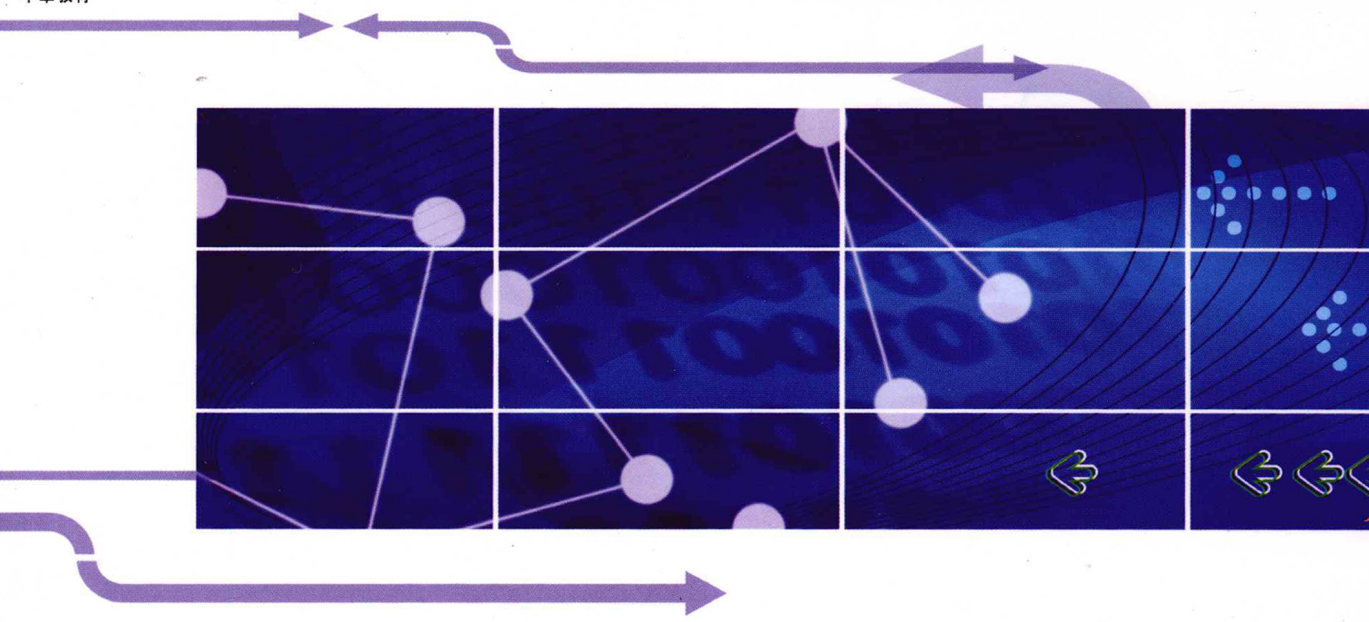
HZ BOOKS
华章教育

计算机应用技术规划教材



软件工程实用教程

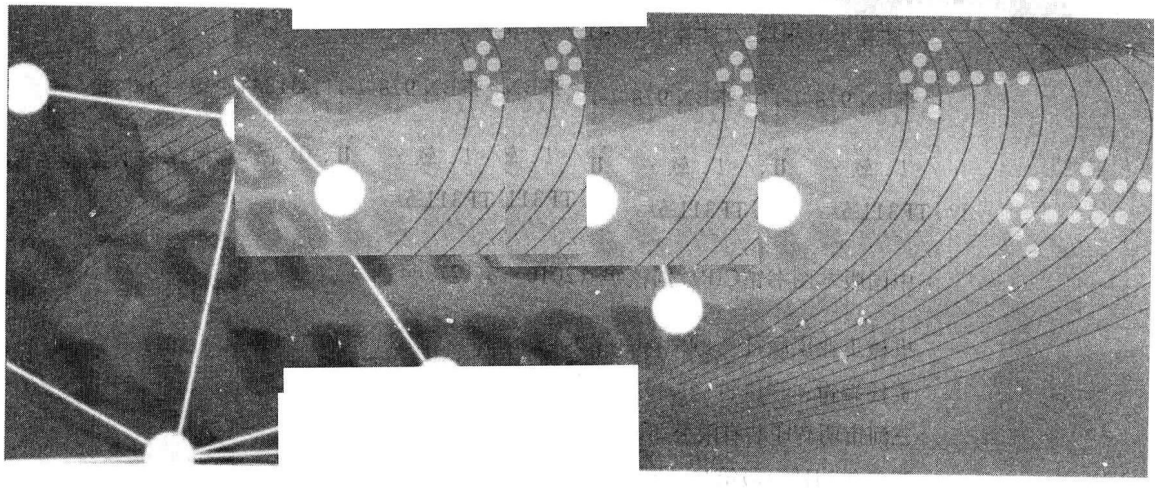
吕云翔 王洋 王昕鹏 编著

机械工业出版社
China Machine Press

计算机应用技术规划教材

软件工程实用教程

吕云翔 王洋 王昕鹏 编著



机械工业出版社
China Machine Press

本书按照典型的软件开发过程来组织内容，旨在培养学生具备软件工程思想以及实际软件开发的能力。全书共8章，主要内容包括：软件工程的起源，软件工程相关概念，软件工程方法、过程和工具；软件可行性研究及软件需求分析，软件设计，软件编码及实现，软件测试与维护；面向对象的软件工程；软件工程中涉及的管理方面的相关内容，如项目计划、软件资源管理、进度管理、人员管理、风险管理等内容。

本书可作为普通高校计算机相关专业“软件工程”课程的教材，也可供业余计算机和软件开发爱好者参考。

封底无防伪标均为盗版

版权所有，侵权必究

本书法律顾问　北京市展达律师事务所

图书在版编目（CIP）数据

软件工程实用教程 / 吕云翔，王洋，王昕鹏编著. —北京：机械工业出版社，2010.10
（计算机应用技术规划教材）

ISBN 978-7-111-31844-6

Ⅰ. 软… Ⅱ. ① 吕… ② 王… ③ 王… Ⅲ. 软件工程-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（2010）第177964号

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：李俊竹
三河市明辉印装有限公司印刷
2011年1月第1版第1次印刷
185mm×260mm · 15.25印张
标准书号：ISBN 978-7-111-31844-6
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
客服热线：（010）88378991；88361066
购书热线：（010）68326294；88379649；68995259
投稿热线：（010）88379604
读者信箱：hzjsj@hzbook.com

前　言

20世纪60年代，为了解决出现的“软件危机”，人们提出了软件工程的概念，并将其定义为“为了经济地获得可靠的和能在实际机器上高效运行的软件，而建立和使用的健全的工程规则”。经过40多年的发展，人们对软件工程有了更全面更科学的认识，软件工程已经成为一门包括理论、方法、过程等内容的独立的学科，并出现了相应的软件工程支撑工具。

然而即使在21世纪的今天，软件危机的种种表现依然没有彻底地得到解决，现实中的很多项目依然挣扎在无法完成或无法按照规定的时间、成本、按预期的质量完成的泥潭中，面临着失败的危险。究其原因，是软件工程的思想和方法并未深入到计算机科学技术，特别是软件开发领域中，并指导人们的开发行为。

为了振兴中国的计算机和软件产业，培养具备软件工程思想和技术，并具有相应开发经验的人才，国家近年来十分重视软件工程相关课程的建设和人才培养。除了建立专门的软件工程专业，也倡导在计算机科学与技术及相关专业开设软件工程课程，使得软件工程思想和技术在中国的IT人才中得到普及。

本书面向普通大学计算机及相关专业的学生，也可供计算机和软件开发爱好者自学使用。为了体现软件工程知识体系的层次，并具有更好的实践指导意义，本书除了介绍软件工程的基本概念、理论、方法和过程，还十分重视软件工程相关工具使用方法的介绍，并通过实际案例来讲述软件工程在实际软件项目开发中的应用和体现。

本书按照典型的软件开发过程来组织内容，全书分为8章。第1章概要介绍软件工程的起源，软件工程相关概念，软件工程方法、过程和工具；第2～5章分别介绍软件可行性研究及软件需求分析，软件设计，软件编码及实现，软件测试与维护；第6章详细介绍了面向对象的软件工程；第7章介绍了软件工程中涉及的管理方面的相关内容，包括项目计划、软件资源管理、进度管理、人员管理、风险管理等内容。

每章开头的“本章目标”概述了该章的主要内容；每章基本按照基础理论和知识介绍、相关技术和方法介绍、软件工程工具使用介绍和实际软件项目应用介绍的顺序来组织内容；每章小结对该章主要内容进行总结和回顾；在每章的最后给出了一定数量的练习题，帮助读者检验该章的学习效果，加深重点知识的印象。

为了进一步加强本书的实践指导意义，本书在第8章安排进行一个相对完整的项目开发，贯穿面向对象的软件工程相关内容，并在该章给出了一些建议练习项目，读者可以选择一些项目进行实际开发，并在这个过程中体会软件工程的基本知识和相关工具的应用。

本书作者一直在北京航空航天大学软件学院担任软件工程课程的教学工作，进行了大量的教学探索和研究。在编写本书的过程中，大量借鉴了笔者和同事在北航软件学院教学中的相关经验。在此感谢北航软件学院同仁的支持和帮助，以及提供的各种宝贵资料和建议。

由于作者的学习能力和水平有限，书中难免有错误和缺陷，恳请广大读者给予批评指正，也希望各位能将教学和学习过程中的经验、心得与我们交流。

编者

2010年6月于北京

yunxianglu@hotmail.com

试读结束：需要全本请在线购买： www.ertongbook.com

教学建议

教学内容	学习要点及教学要求	课时安排
第1章 软件工程概述	• 了解软件的概念、特点及主要分类。 • 了解软件危机的产生原因及其表现。 • 掌握软件工程的概念，以及软件工程的基本原则。 • 掌握软件过程的定义和基本活动。 • 熟知常用的几种软件生存周期模型。 • 了解常用软件工具及其作用。 • 了解与软件工程相关的资源。 • 了解课程案例概要情况。	4
第2章 可行性研究及软件需求分析	• 了解可行性研究的目的、意义和内容。 • 掌握可行性研究的主要步骤。 • 了解需求分析的任务。 • 熟悉进行需求分析的步骤和方法。 • 掌握需求分析的原则。 • 熟悉需求分析的两种方法。 • 掌握结构化需求分析的几种常用建模方法。 • 熟悉Visio的功能和基本用法。 • 掌握绘制数据流图的方法。	6
第3章 软件设计	• 了解软件设计的意义和目标。 • 掌握软件设计的原则。 • 了解软件体系结构的定义和建模方法。 • 熟悉常见的软件体系结构风格。 • 了解软件的质量属性。 • 掌握面向数据流的软件设计方法。 • 了解概要设计和详细设计的主要内容和它们的区别。 • 熟悉数据设计、构件设计和界面设计的工具和方法。	6
第4章 软件编码及实现	• 了解程序设计语言的发展。 • 掌握程序设计语言的分类。 • 了解常见的程序设计语言。 • 了解在选择程序设计语言时要考虑的因素。 • 掌握良好的编码风格。 • 熟悉Visual Studio 2010的使用方法。	2
第5章 软件测试与维护	• 掌握软件测试的原则。 • 了解软件测试的常用模型。 • 了解软件测试的分类。 • 熟悉软件测试的一般步骤。 • 了解测试用例和测试用例设计方法 • 掌握等价类划分技术和基本路径测试技术。 • 了解软件维护的分类。 • 了解软件的可维护性概念。 • 掌握Visual Studio中Unit Test工具的使用方法。	4

（续）

教学内容	学习要点及教学要求	课时安排
第6章 面向对象的软件工程	• 掌握面向对象的基本概念。 • 理解面向对象与面向过程的区别。 • 掌握面向对象的软件开发过程。 • 了解UML的5个视图和13类图。 • 掌握面向对象分析和设计的方法。 • 掌握用例图的绘制方法。 • 熟悉顺序图的绘制方法。 • 掌握类图的绘制方法。	6
第7章 软件工程管理	• 了解软件项目管理的内容。 • 了解软件项目计划、范围管理、成本管理和时间管理主要内容。 • 了解常见的软件组织。 • 了解团队建设的过程 • 熟悉CMM/CMMI相关内容。 • 了解软件配置管理的内容。 • 了解风险管理的内容。 • 掌握软件文档的作用及分类。 • 掌握Project的基本用法。	4
第8章 项目综合实践	• 通过“图书馆信息管理系统”了解面向对象分析过程和方法。 • 通过“图书馆信息管理系统”了解面向对象设计过程和方法。 • 通过“图书馆信息管理系统”了解面向对象实现和测试过程及方法。	4
	教学总学时建议	36

说明：

① 本书为计算机及相关专业“软件工程”课程的教材，理论授课学时数为36学时，不同专业可根据不同的教学要求和计划教学时数酌情对教材内容进行适当取舍。

② 非计算机类专业使用本书可适当降低教学要求。

③ 本书理论授课学时数36学时，包含课堂讨论、练习等必要的课内教学环节。

④ 建议授课时间比例为：基础部分50%，工具部分20%，实践部分30%。

目　录

第1章　软件工程概述

【本章目标】

- 了解软件的概念、特点及主要分类。
- 了解软件危机的产生原因及其表现。
- 掌握软件工程的概念，以及软件工程的基本原则。
- 掌握软件过程的定义和基本活动。
- 熟悉常用的几种软件生存周期模型。
- 了解常用软件工具及其作用。
- 了解与软件工程相关的资源。
- 了解“学生档案管理系统”。

1.1　软件概述

1.1.1　软件的概念及特点

软件是计算机系统中不可或缺的一部分，它与硬件合为一体，从而完成特定的系统功能。人们对软件的认识也是在不断发展的。在计算机发展的初期，计算机的功能主要是由计算机的各个硬件部件通过有机地协调工作来完成的。当时所谓的软件就是程序，它的作用并没有得到人们足够的重视。

随着计算机技术的发展，人们越来越认识到高质量的软件会使计算机系统的功能和效率大大提高。于是，程序在计算机系统中的作用也日益重要。人们通常把各种不同功能的程序，包括系统程序、应用程序、用户自己编写的程序等称为软件。然而，当计算机的应用日益普及，软件日益复杂，规模日益增大，人们意识到软件并不仅仅等于程序。

程序是人们为了完成特定的功能而编制的一组指令集，它由计算机的语言描述，并且能在计算机系统上执行。而软件不仅包括程序，还包括程序的处理对象——数据，以及与程序开发、维护和使用有关的图文资料，即文档。

计算机系统由软件和硬件组成。当制造硬件时，人的创造性过程最终被转换成有形的形式。任何事物都有自己的特点，这是区别于其他事物的根本。理解事物的特点有利于人们更加深刻、更加准确地认识事物的本质。作为计算机系统的重要组成部分，计算机软件功能的发挥依赖于计算机硬件的支持，与硬件相比，它具有以下一些特点：

1）软件是一种逻辑实体，具有抽象性。硬件是有形的设备，而软件不像硬件那样具有明显的可见性。人们可以把软件记录在介质上，但是却无法直观地观察到它的形态，而必须通过在计算机上实际地运行才能了解它的功能、性能及其他特性。

2）软件的生产与硬件的制造不同。它更多地渗透了人类的智能活动，是人类可以劳

试读结束：需要全本请在线购买：　www.ertongbook.com

动的产物。软件是被开发或设计的，而不是传统意义上被制造的。软件成本集中于开发上，这意味着软件项目不能像制造项目那样管理。

3）软件在运行使用过程中不会磨损。在软件的运行和使用期间，它不会产生像硬件那样的磨损和老化现象，然而却存在着缺陷维护和技术更新的问题。软件不会磨损，但是它会退化，而软件的退化是由于修改造成的。因此，软件维护比硬件维护要复杂得多。图1-1和图1-2分别展示了硬件的失效率和使用时间的关系以及软件的失效率和使用时间的关系。

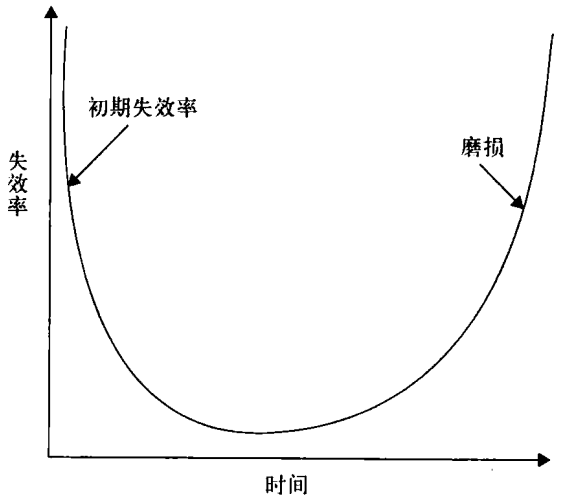


图1-1 硬件失效曲线图

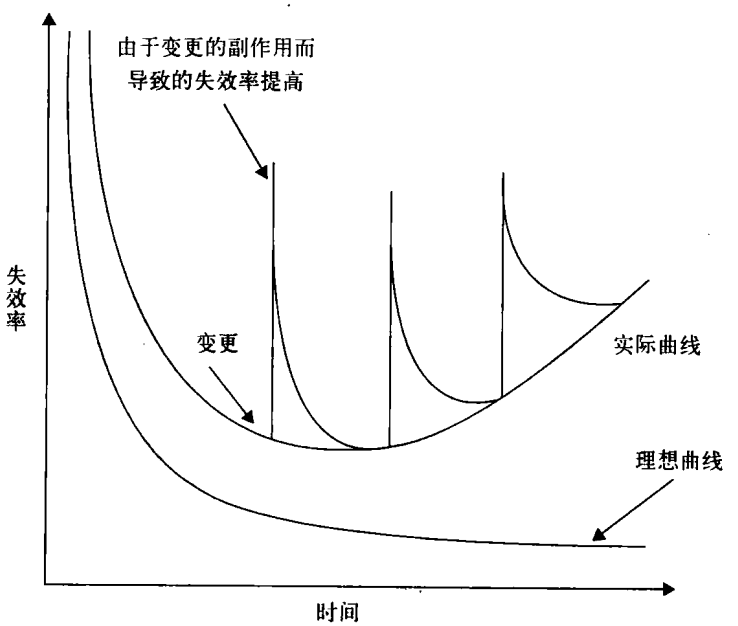


图1-2 软件失效曲线图

4）软件的开发至今尚未完全摆脱手工的开发方式。在硬件世界，构件复用是工程过程的自然的一部分；而在软件世界，构件复用则刚刚开始起步。虽然软件产业正在向基于

构件的组装发展，但大多数软件仍是定制的。

5）软件的开发和运行必须依附于特定的计算机系统环境。它不像有些设备那样能够独立地工作，而是受到了物理硬件、网络配置、支撑软件等因素的制约。由此引发了软件的可移植性问题。

1.1.2 软件的分类

随着计算机软件复杂性的增加，在某种程度上人们很难对软件给出一个通用的分类，但是可以按照不同的角度对软件进行分类。按照功能的不同，软件可以分为系统软件、支撑软件和应用软件三类。系统软件是居于计算机系统中最靠近硬件的一层，为其他程序提供最底层系统服务，它与具体的应用领域无关，如编译程序和操作系统等。支撑软件以系统软件为基础，以提高系统性能为主要目标，支撑应用软件的开发与运行，主要包括环境数据库、各种接口软件和工具组。应用软件是提供特定应用服务的软件，如字处理程序等。系统软件、支撑软件和应用软件之间既有分工又有合作，是不可截然分开的。

基于规模的不同，软件可以划分为微型、小型、中型、大型和超大型软件。一般情况下，微型软件只需要一名开发人员，在4周以内完成开发，并且代码量不超过500行。这类软件一般仅供个人专用，没有严格的分析、设计和测试资料。小型软件开发周期可以持续到半年，代码量一般控制在5000行以内。这类软件通常没有预留与其他软件的接口，但是需要遵循一定的标准，附有正规的文档资料。中型软件的开发人员控制在10人以内，要求在2年以内开发5000到50 000行代码。这种软件的开发不仅需要完整的计划、文档及审查，还需要开发人员之间、开发人员和用户之间的交流与合作。大型软件是10到100名开发人员在1到3年的时间内完成开发的，具有50 000到100 000行代码的软件产品。在这种规模的软件开发中，统一的标准、严格的审查制度及有效的项目管理都是必需的。超大型软件往往涉及上百名甚至上千名成员的开发团队，开发周期可以持续到3年以上，甚至5年。这种大规模的软件项目通常被划分为若干个小的子项目，由不同的团队开发。

根据软件服务对象的不同，软件还可以分为通用软件和定制软件。通用软件是由特定的软件开发机构开发，面向市场公开销售的独立运行的软件系统，如操作系统、文档处理系统和图片处理系统等。定制软件通常是面向特定的用户需求，由软件开发机构在合同的约束下开发的软件，如为企业定制的办公系统、交通管理系统和飞机导航系统等。

按照工作方式，计算机软件还可以划分为实时软件、分时软件、交互式软件和批处理软件。

软件的分类示意图如图1-3所示。

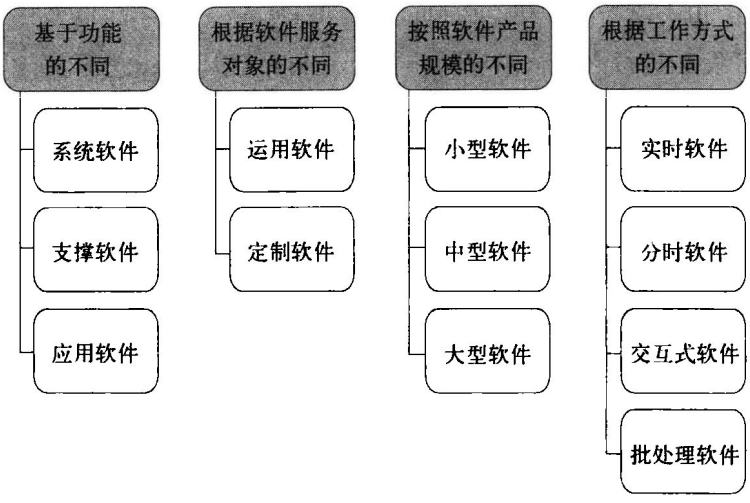


图1-3 软件的分类

1.2 软件危机

1.2.1 软件危机的表现与原因

软件危机是指人们在开发软件和维护软件过程中所遇到的一系列的问题。在20世纪60年代中期，随着软件规模的扩大、复杂性的增加、功能的增强，使得高质量的软件开发变得越来越困难。在软件开发的过程中，会经常出现一些不能按时完成任务、产品质量得不到保证、工作效率低下和开发经费严重超支等现象。这些情况逐渐使人们意识到软件危机的存在及其重要性。计算机软件的开发、维护和应用过程中普遍出现的一些严重的问题，主要表现为以下几个方面：

1）开发出来的软件产品不能满足用户的需求，即产品的功能或特性与用户需求不符。这主要是由于开发人员与用户之间不能充分有效地交流造成的，使得开发人员对用户需求的理解存在着差异。

2）相比越来越廉价的硬件，软件代价过高。

3）软件质量难以得到保证，且难以发挥硬件潜能。如果开发团队缺少完善的软件质量评审体系以及科学的软件测试规程，则最终的软件产品会存在诸多缺陷。

4）难以准确估计软件开发、维护的费用以及开发周期。软件产品往往不能在预算范围之内，按照计划完成开发。很多情况下，软件产品的开发周期或经费会大大超出预算。

5）难以控制开发风险，开发速度赶不上市场变化。

6）软件产品修改维护困难，集成遗留系统更困难。

7）软件文档不完备，并且存在着文档内容与软件产品不符的情况。软件文档是计算机软件的重要组成部分，它为在软件开发人员之间以及开发人员与用户之间信息的共享提供了重要的平台。软件文档的不完整和不一致的问题会给软件的开发和维护等工作带来很多麻烦。

这些问题严重影响了软件产业的发展，制约着计算机应用。OS 360经常被当做一个典

型的案例，来形象地描述软件危机。这是一个超大型的软件项目，约1000名程序员参与了开发。在经历了数十年的开发之后，极度复杂的软件项目甚至产生了一套不包括在原始设计方案之中的工作系统。Frederick P. Brooks是这个项目的管理者，他在自己的著作《人月神话》(The Mythical Man-Month）中曾经承认，自己犯了一个价值数百万美元的错误。

软件危机的出现和日益严重的趋势充分暴露了软件产业在早期的发展过程中存在的各种各样的问题。可以说，人们对软件产品认识的不足，以及对软件开发的内在规律理解的偏差是软件危机出现的本质原因。具体来说，软件危机出现的原因可以概括为以下几点：

1）忽视软件开发前期的需求分析。

2）开发过程缺乏统一的、规范化的方法论的指导。软件开发是一项复杂的工程，人们需要用科学的、工程化的思想来组织和指导软件开发的各个阶段。而这种工程学的视角正是很多软件开发人员所缺乏的，他们往往简单地认为软件开发就是程序设计。

3）文档资料不齐全或不准确。软件文档的重要性没有得到软件开发人员和用户的足够重视。软件文档是软件开发团队成员之间交流和沟通的重要平台，还是软件开发项目管理的重要工具。如果人们不能充分重视软件文档的价值，这样势必会给软件开发带来很多不便。

4）忽视与用户、开发组成员之间的交流。

5）忽视测试的重要性。

6）不重视维护或由于上述原因造成维护工作的困难。由于软件的抽象性和复杂性，使得软件在运行之前对开发过程的进展情况很难估计。再加上软件错误的隐蔽性和改正的复杂性，这些都给软件开发和维护增加了难度。

7）从事软件开发的专业人员对这个产业认识不充分，缺乏经验。软件产业相对于其他工业产业而言，是一个比较年轻、发展不成熟的产业，人们在对它的认识上缺乏深刻性。

8）没有完善的质量保证体系。完善的质量保证体系的建立需要有严格的评审制度，同时还需要有科学的软件测试技术及质量维护技术。软件的质量得不到保证，使得开发出来的软件产品往往不能满足人们的需求，同时人们还可能需要花费大量的时间、资金和精力去修复软件的缺陷，从而导致了软件质量的下降和开发预算超支等后果。

1.2.2 软件危机的启示

软件危机给我们的最大启示，是使我们更加深刻地认识到软件的特性以及软件产品开发的内在规律，具体包括以下几个方面：

1）软件产品是复杂的人造系统，具有复杂性、不可见性和易变性，难以处理。

2）开发小型软件时使用到的非常有效的编程技术和过程，在开发大型的复杂系统时难以发挥同样的作用。

3）从本质上讲，软件开发的创造性成分很大，发挥的余地也很大。它介于艺术与工程之间，并逐步向工程化发展，但又很难发展到完全的工程。

4）计算机和软件技术的快速发展，提高了用户对软件的期望，促进了软件产品的

演化，为软件产品提出了新的、更多的需求，难以在可接受的开发进度内保证软件的质量。

5）几乎所有的软件项目都是新的，而且是不断变化的。项目需求在开发过程中会发生变化，而且会出现很多其他问题，使得对设计和实现手段进行适当的调整是不可避免的。

6）“人月神话”现象——生产力与人数并不成正比。

为了解决软件危机，人们开始尝试着用工程化的思想去指导软件开发，于是软件工程诞生了。

1.3 软件工程

1.3.1 软件工程概念

1968年，在北大西洋公约组织举行的一次学术会议上，人们首次提出了软件工程这个概念。当时，该组织的科学委员们在开会讨论软件的可靠性与软件危机的问题时，提出了“软件工程”的概念，并将其定义为“为了经济地获得可靠的和能在实际机器上高效运行的软件，而建立和使用的健全的工程规则”。这个定义肯定了工程化的思想在软件工程中的重要性，但是并没有提到软件产品的特殊性。

经过40多年的发展，软件工程已经成为一门独立的学科，人们对软件工程也逐渐有了更全面更科学的认识。

IEEE对软件工程的定义为：1）将系统化、严格约束的、可量化的方法应用于软件的开发、运行和维护，即将工程化应用于软件。2）对1）中所述方法的研究。

具体来说，软件工程是以借鉴传统工程的原则、方法，以提高质量、降低成本为目的，指导计算机软件开发和维护的工程学科。这是一种层次化的技术，如图1-4所示。

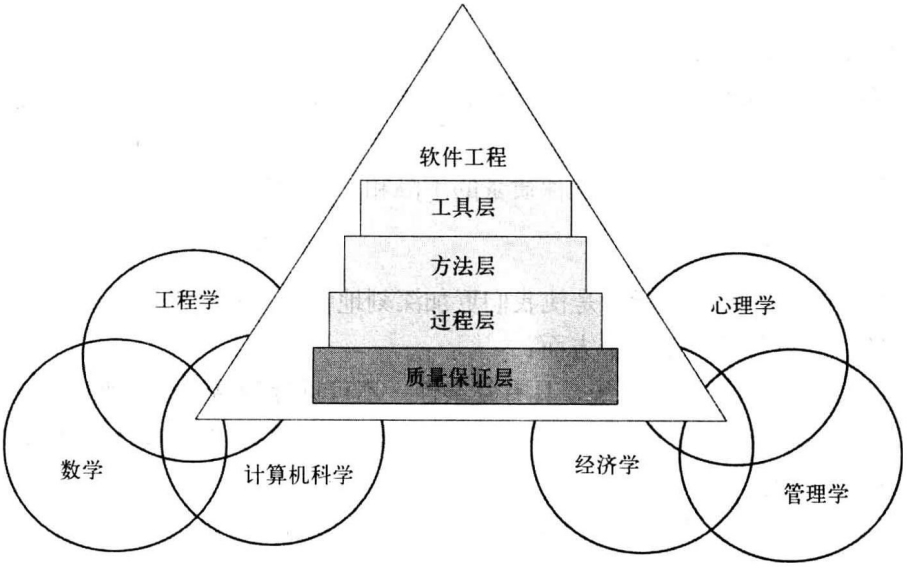


图1-4 软件工程层次图

软件工程的根基是质量保证层；软件工程的基础是过程层，它定义了一组关键过程区域的框架，使得软件能够被合理和及时地开发；方法层提供了建造软件在技术上需要“做什么”，它覆盖了一系列的任务，包括需求分析、设计、编程、测试和支持等；工具层对过程层和方法层提供了自动的或半自动的支持。而软件工程本身是一门交叉学科，涉及多种学科领域的相关知识，包括工程学、数学、计算机科学、经济学、管理学、心理学等。

1.3.2 软件工程发展

2006年，美国国家工程院院士Barry Boehm发表了一篇名为“20世纪和21世软件工程概览”的论文。Boehm博士曾对软件领域做出了杰出贡献，其中包括COCOMO模型、软件过程中的螺旋模型、适用于软件管理和需求决断的w理论等。在这篇论文中，他结合个人在软件工程领域几十年的切身体会，应用黑格尔的哲学思想，对20世纪50年代以来软件工程的发展历程进行了全面而深刻的回顾，并对软件工程未来20年的发展进行了设想。这篇论文一时间在国内外引起了巨大反响，有人说其在软件工程领域的影响可以和《人月神话》等论著媲美。本节就借鉴论文中的一些观点和思路，对软件工程的发展进行概述。

20世纪50年代，软件已经出现，但其作用和人们对其重视程度远远不如硬件，从事软件开发的人员，基本都是硬件工程师或者数学家，人们用硬件工程的理论和思想进行着软件的开发，当一个软件编写完成时，往往需要程序员对其反复进行人工检查和模拟运行，确认后，才送到硬件上去真正执行。

到了20世纪60年代，人们开始发现软件和硬件在许多方面都存在着不同。

首先，软件比硬件更容易得到模型，不需要为了产品的再生产提供昂贵的生产线，且易于修改。这种特点使得编程人员开始采用一种“编码和修改”的方式来开发软件，而不再像硬件工程那样为了防止生产线和产品生产出错而进行详尽的评估、设计和复查。

其次，软件不会消耗，它的维护和硬件的维护也存在着很大的区别；软件是不可见的，但是在它上面的花费却很多。这使得软件不能使用硬件工程中的模型和方法对其开发维护成本进行衡量和估算。

此外，随着软件需求的范围快速扩大，开发软件所需的知识远远超出了工程师和数学家的能力范围。一些大的项目为了开发软件，开始把人类学、社会科学、外语和艺术等也作为主修课，以培养和训练开发人员。导致许多没有工程经验的人由于企业、政府和服务业对于软件的需要，而涌入软件开发行业。这使得软件从业人员构成复杂，知识综合交错。

许多没有计算机科学专业的大学和情报部门开始重视软件，一些付费软件和软件公司也随之出现。软件开发的成功与否在很大程度上依赖于程序员的个人能力，因此出现了“黑客文化”和“牛仔编程员”，他们富有创造力，采用开发、自由和灵活的编程思想，更喜欢按照自己的想法去实现工程，而不是按照雇用他们的公司的想法；他们相信个人的编程能力，经常在最后的时间里用整晚的时间去满足客户的需要，完成代码的编写。

然而，随着软件项目的规模和难度逐渐增大，以个人能力为基础的软件开发的弊端逐渐体现，随之出现了著名的“软件危机”。在这种情况之下，北大西洋公约组织（North

试读结束：需要全本请在线购买： www.ertongbook.com