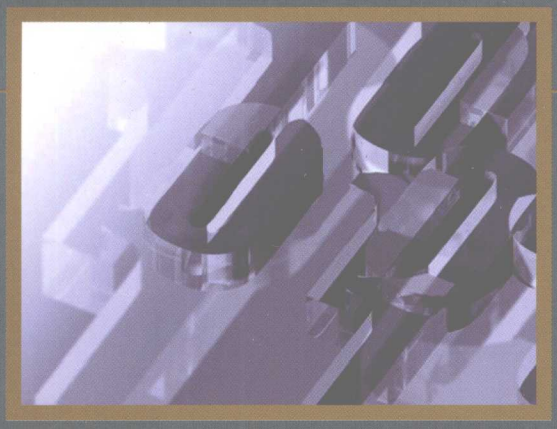
高等学校计算机科学与技术教材

软件工程基础

□ 张权范 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

软件工程基础

张权范　主编

清华大学出版社
北京交通大学出版社
·北京·

内 容 简 介

本书从面向数据流、面向数据结构、面向对象三个层面由浅入深地对软件工程进行了系统的介绍。本书最大的特色是：语言简单明了，概念清晰，内容丰富且实用，对每一个具体的知识点一般都有具体的真实的现场工作场景的案例来帮助读者理解相关的理论知识，跟踪了目前软件工程领域的最新成果。当然，本书的重点还是对传统的软件工程思想的描述，而对于软件测试以及面向对象的章节，因为目前在高等院校中都把它们作为单独的课程来开设，所以，仅以介绍为主，但也都给出了比较好的案例；对软件生命周期各环节的文档附有规格说明书，主要的文档还附有具体案例。

全书共分10章，是按软件工程的三个层面与软件生命周期的顺序来组织的，本书可作为高等院校计算机专业本科层次的教材，也可以作为大专层次院校的教材或者作为社会上广大读者的自学参考书。

图书在版编目(CIP)数据

软件工程基础/张权范编著. —北京：清华大学出版社；北京交通大学出版社，2009.8
(高等学校计算机科学与技术教材)
ISBN 978－7－81123－744－3

Ⅰ. 软…　Ⅱ. 张…　Ⅲ. 软件工程－高等学校－教材　Ⅳ. TP311.5

中国版本图书馆CIP数据核字(2009)第123469号

责任编辑：谭文芳
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：北京东光印刷厂
经　　销：全国新华书店
开　　本：185×260　印张：21　字数：534千字
版　　次：2009年8月第1版　2009年8月第1次印刷
书　　号：ISBN 978－7－81123－744－3/TP·508
印　　数：1～5 000册　定价：34.00元

前　言

软件工程是以软件的说明、开发、维护和管理为内容，利用工程学的原理和方法来组织和管理软件生产，以保证软件产品的质量，提高软件生产率。随着计算机应用的普及，计算机软件无处不在。该学科已成为计算机科学的一个重要分支与信息产业的一个支柱，软件工程已逐渐为人们所熟悉并得到广泛应用。人们都认识到，在项目开发过程中必须遵循软件工程原则。软件工程课程是计算机相关专业学生参加工作以后最能直接应用的一门专业课。在本书的编写过程中尽可能坚持简单明了与实用的原则讲述软件工程的基本概念、原理、方法和工具，介绍目前较成熟的、广泛使用的软件工程技术。

软件工程讲述“软件开发”和“做程序员”的道理，在一个团队中的合作精神。古人说：“读书明理”。我认为其中的重要内涵，是要有积极的人生观，以贡献社会为己任。本书是作者多年的软件工程教学与实践经验的总结，希望本书的出版能为培养我国软件专业人才发挥一定的作用。软件工程的观念、方法、策略和规范都是朴实无华的，并非要具备超凡的智慧才可领会，关键在于实践。要抱着科学的态度来看待它，因为它不是小说，没有引人入胜的情节。

如何正确安排软件的结构，合理组织、管理软件的生产，不仅仅是从事软件开发专业人员的事，广大计算机应用人员也需要掌握这方面的知识。本书可作为高等院校计算机专业的教材，也可供从事计算机软件开发及应用的广大科技人员做参考。

作者建议在学习软件工程课程之前，学生应当具备高级语言、数据结构、操作系统和数据库技术等方面的知识，并且要从宏观与微观相结合的角度来了解该课程。从宏观的角度来讲，可以从面向数据流、面向数据结构和面向对象等层次来整体把握该课程；从微观的角度来讲，要掌握各层次的各种软件工程工具的具体运用。本课程可安排在专业课学习的后期，毕业设计之前。使学生在毕业实习、毕业设计实践中，运用软件工程学的原理、方法和工具。

本书建议学时为 64 学时，各章内容安排及学时分配如下。

第 1 章　概述软件工程发展史，软件工程基本概念，软件开发常用模型（瀑布模型、快速原型、喷泉模型、软件重用开发模型和螺旋模型等），软件开发方法（4 学时）。

第 2 章　软件计划（4 学时）。

第 3 章　软件需求分析（10 学时）。

第 4 章　软件总体设计（4 学时）。

第 5 章　软件详细设计（4 学时）。

第 6 章　软件编码（4 学时）。

第 7 章　软件测试（4 学时）。

第 8 章　软件实施与维护（10 学时）。

第 9 章　软件项目管理（4 学时）。

第 10 章　面向对象软件工程技术（16 学时）。

本书各章配有与软件工程各个阶段对应的文档规范与适当数量的习题，供读者练习。

作　者

2009 年 6 月

目　录

第1章 概 述

本章要点：

- ☑ 软件
- ☑ 软件工程
- ☑ 软件危机
- ☑ 软件生存周期
- ☑ 软件开发模型
- ☑ 软件开发方法

1.1 软件工程学的几个基本概念

1.1.1 软件与软件工程

本节主要介绍软件与软件工程的概念、软件工程产生的背景。

1. 软件及其组成

软件是计算机系统中与硬件对等的一部分，是程序及相关文档资料的集合。因此，软件有两大组成要素：一是存储介质上的程序，它们是可执行的，并可产生用户需要的结果；二是相关的文档资料，它们既是软件开发过程中的质量保证，又是软件使用与维护的依据。两大组成要素详述如下。

（1）可执行部分

应用程序——是面向用户的解决各种特定实际问题的程序，如工程管理或科学计算程序、信息管理程序及实时监控程序等。

系统程序——是面向计算机系统的为应用程序服务的程序的综合，它们支撑应用程序的运行，如操作系统。

（2）不可执行部分

面向用户的文档——告诉用户如何使用、维护和修改程序，如用户手册、操作手册及程序维护手册等。

面向开发方的文档——提供软件开发过程的质量保证，如系统可行性论证报告、软件计划说明书、需求规格说明书、数据库设计说明书，以及测试计划、测试分析报告等。

项目越大，与其相关的文档资料也越重要，从以后的介绍中可以看出，程序编写在整个项目的开发过程中只占很少的一部分。

2. 软件危机与软件工程

（1）软件危机

软件危机是指在软件开发和维护过程中所遇到的一系列严重问题。总的来说包括两个方面，即如何满足日益增长的软件需求，以及如何维护数量不断膨胀的已有软件。软件危机具体表现如下。

① 供求矛盾。

② 软件成本和开发进度难以估计。

③ 软件产品不符合用户的实际需要。

④ 软件的可靠性差。

⑤ 软件的维护费急剧上升。

（2）软件危机产生的原因

产生软件危机的原因可以从内部与外部的角度来分析，就内因而言可归结为以下几方面。

① 目标不清。在软件开发过程中，对整个组织的目标没有一个明确、全面和定量的概念。这种目标的不明确可能是组织本身的问题，也可能是开发方对于组织目标的理解不清楚，在软件开发过程中用自己的主观理解加以替代，导致软件目标与组织目标相背离。

② 情况不明。软件是为需求方服务的，需求决定了软件的结构和功能。因此，开发方必须与相应层次的人员进行认真调研，通过与用户的反复交流，全面了解情况，否则会导致结果和客观情况不符合。

③ 通信误解。这是指需求方与开发方的工作人员的知识背景和工作经历不同，他们之间的交流会存在一定的难度。

④ 步骤混乱。软件开发是一个长期复杂的过程，它的各个工作环节之间有着很强的逻辑关系，若不遵守这种关系，就会产生不良后果。例如，没有完成软件的全面设计就开始编写程序，不仅无法保证各部分的正确衔接，而且肯定会造成返工。

⑤ 不遵守统一的标准。软件本质上是一种产品，其生产过程必须与其他工业产品的生产过程一样有其一套标准，开发人员在实际工作过程中必须加以严格遵守。

上述的原因，使人们懂得探索软件研制工作的正确方法的必要性。同时也说明我们对软件本身的认识不够深刻，也缺乏科学的研制方法和工具。

（3）解决软件危机的途径

根据软件危机产生的原因，寻找解决软件危机的途径并归纳如下。

① 组织良好、管理严密、各类人员协同配合。

② 统一开发标准。

③ 使用软件开发辅助工具，如 CASE（Computer Aided Softeware Engineering，计算机辅助软件工程）。

3. 软件工程

软件工程是采用工程的概念、原理、技术和方法来研制和维护软件，把正确的管理技术和当前最好的技术方法结合在一起。软件工程的目标在于研究一套科学的工程方法并与之相适应，发展一套方便的工具系统，力求用较少的投资获得高质量的软件。

1.1.2 软件生存周期

就像一个人从出生开始经过幼儿期、少年期、中年期、老年期直到死亡的过程一样，一个软件从提出设想到完成使命、报废为止，也经历了一个漫长的时期。通常把软件经历的这个漫长时期称为软件生存周期。一般软件生存周期主要经历几个阶段，即软件计划、软件需求分析、软件总体设计、软件详细设计、软件编码、软件测试和软件维护等。

（1）软件计划

这一阶段的工作是确定将要开发的软件系统是做什么的，在经济上、技术上和操作上是否可行。具体地说，就是要确定软件系统的工作范围；预测开发的系统所需要的资源，包括硬件、软件和人员；对软件开发成本进行初步估计，并写出软件计划任务书。

（2）软件需求分析

这一阶段的工作是对用户的要求进行分析和综合，确定软件的基本目标和逻辑功能要求，解决系统“做什么”的问题，并写出软件需求规格说明书。这份文件资料是软件工程中最重要的文件，是用户和软件研制人员之间相互共同约定和开发的基础。

（3）软件总体设计

这一阶段的主要任务是解决系统“怎么做”的问题。总体设计决定软件系统的总体结构即模块结构，并给出模块的相互调用关系、模块间传递的数据及每个模块的功能说明。这个阶段的文件资料是软件结构图和模块功能说明。

（4）软件详细设计

这一阶段给出每一个模块内部过程的描述，并写出软件详细设计说明书。

（5）软件编码

这一阶段是把软件设计方案加以具体实施。即根据软件详细设计说明书的要求，为软件系统中每一个功能模块编写程序并进行模块调试。这一阶段的文件资料是程序说明书、源程序。

（6）软件测试

软件测试阶段的主要任务是发现和排除错误，也就是对软件系统进行从上到下全面的测试和检验，看它是否符合软件总体设计方案规定的功能要求。这期间要提出测试标准，制定测试计划，确定测试方法。经过测试，纠错得到可运行的程序，同时写出软件测试报告。

（7）软件维护

由于经过测试的软件仍然可能有错，用户的需求和软件的操作环境也可能发生变化。因此，交付运行的软件系统仍然需要维护，所以软件维护的实质是对软件继续进行查错、纠错和修改。一般维护分为以下几方面。

① 改正性维护：对软件性能、功能、处理和实现中出现的错误进行纠正。

② 适应性维护：当软件处理对象或数据环境变化时，依某些适应性进行修改。

③ 完善性维护：为了提高软件的性能或者对可维护性方面所做的某些修改。

④ 预防性维护：为了改善将来的可靠性或可维护性，而对软件进行的修改或补充。

软件生存周期划分为上述7个阶段，这为工程化地开发软件系统提供了一个框架。但是，实际的开发工作不可能是直线的，常常存在着反复。例如，在软件设计阶段发现软件规格说明书有不完整或定义不确切之处，就要回到需求分析阶段进行“再分析”，测试阶段发

现模块内部或接口中的错误，就要回到设计阶段对原来的设计进行修改。

实际的软件工程设计流程如图 1 - 1 所示。

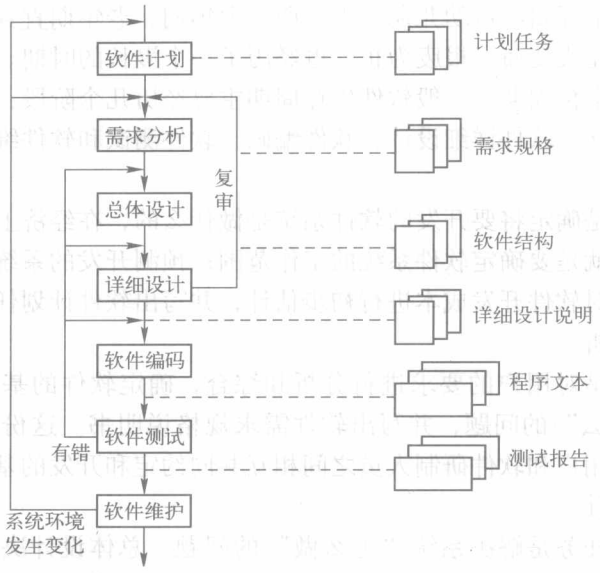


图 1 - 1 软件工程设计流程

软件生存周期中，软件计划、需求分析、软件总体设计、软件详细设计、软件编码和软件测试常称为软件的开发期，最后一个阶段称为软件的维护期。在软件开发期中，测试阶段工作量约占整个开发期总工作量的 40%，而在软件的整个生存周期中软件维护的周期最长，工作量也最大。

1.1.3 软件开发模型

根据软件生产工程化的需要，生存周期的划分也有所不同，从而形成了不同的软件生存周期模型（SoftWare Life Cycle Model），或称软件开发模型。

软件开发模型总体来说有传统的瀑布模型和后来兴起的少速原型模型。具体分为：瀑布模型、快速原型、喷泉模型、软件重用开发模型和螺旋模型。

1. 瀑布模型（Waterfall Model）

瀑布模型遵循软件生存周期的划分，明确规定每个阶段的任务。各阶段的工作顺序展开，恰如奔流不息、拾级而下的瀑布，如图 1 - 2 所示。

瀑布模型把软件生存周期分为计划时期、开发时期、运行时期 3 个时期。这 3 个时期又可细分为若干个阶段：计划时期可分为问题定义、可行性研究两个阶段；开发时期可分为需求分析、概要设计、详细设计、编码和测试 5 个阶段；运行时期可分为运行、维护两个阶段。瀑布模型软件开发有以下几个特点。

（1）软件生存周期的顺序性

只有前一阶段工作完成以后，后一阶段的工作才能开始，前一阶段的输出文档，就是后一阶段的输入文档。只有前一阶段有正确的输出，后一阶段才可能有正确的结果。如果在生

存周期的某一阶段出现了错误，往往要追溯到在它之前的一些阶段。

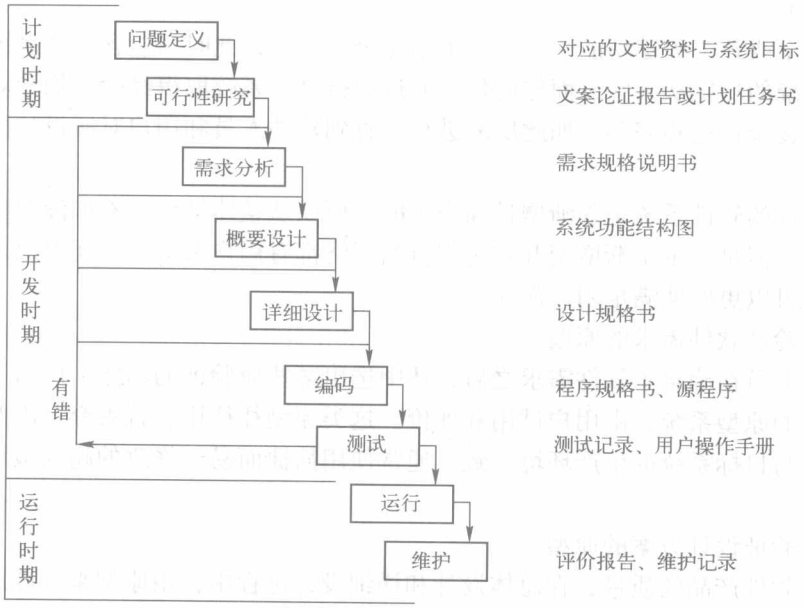


图1-2 适用于结构化开发技术的典型的瀑布模型

瀑布模型开发适合于在软件需求比较明确、开发技术比较成熟、工程管理比较严格的场合下使用。

（2）尽可能推迟软件的编码

程序设计也称为编码。实践表明，大、中型软件编码开始得越早，完成所需的时间反而越长。瀑布模型在编码之前安排了需求分析、总体设计、详细设计等阶段，从而把逻辑设计和编码清楚地划分开来，尽可能推迟程序编码阶段。

（3）保证质量

为了保证质量，瀑布模型软件开发在每个阶段都要完成规定的文档，每个阶段都要对已完成的文档进行复审，以便及早发现隐患，排除故障。

本书以瀑布模型为典型开发模型，介绍各阶段工作的具体方法、步骤和工具，其他模型可以参照执行。

2. 快速原型（Rapid Prototype Model）

正确的需求定义是系统成功的关键。但是许多用户在开始时往往不能准确地叙述他们的需要，软件开发人员需要反复多次地和用户交流信息，才能全面、准确地了解用户的要求。当用户实际使用了目标系统以后，也常常会改变原来的某些想法，对系统提出新的需求，以便使系统更加符合他们的需要。

理想的做法是先根据需求分析的结果开发一个原型系统，请用户试用一段时间，以便能正确地认识到他们的实际需要是什么，这相当于工程上先制作“样品”试用后，做适当改进，然后再批量生产一样，这就是快速原型法。虽然此法要额外花费一些成本，但是可以尽早获得更正确完整的需求，可以减少测试和调试的工作量，提高软件质量。因此快速原型法如果使用得当，能减少软件的总成本，缩短开发周期，是目前比较流行的实用开发模式。