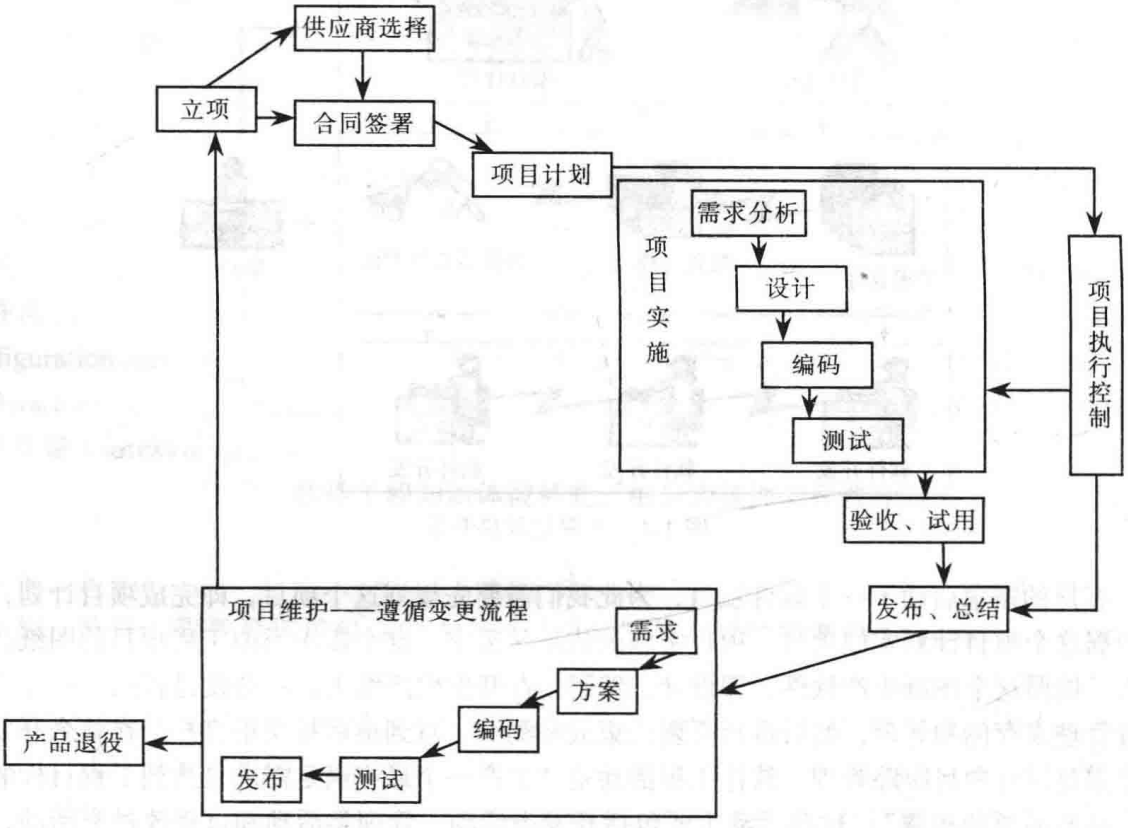


图 1-3 软件工程各个阶段过程之间的关系

按照项目的初始、计划、执行、控制、结束五个阶段，可以总结出软件工程的相关过程如下：

1）初始阶段的过程。包括：立项，供应商选择，合同签署。

2）计划阶段的过程。包括：范围计划，时间计划，成本计划，质量计划，风险计划，沟通计划，人力资源计划，合同计划，配置管理计划。

3）执行阶段的过程。包括：需求分析，概要设计，详细设计，编码，单元测试，集成测试，系统测试，项目验收，项目维护。

4）控制阶段的过程。包括：范围计划控制，时间计划控制，成本计划控制，质量计划控制，风险计划控制，沟通计划控制，人力资源计划控制，合同计划控制，配置管理计划控制。

5）结束阶段的过程。包括：合同结束，项目总结。

这些过程活动分布在软件工程的软件项目管理、软件项目开发、软件过程改进三条线索中。

“软件工程”与其他行业的工程有所区别，其模式或者标准很难统一为一个模型，所以，软件工程的模型是弹性的，标准是一个相对的标准。按照软件项目的初始、计划、执行、控制、结束五个阶段，我们建立一个基于过程元素的软件工程模型，如图 1-4 所示。模型用虚线分割成两部分，第一部分是过程构建和过程改进，其中的过程库是软件项目的标准过程积累；第二部分为基于过程的软件项目实施过程。

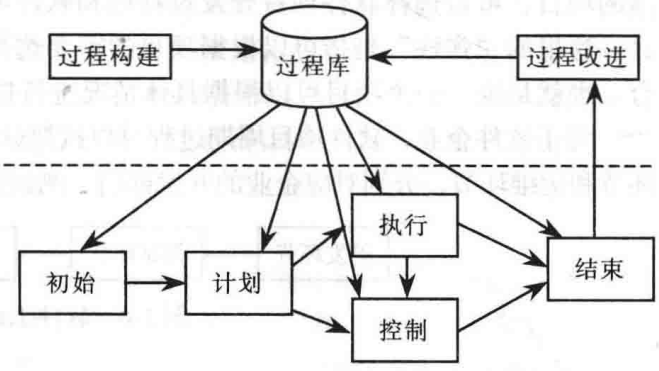


图 1-4　软件工程模型

我们将图 1-4 中的虚线下面部分展开为含有过程的流程模式，五个阶段中可以包含一些过程，这些过程元素存储在过程库中，这样就形成了一个基于过程的弹性软件工程模型，如图 1-5 所示。

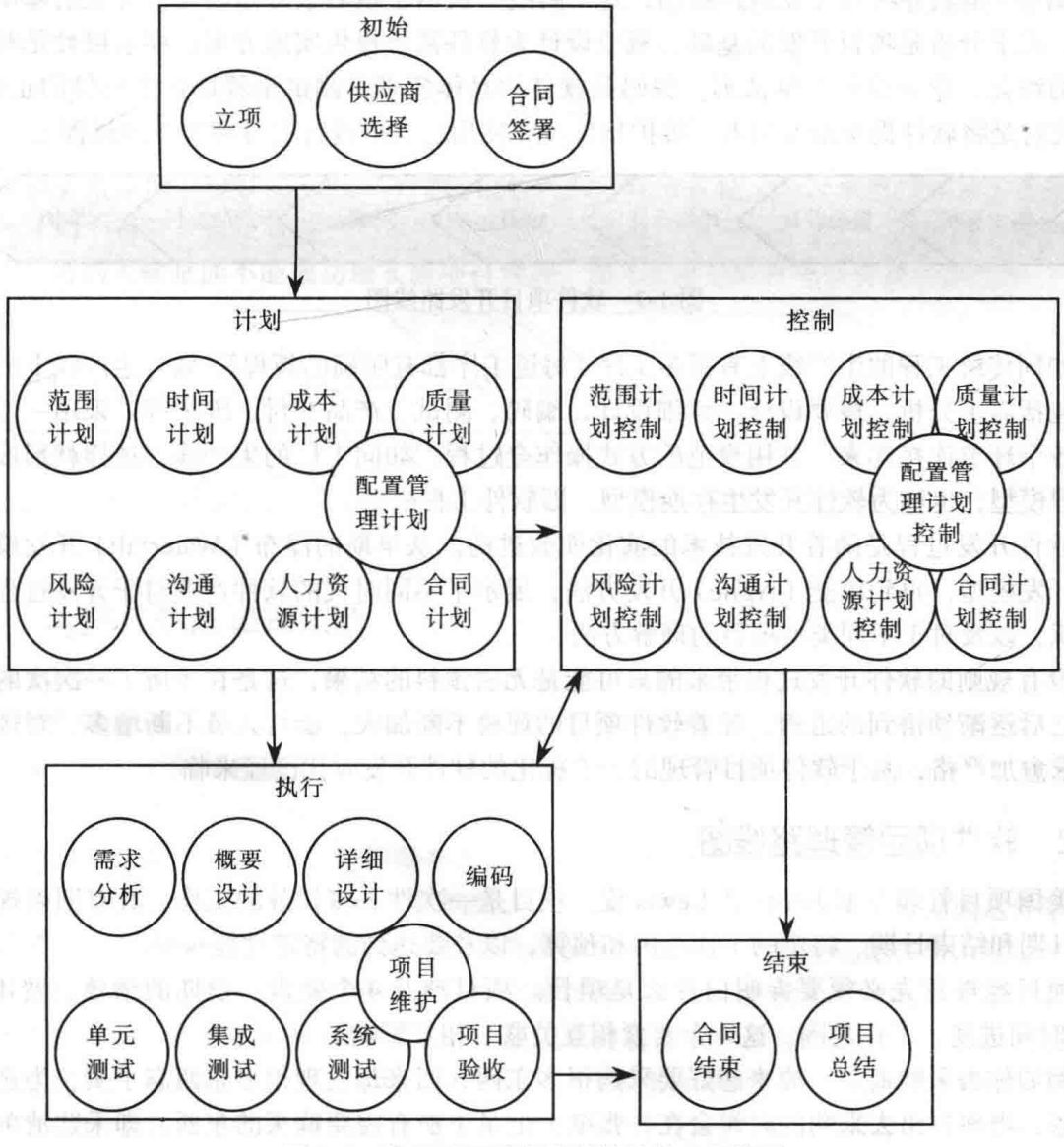


图 1-5　基于过程元素的弹性软件工程模型

这个弹性软件工程模型包含了软件工程的开发、管理、过程改进三个方面，对于一个具体的项目，可以选择软件项目开发过程组和软件项目管理过程组中的过程进行组合来完成项目。这里的“弹性”是指可以根据项目的需要选择过程，而软件过程又可以根据需要进行组合。也就是说，一个项目可以根据具体情况进行排列和取舍，形成特定项目的模型。

对于软件企业，软件项目周期过程可以按照软件工程流程分为开发环节、测试环节、生产环节和运维环节，分别对应企业的开发部门、测试部门、生产部门和运维部门，如图 1-6 所示。

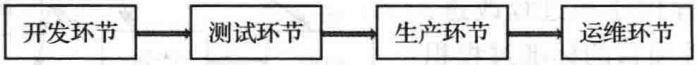


图 1-6 软件工程化环节

1.4.1 软件项目开发路线图

软件项目开发过程是软件工程的核心过程，通过这个生产线可以生产出用户满意的产品。软件工程提供了一整套工程化的方法来指导软件人员的工作。

图 1-7 是软件项目开发的路线图，这个路线图展示了从需求开始的软件开发的基本工艺流程。需求分析是项目开发的基础；概要设计为软件需求提供实施方案；详细设计是对概要设计的细化，它为编码提供依据；编码是软件的具体实现；测试是验证这个软件的正确性；产品交付是将软件提交给使用者；维护指软件在使用过程中进行完善和改进的过程。

图 1-7 软件项目开发路线图

如同传统工程的生产线上有很多工序（每道工序都有明确的规程），软件生产线上的工序主要包括需求分析、概要设计、详细设计、编码、测试、产品交付、维护等。采用一定的流程将各个环节连接起来，并用规范的方式操作全过程，如同工厂的生产线，这样就形成了软件工程模型，也称为软件开发生存期模型，即软件工程模型。

软件开发过程是随着开发技术的演化而改进的。从早期的瀑布（Waterfall）开发模型到迭代开发模型，再到敏捷（Agile）开发方法，展示了不同时代的软件产业对于开发过程的不同认识，以及对于不同类型项目的理解方法。

没有规则的软件开发过程带来的只可能是无法预料的结果，这是在经历了一次次的项目失败之后逐渐领悟到的道理。随着软件项目的规模不断加大，参与人员不断增多，对规范性的要求愈加严格，基于软件项目管理的、工程化的软件开发时代已经来临。

1.4.2 软件项目管理路线图

美国项目管理专家 James P. Lewis 说：项目是一次性、多任务的工作，具有明确规定的开始日期和结束日期、特定的工作范围和预算，以及要达到的特定性能水平。

项目经理首先必须要弄明白什么是项目。项目涉及 4 个要素：预期的绩效、费用（成本）、时间进度、工作范围。这 4 个要素相互关联、相互影响。

例如你去采购商品，原来想好要采购很多东西，回来却发现很多东西忘了买，为避免这种问题，当你再出去采购的时候会在一张纸上记录下所有需要购买的东西，即采购清单，你可以“完成一个采购项，在采购清单上打一个钩”，如果清单中每项都打钩了，就表示所有

的采购任务完成了，这个采购清单就是你的计划，你通过不断在采购清单上打钩来控制“采购”这个项目很好地完成。再举一个我们熟悉的例子，假如让你负责一个聚会活动，那么你就是这个“聚会活动”的项目经理，如何使这个项目成功就是你的任务。为了很好地完成这个任务，你需要知道聚会中有哪些活动、费用如何、如何安排时间等，在聚会进行过程中，你需要控制哪些活动完成了，哪些活动没有完成，进度进展得如何，费用花费得如何等。经历几次没有计划的聚会后，你会觉得必须要事前制订好一个节目单——相当于一个计划，记录有哪些活动，安排时间，控制花费等。

同理，软件项目管理也是这样，软件项目管理就是如何管理好软件项目的范围、时间、成本、质量，也就是管理好项目的内容、花费的时间（进度）、花费的代价（规模成本）、产品质量，为此需要制订一个好的项目计划，然后控制好这个计划，即软件项目管理的实质是软件项目计划的编制和项目计划的跟踪控制。计划与跟踪控制是相辅相成的关系：计划是项目成功实施的指南和跟踪控制的依据，而跟踪控制又用来保证项目计划的成功执行。

实际上，要做到项目计划切合实际是一个非常高的要求，需要对项目的需求进行详细分析，根据项目的实际规模制订合理的计划。计划的内容包括进度安排、资源调配、经费使用等，为了降低风险，还要进行必要的风险分析与制订风险管理计划，同时要对自己的开发能力有非常准确的了解，制订切实可行的质量计划和配置管理计划等。这来源于项目经理的职业技能和实践经验的持续积累。

制订了合适的项目计划之后，才能进行有效的跟踪与监督。当发现项目计划的实际执行情况与计划不符时，要进行适当、及时的调整，确保项目按期、按预算、高质量地完成。项目成功与否的关键是能不能成功地实施项目管理。图 1-8 便是软件项目管理的路线图。

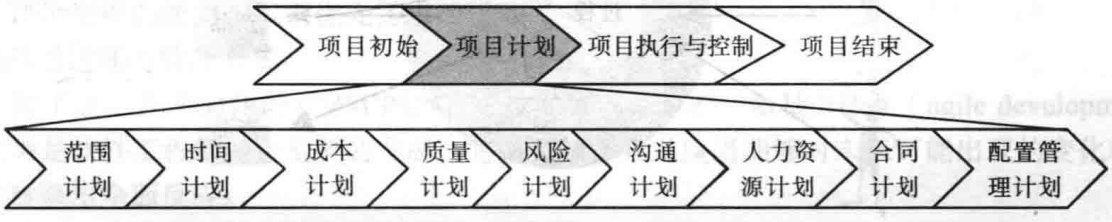


图 1-8　软件项目管理的路线图

1.4.3　软件过程改进路线图

自 20 世纪 70 年代软件危机以来，人们不断地展开新方法和新技术的研究与应用，但未取得突破性的进展。直到 20 世纪 80 年代末，人们得出这样一个结论：一个软件组织的软件能力取决于该组织的过程能力。一个软件组织的过程能力越成熟，该组织的软件生产能力就越有保证。

所谓过程，简单来说就是我们做事情的一种固有的方式，我们做任何事情都有过程存在，小到日常生活中的琐事，大到工程项目。对于做一件事，有经验的人对完成这件事的过程会很了解，他会知道完成这件事需要经历几个步骤，每个步骤都完成什么事，需要什么样的资源、什么样的技术等，因而可以顺利地完成工作；没有经验的人对过程不了解，就会有无从下手的感觉。图 1-9 和图 1-10 可以形象地说明过程在软件开发中的地位。如果项目人员将关注点只放在最终的产品上，如图 1-9 所示，不关注期间的开发过程，那么不同的开发队伍或者个人可能就会采用不同的开发过程，结果导致开发的产品有的质量高，有的质量差，完全依赖个人的素质和能力。