中国计算机软件专业技术水平考试指定用书

中国计算机软件专业技术资格和水平考试中心组织编写

软件工程（高级）

郑人杰 主编

清华大学出版社
http://www.tup.tsinghua.edu.cn

中国计算机软件专业技术水平考试指定用书

软 件 工 程

（高 级）

中国计算机软件专业技术资格和水平考试中心 组织编写

郑人杰 主编

清 华 大 学 出 版 社

(京)新登字 158 号

内 容 简 介

本书是中国计算机软件专业技术水平考试指定用书之一。本书是软件工程的高级读物，全书分为三个部分，即软件工程技术、软件质量管理与质量保证及软件工程管理。主要内容包括：软件生存期过程和软件工程，软件需求分析，软件复用技术，软件测试，软件维护与软件再工程，软件工具与软件开发环境，软件质量保证，软件工程标准化和软件文档，软件过程能力评估，软件工程项目管理，软件度量，软件配置管理，软件人员组织与管理，软件知识产权保护等。

本书是"中国计算机软件专业技术软件工程(高级)"水平考试的必读教材，也可作为相应培训班的教材，通过该级考试的考生具有软件工程高级工程师的相应水平。

版权所有，翻印必究。

本书封面贴有清华大学出版社激光防伪标签，无标签者不得销售。

书　名：软件工程(高级)
主　编：郑人杰
出版者：清华大学出版社（北京清华大学校内，邮编 100084）
http：//www.tup.tsinghua.edu.cn
印刷者：北京市清华园胶印厂
发行者：新华书店总店北京发行所
开　本：787×1092 1/16　**印张**：25.25　**字数**：596 千字
版　次：1999 年 8 月 第 1 版　1999 年 8 月 第 1 次印刷
书　号：ISBN 7-302-03534-2/TP・1937
印　数：00001～31000
定　价：37.00 元

序

科学技术的日新月异，信息化时代的来临，使以计算机技术为基础的信息科学在经济和社会生活各个领域得到了极为广泛的应用，其发展水平成为衡量国家经济与科技实力的重要标志已是不争之事实。各国都把培养大量高水平计算机专业人才作为21世纪经济和科技发展的重要战略目标之一。一些经济发达国家通过开展对计算机专业人才的教育培训，尤其是开展不同层次、不同规模的计算机水平测试，吸引、储备了大量计算机高级人才，为迎接日趋激烈的科技竞争奠定了坚实基础。这些国家的成功经验值得我们学习、借鉴。

中国计算机软件专业技术资格和水平考试自1991年开始实施至今已走过了8年的历程，共有近40万人参加考试，在国内外已产生较大影响。特别是1999年度中国计算机软件专业技术水平考试将由原来的一个专业扩展为程序设计、软件工程、数据库技术、计算机网络、多媒体技术5个专业，这无疑是中国计算机软件专业技术水平考试发展的一个质的飞跃，必将把中国计算机软件专业技术水平考试推向新的阶段。

我相信，新近编写出版的《中国计算机软件专业技术水平考试指定用书》能对广大应试者起到很好的指导作用。

我更加希望，在世纪之交，中国计算机软件专业技术水平考试能够抓住机遇，迎接竞争与挑战，为促进我国科教兴国战略的贯彻实施做出应有的贡献。

赵东宏

编者的话

软件工程是个年轻的学科，但它具有鲜明的实践性。近年来在计算机领域中，它的地位显得越来越重要，不仅大型软件项目离不开它，就是一般的，甚至小型软件项目也必须运用它的概念、原则和方法。软件分析人员、软件设计、编程、测试、维护人员，以及软件管理人员都必须很好地掌握软件工程知识，才能适应岗位工作的要求。特别是在当前，软件产业已经被中央、各地方政府和有关部门当作国民经济中新的增长点的形势下，软件工程更加受到重视。同时，事实也一再表明，不掌握软件工程知识，不按照软件工程的要求去做软件项目，提供给用户的产品只能是低质量的、难于维护的。

这套软件工程教材是为适应软件人员水平考试要求编写的。分为初级、中级和高级三册，分别面向软件的编程人员、以软件设计与开发为主的软件工程师及软件分析人员或软件管理人员。

本书是软件工程的高级读物，全书分为三个部分，即软件工程技术、软件质量管理与质量保证及软件工程管理。其中第3章“软件复用技术”由中科院软件所研究员卢慧琼编写，第6章“软件工具与软件开发环境”由上海复旦大学计算机系教授钱乐秋、任杰编写，第11章“软件过程能力评估”由中国航天工业总公司第二研究院计算机应用和仿真技术研究所研究员王纬编写，第15章15.2“软件人员能力成熟度模型”一节由华北电力大学信息系副教授徐磊编写，15.3“软件工程师道德和职业活动规范”一节由高等教育出版社副编审刘建元编写，第16章“软件知识产权保护”由中国软件登记中心主任李维编写。其余各章由郑人杰编写。

限于时间过于紧促，书中不当之处欢迎读者和同行批评指正，以利改进和提高。

郑人杰

1999年3月于清华园

此为试读，需要完整PDF请访问：www.ertongbook.com

目　录

第 1 篇　软件工程技术

第 2 篇　软件质量管理与质量保证

第 3 篇 软件工程管理

此为试读,需要完整PDF请访问: www.ertongbook.com

第1篇　软件工程技术

第1章　软件生存期过程和软件工程

在计算机软件的发展历史上曾经出现过若干个不同的时期，各个发展的历史时期反映了人们对软件的认识在逐步扩展。

在60年代以前甚至还没有软件开发的说法，那时只有程序设计(programming)的概念，最多在写出程序时配有程序结构说明和使用说明。

70年代以来人们认识到软件的工作不能只是编制程序，于是出现了软件生存期的概念。认为软件开发工作在程序编制之前和在其之后还有许多重要的工作不能忽视，例如需求分析、测试等等。与此同时，在总结"软件危机"的教训时，认识到必须建立软件工程的观念，摒弃了那种只有充满编程技巧的程序才能高水平地发挥个人才能的观念，强调程序的可读性、可理解性、可测试性和可修改性等工程原则。

90年代开始人们更加强调软件开发的效率和软件的质量以及与其相关的软件管理。软件开发固然是重要的、主要的软件工作，但一些支持性、管理性的工作也不可忽视，于是建立了软件过程的概念。软件工程过程或软件生存期过程使我们更全面、系统和深刻地理解了软件的有关工作。

本章将在软件生存期及其模型的基础上，围绕国际标准阐述软件生存期过程的有关内容。最后简要讨论软件工程的目标和软件工程的定义。

1.1　软件生存期及其模型

1.1.1　软件生存期(life cycle)

同任何事物一样，软件也有一个孕育、诞生、成长、成熟、衰亡的许多阶段，一般称其为计算机软件的生存期。根据这一思想，把上述基本的过程活动进一步展开，可以得到软件生存期的6个阶段工作，即制定计划、需求分析、设计、程序编制、测试及运行维护。以下对这6个阶段的任务作一概括的描述。

1. 制定计划 (planning)

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；由系统分析员和用户合作，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，

并对可利用的资源(计算机硬件、软件、人力等)、成本、可取得的效益、开发的进度做出估计,制定出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. 需求分析和定义 (requirement analysis and definition)

对待开发软件提出的需求进行分析并给出详细的定义。软件人员和用户共同讨论决定:哪些需求是可以满足的,并对其加以确切地描述。然后编写出软件需求说明书或系统功能说明书,及初步的系统用户手册,提交管理机构评审。

3. 软件设计 (software design)

设计是软件工程的技术核心。在设计阶段中,设计人员把已确定了的各项需求转换成一个相应的体系结构。结构中的每一组成部分都是意义明确的模块,每个模块都和某些需求相对应,即概要设计。进而对每个模块要完成的工作进行具体的描述,为源程序编写打下基础,即详细设计。所有设计中的考虑都应以设计说明书的形式加以描述,以供后继工作使用并提交评审。

4. 程序编写 (coding/programming)

把软件设计转换成计算机可以接受的程序代码,即写成以某一种特定程序设计语言表示的"源程序清单"。这一步工作也称为编码。自然,写出的程序应当是结构良好、清晰易读的,且与设计相一致的。

5. 软件测试 (testing)

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。首先是进行单元测试,查找各模块在功能和结构上存在的问题并加以纠正;其次是进行组装测试,将已测试过的模块按一定顺序组装起来;最后按规定的各项需求,逐项进行有效性测试,决定已开发的软件是否合格,能否交付用户使用。

6. 运行/维护 (running/maintenance)

已交付的软件投入正式使用,便进入运行阶段。这一阶段可能持续若干年甚至几十年。软件在运行中可能由于多方面的原因,需要对它进行修改。其原因可能有:运行中发现了软件中的错误需要修正;为了适应变化了的软件工作环境,需做适当变更;为了增强软件的功能需做变更。

1.1.2 软件生存期模型 (Software Life Cycle Model)

类似于其他工程项目中安排各道工序那样,为反映软件生存期内各种活动应如何组织,上节所给出的 6 个步骤应如何衔接,需要用软件生存期模型做出直观的图示表达。

软件生存期模型是从软件项目需求定义直至软件经使用后废弃为止,跨越整个生存期的系统开发、运行和维护所实施的全部过程、活动和任务的结构框架。

到现在为止,已经提出了多种软件生存期模型。例如,瀑布模型、演化模型、螺旋模型、喷泉模型和智能模型等。以下对这些模型作一简要描述。

1. 瀑布模型 (Waterfall Model)

瀑布模型规定了各项软件工程活动,包括:制定开发计划、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。并且规定了它们自上而下、相互衔接的固定次序,如同瀑布流水,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。实际情况是，每项开发活动均应具有以下特征：

(1) 从上一项活动接受该项活动的工作对象，作为输入；

(2) 利用这一输入实施该项活动应完成的内容；

(3) 给出该项活动的工作成果，作为输出传给下一项活动；

(4) 对该项活动实施的工作进行评审。若其工作得到确认，则继续进行下一项活动，在图 1.1 中用向下指的箭头表示；否则返回前项，甚至更前项的活动进行返工，在图 1.1 中由向上指的箭头表示。

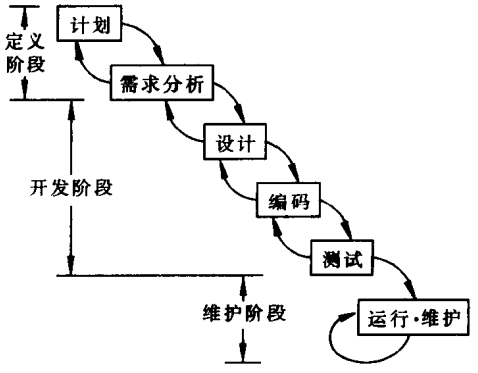

图 1.1 软件生存期的瀑布模型

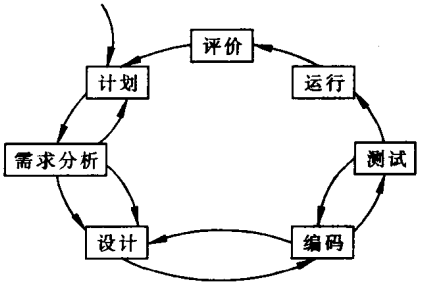

图 1.2 软件生存期循环

需要注意：软件维护在软件生存期中有它的特点。一方面，维护的具体要求是在软件投入运行以后提出来的，经过“评价”，确定变更的必要性，才进入维护工作。另一方面，维护中对软件的变更仍然要经历上述软件生存期在开发中已经历过的各项活动。如果把这些活动一并表达，就构成了生存期循环，如图 1.2 所示。事实上，有人把维护称为软件的二次开发，正是出于这种考虑。由于软件在投入使用以后可能经历多次变更，为把开发活动和维护活动区别开来，便有了 b 形的软件生存期表示，如图 1.3 所示。它与前述的软件生存期循环一样，都是软件生存期瀑布模型的变种。

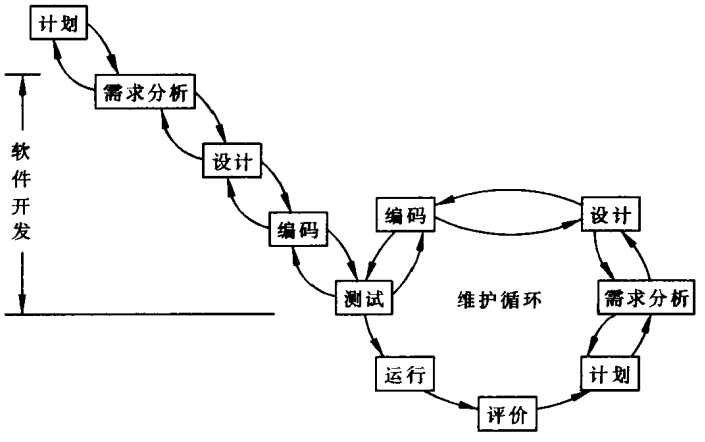

图 1.3 具有维护循环的软件生存期

瀑布模型为软件开发和软件维护提供了一种有效的管理图式。根据这一图式制定开发计划、进行成本预算、组织开发力量，以项目的阶段评审和文档控制为手段有效地对整个开发过程进行指导，从而保证了软件产品及时交付，并达到预期的质量要求。瀑布模型20多年来之所以广为流行，是因为它在消除非结构化软件、降低软件的复杂度、促进软件开发工程化方面起着显著作用。与此同时，瀑布模型在大量的软件开发实践中也逐渐暴露出它的严重缺点。其中最为突出的缺点是该模型缺乏灵活性，特别是无法解决软件需求不明确或不准确的问题。这些问题的存在对软件开发会带来严重影响，最终可能导致开发出的软件并不是用户真正需要的软件，并且这一点在开发过程完成后才有所察觉。面对这些情况，无疑需要进行返工或是不得不在维护中纠正需求的偏差。但无论上述哪一种情况都必须付出高额的代价，并将为软件开发带来不必要的损失。另一方面，随着软件开发项目规模的日益庞大，由于瀑布模型不够灵活等缺点引发出的上述问题显得更为严重。

为弥补瀑布模型的不足，近年来已经提出了多种其他模型。

2. 演化模型 (Evolutional Model)

由于在项目开发的初始阶段人们对软件的需求认识常常不够清晰，因而使得开发项目难于做到一次开发成功，出现返工再开发在所难免。有人说，往往要“干两次”后开发出的软件才能较好地令用户满意。第一次只是试验开发，其目标只是在于探索可行性，弄清软件需求；第二次则在此基础上获得较为满意的软件产品。通常把第一次得到的试验性产品称为“原型”。显然，演化模型在克服瀑布模型缺点、减少由于软件需求不明确而给开发工作带来风险方面，确有显著的效果。

3. 螺旋模型 (Spiral Model)

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，并且加入被两种模型都忽略了的风险分析，弥补了两者的不足。

在此先简要说明什么是风险分析。“软件风险”是普遍存在于任何软件开发项目中的实际问题。对于不同的项目，其差别只是风险有大有小而已。在制定软件开发计划时，系统分析员必须回答：项目的需求是什么，需要投入多少资源以及如何安排开发进度等一系列问题。然而，若要他们当即给出准确无误的回答是不容易的，甚至是不可能的。但系统分析员又不可能完全回避这一问题。凭借经验的估计给出初步的设想便难免带来一定风险。实践表明，项目规模越大，问题越复杂，资源、成本、进度等因素的不确定性越大，承担项目所冒的风险也越大。总之，风险是软件开发不可忽视的潜在不利因素，它可能在不同程度上损害到软件开发过程或软件产品的质量。软件风险驾驭的目标是在造成危害之前，及时对风险进行识别、分析，采取对策，进而消除或减少风险的损害。

螺旋模型沿着螺线旋转，如图 1.4 所示，在笛卡尔坐标的 4 个象限上分别表达了 4 个方面的活动，即：

(1) 制定计划：确定软件目标，选定实施方案，弄清项目开发的限制条件；

(2) 风险分析：分析所选方案，考虑如何识别和消除风险；

(3) 实施工程：实施软件开发；

(4) 客户评估：评价开发工作，提出修正建议。

沿螺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。例如，在第一

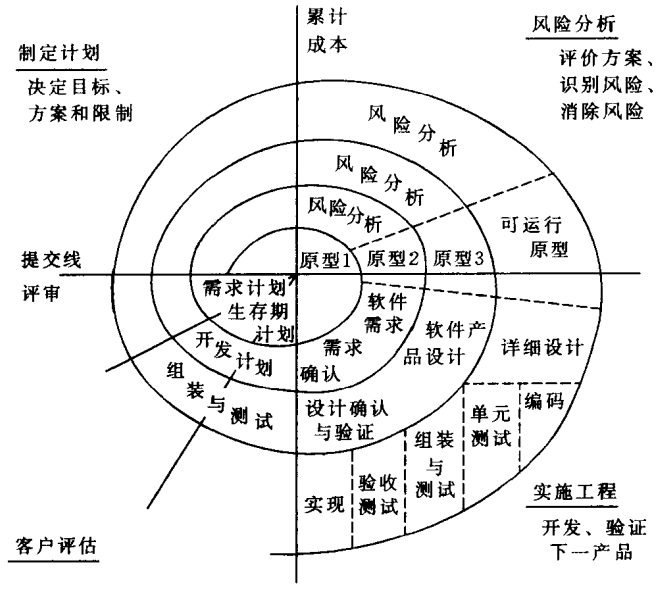

图 1.4　螺旋模型

圈，确定了初步的目标、方案和限制条件以后，转入右上象限，对风险进行识别和分析。如果风险分析表明，需求有不确定性，那么在右下的工程象限内，所建的原型会帮助开发人员和客户，考虑其他开发模型，并对需求做进一步修正。

客户对工程成果做出评价之后，给出修正建议。在此基础上需再次计划，并进行风险分析。在每一圈螺线上，做出风险分析的终点是是否继续下去的判断。假如风险过大，开发者和用户无法承受，项目有可能终止。多数情况下沿螺线的活动会继续下去，自内向外，逐步延伸，最终得到所期望的系统。图 1.5 给出了螺旋模型的另一图式。

如果软件开发人员对所开发项目的需求已有了较好的理解或较大的把握，则无需开发原型，可采用普通的瀑布模型。这在螺旋模型中可认为是单圈螺线。与此相反，如果对所开发项目的需求理解较差，则需要开发原型，甚至需要不止一个原型的帮助，那就需要经历多圈螺线。在这种情况下，外圈的开发包含了更多的活动。也可能某些开发采用了不同的模型。

螺旋模型适合于大型软件的开发，应该说它是最为实际的方法，它吸收了软件工程“演化”的概念，使得开发人员和客户对每个演化层出现的风险有所了解，继而做出应有的反应。

螺旋模型的优越性比起其他模型来说是明显的，但并不是绝对的。要求许多客户接受和相信演化方法并不容易。这个模型的使用需要具有相当丰富的风险评估经验和专门知识。如果项目风险较大，又未能及时发现，势必造成重大损失。此外，螺旋模型是出现较晚的新模型，远不如瀑布模型普及，要让广大软件人员和用户充分肯定它，还有待于更多的实践。

4. 喷泉模型（Water Fountain Model）

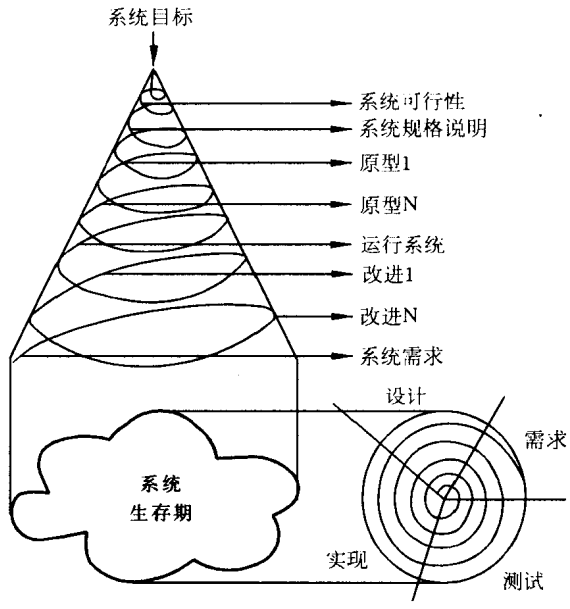

图 1.5 螺旋模型的另一图式

瀑布模型的另一个不足之处在于，它对软件复用和生存期中多项开发活动的集成并未提供支持，因而难于支持面向对象的开发方法。“喷泉”一词体现了迭代和无间隙特性。喷泉模型在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。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。

5. 智能模型（Intelligence Model）

智能模型也称为基于知识的软件开发模型，它综合了上述若干模型，并把专家系统结合在一起。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。为此，建立了知识库，将模型、软件工程知识与特定领域的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含有应用领域知识规则的其他专家系统相结合，构成了这一应用领域软件的开发系统。

1.2 软件生存期过程

软件生存期概念的出现曾经帮助我们较为全面地认识软件开发（包括软件的运行与维护）。这主要表现在 1988 年制订和公布的国家标准《GB 8566-88 计算机软件开发规范》中，该标准曾将软件生存期划分为 8 个阶段，即：可行性研究和计划、需求分析、概要设计、详细设计、实现、组装测试、确认测试、使用和维护。

该标准为每个阶段规定了任务、实施步骤、实施要求以及完成的标志。对软件生存期按此方式做 8 个阶段的划分大致符合也适应前述的瀑布模型，反映了 80 年代人们对软件工程的认识。

此后，于 20 世纪 90 年代初提出了软件工程过程的概念。认为软件工程过程是为了获

此为试读，需要完整PDF请访问：www.ertongbook.com