



高等职业教育信息技术类项目式系列规划教材
首批国家示范性高等职业院校建设成果教材

软件工程与 项目实践教程

高伟锋 刘 英 王凤岭 蒋文沛 编著



科学出版社
www.sciencep.com



免费提供电子课件

高等职业教育信息技术类项目式系列规划教材
首批国家示范性高等职业院校建设成果教材

软件工程与项目实践教程

高伟锋 刘 英 王凤岭 蒋文沛 编著

科学出版社

北 京

内 容 简 介

本书从实用角度介绍软件工程的基础知识和软件工程技术方法。本书的编写力求做到结合实际,注重培养实际开发能力和文档写作能力,注意内容的新颖性和系统性。

本书主要介绍了软件工程的基本概念、原理和应用,内容包括软件策划、软件需求分析、系统设计、软件的编码和测试、软件的实施与部署、面向对象方法学的应用以及统一建模语言(UML)。

本书可作为高职高专院校计算机专业或信息管理专业的教材,也可供软件培训人员和软件开发人员阅读参考。

图书在版编目(CIP)数据

软件工程与项目实践教程/高伟锋等编著. —北京:科学出版社, 2009

(高等职业教育信息技术类项目式系列规划教材·首批国家示范性高等职业院校教育院校建设成果教材)

ISBN 978-7-03-026297-4

I. 软… II. 高… III. ①软件工程—高等学校:技术学校—教材 ②软件开发—项目管理—高等学校:技术学校—教材 IV. TP311.5

中国版本图书馆CIP数据核字(2009)第240302号

责任编辑:李太铨 吕建忠/责任校对:赵燕

责任印制:吕春珉/封面设计:耕者设计工作室

科学出版社出版

北京东黄城根北街16号

邮政编码:100717

<http://www.sciencecp.com>

源海印刷有限责任公司印刷

科学出版社发行 各地新华书店经销

*

2010年2月第 一 版 开本:787×1092 1/16

2010年2月第一次印刷 印张:12 3/4

印数:1—3 000 字数:290 000

定价:20.00元

(如有印装质量问题,我社负责调换〈环伟〉)

销售部电话 010-62134988 编辑部电话 010-62135120-8220

版权所有,侵权必究

举报电话:010-64030229; 010-64034315; 13501151303

编 委 会

主 任 陈建新

副主任 张宁东 周 旺 唐锡海 王凤岭 易著梁

委 员 (按姓氏笔画排序)

于小川 王永琦 王汝凉 韦小波 宁爱民

孙 湧 孙宁青 曲桂东 李 洛 陈承欢

巫家敏 庞松鹤 钟 诚 梁 裕 梁京章

梁锦叶 黄欣萍 黄富佳 蒋年华

总策划 王凤岭 易著梁

序

职业教育作为一种教育类型，其课程也必须有自己的类型特征。从教育学的观点来看，当且仅当课程内容的选择以及所选内容的序化都符合职业教育的特色和要求之时，职业教育的课程改革才能成功。这里，改革的成功与否有两个决定性的因素：一个是课程内容的选择，一个是课程内容的序化。这也是职业教育教材编写的基础。

首先，课程内容的选择涉及的是课程内容选择的标准问题。

一般来说，课程内容涉及两大类知识：一类是涉及事实、概念以及规律、原理方面的“陈述性知识”，一类是涉及经验以及策略方面的“过程性知识”。“事实与概念”解答的是“是什么”的问题，“规律与原理”回答的是“为什么”的问题；而“经验”指的是“怎么做”的问题，“策略”强调的则是“怎样做更好”的问题。

由专业学科构成的以结构逻辑为中心的学科体系，侧重于传授实际存在的显性知识即理论性知识，主要解决“是什么”（事实、概念等）和“为什么”（规律、原理等）的问题，这是培养科学型人才的一条主要途径。

由实践情境构成的以过程逻辑为中心的行动体系，强调的是获取自我建构的隐性知识即过程性知识，主要解决“怎么做”（经验）和“怎样做更好”（策略）的问题，这是培养职业型人才的一条主要途径。

个体所具有的智力类型大致分为两大类：一是抽象思维，一是形象思维。职业教育的教育对象，依据多元智能理论分析，其逻辑数理方面的能力相对较差，而空间视觉、身体动觉以及音乐节奏等方面的能力则较强。职业教育的教育对象多数是具有形象思维特点的个体。

因此，职业教育课程内容选择的标准应该以职业实际应用的经验和策略的习得为主，以适度、够用的概念和原理的理解为辅，即以过程性知识为主、陈述性知识为辅。

其次，课程内容的序化涉及的是课程内容序化的标准问题。

知识只有在序化的情况下才能被传递，而序化意味着确立知识内容的框架和顺序。职业教育课程所选取的内容，由于既涉及过程性知识，又涉及陈述性知识，因此，寻求这两类知识的有机融合，就需要一个恰当的参照系，以便能以此为基础对知识实施“序化”。

按照学科体系对知识内容序化，课程内容的编排呈现出一种“平行结构”的形式。学科体系的课程结构常会导致陈述性知识与过程性知识的分割，理论知识与实践知识的分割，以及知识排序方式与知识习得方式的分割。这不仅与职业教育的培养目标相悖，而且与职业教育所追求的整体性学习的教学目标相悖。

按照行动体系对知识内容序化，课程内容的编排则呈现一种“串行结构”的形式。在学习过程中，学生认知的心理顺序与专业所对应的典型职业工作顺序，或是对多个职业工作过程加以归纳整合后的职业工作顺序（即行动顺序），都是串行的。这样，针对行动顺序的每一个工作过程环节来传授相关的课程内容，实现实践技能与理论知识的整合，将收到事半功倍的效果。



鉴于每一行动顺序都是一种自然形成的过程序列,而学生认知的心理顺序也是循序渐进自然形成的过程序列,这表明,认知的心理顺序与工作过程顺序在一定程度上是吻合的。

需要特别强调的是,按照工作过程来序化知识,即以工作过程为参照系,将陈述性知识与过程性知识整合、理论知识与实践知识整合,其所呈现的知识从学科体系来看是离散的、跳跃的和 discontinuous 的,但从工作过程来看,却是不离散的、非跳跃的和连续的了。因此,参照系在发挥着关键的作用。课程不再关注建筑在静态学科体系之上的显性理论知识的复制与再现,而更多的是着眼于蕴含在动态行动体系之中的隐性实践知识的生成与构建。这意味着,理论知识在数量上未变,但其排序的方式发生变化;理论知识的质量却发生了变化,不是知识的位移而是知识与实践的紧密融合。这正是对行动体系下强调工作过程系统化这一全新的职业教育课程开发中所蕴含的革命性变化的本质概括。

由此,我们可以得出这样的结论:如果“工作过程导向的序化”获得成功,那么传统的学科课程序列就将“出局”,通过对其保持适当的“有距离观察”,就有可能解放与扩展传统的课程视野!寻求现代的知识关联与分离的路线,确立全新的内容定位与支点,从而凸显课程的职业教育特色。

因此,“工作过程导向的序化”是一个与已知的序列范畴进行的对话,也是与课程开发者的立场和观点进行对话的创造性行动。这一行动并不是简单地排斥学科体系,而是通过“有距离观察”,在一个全新的架构中获得对职业教育课程论的元层次认知。所以,“工作过程导向的课程”的开发过程,实际上是一个伴随学科体系的解构而凸显行动体系的重构过程。然而,学科体系的解构并不意味着学科体系的“肢解”,而是依据职业情境对知识实施行动性重构,进而实现新的体系——行动体系的构建过程。不破不立,学科体系解构之后,在工作过程基础上的系统化和结构化的产物——行动体系也就“立在其中”了。

非常高兴,南宁职业技术学院信息工程学院针对高职计算机类专业职业岗位任务的项目化特点,从高等职业教育人才培养模式的角度出发,创新性地将传统“项目教学法”提升和拓展为基于工作过程为导向的“项目教学”人才培养模式。这就是以项目为纽带,加强学校和企业 in 课程体系建设和教学内容改革方面的深度合作,以校企联合成立的“计算机应用研究所”为孵化器,将真实的企业项目整合为项目教学资源;校企互通,共建项目教学团队,以项目为主线贯穿整个教学过程;吸引 IT 企业进驻校园,开展生产性实训,以项目教学来培养学生的职业能力和职业素质,并开发了一系列基于工作过程的课程、教材及其教学模式,为高职计算机类专业教学改革起到了很好的示范作用。从内涵来看,系统化的“项目教学”应该是工作过程系统化理论的具体实施。

前言

随着信息科学的发展,计算机应用的范围越来越广,计算机软件的社会需求量也越来越大。软件产业是信息产业的核心,软件产业化的一个重要方面是软件开发工程化,采用先进的工程化方向进行软件开发是实现软件产业化的关键技术手段。软件工程是用科学知识和技术原理来定义、开发和维护软件的一门学科。

本书将以面向对象软件工程的思想为基础,基于软件项目开发的工作过程,利用一个较完整的项目贯穿讲解整个软件生存周期,并通过实训项目解释和介绍软件开发的关键概念和工具。本书的目的是帮助学生获得较多的软件项目实践经验,实现“实践—理论—实践”的良性循环,从而较好地将高校的软件工程教育与IT企业的人才需求平滑接轨,使大学毕业生一走上就业岗位就能直接参与软件开发工作,较快地成为软件开发高手。

本书吸取了国内外多家软件企业先进的软件开发经验,借鉴了企业的软件开发模式,由具有丰富企业软件开发和项目管理工作经验的企业专家和老师共同编写,旨在把最新的软件开发技术和软件工程流程引入到学校课堂教学中。

全书围绕“网上书店”系统,深入浅出地介绍了软件工程及项目的知识技能。每个项目任务均安排了以下几个模块。

项目问题: 提出一个实际应用中的问题,并提出解决方案。

工作情境: 叙述完成任务的详细操作步骤及少量必备的知识点。

归纳总结: 对完成任务过程中涉及的新知识、新技能等进一步归纳总结,使学生在实现任务的过程中建立起来的感性认识得以梳理,从而帮助学生实现掌握知识和技能的目的。

拓展提高: 对同类任务中可能会遇到的问题、方法、技能或新知识等进行拓展性的介绍,使学生知识和技能得到进一步的提高。

思考提高: 在思考提高部分提出类似的另一个项目任务,要求学生自主思考完成实训任务,达到灵活运用、举一反三的效果。

本书项目1至项目5由高伟锋编写,项目6和项目7由刘英编写,王凤岭和蒋文沛负责全书的审核工作。在本书的编写过程中得到了南宁职业技术学院计算机应用研究所的大力支持,在此表示衷心感谢。

本书配有电子课件、源码等相关资源,可在科学出版社职教技术出版中心网站www.abook.cn 搜索下载。

由于编写时间仓促,作者水平有限,书中难免存在不足之处,敬请专家和广大读者批评指正。

目 录

绪论 软件项目开发概述	1
0.1 项目问题	1
0.2 解决方案	2
0.3 归纳总结	2
0.3.1 软件工程概述	2
0.3.2 软件项目管理	6
0.4 思考与提高	8
项目 1 保障软件项目质量	9
1.1 项目案例分析	9
1.1.1 项目问题	9
1.1.2 解决方案	9
1.2 工作任务	9
1.2.1 使用 Visual SourceSafe 进行版本控制	9
1.2.2 使用 JIRA 进行缺陷管理	18
1.3 归纳总结	23
1.3.1 版本控制	23
1.3.2 软件质量	24
1.3.3 JIRA 中的概念	25
1.4 知识拓展	28
1.4.1 常见配置管理工具	28
1.4.2 常见缺陷管理工具	29
1.5 思考与提高	33
项目 2 策划网上书店	34
2.1 项目案例分析	34
2.1.1 项目问题	34
2.1.2 解决方案	34
2.2 工作任务	35
2.2.1 可行性研究	35
2.2.2 使用 Project 制定项目计划	41
2.3 归纳总结	44
2.3.1 为什么要做计划	44
2.3.2 WBS 分解	44
2.3.3 制定计划的原则	45



2.4	知识拓展	45
2.5	思考与提高	47
项目 3	分析网上书店	48
3.1	项目案例分析	48
3.1.1	项目问题	48
3.1.2	解决方案	49
3.2	工作任务	49
3.2.1	使用 StarUML 捕获用例	49
3.2.2	编写需求规格说明书	57
3.3	归纳总结	59
3.3.1	用例模型	59
3.3.2	参与者	59
3.3.3	用例	60
3.3.4	建立用例模型的步骤	61
3.4	知识拓展	62
3.4.1	快速原型法	62
3.4.2	数据流图	63
3.5	思考与提高	64
项目 4	设计网上书店	65
4.1	项目案例分析	65
4.1.1	项目问题	65
4.1.2	解决方案	65
4.2	工作任务	66
4.2.1	应用 UML 构建系统分析模型	66
4.2.2	编写概要设计报告和详细设计报告	76
4.2.3	设计数据库概念数据模型, 画出 E-R 图	80
4.3	归纳总结	87
4.3.1	系统架构设计	87
4.3.2	软件设计	89
4.3.3	UML 图	90
4.3.4	E-R 模型	92
4.4	知识拓展	93
4.4.1	结构化程序设计	93
4.4.2	详细设计描述工具	94
4.5	思考与提高	96
项目 5	实现网上书店	97
5.1	项目案例分析	97
5.1.1	项目问题	97
5.1.2	解决方案	97

5.2	工作任务	98
5.2.1	编程规约和规范的审查	98
5.2.2	应用 NUnit 进行单元测试	100
5.3	归纳总结	104
5.3.1	标准化的重要性	104
5.3.2	单元测试的优点	105
5.3.3	单元测试的内容	105
5.4	知识拓展	106
5.4.1	静态测试	106
5.4.2	程序设计语言的选择	106
5.5	思考与提高	108
项目 6	测试网上书店	109
6.1	项目案例分析	109
6.1.1	项目问题	109
6.1.2	解决方案	109
6.2	工作任务	109
6.2.1	编写测试计划和测试用例	109
6.2.2	执行测试用例	114
6.2.3	性能测试	116
6.2.4	填报软件缺陷、跟踪缺陷	125
6.2.5	编写测试报告	128
6.3	归纳总结	130
6.3.1	基于 Web 的系统测试方法	130
6.3.2	测试错误类型	134
6.4	知识拓展	134
6.4.1	黑盒测试	134
6.4.2	白盒测试	137
6.5	思考与提高	137
项目 7	发布网上书店	138
7.1	项目案例分析	138
7.1.1	项目问题	138
7.1.2	解决方案	138
7.2	工作任务	138
7.2.1	系统实施	138
7.2.2	客户培训	157
7.2.3	客户验收	160
7.3	归纳总结	162
7.3.1	系统实施过程	162
7.3.2	系统培训过程	163



7.3.3 系统验收流程	163
7.4 知识拓展	164
7.4.1 项目维护	164
7.4.2 过程改进	166
7.5 思考与提高	167
附录1 C#编码规范	168
附录2 数据库命名规范	187
参考文献	189

绪论

软件项目开发概述

0.1 项目问题

某软件公司刚成立，从学校招聘了一些新的开发人员。最近，该公司接到一个网上书店的软件项目。由于公司刚成立，新招的开发人员都只有编程的经验，没有完整的项目开发经验。该公司该如何开展网上书店的项目开发呢？

通常在计算机软件的开发和维护过程中会遇到的一系列严重问题。

1) 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，实际进度比预定计划一再拖延的现象并不罕见。

2) 用户对“已完成”系统不满意的现象经常发生。

3) 软件产品的质量往往靠不住。错误一大堆，补丁一个接一个。

4) 软件的可维护程度非常低。

5) 软件通常没有适当的文档资料。

6) 软件的成本不断提高。

7) 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。

造成上述问题的原因主要有以下两个方面。

(1) 与软件本身的特点有关

软件同传统的工业产品相比，有如下一些特性。

1) 软件是一种逻辑实体，具有抽象性。这个特点使它与其他工程对象有着明显的差异。人们可以用纸、内存、磁盘或者光盘进行记录，但却无法看到软件本身的形态，必须通过观察、分析、思考、判断，才能了解它的特性。

2) 软件没有明显的制造过程。一旦研制开发成功，就可以大量复制，所以对软件的质量控制，必须着重在软件开发方面。

3) 软件在使用过程中，没有磨损、老化的问题。软件在生存周期后期不会因为磨损而老化，但会为了适应硬件、环境以及需求的变化而进行修改，而这些修改会不可避免地引入错误，导致软件失效率升高，从而引起软件退化。当修改的成本变得难以接受时，软件就被抛弃。

4) 软件对硬件和环境有着不同程度的依赖性，这导致了软件移植的问题。

5) 软件的开发至今尚未完全摆脱手工作坊式的开发方式，生产效率低。

6) 软件是复杂的，而且以后会更加复杂。软件是人类有史以来生产的复杂度最高的工业产品之一。软件涉及人类社会的各行各业、方方面面，软件开发常常涉及其他领域



的专业知识，这对软件工程师提出了很高的要求。

7) 软件的成本相当昂贵。软件开发需要投入大量、高强度的脑力劳动，成本非常高，风险也大。现在软件的开销已大大超过了硬件的开销。

8) 软件工作牵涉到很多社会因素。许多软件的开发和运行涉及机构、体制和管理方式等问题，还会涉及到人们的观念和心理。这些人为的因素，常常成为软件开发的困难所在，直接影响到项目开发的成败。

(2) 与软件开发和维护的方法不正确有关

软件开发和维护的不正确方法主要表现为：忽视软件开发前期的需求分析；开发过程没有统一的、规范的方法作指导，文档资料不齐全，忽视人与人的交流；忽视测试阶段的工作，提交用户的软件质量差；轻视软件的维护。这些大多数都是软件开发过程管理方面的原因。

0.2 解决方案

在计算机系统发展的初期，硬件通常用来执行某个单一的程序，而这个程序又是为一个特定的目的而编制的。早期当通用硬件成为平常事情的时候，软件的通用性却是有限的。大多数软件是由使用该软件的个人或机构研制的，软件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。

从 20 世纪 60 年代中期到 70 年代中期是计算机系统发展的第二个时期，在这一时期软件开始作为一种产品被广泛使用，“软件作坊”出现了，专职应别人的需求写软件。这一软件开发的方法基本上沿用早期的个体化软件开发方式，但软件的数量急剧膨胀，软件需求日趋复杂，维护的难度越来越大，开发成本令人吃惊地增多，而失败的软件开发项目却屡见不鲜。“软件危机”就这样开始了！

“软件危机”使得人们开始对软件及其特性进行更深一步的研究，改变了人们早期对软件的不正确看法。早期那些被认为是优秀的程序常常很难被别人看懂，通篇充满了程序技巧；而现在人们普遍认为优秀的程序除了功能正确、性能优良之外，还应该容易看懂、容易使用、容易修改和扩充。人们针对软件危机的表现和原因，经过不断的实践和总结，越来越认识到：按照工程化的原则和方法组织软件开发工作，是摆脱软件危机的一个主要出路。

0.3 归纳总结

0.3.1 软件工程概述

1. 软件工程的介绍

软件工程是一门研究如何用系统化、规范化、数量化等工程原则和方法去进行软件开发和维护的学科。

软件工程包括两个方面的内容：软件开发技术和软件项目管理。软件开发技术包括

软件开发方法学、软件工具和软件工程环境。软件项目管理包括软件度量、项目估算、进度控制、人员组织、配置管理、项目计划等。

统计数据表明,大多数软件开发项目的失败,并不是由于软件开发技术方面的原因。它们的失败是由于不适当的管理造成的。遗憾的是,尽管人们对软件项目管理重要性的认识有所提高,但在软件管理方面的进步远比在设计方法学和实现方法学上的进步小,至今还提不出一套管理软件开发的通用指导原则。

2. 软件生存期

软件生命周期是指一个从用户需求开始,经过开发、交付使用,以及在使用中不断地增补修订,直至软件报废的全过程,亦称软件生存期。我国颁布的《计算机软件文档编制规范》(GB/T 8567-2006)中规定,软件生命周期分为七个阶段。

(1) 可行性研究和项目开发计划

可行性研究和项目开发计划阶段必须要回答的问题是“要解决的问题是什么”。

(2) 需求分析

需求分析阶段的任务不是具体解决问题,而是准确地确定“软件系统必须做什么”,确定软件系统必须具备哪些功能。

(3) 概要设计

概要设计就是设计软件的结构,即软件应由哪些模块组成,这些模块的层次结构是怎样的,这些模块的调用关系是怎样的,每个模块的功能是什么。同时,还要设计该项目的应用系统的总体数据结构和数据库结构,即应用系统要存储什么数据,这些数据是什么样的结构,它们之间有什么关系等。

(4) 详细设计

详细设计阶段就是为每个模块完成的功能进行具体描述,要把功能描述变为精确的、结构化的过程描述。

(5) 编码

编码阶段就是把每个模块的控制结构转换成计算机可接受的程序代码,即写成以某特定程序设计语言表示的“源程序清单”。

(6) 测试

测试是保证软件质量的重要手段,其主要方式是在设计测试用例的基础上检验软件的各个组成部分。测试分为模块测试、组装测试和确认测试。

(7) 维护

软件维护是软件生存期中时间最长的阶段。已交付的软件投入正式使用后,便进入软件维护阶段,它可以持续几年甚至几十年。

在大部分文献中将生存周期划分为五个阶段,即需求定义、设计、编码、测试及维护。其中,需求定义阶段包括可行性研究和项目开发计划、需求分析,设计阶段包括概要设计和详细设计。

3. 软件生存期模型

为了描述软件生存期的活动,提出了多种生存期模型。例如,瀑布模型、循环模型、



增量模型、螺旋模型等。

(1) 瀑布模型

瀑布模型是由 1970 年 W.Royce 提出的、最早的软件开发模型。瀑布模型是将软件开发活动中的各项活动规定为依线性顺序连接的若干阶段工作,形如瀑布流水。换句话说,它将软件开发过程划分成若干个互相区别而又彼此联系的阶段,每个阶段中的工作都以上一个阶段工作的结果为依据,同时作为下一个阶段的工作基础。

该模型说明整个软件开发过程是按图中的各个阶段进行的,每个阶段的任务完成之后,产生相应的文档。它是以文档作为驱动、适合于需求很明确的软件项目开发的模型。瀑布模型得到了广泛的应用,它在消除非结构化软件、降低软件的复杂性、促进软件开发工程化方面起了很大的作用。但在大量的软件开发实践中也逐渐暴露出它的缺点。由于瀑布模型是一种理想的线性开发模式,缺乏灵活性,也无法解决软件需求不准确或者不明确的问题。这些缺点对软件开发带来了严重影响,由于需求不明确,会导致开发的软件不符合用户的需求而夭折。

(2) 循环模型

瀑布模型的缺陷在于将充满回溯、相互重叠的软件开发过程硬性地割裂为几个阶段,随着开发的软件规模越来越大,造成的危害也更加严重。循环模型是为了描述软件开发过程中可能的回溯,对瀑布模型进行了改进,开发各阶段可能循环重复。

(3) 增量模型

增量模型是一种非整体开发的模型。根据增量的方式和形式的不同,分为基于瀑布模型的渐增模型和基于原型的快速原型模型。

该模型具有较大的灵活性,适合于软件需求不明确、设计方案有一定风险的软件项目。增量模型和瀑布模型之间的本质区别是:瀑布模型属于整体开发模型,它规定在开始下一个阶段的工作之前,必须完成前一阶段的所有细节;而增量模型属于非整体开发模型,它推迟某些阶段或所有阶段中的细节,从而较早地产生工作软件。

(4) 螺旋模型

对于大型软件,只开发一个原型往往达不到要求。螺旋模型将瀑布模型和增量模型结合起来,并加入了风险分析。它是由 TRW 公司的 B.Boehm 于 1988 年提出的。该模型将开发划分为制定计划、风险分析、实施工程和客户评估四类活动。沿着螺旋线每转一圈,表示开发出一个更完善的新的软件版本。如果开发风险过大,开发机构和客户无法接受,项目有可能就此终止;多数情况下,会沿着螺旋线继续下去,自内向外逐步延伸,最终得到满意的软件产品。

螺旋模型将开发过程分为几个螺旋周期,每个螺旋周期可分为如下四个工作步骤。

- 1) 制定计划:确定目标、方案和限制条件。
- 2) 风险分析:评估方案、标识风险和解决风险。
- 3) 实施工程:开发确认产品。
- 4) 客户评估:计划下一周期工作。

(5) 喷泉模型

该模型是由 B.H.Sollers 和 J.M.Edwards 于 1990 年提出的开发模型,主要用于采用对象技术的软件开发项目。它克服了瀑布模型不支持软件重用和多项开发活动集成的局

限性。喷泉模型使开发过程具有迭代性和无间隙性。软件的某个部分常常被重复工作多次，相关对象在每次迭代中随之加入渐进的软件成分，即为迭代的特性；而分析和设计活动等各项活动之间没有明显的边界，即为无间隙的特性。

喷泉模型是以面向对象的软件开发方法为基础，以用户需求作为喷泉模型的源泉。从右边喷泉模型的图中可以看出其特点如下。

- 1) 喷泉模型规定软件开发过程有四个阶段，即分析、系统设计、软件设计和实现。
- 2) 喷泉模型的各阶段相互重叠，它反映了软件过程并行性的特点。
- 3) 喷泉模型以分析为基础，资源消耗成塔型，在分析阶段消耗的资源最多。
- 4) 喷泉模型反映了软件过程迭代性的自然特性，从高层返回低层无资源消耗。
- 5) 喷泉模型强调增量开发，依据“分析一点，设计一点”的原则，并不要求一个阶段的彻底完成，整个过程是一个迭代的逐步提炼的过程。

6) 喷泉模型是对象驱动的过程，对象是所有活动作用的实体，也是项目管理的基本内容。

7) 喷泉模型在实现时，由于活动不同，可分为系统实现和对象实现，这既反映了全系统的开发过程，也反映了对象族的开发和重用过程。

(6) RUP 统一建模过程

RUP 软件开发生命周期是一个二维的软件开发模型，如图 0.1 所示。纵轴代表核心 workflow 是静态的一面，横轴代表时间显示过程动态的一面，用周期、阶段、迭代、里程碑等名词描述。

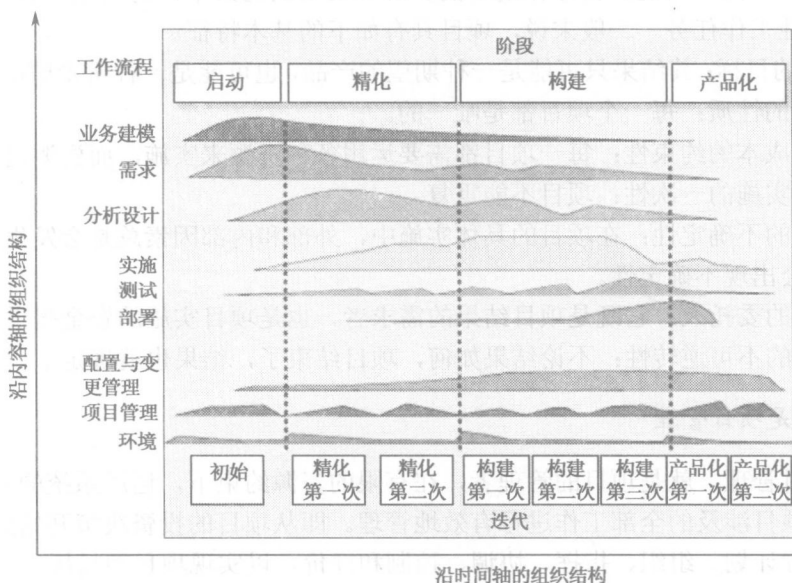


图 0.1 RUP 统一过程

从横轴来看，RUP 把软件开发生命周期划分为多个循环，每个循环生成产品的一个新版本，每个循环由如下四个连续阶段组成。

- 1) 初始阶段：定义最终产品视图和业务模型，确定系统范围。



- 2) 细化阶段: 设计、确定系统的体系结构, 制定工作计划即资源要求。
- 3) 构造阶段: 构造产品并继续演进需求、体系结构、计划直至产品提交。
- 4) 移交阶段: 把产品提交给用户使用。

RUP 的九个核心工作流如下。

- 1) 业务建模: 理解待开发系统所在的机构及其商业运作, 确保所有人员对它有共同的认识, 评估待开发系统对结构的影响。
- 2) 需求: 定义系统功能及用户界面, 为项目预算及计划提供基础。
- 3) 分析与设计: 把需求分析结果转换为分析与设计模型。
- 4) 实现: 把设计模型转换为实现结果, 并做单元测试, 集成为可执行系统。
- 5) 测试: 验证所有需求是否已经被正确实现, 对软件质量提出改进意见。
- 6) 部署: 打包、分发、安装软件, 培训用户及销售人员。
- 7) 配置与变更管理: 跟踪并维护系统开发过程中产生的所有制品的完整性和一致性。
- 8) 项目管理: 为软件开发项目提供计划、人员分配、执行、监控等方面指导, 为风险管理提供框架。
- 9) 环境: 为软件开发机构提供软件开发环境。

0.3.2 软件项目管理

1. 什么是项目

所谓项目, 简单地说, 就是在既定的资源和要求的约束下, 为实现某种目的而相互联系的一次性工作任务。一般来说, 项目具有如下的基本特征。

- 1) 明确的目标: 其结果只可能是一种期望的产品, 也可能是一种所希望得到的服务。
- 2) 独特的性质: 每一个项目都是唯一的。
- 3) 资源成本的约束性: 每一项目都需要运用各种资源来实施, 而资源是有限的。
- 4) 项目实施的一次性: 项目不能重复。
- 5) 项目的不确定性: 在项目的具体实施中, 外部和内部因素总是会发生一些变化, 因此项目也会出现不确定性。
- 6) 特定的委托人: 它既是项目结果的需求者, 也是项目实施的资金提供者。
- 7) 结果的不可逆转性: 不论结果如何, 项目结束了, 结果也就确定了。

2. 什么是项目管理

所谓项目管理, 就是项目的管理者, 在有限的资源约束下, 运用系统的观点、方法和理论, 对项目涉及的全部工作进行有效地管理。即从项目的投资决策开始到项目结束的全过程进行计划、组织、指挥、协调、控制和评价, 以实现项目的目标。

项目管理内容如下。

- 1) 项目范围管理是为了实现项目的目标, 对项目的工作内容进行控制的管理过程。它包括范围的界定、范围的规划和范围的调整等。
- 2) 项目时间管理是为了确保项目最终按时完成的一系列管理过程。它包括具体活动界定、活动排序、时间估计、进度安排及时间控制等工作。