“十二五”高等职业教育计算机类专业规划教材

软件工程案例教程

RUANJIAN GONGCHENG ANLI JIAOCHENG

汪作文　编著

中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

“十二五”高等职业教育计算机类专业规划教材

软件工程案例教程

汪作文　编著

中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

内 容 简 介

本书比较系统地介绍了软件工程的概念、技术和方法，内容包括软件工程概述、可行性分析、需求分析基础、结构化分析方法、结构化的设计方法、面向对象的分析与设计、软件编码与实现、用户界面设计、软件测试等。在面向对象的分析和设计方法中，还讲述了统一建模语言UML。全书采用实际的软件项目——“尚品购书网站”系统作为案例，通过该项目在软件开发的各个阶段的文档设计，系统地介绍了软件项目的开发过程，为学习软件开发技术的学生和技术人员提供了帮助和借鉴。书中还附带一定的例题和习题，便于教学与自学。

本书可作为高等职业院校计算机专业的教材或教学参考书，也可作为软件开发人员的参考书。

图书在版编目（CIP）数据

软件工程案例教程/汪作文编著. —北京：中国铁道出版社，2013.10

“十二五”高等职业教育计算机类专业规划教材

ISBN 978-7-113-17463-7

Ⅰ. ①软… Ⅱ. ①汪… 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(2013)第 240527 号

书　　名：软件工程案例教程
作　　者：汪作文　编著

策划编辑：翟玉峰　　**读者热线：**400-668-0820
责任编辑：翟玉峰　何　佳
封面设计：白　雪
责任印制：李　佳

出版发行：中国铁道出版社（100054，北京市西城区右安门西街 8 号）
网　　址：http://www.51eds.com
印　　刷：北京华正印刷有限公司印刷
版　　次：2013 年 10 月第 1 版　　2013 年 10 月第 1 次印刷
开　　本：787 mm×1 092 mm　1/16　**印张：**12.5　**字数：**300 千
印　　数：1～3 000 册
书　　号：ISBN 978-7-113-17463-7
定　　价：26.00 元

FOREWORD

前 言

软件开发技术是一门新兴的技术，而软件工程则是指导软件开发与维护的工程学科。自 20 世纪 60 年代末期以来，人们为克服“软件危机”在这一领域进行了深入的研究，逐渐形成了系统的软件开发与维护理论、技术和方法，这些理论、技术和方法在指导软件的开发实践中发挥了重要作用。

软件工程学的研究范围非常广泛，学科内的新技术、新方法不断涌现。本书着重借助一个实际的软件开发案例，从实用角度讲解软件开发技术的相关概念、基本原理和技术方法，同时也注意了全书的系统性和先进性。

本书共 9 章。第 1 章介绍了软件工程的概念、软件工程的发展和软件危机，着重介绍了软件生存周期、软件开发模型及软件工程的基本概念和基本内容。第 2 章介绍了可行性分析的具体内容，从第 3 章到第 9 章是本书的重点，分别论述了需求分析、结构化的分析和设计、面向对象的分析和设计、软件编码与实现、用户界面设计、软件测试等各个阶段的各种方法和技术，对 SAD 方法、数据流图、数据字典、结构图、N-S 结构图、PDL 语言、判定树、判定表等作了比较详细的介绍。同时也介绍了 OOA、OOD、OOP 等面向对象的分析和设计技术以及标准建模语言 UML、UML 的静态建模机制、UML 的动态建模机制、UML 软件开发过程等内容。全书使用实际的软件项目——“尚品购书网站”系统作为案例，通过该项目在软件开发的各个阶段的文档设计，系统地介绍了软件项目的开发过程。

本书内容新颖，将实际的软件开发案例贯穿整个教学过程，语言文字通俗易懂；各章重点、难点突出，原理、技术和方法的阐述融于丰富的实例之中；每一章首都有要点，从第 2 章开始，每一章都给出了相应的软件文档。各章都有习题，便于教学与自学。本书可作为高等院校“软件工程”课程教材或教学参考书，也可供从事软件开发与应用的工程技术人员和软件项目管理人员阅读参考。

在写作的过程中还得到了何婧、程琼老师的支持，在此表示感谢。

由于编者水平有限，编写时间仓促，书中难免有疏漏之处，恳请专家和读者批评斧正。

编 者

2013 年 6 月

CONTENTS

目 录

第1章 软件工程概述

- 软件与软件危机
- 软件工程的概念
- 软件生存周期
- 软件体系结构
- 软件开发模型
- 计算机辅助软件工程

在信息社会里，信息的获取、处理、交流和决策都需要高质量的软件，这样就促使人们对计算机软件的品种、数量、质量、成本和开发时间等提出越来越高的要求。而随着计算机应用的逐步扩大，软件需求量迅速增加，规模也日益增大，长达数万行、数十万行乃至百万行以上的软件，也越来越多。软件规模的膨胀，带来了它的复杂度的增加，而开发一个数万以至数百万行的软件，即使是富有经验的软件开发技术人员，也难免顾此失彼。其结果是，软件开发计划一拖再拖，成本失去控制，软件质量得不到保证。

为了扭转这种被动局面，自20世纪60年代末期以来，人们十分重视软件开发技术的研究，十分重视软件开发的方法、工具和环境的研究，并在软件开发技术这一领域取得了重要的成果，逐步形成了计算机科学中一门新学科——软件工程学。

1.1 软件工程背景

1.1.1 软件的定义

计算机软件是与计算机系统操作有关的程序、规程、规则及任何与之相关的文档及数据。它由两部分组成：一是计算机可执行的程序及有关数据；二是计算机不可执行的，与软件开发、运行、维护、使用相关的文档。

程序（Program）是用程序设计语言描述的、适合计算机处理的语句序列。它是软件开发人员根据用户需求开发出来的。目前的程序设计语言有三种类型：依赖于具体计算机的机器语言、汇编语言，独立于机器的面向过程的语言，以及独立于机器的面向对象的语言。机器语言是用中央处理器（CPU）指令集表示的符号语言，优秀的软件开发人员使用机器语言可以开发出时空开销较小的高质量程序。但是，用机器语言编写程序时工作效率低，程序难以阅读和调试，不利于软件的维护，也难以在不同的CPU系统中推广使用。而高级语言与机器

无关，其表达能力强，容易阅读和修改，大大提高了软件开发效率。今天，世界上的程序设计语言有几百种，但广泛使用的不过十余种，如支持现代软件开发的 C 语言，支持面向对象设计方法的 C++语言，支持网络计算的面向对象程序设计语言 Java，还有支持 Web 应用的 C#语言等。

文档（Document）是一种数据媒体和其上所记录的数据。文档记录软件开发活动和阶段性成果，它具有永久性并能供人或机器阅读。它不仅用于专业人员和用户之间的通信和交流，而且还可以用于软件开发过程的管理和运行阶段的维护。为了提高软件开发的效率，提高软件产品的质量，许多国家对软件文档都制订了详尽、具体的规定，颁布了各种规范和标准。我国也制订了相应的规范和标准，如：《计算机软件开发规范》《计算机软件需求说明编制指南》《计算机软件测试文件编制规范》等。

1.1.2 软件的特点和分类

1. 软件产品的特点

软件是逻辑产品而不是物理产品。因此，软件在开发、生产、维护和使用等方面与硬件相比均存在明显的差距。

软件开发与硬件相比，更依赖于开发人员的业务素质、智力、人员的组织、合作和管理。一般情况下，软件的开发、设计几乎都是从头开始的，开发的成本和进度很难估计。软件在提交使用之前，尽管经过了严格的测试和试用，但仍然不能保证软件没有潜在的错误。而硬件产品在经过生产、组装、测试、试用后，设计过程的错误一般是可以排除的。

硬件试制成功以后，批量生产需要建生产线，投入大量的人力、物力和资金。生产过程中还要进行严格的质量控制，对每件产品进行检验。而软件产品开发成功以后，只需要对原版软件进行复制即可。但是，软件在使用过程中的维护工作却比硬件复杂得多。另外，软件产品是逻辑的而不是物理的，所以软件产品不会磨损和老化。这与硬件产品也是不一样的。

2. 软件产品的分类

1）按软件的功能进行划分

（1）系统软件。能与计算机硬件紧密配合在一起，使计算机系统各个部件、相关的软件和数据协调、高效地工作的软件。例如，操作系统、数据库管理系统、设备驱动程序以及通信处理程序等。系统软件在运行时需要频繁地与硬件交互，以提供有效的用户服务，共享资源的共享，其间伴随着复杂的进程管理和复杂的数据结构处理。系统软件是计算机系统必不可少的一个组成部分。

（2）支撑软件。是协助用户开发软件的工具性软件，其中包括帮助程序开发人员开发软件产品的工具，也包括帮助管理人员控制开发的进程的工具。如：各种软件包和专用程序等。

（3）应用软件。是在特定领域内开发，为特定目的服务的一类软件。现在几乎所有的国民经济领域都使用了计算机，为这些计算机应用领域服务的应用软件种类繁多。其中商业数据处理软件是所占比例最大的一类，工程与科学计算软件大多属于数值计算问题。

此外，应用软件在计算机辅助设计／制造（CAD／CAM）、系统仿真、智能产品嵌入软件（如汽车油耗控制、仪表盘数字显示、刹车系统），以及人工智能软件（如专家系统、模式识别）等方面表现卓越，使得传统的产业部门面目一新，带给人们惊人的生产效率和巨大的经济效益。而用于事务管理、办公自动化方面的软件也在企事业机关迅速推广，中文信息处理、计算机辅助教学（CAI）等软件使得计算机向家庭普及，甚至连幼儿也能在计算机上学习和游戏。

2）按软件规模进行划分

按开发软件所需的人力、时间以及完成的源程序行数，可确定六种不同规模的软件，六种软件分类如表 1-1 所示。

表 1-1　软件规模的分类

类　别	参加人员数	研制期限	产品规模（源程序行数）
微型	1	1～4 周	0.5K
小型	1	1～6 月	1K～2K
中型	2～5	1～2 年	5K～50K
大型	5～20	2～3 年	50K～100K
甚大型	100～1000	4～5 年	1M（=1024K）
极大型	2000～5000	5～10 年	＞1M

规模大、时间长、很多人参加的软件项目，其开发工作必须要有软件工程的知识做指导。而规模小、时间短、参加人员少的软件项目也得有软件工程概念，遵循一定的开发规范。其基本原则是一样的，只是对软件工程技术依赖的程度不同而已。

3）按软件服务对象的范围划分

（1）项目软件：也称定制软件，是受某个特定客户（或少数客户）的委托，由一个或多个软件开发机构在合同的约束下开发出来的软件。例如，军用防空指挥系统、卫星控制系统。

项目软件中有的软件带有试验研究性质，项目完成后根据需要可能在此基础上做进一步开发。

（2）产品软件：是由软件开发机构开发出来直接提供给市场，或是为千百个用户服务的软件。例如，文字处理软件、文本处理软件、财务处理软件、人事管理软件等。

由于产品软件要参与市场竞争，其功能、使用性能以及培训和售后服务显得尤为重要。

1.1.3　软件的发展

自从第一台计算机问世以来，软件的发展如表 1-2 所示，可以划分为三个阶段：

程序设计阶段，约为 20 世纪 50 至 60 年代。

程序系统阶段，约为 20 世纪 60 至 70 年代。

软件工程阶段，约为 20 世纪 70 年代以后。

表 1-2　计算机软件发展的三个时期及其特点

时期 / 特点	程序设计	程序系统	软件工程
软件所指	程序	程序及说明书	程序、文档、数据
主要程序设计语言	汇编及机器语言	高级语言	软件语言*
软件工作范围	程序编写	包括设计和测试	软件生存期
软件使用者	程序设计者本人	少数用户	市场用户
软件开发组织	个人	开发小组	开发小组及大中型软件开发机构
软件规模	小型	中小型	大中小型
决定质量的因素	个人编程技术	小组技术水平	技术水平及管理水平
开发技术和手段	子程序和程序库	结构化程序设计	数据库、开发工具、开发环境、工程化开发方法、标准和规范、网络及分布式开发、面向对象技术及软件复用
维护责任者	程序设计者	开发小组	专职维护人员
硬件特征	价格高，存储容量小，工作可靠性差	价格有所下降，速度、容量及工作可靠性有明显提高	向超高速，大容量，微型化及网络化方向发展
软件特征	完全不受重视	软件技术的发展不能满足需要，出现软件危机	开发技术有进步，但未获突破性进展，价格高，未完全摆脱软件危机

说明：这里软件语言包括需求定义语言、软件功能语言、软件设计语言、程序设计语言等。

从表 1-2 中可以看到三个发展时期主要特征的对比。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 年代到 60 年代时，程序设计曾经被看作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是低级阶段的表现。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提出的要求，软件质量得不到保证，软件成本不断上升，软件开发的生产率无法提高。致使问题积累起来，形成了日益尖锐的矛盾。这就导致了软件危机。

1.1.4　软件危机

所谓软件危机，是指在软件开发过程中遇到的一系列严重的问题。

软件危机产生的原因：

（1）软件不同于硬件，它是计算机系统的逻辑部件而不是物理部件。在写出程序代码并在计算机上试运行之前，软件开发过程的进展情况较难衡量。很难检验开发的正确性且软件开发的质量也较难评价。因此，控制软件开发过程相当困难。此外，在软件运行过程中发现错误，很可能是遇到了一个在开发期间引入的、但在测试阶段没有能够检测出来的错误，所以软件维护常常意味着修改原来的设计。这样，维护的费用十分惊人，客观上使得软件较难维护。

（2）软件开发的过程是多人分工合作，分阶段完成的过程，参与人员之间的沟通和配合十分重要。但是，相当多的软件开发人员对软件的开发和维护存在不少错误的观念，在实践的过程中没有采用工程化的方法，或多或少采用了一些错误的方法和技术，这是造成软件危机的主要原因。

（3）开发和管理人员只重视开发而轻视问题的定义，使软件产品无法满足用户的要求。对用户的要求没有完整准确的认识就急于编写程序。这是许多软件开发失败的另一主要原因。事实上，许多用户在开始时并不能准确具体地叙述他们的需要，软件人员需要做大量深入细致的调查研究工作，反复多次与用户交流信息，才能真正全面、准确、具体地了解用户的要求。

（4）软件管理技术不能满足现代软件开发的需要，没有统一的软件质量管理规范。首先是文档缺乏一致性和完整性，从而失去管理的依据。因为程序只是完整软件产品的一个组成部分，一个软件产品必须由一组的配置组成，不能只重视程序而应当特别重视软件配置。其次，由于成本估计不准确，资金分配混乱，人员组织不合理，进度安排无序，导致软件技术无法实施。

（5）在软件的开发和维护关系问题上存在错误的观念。软件维护工作通常是在软件完成之后进行的，因此是极端艰巨复杂的工作，需要花费很大的代价。所以做好软件的定义工作，是降低软件成本，提高软件质量的关键。如果软件人员在定义阶段没有正确、全面地理解用户要求，直到测试阶段才发现软件产品不完全符合用户的需要，这时再修改就为时已晚了。另外，在软件生存期的不同节点进行修改需要付出的代价是很不相同的。在早期引入变更，涉及面较小，付出的代价较低；在开发的中期，软件配置的许多成分已经完成，引入一个变更可能需要对所有已完成的配置成分都做相应的修改，不仅工作量大，而且逻辑上更复杂，因而付出的代价剧增。在软件“已经完成”后再引入变更，则需要付出更高得多的代价。因此，必须把软件工程的观念引入软件开发的各个阶段，建立起软件开发与维护的工程化的观念。

1.2 软件工程的基本原理

1.2.1 软件工程的定义

软件工程是一门综合性的交叉学科，它涉及计算机科学、工程科学、管理科学和数学等。计算机科学中的研究成果都可以用于软件工程，但计算机科学着眼于原理和理论，软件工程着眼于如何建造一个软件系统。此外，软件工程要用工程科学中的技术来进行成本估算、安排进度及制订计划和方案；软件工程还要利用管理科学中的方法、原理来实现软件生产的管

理；并用数学的方法建立软件开发中的各种模型和算法，如可靠性模型、说明用户需求的形式化模型等。

开发一个软件，除去那些规模很小的项目以外，通常要由多个软件人员分工合作、共同完成；开发阶段之间的工作也应有很好的衔接；开发工作完成以后，软件成果要面向用户，在应用中接受用户的检验。所有这些活动都要求人们改变过去那种把软件当作个人才智产物的观点，抛弃那些只按自己工作习惯，不顾与周围其他人员配合关系的做法。

在这一点上，软件开发与计算机硬件研制，甚至与高楼建设没有本质的差别。任何参加这些工程项目的人员，它们的才能只有在工程项目的总体要求和技术规范的约束下充分发挥和施展。

许多计算机和软件科学家尝试，把其他工程领域中行之有效的工程学知识运用到软件开发工作中来。经过不断实践和总结，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，也是摆脱软件危机的一个主要出路。

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。

1.2.2 软件工程的目标和原则

组织实施软件工程项目，从技术上和管理上采取了多项措施以后，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）实现要求的软件功能；
（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。

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。而且上述目标很可能是互相冲突的。例如，若降低开发成本，很可能同时也降低了软件的可靠性。另一方面，如果过于追求提高软件的性能，可能造成开发出的软件对硬件有较大的依赖，从而直接影响到软件的可移植性。

图 1-1 表明了软件工程目标之间存在的相互关系。其中有些目标之间是互补关系，例如，易于维护和高可靠性之间，低开发成本与按时交付之间。还有一些目标是彼此互斥的，例如，低开发成本与软件可靠性之间，提高软件性能与软件可移植性之间，就存在冲突。

这里提到的几个目标很自然地成为判断软件开发方法或管理方法优劣的衡量尺度。如果提出一种新的开发方法，通常关心的是它对满足哪些目标比现有的方法更为有利。实际上，实施软件开发项目就是力图在以上目标的冲突取得一定程度的平衡。

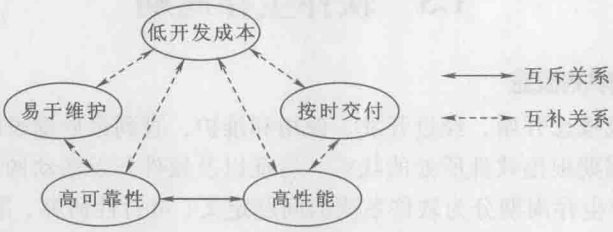

图 1-1　软件工程目标之间的关系

1.2.3　软件工程的基本原理

自从“软件工程”这一术语问世以来，研究软件工程的专家陆续提出了许多软件工程的准则或信条。美国著名软件工程专家 Boehm 于 1983 年提出了软件工程的 7 条基本原理，并认为这 7 条原理是确保软件产品质量和开发效率的最有效的原理。

1. 用分解阶段的生命周期计划严格管理

相关的统计数据表明，失败的软件项目 50%以上是由于计划不周造成的。这条原理表明，应该把软件生命周期分成若干阶段，并相应制订出切实的计划，然后严格按照计划对软件的开发和维护进行管理。Boehm 认为，在整个的软件生命周期内，应制订并严格执行的计划包括：项目概要计划、里程碑计划、项目控制计划、产品控制计划、验证计划和运行维护计划。

2. 坚持进行阶段评审

对一些软件项目开发的统计表明，大部分的错误属于设计方面的错误，约占 63%。错误发现的越晚，改正错误付出的代价就越大。因此，软件的质量保证工作不能等到编码结束之后再进行，应坚持进行严格的阶段评审，以便尽早发现错误。

3. 实行严格的产品控制

软件的需求变更是不可避免的，软件开发人员应采用科学的产品控制技术来顺应这种需求。当需求发生变更时，其他各个阶段的文档或代码应随之相应改动，以保证软件质量。

4. 采纳现代程序设计技术

采用先进的技术既可以提高软件的开发效率，又可以减少软件维护的成本。

5. 结果应能清楚地审查

由于软件产品的一些独特的性质，为更好地进行管理，应根据软件开发的总目标及完成的期限，尽量明确规定开发小组的责任和产品标准，从而使所得到的标准能清楚地审查。

6. 开发小组的人员应少而精

开发人员的素质和数量是影响软件质量和开发效率的重要因素，应该少而精。

7. 承认不断改进软件工程实践的必要性

Boehm 提出应把不断改进软件工程实践的必要性作为软件工程的第 7 条原理。根据这条原理，不仅要积极采纳新的软件开发技术，还要注意不断总结经验，收集进度和消耗等数据，进行出错类型和问题报告统计。这些数据既可以用来评估新的软件技术的效果，又可以用来指明必须着重注意的问题和应该优先进行研究的工具和技术。

1.3 软件生存周期

1. 软件生存周期的概念

软件产品从形成概念开始，经过开发、使用和维护，直到最后退役的全过程称为软件生存周期。软件生成周期根据软件所处的状态、特征以及软件开发活动的目的、任务可以划分为若干个阶段。软件生存周期分为软件系统的问题定义、可行性研究、需求分析、概要设计、详细设计、软件编码、软件测试、软件维护八个阶段。下面介绍各个阶段的主要任务、完成任务的途径及其阶段性产品。

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六个步骤：

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；制订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。

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。

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过程性描述。编写设计说明书，提交评审。

程序编写：把软件设计的过程性描述转换成计算机可以接受的程序代码。

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。

运行/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。

2. 目的和阶段

软件工程过程没有规定一个特定的软件生存周期模型或软件开发方法，各个软件开发机构可以为自己的开发项目选择一种生存周期模型，并将软件工程过程所包含的各种过程、活动和任务映射到该模型中。也可以选择和使用软件开发方法来执行适合于其软件项目的活动和任务。

研究软件生存周期的目的是为了科学、有效地组织和管理软件的生产，从而使软件生产更可靠、更经济。采用软件生存周期来划分软件的工程化开发，使软件开发分阶段依次进行。前一个阶段任务的完成是后一个阶段工作的前提和基础，而后一个阶段通常将前一个阶段提出的方案进一步具体化。每一个阶段结束之前都要接受严格的技术评审和管理评审。采用这种划分，使得每一个阶段的工作相对独立，有利于简化整个问题的解决，且便于不同人员分工协作。而且其严格的科学的评审制度保证了软件的质量，提高了软件的可维护性，从而大大提高了软件开发的生产率和成功率。

软件生存周期的各阶段有不同的划分。软件规模、种类、开发模式、开发环境和开发方法都影响软件生存周期的划分。在划分软件生存周期阶段时，应遵循以下规则，即：各阶段的任务应尽可能相对独立，同一阶段各项任务的性质应尽可能相同，从而降低每个阶段任务的复杂程度，简化不同阶段之间的联系，有利于软件项目开发的组织和管理。

1.4 软件体系结构

软件体系结构到底是什么？软件体系结构的思想最早是由 Dijsktra 等人提出的，Shaw、Perry 及 Wolf 等人在 20 世纪 80 年代末期作了进一步研究。虽然软件体系结构已经成为软件工程研究的重点，但是许多研究人员都是基于自己的经验从不同的角度、不同侧面对体系结构进行刻画的。

Perry 及 Wolf 等人认为软件体系结构由一组具有特定形式的体系结构元素组成，包括处理元素、数据元素和连接元素等三种。

Garlan 和 Perry 则指出，软件体系结构包括一个系统的构件结构、构件间的相互关系，以及控制构件设计与演化的原则及规范三个方面。

Shaw 和 Garlan 认为，体系结构是对构成系统的元素、这些元素间的交互、它们的构成模式，以及这些模式之间的限制的描述。

目前一个比较统一的定义是：软件体系结构是一个系统的高层结构共性的抽象，是建立系统时的构造模型、构造风格、构造模式。

目前，在商业软件开发中常用的软件体系结构有：层次结构、C/S 结构和 B/S 结构。

1. 层次结构

所谓层次结构，就是将软件的实现分成多个层次，低层的模块实现相对单纯的功能，多个低层模块组合成一个较高的模块，实现相对多的功能，最后所有的模块组合起来完成整个的软件的功能。

层次系统要求上层子系统调用下层子系统的功能，而下层子系统不能够调用上层子系统的功能。一般下层每个程序接口执行当前的一个简单的功能，而上层通过调用不同的下层子程序，并按不同的顺序来执行这些下层程序，层次结构就是以这种方式来完成多个复杂业务功能的。层次结构主要用于单机系统。

2. C/S 结构

客户-服务器结构简称 C/S 结构或两层体系结构，由服务器提供应用（数据）服务，多台客户机进行连接。如图 1-2 所示。

图 1-2 C/S 结构

3. B/S 结构

浏览器-服务器结构简称 B/S 结构，如图 1-3 所示。

在这种结构下，主要事务逻辑在服务器端（Server）实现，极少部分事务逻辑在前端浏览器（Browser）实现。客户机统一采用浏览器，用户工作界面是通过 WWW 浏览器来实现的。

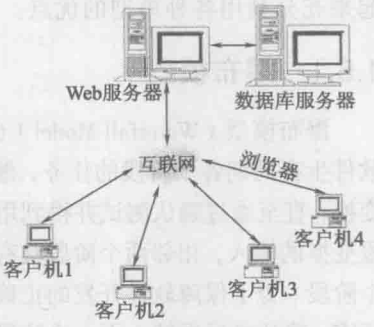

图 1-3 B/S 结构

4. B/S 和 C/S 结构的比较

1）响应速度

C/S 结构的软件系统比 B/S 结构的软件系统在客户端响应方面速度快，能充分发挥客户端的处理能力，很