全国高等职业教育规划教材

软件工程与团队开发实战

主　编　张　恺
副主编　陈自力　郑　晶　张瑞英

电子教案下载网址www.cmpedu.com

机械工业出版社
CHINA MACHINE PRESS

全国高等职业教育规划教材

软件工程与团队开发实战

主　编　张　恺

副主编　陈自力　郑　晶　张瑞英

参　编　李伙钦　王　敏　潘燕燕

张传娟　陈清水　王军祥

主　审　陈常晖

机械工业出版社

本书围绕软件项目的开发过程，运用软件工程的方法，结合企业的实际项目案例与团队构成，从项目团队组建、项目计划与进度管理、软件项目需求管理、配置管理、项目规范、系统设计、软件测试、软件项目缺陷管理、验收交付与过程改进等几个方面对软件项目的开发过程进行了阐述。

作者根据自身多年的开发和教学经验，按照软件项目的开发流程、企业的人才需求和学生的认知规律精心编写了本书的内容。本书通过一系列团队开发的案例，力求将软件工程思想与实际软件项目开发融为一体，既有对多种常见方法的全面概括介绍，又有对一种典型方法的深入介绍，可以作为高职高专软件技术、应用控制技术、网络技术、信息管理和电子商务等专业的教材，也可作为计算机培训班的教材及软件行业程序员自学者的参考书。

图书在版编目（CIP）数据

软件工程与团队开发实战/张恺主编．—北京：机械工业出版社，2011.8
全国高等职业教育规划教材
ISBN 978-7-111-34433-9

Ⅰ.①软… Ⅱ.①张… Ⅲ.①软件工程-高等职业教育-教材
Ⅳ.①TP311.5

中国版本图书馆 CIP 数据核字（2011）第 149492 号

机械工业出版社（北京市百万庄大街 22 号 邮政编码 100037）
责任编辑：鹿 征 杨 硕
责任印制：乔 宇
三河市宏达印刷有限公司印刷

2011 年 9 月第 1 版·第 1 次印刷
184mm×260mm·15.25 印张·373 千字
0001-3000 册
标准书号：ISBN 978-7-111-34433-9
定价：29.00 元

凡购本书，如有缺页、倒页、脱页，由本社发行部调换

电话服务
社服务中心：（010）88361066
销售一部：（010）68326294
销售二部：（010）88379649
读者购书热线：（010）88379203

网络服务
门户网：http://www.cmpbook.com
教材网：http://www.cmpedu.com

全国高等职业教育规划教材
计算机专业编委会成员名单

主　　任　周智文

副 主 任　周岳山　林　东　王协瑞　张福强
陶书中　龚小勇　王　泰　李宏达
赵佩华

委　　员　（按姓氏笔画排序）

马　伟　马林艺　万雅静　万　钢
卫振林　王兴宝　王德年　尹敬齐
史宝会　宁　蒙　刘本军　刘新强
刘瑞新　余先锋　张洪斌　张　超
李　强　杨　莉　杨　云　罗幼平
贺　平　赵国玲　赵增敏　赵海兰
钮文良　胡国胜　秦学礼　贾永江
徐立新　唐乾林　陶　洪　顾正刚
康桂花　曹　毅　眭碧霞　梁　明
黄能耿　裴有柱

秘 书 长　胡毓坚

出 版 说 明

根据《教育部关于以就业为导向深化高等职业教育改革的若干意见》中提出的高等职业院校必须把培养学生动手能力、实践能力和可持续发展能力放在突出的地位，促进学生技能的培养，以及教材内容要紧密结合生产实际，并注意及时跟踪先进技术的发展等指导精神，机械工业出版社组织全国近60所高等职业院校的骨干教师对在2001年出版的“面向21世纪高职高专系列教材”进行了全面的修订和增补，并更名为“全国高等职业教育规划教材”。

本系列教材是由高职高专计算机专业、电子技术专业和机电专业教材编委会分别会同各高职高专院校的一线骨干教师，针对相关专业的课程设置，融合教学中的实践经验，同时吸收高等职业教育改革的成果而编写完成的，具有定位准确、注重能力、内容创新、结构合理、叙述通俗的编写特色。在几年的教学实践中，本系列教材获得了较高的评价，并有多个品种被评为普通高等教育“十一五”国家级规划教材。在修订和增补的过程中，除了保持原有特色外，针对课程的不同性质采取了不同的优化措施。其中，核心基础课的教材在保持扎实的理论基础的同时，增加实训和习题；实践性较强的课程强调理论与实训紧密结合；涉及实用技术的课程则在教材中引入了最新的知识、技术、工艺和方法。同时，根据实际教学的需要对部分课程进行了整合。

归纳起来，本系列教材具有以下特点：

1）围绕培养学生的职业技能这条主线来设计教材的结构、内容和形式。

2）合理安排基础知识和实践知识的比例。基础知识以“必需、够用”为度，强调专业技术应用能力的训练，适当增加实训环节。

3）符合高职学生的学习特点和认知规律。对基本理论和方法的论述容易理解、清晰简洁，多用图表来表达信息；增加相关技术在生产中的应用实例，引导学生主动学习。

4）教材内容紧随技术和经济的发展而更新，及时将新知识、新技术、新工艺和新案例引入教材。同时注重吸收最新的教学理念，并积极支持新专业的教材建设。

5）注重立体化教材建设。通过主教材、电子教案、配套素材光盘、实训指导和习题及解答等教学资源的有机结合，提高教学服务水平，为高素质技能型人才的培养创造良好的条件。

由于我国高等职业教育改革和发展的速度很快，加之我们的水平和经验有限，因此在教材的编写和出版过程中难免出现问题和错误。我们恳请使用这套教材的师生及时向我们反馈质量信息，以利于今后不断提高教材的出版质量，为广大师生提供更多、更适用的教材。

机械工业出版社

前　言

通过对 IT 职业教育现状的调查，结合企业人才的需求，发现目前 IT 从业人员的工作能力低下和团队协作能力欠缺是 IT 公司普遍存在的问题，而解决这一问题，也成为对 IT 教育领域的严重挑战。目前最欠缺的手段和方式，就是如何安排具有鲜明针对性的、以软件工程方法为基础、团队协作氛围下的 IT 开发，以及以企业项目开发环境为真实背景的实战训练。软件工程是计算机科学技术的一个重要分支，是一个异常活跃的研究领域。严格遵循软件工程方法论可以大大提高软件开发的成功率，能够显著减少软件开发和维护中的问题。

经典的软件工程书籍厚得像砖头，或让人望而却步，或让人看了云里雾里。本书不偏重于理论层面，主要以职业为导向、就业为目标，没有过多知识框架的限制，也不是针对某一个特定语言或特定框架的使用进行介绍，完全是类似模拟沙盘的战斗演习，让读者了解企业中是如何开发一个项目的，真正的开发项目包括哪些阶段，真正的项目开发中需要什么样的人才，团队开发过程中有哪些常用的技术与手段，通过亲身体验，获得宝贵经验。经过精心组织设计的本书内容，体现了“在做中学、学以致用”的基本理念。

本书共分两大部分、11 个章节，第 1 部分即第 1 ~9 章，主要介绍软件工程的基本原理、概念，软件生存周期各阶段的任务、过程以及用到的方法和技术。第 2 部分即第 10、11 章，主要介绍了软件团队开发的实战技术和案例。

本书由福建交通职业技术学院张恺主编，并完成全部书稿的统稿工作。其中第 1 ~9 章由张恺编写，第 10 章由福建交通职业技术学院陈自力和北京信息职业技术学院张瑞英编写，第 11 章由福建江夏学院郑晶和北京信息职业技术学院张瑞英编写，福州博洋教育的陈清水参与了部分教学案例的编写，福建交通职业技术学院的李伙钦、王敏、潘燕燕、张传娟、王军祥参与了电子课件的制作。

编者

目　录

第 2 部分　团队开发项目实战

·第 1 部分·

软件工程与团队开发

第1章　软件工程的基本概念

开篇小故事

有一对兄弟，他们的家住在80层楼上。有一天他们外出旅行回家，发现大楼停电了！虽然他们背着大包的行李，但看来没有什么别的选择，于是哥哥对弟弟说，我们就爬楼梯上去！于是，他们背着两大包行李开始爬楼梯。爬到20楼的时候他们开始累，哥哥说："包太重了，不如这样吧，我们把包放在这里，等来电后坐电梯来拿。"于是，他们把行李放在了20楼，这样轻松多了，继续向上爬。

他们有说有笑地往上爬，但是好景不长，到了40楼，两人实在太累了。想到还只爬了一半，两人开始互相埋怨，指责对方不注意大楼的停电公告，才会落得如此下场。他们边吵边爬，就这样一路爬到了60楼。到了60楼，他们累得连吵架的力气也没有了。弟弟对哥哥说："我们不要吵了，爬完它吧。"终于80楼到了！兴奋地来到家门口，兄弟俩才发现他们的钥匙留在20楼的包里了……

20岁之前，我们活在家人、老师的期望之下，背负着很多压力、包袱，自己也不够成熟，能力不足，因此步履难免不稳。20岁之后，离开了众人的压力，卸下了包袱，开始全力以赴地追求自己的梦想，就这样愉快地过了20年。可是到了40岁，发现青春已逝，不免产生许多遗憾和悔恨，于是开始遗憾这个、惋惜那个、抱怨这个、嫉恨那个，就这样在抱怨中度过了20年。到了60岁，发现人生已所剩不多，于是告诉自己不要再抱怨了，就珍惜剩下的日子吧！于是默默地走完了自己的余年。到了生命的尽头，才想起自己好像什么事情都没有完成……

原来，我们所有的梦想都留在了20岁的青春岁月，还没有来得及完成……

有梦在远方，是年轻人，同样也是所有人的人生最大力量。

本书的内容即将展开，希望能够为年轻一代的准IT人一些帮助，帮助每个人实现自己的IT梦想。

1.1　软件工程的定义与目标

自20世纪40年代中出现了世界上第一台计算机以后，就有了程序的概念。其后经历了几十年的发展，计算机软件经历了三个发展阶段：

- 程序设计阶段，20世纪50～60年代。
- 程序系统阶段，20世纪60～70年代。

• 软件工程阶段，20 世纪 70 年代以后。

几十年来最根本的变化体现在：

1）人们改变了对软件的看法。20 世纪 50 ~ 60 年代时，程序设计曾经被看做是一种任人发挥创造才能的技术领域。当时人们认为，写出的程序只要能在计算机上得出正确的结果，程序的写法可以不受任何约束。随着计算机的广泛使用，人们要求这些程序容易看懂、容易使用，并且容易修改和扩充。于是，程序便从个人按自己意图创造的“艺术品”转变为能被广大用户接受的工程化产品。

2）软件的需求是软件发展的动力。早期的程序开发者只是为了满足自己的需要，这种自给自足的生产方式仍然是其低级阶段的表现。进入软件工程阶段以后，软件开发的成果具有社会属性，它要在市场中流通以满足广大用户的需要。

3）软件工作的范围从只考虑程序的编写扩展到涉及整个软件生存周期。

在软件技术发展的第二阶段，随着计算机硬件技术的进步，要求软件能与之相适应。然而软件技术的进步一直未能满足形势发展的要求，致使问题积累起来，形成了日益尖锐的矛盾，进而导致了软件危机。问题归结起来有：

1）缺乏软件开发的经验和有关软件开发数据的积累，使得开发工作的计划很难制定。致使经费预算常常超支，进度无法遵循计划，开发完成的期限一拖再拖。

2）软件需求在开发初期未能明确提出，或是未能得到确切的表达。开发工作开始后，软件人员和用户又未能及时交换意见，造成开发后期矛盾的集中暴露。

3）开发过程没有统一的、公认的方法论和规范指导，参与人员各行其是，加之设计和实现过程的资料不完整，或忽视了每个人的工作与其他人工作的接口，使得软件很难维护。

4）未能在测试阶段充分做好检测工作，提交用户的软件质量差，在运行中暴露出大量的问题。

如果这些障碍不能突破，从而摆脱困境，软件的发展是没有出路的。

软件工程由此应运而生。

1.1.1 软件工程的定义

“软件工程”这一术语首次出现在 1968 年 NATO 会议上。Fritz Bauer 曾经为软件工程下了定义：“软件工程是为了经济地获得能够在实际机器上有效运行的可靠软件而建立和使用的一系列完善的工程化原则。”1983 年 IEEE 给出的定义为：“软件工程是开发、运行、维护和修复软件的系统方法”。其中，“软件”的定义为：计算机程序、方法、规则、相关的文档资料以及在计算机上运行时所必需的数据。

后来尽管又有一些人提出了许多更为完善的定义，但主要思想都是强调在软件开发过程中需要应用工程化原则的重要性。简单地说，软件工程就是指采用工程的概念、原理、技术和方法来开发与维护软件，把经过时间考验而证明正确的管理技术和当前能够得到的最好的技术方法结合起来。经过不断实践和总结，研究者最后得出一个结论：按工程化的原则和方法组织软件开发工作是有效的，是摆脱软件危机的一条主要出路。

1.1.2 软件工程的目标

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。质量是软件

需求方最关心的问题，用户缴纳了费用，当然希望能够得到一个实用的产品。生产率是软件开发方最关心的问题，老板和员工都想用更少的时间挣更多的钱。质量与生产率之间有着内在的联系，高生产率必须以质量合格为前提。如果质量不合格，对供需双方都是坏事。从短期效益看，追求高质量会延长软件开发时间并且增大费用，似乎降低了生产率。从长期效益看，高质量将保证软件开发的全过程更加规范流畅，大大降低了软件的维护代价，实质上是提高了生产率，同时也会为公司获得良好的信誉。质量与生产率之间不存在根本的对立，如果真的要比较质量与生产率的重要等级的话，那么应该是质量第一，生产率第二，原因有三点：①质量直接体现在软件的每段程序中，高质量自然是开发人员的技术追求，也是职业道德的要求；②高质量对所有的用户都有价值，而高生产率只对开发方有意义；③如果一开始就追求高生产率，容易使人急功近利，留下隐患。宁可进度慢些，也要保证每个环节的质量，以图长远利益。

组织实施软件工程项目，最终希望得到项目的成功。所谓成功指的是达到以下几个主要的目标：

1）付出较低的开发成本。

2）达到要求的软件功能。

3）取得较好的软件性能。

4）开发的软件易于移植。

5）需要较低的维护费用。

6）能按时完成开发工作，及时交付使用。

导致这些目标的因素相互重叠，有些因素互补，而有些因素是相互抵触的，因此在具体项目的实际开发中，企图让以上几个目标都达到理想的程度往往是非常困难的。例如，低开发成本与软件可靠性之间，低开发成本与高性能之间都存在冲突。

好的软件工程方法可以同时提高质量与生产率，减少项目成功因素之间的互斥，找到最佳的平衡点。

1.2 软件工程的常用模型

软件工程过程通常包含四种基本的过程活动：

1）P（Plan）：软件规格说明。规定软件的功能及其运行的限制。

2）D（Do）：软件开发。产生满足规格说明的软件。

3）C（Check）：软件确认。确认软件能够完成客户提出的要求。

4）A（Action）：软件演进。为满足客户的变更要求，软件必须在使用的过程中演进，版本推陈出新。

事实上，软件工程过程是一个软件开发机构针对某一类软件产品为自己规定的工作步骤，它应当是科学的、合理的，否则必将影响到软件产品的质量。

如同任何事物一样，软件也有一个孕育、诞生、成长、成熟、衰亡的生存过程，被称为计算机软件的生存周期。根据这一思想，把上述基本的过程活动进一步展开，可以得到软件生存周期的以下六个步骤：

1）制订计划：确定要开发软件系统的总目标，给出它的功能、性能、可靠性，以及接

口等方面的要求；研究完成该项软件任务的可行性，探讨解决问题的可能方案；制定完成开发任务的实施计划，连同可行性研究报告，提交管理部门审查。

2）需求分析：对待开发软件提出的需求进行分析并给出详细的定义。编写出软件需求说明书及初步的用户手册，提交管理机构评审。

3）软件设计：把已确定了的各项需求转换成一个相应的体系结构。进而对每个模块要完成的工作进行具体的描述。编写设计说明书，提交评审。

4）程序编写：把软件设计转换成计算机可以接受的程序代码。

5）软件测试：在设计测试用例的基础上检验软件的各个组成部分。

6）运行与维护：已交付的软件投入正式使用，并在运行过程中进行适当的维护。

六个步骤在软件工程模型中可以使用一定的流程将其连接起来，并用规范的方式操作全过程，如同工厂的生产线。常见的软件工程模型有：瀑布模型、增量模型、螺旋模型、喷泉模型等。

1.2.1 瀑布模型

最早出现的软件工程模型是瀑布模型，又称线性模型。它规定了各项软件工程活动，包括：制订开发计划，进行需求分析和说明，软件设计，程序编码，测试及运行维护，参看图1-1。各项活动按自上而下，相互衔接的固定次序，如同瀑布逐级下落。

然而软件开发的实践表明，上述各项活动之间并非完全是自上而下、呈线性图式的。实际情况中，每项开发活动均处于一个质量环（输入－处理－输出－评审）中。只有当其工作得到确认，才能继续进行下一项活动，在图1-1中用向下的箭头表示；否则返工，在图1-1中由向上的箭头表示。每项活动均处于一个质量环（输入－处理－输出－评审）中。

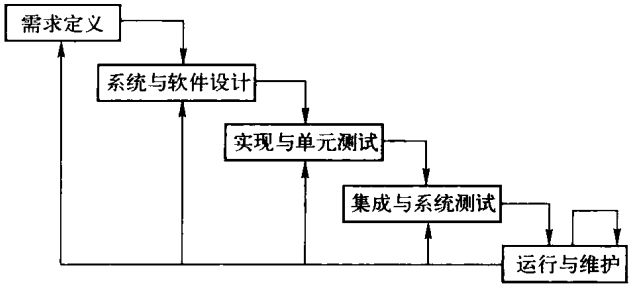

图1-1　软件工程的瀑布模型

线性模型是最早提出的软件工程模型，太简单并过于理想化，目前已经不能适应大部分的软件开发模式，几乎被业界所抛弃，只是偶尔在教科书中提起。但是“线性”毕竟是大部分人最熟悉并能最快熟练掌握和应用的一种思维方式，当人们遇到一个复杂的“非线性”问题时，总是需要将其转换成一系列简单的“线性”问题予以逐个解决。一个软件系统整体可能是复杂的，但是其中的单个子函数或者子程序肯定是可以用线性的方式予以解决的。随着软件工程的发展，软件模型越来越多，但是都可以从中找到线性模型的影子。

1.2.2 增量模型

增量模型可以说是一种分段式的线性模型，它先完成一个系统子集的开发，再按同样的

开发步骤增加功能（系统子集），如此递增下去直至满足全部系统需求，如图 1-2 所示。

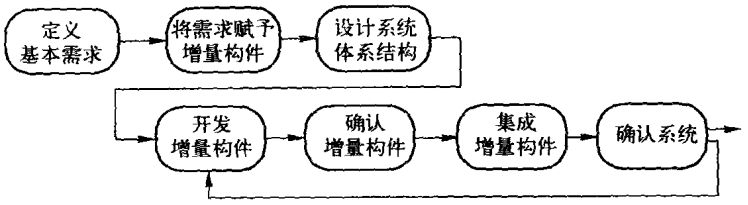

图 1-2　软件工程的增量模型

增量开发方法的新演进版本叫做“极限程序设计（Extreme Programming）”。

1.2.3　螺旋模型

对于复杂的大型软件，开发一个原型往往达不到要求。螺旋模型将瀑布模型与演化模型结合起来，并且加入两种模型均忽略了的风险分析。螺旋模型沿着螺线旋转，如图 1-3 所示，在笛卡儿坐标系的四个象限上分别表达了四个方面的活动，即：

- 制订计划——确定软件目标，选定实施方案，弄清项目开发的限制条件。
- 风险分析——分析所选方案，考虑如何识别和消除风险。
- 实施工程——实施软件开发。
- 客户评估——评价开发工作，提出修正建议。

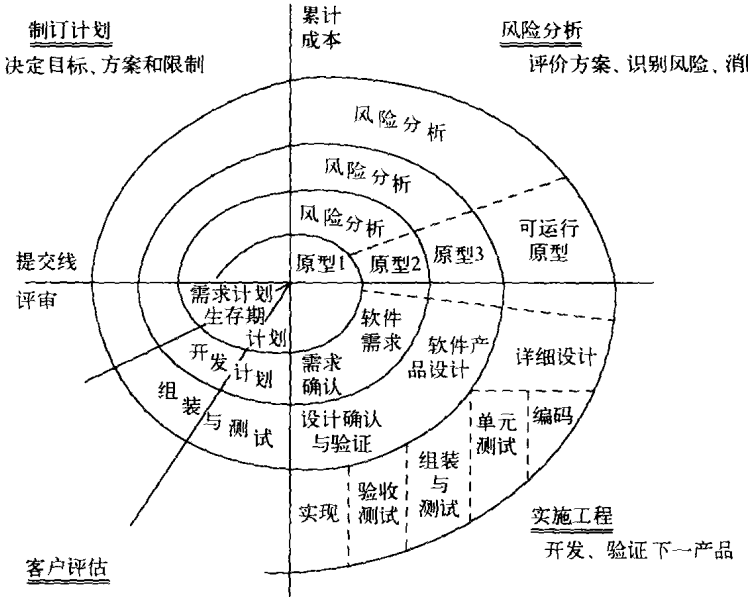

图 1-3　软件工程的螺旋模型

沿螺旋线自内向外每旋转一圈便开发出更为完善的一个新的软件版本。

1.2.4　喷泉模型

喷泉模型对软件复用和生存周期中多项开发活动的集成提供了支持，主要支持面向

对象的开发方法。“喷泉”一词本身体现了迭代和无间隙特性。系统某个部分常常重复工作多次，相关功能在每次迭代中随之加入演进的系统。所谓无间隙是指在开发活动，即分析、设计和编码之间不存在明显的边界。如图 1-4 所示，喷泉模型是对象驱动的过程。

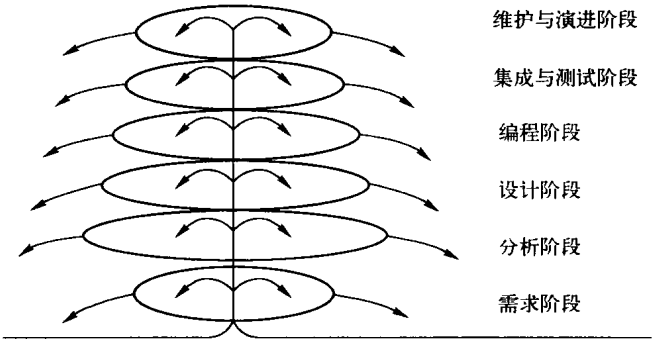

图 1-4　软件工程的喷泉模型

1.2.5　智能模型

智能模型是基于知识的软件开发模型，它把瀑布模型和专家系统综合在一起。该模型在各个开发阶段都利用了相应的专家系统来帮助软件人员完成开发工作。

智能模型建立了各个阶段的知识库，将模型、相应领域知识和软件工程知识分别存入数据库。以软件工程知识为基础的生成规则构成的专家系统与包含应用领域知识规则的其他专家系统相结合，构成该应用领域的开发系统。该模型应用基于规则的系统，采用归约和推理机制，帮助软件人员完成开发工作，并使维护在系统规格说明一级进行。

1.3　软件开发的基本策略

每个人都有自己的一套解决问题的思路，并能自然地在工作、学习和生活的过程中体现出来。同样，每个程序员自身所具有的软件工程观念会无形地支配其怎么去做事情。软件工程四十年的发展，已经积累了相当多的方法，但这些方法不是严密的理论，只是前人在不断的实践过程中针对各种软件问题的方法总结出来的解决方法。程序员在使用的过程中切不可教条地套用方法，更重要的是学会“选择合适的方法”和“产生新方法”。几千年前，我们的祖先就写下了很多他们自己的心得体会，例如《三十六计》、《孙子兵法》等，被现代人很好地运用于工业和商业实际中。

本节主要介绍软件开发中的三种基本策略：“软件复用”、“分而治之”与“优化——折中”。

1.3.1　软件复用

复用就是指“利用现成的东西”，文人称之为“拿来主义”。被复用的对象可以是有形的物体，也可以是无形的成果。复用不是人类懒惰的表现，而是智慧的表现。因为人类总是

在继承了前人的成果，不断加以利用、改进或创新后才会进步。

在工程生产中，复用的内涵包括了提高质量与生产率两方面。由经验可知，在一个新产品中，大部分的内容是成熟的、已经成型或者可以借鉴的，只有小部分内容是创新的。一般情况下，可以相信成熟的东西总是比较可靠的（即具有高质量），而大量成熟的工作可以通过复用来快速实现（即具有高生产率）。勤劳并且聪明的人们应该把大部分的时间用在小比例的创新工作上，而把小部分的时间用在大比例的成熟工作中，这样才能把工作做得又快又好。例如一双新款西服的生产，面料、剪裁、制衣的工艺都是很成熟的，大部分的时间是用在新款衣服的设计等创新工作上。

把复用的思想用于软件开发，称为软件复用。据统计，世上已有 1000 亿多行程序，无数功能被重写了成千上万次，很是浪费。面向对象（Object Oriented）学者的口头禅就是“请不要再发明相同的车轮子了”。

具有一定集成度并可以重复使用的软件组成单元称为软构件（Software Component）。软件复用可以表述为：构造新的软件系统可以不必每次从零做起，直接使用已有的软构件，即可组装（或加以合理修改）成新的系统。复用方法合理化并简化了软件开发过程，减少了总的开发工作量与维护代价，既降低了软件的成本又提高了生产率。另一方面，由于软构件是经过反复使用验证的，自身具有较高的质量。因此由软构件组成的新系统也具有较高的质量。利用软构件生产应用软件的过程如图 1-5 所示。

软件复用不仅要使自己拿来方便，还要让别人拿去方便，是“拿来拿去主义”。面向对象方法、Microsoft 公司的 COM 规范［Roberson 1999］都能很好地应用于实现大规模的软件复用中。目前越来越多的开源第三方组件也承担着软构件的作用，比如在开发过程中经常使用的 EXT、DBNET、JQUERY 等。中间件的生产厂商也日益增多，都是看准了软件复用这块庞大的商业市场。

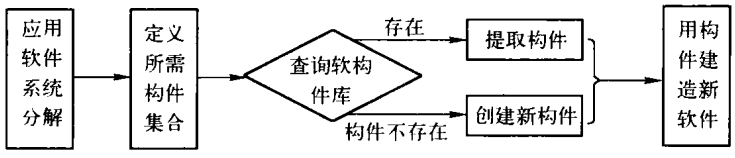

图 1-5　利用软构件生产应用软件的过程

1.3.2　分而治之

分而治之是指把一个复杂的问题分解成若干个简单的问题，然后逐个解决。这种朴素的思想来源于人们生活与工作的经验，完全适合于技术领域。软件人员在执行分而治之的时候，应该着重考虑：复杂问题分解后，每个问题能否用程序实现？所有程序最终能否集成为一个软件系统并有效解决原始的复杂问题？

图 1-6 表示了软件领域的分而治之策略。诸如软件的体系结构设计、模块化设计都是分而治之的具体表现。软件的分而治之不可以“硬分硬治”。不像是为了吃一个西瓜或是一只鸡，先切成 n 块，再把每块放进嘴里咀嚼，然后交由胃肠来消化吸收，象征复杂问题的西瓜或是鸡也就此消失了。