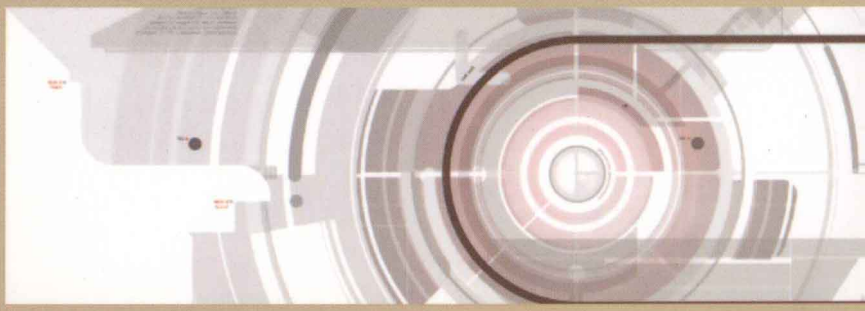
总装备部“1153”人才工程专项经费资助项目



软件工程原理及应用

韦 群 □ 编著

国防工业出版社
National Defense Industry Press

总装备部“1153[redacted]助项目

软件工程原理及应用

韦群　编著

国防工业出版社

·北京·

内容简介

本书在对软件工程基本概念进行介绍的基础上，全面系统地介绍了软件开发的基本原理、基本方法及相关技术。以传统的软件工程和面向对象的软件工程为主线，根据软件开发“工程化”思想，重点介绍了结构化开发方法和面向对象开发方法，强调了软件体系结构在软件开发中的作用，通过对软件测试及软件管理技术等内容的介绍，确保软件开发质量。针对软件生命周期的主要阶段，结合具体案例，给出了基本原理和技术的应用实例。教材内容新颖、全面，对软件开发具有指导性作用。

本书适合高等院校计算机科学与技术专业本科或研究生、信息专业各类继续教育人员阅读，也可作为从事软件开发的科技人员的参考书、培训教材等。

图书在版编目(CIP)数据

软件工程原理及应用/韦群编著. —北京：国防工业出版社，2012.8

ISBN 978-7-118-08194-7

Ⅰ.①软... Ⅱ.①韦... Ⅲ.①软件工程－教材 Ⅳ.①TP311.5

中国版本图书馆 CIP 数据核字(2012)第 171907 号

※

国防工业出版社出版发行

（北京市海淀区紫竹院南路 23 号 邮政编码 100048）

北京嘉恒彩色印刷有限责任公司

新华书店经售

*

开本 710×960 1/16 印张 20¾ 字数 368 千字

2012 年 8 月第 1 版第 1 次印刷 印数 1—3000 册 定价 58.00 元

（本书如有印装错误，我社负责调换）

国防书店：(010)88540777 发行邮购：(010)88540776

发行传真：(010)88540755 发行业务：(010)88540717

前　言

软件工程是一门迅速发展的新兴学科，现已成为计算机科学的一个重要分支，软件工程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。

工程是将理论和知识应用于实践的科学。软件工程是软件生产和软件管理的工程科学，它借鉴了传统工程的原则和方法，以求高效地开发高质量的软件。其中计算机科学和数学用于构造模型与算法，工程科学用于制定规范、设计范型、评估成本及确定权衡，管理科学用于计划、资源、质量和成本的管理。

本书系统地介绍了软件工程的有关概念、原理、方法、技术和相关管理技术。全书共8章，以软件生存周期为主线，对软件工程有关的分析、设计、验证、维护和管理等内容做了详尽阐述，突出结构化技术、面向对象技术和构件技术在软件开发过程中的运用，强调软件体系结构在软件产品开发中的主要作用，通过对软件测试及软件管理技术等内容的介绍，确保软件开发质量。最后，通过一个综合实例，全面实践了软件工程的理论和方法。全书从方法学角度出发，内容紧凑，阐述力求理论联系实际，并通过与实例相结合，深入浅出，循序渐进。

本书主要用做计算机科学与技术专业本科或研究生“软件工程”课程教材，亦可作为高等院校计算机科学与技术专业或信息类相关专业的教学参考书，或作为从事软件开发的科技人员的参考书、培训教材等。

作　者

2012年2月

目　录

第一章　软件工程概述

伴随着电子商务、网络经济的发展，人类迈进了一个崭新的信息时代，计算机技术在各行各业的应用达到了前所未有的广度和深度，一种以计算机软件为主要产品的新兴工业——软件工业浮出水面并逐渐走向成熟。正如当初机器化大生产最终代替了手工作坊，软件工程化的浪潮正席卷全球，软件企业也将生产的重点从盲目追求数量和拼命缩短开发周期转移到提高产品质量及规范生产能力上。在机遇与挑战并存的今天，软件工程成了软件企业克敌制胜的法宝。

软件工程作为一门工程学科，是现代软件产业发展的指导思想和管理原则。软件工程的基本理念是“按工程的概念、原理、技术和方法开发与维护计算机软件”。在软件工程建立之前，软件开发过程主要是程序员个性化心理过程的组合，充满神秘性和不确定性。软件工程作为一门工程学科，为软件开发过程提供了基本的工程化、结构化框架，即在结构化细分软件开发过程的基础上，通过开发技术与管理技术的结合，对每一个开发阶段的工作进行控制和测评，从而使软件开发过程在整体上是可计划、可预期和可管理的。

1.1　软件及其发展

人们对软件的认识经历了一个由浅到深的过程。20 世纪 40 年代，出现了世界上第一台计算机之后，就有了程序的概念。可以认为它是软件的前身，经历了几十年的发展，人们对软件有了更为深刻的认识。

软件的发展经历了三个不同的阶段。20 世纪 50—60 年代为程序设计阶段，主要特征是硬件通用化，个性化的机器语言程序；20 世纪 60 年代中期至 70 年代中期为程序系统阶段，主要特征是硬件的速度、容量、价格及性能的优化与程序开发能力的增长不能匹配，阻碍了计算机系统整体水平的提高，出现软件危机；20 世纪 70 年代以后，为软件工程阶段，主要特征是硬件技术以摩尔定律飞速发展，硬件技术的生产进入高速、全球化、现代化生产方式，相比之下，软件技术仍然处在手工作坊方式，软件危机依然严重。

现在，被普遍接受的软件的定义是：软件是计算机系统中与硬件相互依存的

另一部分，它是包括程序、数据及其相关文档的完整集合。其中，程序是按事先设计的功能和性能要求执行的指令序列；数据是使程序能正常操纵信息的数据结构；文档是与程序开发、维护和使用有关的图文材料。

软件同传统的工业产品相比，有其独特的特性，主要表现在以下 8 个方面：

（1）软件是一种逻辑产品，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。人们通常看到的是它的载体，如记录它的纸张、内存和磁盘、光盘等，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，才能了解它的功能、性能等特性。

（2）与传统的工业产品的生产不同，软件没有明显的制造过程。传统的工业产品在车间里生产，生产过程可见、可触摸，也容易衡量生产过程中的消耗和进展。可是软件的开发过程是在人的大脑里通过智力活动，把知识和技术转化成信息的一种产品，很难度量其进度。

工业产品的生产可以重复制造，在制造过程中进行质量控制。软件产品一旦研制开发成功，就可以大量复制同一内容的副本。所以对软件而言，必须在软件开发过程中进行质量控制。

（3）软件在使用过程中，没有磨损、老化的问题。硬件存在机械磨损，老化问题。硬件的故障率曲线如图 1－1 所示。

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。这些修改又不可避免地会引入错误，导致软件失效率升高，从而使得软件退化。当修改的成本高得难以接受时，软件就被抛弃。软件的实际故障率曲线如图 1－2 所示。

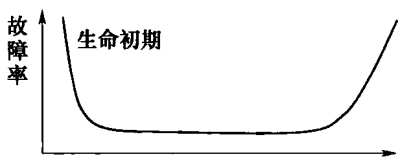


图 1－1　硬件的故障率曲线

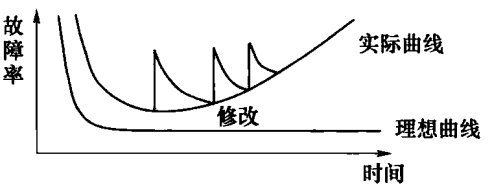


图 1－2　软件的实际故障率曲线

（4）软件对硬件和环境有着不同程度的依赖性。软件的开发和运行常受到计算机系统的限制，这导致了软件移植的问题，这是衡量软件质量的因素之一。

（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。软件产业是一个新型的特殊产业，它与许多传统的产业相比具有一些不同的特点，传统的产业常依赖于特定的原料资源，而软件产业最主要的资源是人，软件

产业本质上又是一个劳力密集型产业。产业的规模常与人员规模联系在一起，这也从某种意义上限制了软件产业的高效益产出。我们不能单靠拼人海战术去拼效益，这将是非常局限和困难的。但我们也要看到软件产业又是一个知识型产业或智力密集型产业。

目前还不能像设计和建造楼房、汽车那样开发软件。软件开发远没有建筑工程、机械工程那样成熟和真正的工程化。尽管对软件复用技术、自动生成技术、软件开发工具或软件开发环境等软件技术新的开发方法进行了大量的研究，但采用的比率低。

（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。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门知识，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，还设计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，直接影响到项目的成败。

1.2　软 件 危 机

软件和硬件是计算机技术的两个相互联系而又密不可分的部分，然而相对硬件来说，软件却很不成熟，也很脆弱，由软件导致的灾难屡见不鲜。1996 年欧洲航天局首次发射“阿丽亚纳”5 号火箭失败，直接损失 5 亿美元，还使耗资已达 80 亿美元的开发计划推迟了近三年，事故的原因是火箭控制系统的软件故障。20 世纪 90 年代后半期，“千年虫”问题震惊世界，各国投入了大量的人力和物力，耗资数千亿美元，“虫害”才基本上得到控制。F－18 战斗机在海湾战争中，飞行控制软件共发生了 500 多次故障，“爱国者”导弹因软件问题误伤了 28 名美国士兵。

随着计算机应用日益普及和深化，计算机软件的数量以惊人的速度急剧膨胀，软件的规模之大，成本之高也超出了最初人们的想象。但是，软件生产的质量没有可靠的保证，软件开发的生产率也远远跟不上普及计算机应用的要求，软件已经成为制约计算机应用发展的“瓶颈”。手工作坊式的开发方式已经严重阻碍了计算机软件的发展，更为严重的是，用错误方法开发出来的许多大型软件几乎根本无法维护，只好提前报废，造成大量人力、物力的浪费。

如果将软件的成功开发定义为“规定的需求大部分得到满足，预算基本符

合实际,基本按计划进行",那么能够称得上是成功的软件项目仅占所有开发项目的26%。项目组成员经常会听到用户的抱怨,与此同时,项目组也是怨声载道。

软件危机包括两个方面问题,一方面是如何开发软件,即研究软件开发方法,以满足对软件的日益增长需求;另一方面是如何维护数量不断增长的已有软件。直到今天,软件危机仍然存在。具体表现在以下方面:

(1) 软件增长与硬件增长失衡,软件的增长能力远远跟不上硬件及网络技术的发展速度和水平,也远远跟不上计算机应用领域深度与广度的发展,成为制约IT产业发展的瓶颈。

(2) 软件项目的开发成本与进度计划常常形同虚设,软件开发中遇到的各种情况,令软件开发过程难以保证按预定的计划实现,为了追赶进度或降低成本所作的快速努力又往往损害了软件的质量或性能,不得不返工或引起用户的不满。

(3) 软件产品质量差。软件产品质量保证技术(审查、复审、测试)不能贯穿于开发全过程,软件产品的质量不但取决于开发人员的技术水平,还取决于开发人员的意志品质和团队精神,软件开发整体上仍然深受非智力的个性心理特征的制约。软件质量问题与其他商品的质量问题有着很多的不同。首先,软件开发仍为早期的个体化方式,其最大的特点是开发过程没有交互性,软件的规划、设计、测试和维护都只能由某一个人负责,对软件质量最具发言权的用户无法也无力参与到软件的质量管理当中;其次,很多软件设计带有太多的随意性,许多功能都是在灵机一动时添加进软件当中的。这是造成软件成本提高和不能令人满意的重要因素。

很多投入了巨资和人力的软件产品不能取得好的成绩,这在软件产业中是一个很普遍的现象,如何控制和管理软件产品的质量,是整个软件行业一开始就面临的问题。

(4) 软件的可维护性、可移植性、可适应性差。软件是逻辑元件,不是一种实物。软件故障是由软件中的逻辑故障所造成的,不是硬件的"用旧"、"磨损"问题,软件维护不是更换某种设备,而是要纠正逻辑缺陷。

软件维护有三类工作:改正性维护、适应性维护和提高性维护。改正性维护约占20%,改正处理上的错误,性能方面、编制程序方面的错误。适应性维护约占25%,适应数据环境、硬件及操作系统,也包括移植工作。提高性维护约占50%,提高处理效率、性能或使用方便,增加及改善输出信息,以便于维护。

软件重复使用还是一个追求中的目标,人们不得不重复开发先前被开发过的软件。

（5）软件的文档管理无法满足项目管理人员、开发人员、测试人员、维护人员、用户的多方面的需要，文档管理不当造成开发过程沟通不充分，增加了软件的成本，降低了软件的品质和可维护性。

（6）软件在整个信息系统总成本中所占的比例不断上升。计算机发展的早期，大型的计算机系统主要用于军事领域，研制费用主要由国家财政提供，很少考虑研制代价问题。

随着计算机市场化产业的发展，代价和成本成为投资者考虑的最主要的问题之一。技术的进步，使得计算机硬件的成本持续降低，而软件成本不断增长，软件成本在计算机系统总成本中所占比例呈现日益扩大的趋势。美国在1985年软件成本已经占系统总成本的90%以上。

软件危机的原因，一方面与软件本身的特点有关；另一方面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、规范的方法论的指导，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上的原因。

面对软件开发和维护过程中遇到的一系列严重问题，1968年，北大西洋公约组织的计算机科学家召开国际会议，第一次提出软件危机的概念，并且在60年代后期开始认真研究解决软件危机的方法，从而逐步形成了计算机科学技术领域中的一门新兴的学科——计算机软件工程学，通常简称为软件工程。

1.3　软 件 工 程

我们已经进入到了信息化时代，作为信息化世界基础的软件技术仍然发展得很不成熟，软件危机并没有成为过去。计算机技术发展到今天，软件仍然是最落后、最脆弱的领域。

最初，编制程序完全是一种技巧，主要依赖于程序员的素质。软件技术落后的原因来自两个方面：一是软件本身的复杂性，有人认为，软件是迄今为止人类设计的最复杂的东西；二是软件开发过程基本上是一种手工劳动，由于研制一个软件系统，特别是大型复杂的软件系统，同研制一台机器、一座楼房有许多共同之处。因此，可以参考机械工程、建筑工程一样来处理软件研制的全过程。

针对20世纪60年代出现的软件危机，北大西洋公约组织于1968年专门召开了一次学术会议，首次提出了“软件工程（Software Engineering）”这一概念并进行了讨论，这在软件技术发展史上是一件划时代的大事。自这一概念提出以来，围绕软件项目，开展了有关开发模型、方法以及支持工具的研究，各种有关软

件的技术、思想、方法和概念不断被提出，软件工程逐渐发展成一个独立的学科，称为“软件工程方法学”或者“软件工程学”。

软件工程的定义有多种，如：

定义1：软件工程是指导计算机软件开发和维护的工程学科。采用工程的概念、原理、技术和方法来开发和维护软件，把经过时间考验而证明正确的管理技术和当前能够得到的最好的技术方法结合起来，这就是软件工程。

定义2：软件工程是研究和应用如何以系统性的、规范性的、可定量的方法去开发、操作和维护软件，即把工程应用到软件上。

定义3：将系统化的、规范的、可度量的方法应用于软件的开发、运行和维护的过程，即将工程化应用于软件中；对上述系统化的、规范的、可度量的方法的研究。

软件工程包括两方面内容：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。软件工程知识体系如图1-3所示。

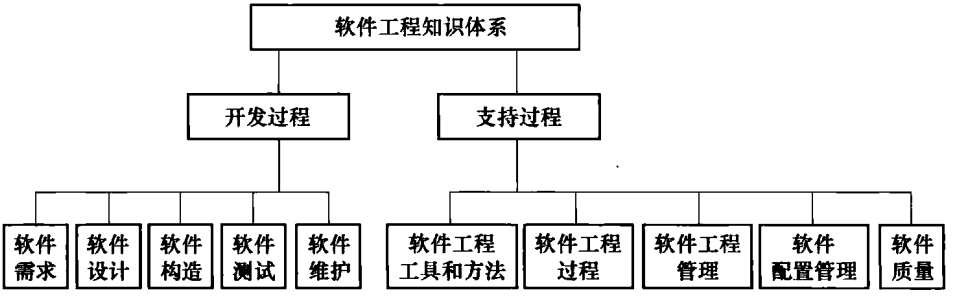


图1-3　软件工程知识体系

软件工程知识体系包括开发过程和支持过程。其中开发过程主要有软件需求、设计、构造、测试和维护5个阶段；软件支持过程有软件工程管理、软件工程过程、软件工程工具和方法、软件配置管理以及软件质量等过程。

软件工程包括三个要素：方法、工具和过程，支持软件工程的根基就在于对质量的关注。软件工程方法为软件开发提供了“如何做”的技术。它包括了多方面的任务，如项目计划与估算、软件系统需求分析、数据结构、系统总体结构的设计、算法的设计、编码、测试以及维护等。软件工具为软件工程方法提供了自动的或半自动的软件支撑环境。目前，已经开发出了许多软件工具，已经能够支持上述的软件工程方法。软件工程的过程则是将软件工程的方法和工具综合起来以达到合理、及时地进行计算机软件开发的目的过程，定义了方法使用的顺

序、要求交付的文档资料、为保证质量和协调变化所需要的管理及软件开发各个阶段完成的里程碑。

自从1968年提出软件工程这一术语以来，研究软件工程的专家学者们陆续提出了100多条关于软件工程的准则或信条。美国著名的软件工程专家Boehm综合这些专家的意见，并总结了TRW公司多年的开发软件的经验，于1983年提出了软件工程的七条基本原理。这7条的基本原理是：

（1）用分阶段的生命周期计划严格管理。这一条是吸取前人的教训而提出来的。统计表明，50%以上的失败项目是由于计划不周而造成的。在软件开发与维护的漫长生命周期中，需要完成许多性质各异的工作。这条原理意味着，应该把软件生命周期分成若干阶段，并相应制定出切实可行的计划，然后严格按照计划对软件的开发和维护进行管理。Boehm认为，在整个软件生命周期中应指定并严格执行6类计划：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、运行维护计划。

（2）坚持进行阶段评审。统计结果显示：大部分错误是在编码之前造成的，大约占63%；错误发现得越晚，改正它要付出的代价就越大，要差2到3个数量级。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。开发人员最痛恨的事情之一就是改动需求。但是实践告诉我们，需求的改动往往是不可避免的。这就要求我们要采用科学的产品控制技术来顺应这种要求。也就是要采用变动控制，又叫基准配置管理。当需求变动时，其他各个阶段的文档或代码随之相应变动，以保证软件的一致性。

（4）采纳现代程序设计技术。从六七十年代的结构化软件开发技术，到最近的面向对象技术，从第一、第二代语言，到第四代语言，人们已经充分认识到采用先进的技术既可以提高软件开发的效率，又可以减少软件维护的成本。

（5）结果应能清楚地审查。软件是一种看不见、摸不着的逻辑产品。软件开发小组的工作进展情况可见性差，难于评价和管理。为更好地进行管理，应根据软件开发的总目标及完成期限，尽量明确地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。这一条基于两点原因：高素质开发人员的效率比低素质开发人员的效率要高几倍到几十倍，开发工作中犯的错误也要少的多；当开发小组为N人时，可能的通信信道为$N(N-1)/2$，可见随着人数N的增大，通信开销将急剧增大。

(7) 承认不断改进软件工程实践的必要性。遵从上述7条基本原理,就能够较好地实现软件的工程化生产。但是,它们只是对现有的经验的总结和归纳,并不能保证赶上技术不断前进发展的步伐。因此,Boehm 提出应把承认不断改进软件工程实践的必要性作为软件工程的第七条原理。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,也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。

Boehm 认为,这7条原理是确保软件产品质量和开发效率的原理的最小集合。它们是相互独立的,是缺一不可的最小集合;同时,它们又是相当完备的。人们当然不能用数学方法严格证明它们是一个完备的集合,但是可以证明,在此之前已经提出的100多条软件工程准则都可以由这7条原理的任意组合蕴含或派生。

1.4 软件过程

1.4.1 软件生存周期

在软件的开发和使用过程中,通常软件的维护费用要远远高出软件的开发费用,因而软件开发不能只考虑开发期间的费用,而应考虑软件生存周期的全部费用。因此,软件生存周期的概念就变得特别重要。考虑软件费用时,不仅要降低开发成本,更要降低整个软件生存周期的总成本。

(1) 软件生存周期定义。任何事物都要经历产生、发展、成熟、消亡的过程,人们一般把这样一个过程称为生存周期,软件也有其自身的生存周期。软件生存周期是软件产品或系统一系列相关活动的全周期。从形成概念开始,经过研制,交付使用,在使用中不断增补修订,直到最后被淘汰,让位于新的软件产品的全过程,即软件生存周期是指软件产品从考虑其概念开始,到该软件产品不再能使用为止的整个时期。一般包括概念阶段、需求阶段、设计阶段、实现阶段、测试阶段、安装阶段以及交付使用阶段、运行阶段和维护阶段,有时还有退役阶段,这些阶段可以有重叠。

(2) 软件开发生存周期。软件开发生存周期是指软件产品从考虑其概念开始到该软件产品交付使用为止的整个时期。一般包括概念阶段、需求阶段、设计阶段、实现阶段、测试阶段、安装阶段以及交付阶段。

(3) 软件开发过程。把用户的要求转变成软件产品的过程叫做软件开发过