



软件

工程实践与项目管理研究

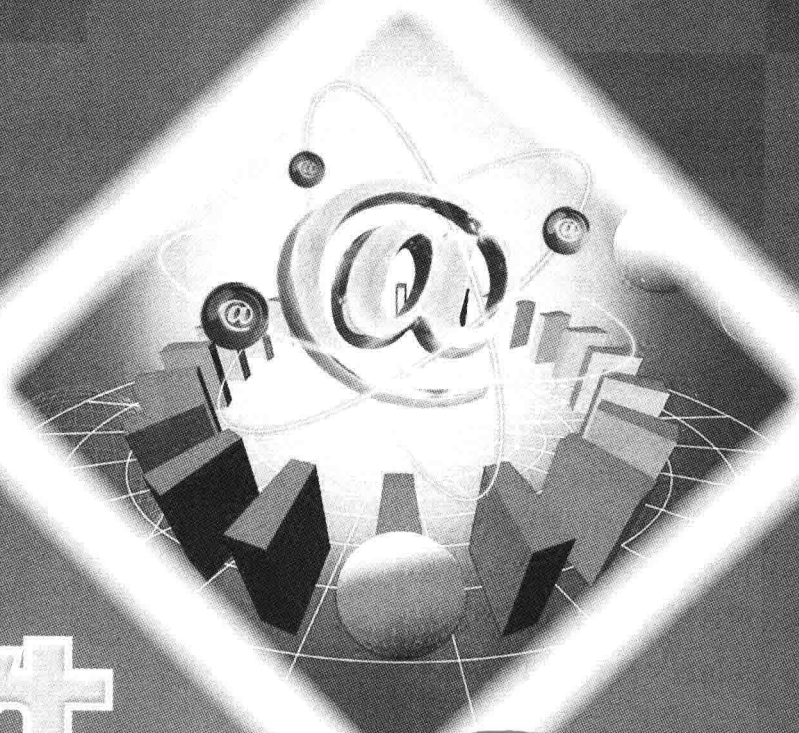
RUANJIAN

GONGCHENG SHIJIAN YU XIANGMU GUANLI YANJIU

主 编 罗自强 张 妍 王 坤
副主编 范卫锋 杨秀荣 王文霞



中国水利水电出版社
www.waterpub.com.cn



软件



工程实践与项目管理研究

RUANJIAN
GONGCHENG SHIJIAN

GUANLI YANJIU

主 编 罗自强 张 妍 王 坤
副主编 范卫锋 杨秀荣 王文霞



中国水利水电出版社
www.waterpub.com.cn

内 容 提 要

软件工程是一门实践性很强的学科,本书重点对软件工程实践与项目管理的内容进行研究分析,首先阐述了可行性研究与需求分析管理、软件的概要设计和详细设计、软件编码与实现、软件测试与维护等方面的内容,并描述了目前较为流行的面向对象方法学,介绍了软件工程标准与开发环境,另外还对项目管理的三要素——质量、进度和成本作了说明,随后讨论了人力资源管理与团队建设,最后是软件项目的收尾与验收管理的内容。

本书可以作为软件开发工程师、项目经理、软件测试工程师或计算机相关专业技术人员的参考用书。

图书在版编目(CIP)数据

软件工程实践与项目管理研究/罗自强,张妍,王

坤主编. —北京:中国水利水电出版社,2013. 10

ISBN 978-7-5170-1362-4

I. ①软… II. ①罗… ②张… ③王… III. ①软件工程—项目管理 IV. ①TP311.5

中国版本图书馆 CIP 数据核字(2013)第 261486 号

策划编辑:杨庆川 责任编辑:杨元泓 封面设计:马静静

书 名	软件工程实践与项目管理研究
作 者	主 编 罗自强 张 妍 王 坤 副主编 范卫锋 杨秀荣 王文霞
出版发行	中国水利水电出版社 (北京市海淀区玉渊潭南路 1 号 D 座 100038) 网址:www. waterpub. com. cn E-mail:mchannel@263. net(万水) sales@waterpub. com. cn 电话:(010)68367658(发行部)、82562819(万水)
经 售	北京科水图书销售中心(零售) 电话:(010)88383994、63202643、68545874 全国各地新华书店和相关出版物销售网点
排 版	北京鑫海胜蓝数码科技有限公司
印 刷	三河市天润建兴印务有限公司
规 格	184mm×260mm 16 开本 25.5 印张 652 千字
版 次	2013 年 10 月第 1 版 2013 年 10 月第 1 次印刷
印 数	0001—3000 册
定 价	86.00 元

凡购买我社图书,如有缺页、倒页、脱页的,本社发行部负责调换

版权所有·侵权必究

前 言

软件产业是增长最快的朝阳产业,是高投入、高产出、无污染、低能耗的绿色产业。软件产业关系到国家经济发展和文化安全,体现了国家的综合实力,是决定 21 世纪国际竞争地位的战略产业。随着 Internet 的日益普及和广泛应用,全球软件产业呈现出网络化、服务化和全球化的发展趋势。软件产业将成为全球高科技产业发展最主要的推动力,这对我国软件产业的发展和国际化软件人才的培养提出了新的挑战。

项目管理是 20 世纪 40 年代后期迅速发展起来的一种计划管理办法,它不仅适用于国防、航天、建筑、工程、金融、制造等领域,而且也适用于电子、计算机、软件开发等 IT 行业。IT 软件产业的发展历程共经历了 3 个时代:最初为 20 世纪 70 年代开始的以结构化分析、设计、评审和结构化测试为特征的结构化生产时代;其次是 90 年代中期以 IT 软件过程为中心的时代;现在,我们正处在第三个时代,即软件工业化生产时代。40 多年来 IT 软件事业的发展使越来越多的人认识到要高效率、高质量、低成本和高回报率(ROI)地完成 IT 软件项目,必须要引入项目管理手段,将人员、技术和工具等有效地组织起来。从实际经验来看,IT 软件项目管理是影响 IT 软件项目全局的主要因素,而技术只影响局部。因此在规模和复杂性都在不断增长的软件项目及 IT 全球化的市场变动中,IT 软件项目管理将起到关键性的作用,同时也是软件工业化生产的必要条件。

随着计算机的日益普及和广泛应用,软件系统的规模和复杂度与日俱增,软件技术面临着新的挑战。大型复杂软件的开发是一项特殊的工程,不仅与传统工程一样,需要按照工程化的方法去组织管理软件的开发,而且软件开发更具有特殊性和复杂性。软件工程作为一门指导计算机软件系统开发和维护的工程学科,其应用水平和普及程度关系到我国的国际竞争力。近年来软件工程对软件产业的支撑作用日渐凸现。

软件工程是一门实践性很强的学科,只有通过软件开发的实践才能真正掌握和应用软件工程的理论知识。本书分为 12 章。第 1 章是绪论,讲述了软件、软件危机,软件工程的开发与原理,以及软件的生命周期的概况。第 2 章介绍了软件的可行性研究与需求分析及方法等内容。第 3 章到第 6 章,分别对软件的概要设计、详细设计、编码与实现以及软件的测试、调试与维护进行了详细的解说。第 7 章对软件中目前最流行的面向对象研究方法进行了描述。第 8 章简单介绍软件工程标准与开发环境的编写。第 9 章到第 11 章则主要对项目管理的三要素,即质量、进度和成本作了说明。第 12 章是软件项目的收尾与验收管理,当一个项目验收完毕,项目就基本上完成了它的历史使命。如何进行有效地项目收尾总结将是提高项目管理水平的关键。

本书编写过程中,阅读和借鉴了大量的国内外相关专家和学者的研究成果,在此向有关作者表示由衷的敬意。同时也得到了许多同行专家的大力支持和帮助,在此表示衷心的感谢。限于编者学识水平,书中的疏漏和讹误在所难免,恳请各位专家同仁予以批评指正。

编者

2013年8月

目 录

第 1 章 绪论	1
1.1 软件与软件危机	1
1.2 软件工程及其基本原理	6
1.3 软件生命周期	11
1.4 软件开发模型	13
第 2 章 可行性研究与需求分析管理	26
2.1 问题定义与可行性研究	26
2.2 需求分析概述	34
2.3 获取需求的方法	41
2.4 结构化分析方法	43
2.5 原型化方法	45
2.6 需求规格说明与评审	47
2.7 需求管理	50
第 3 章 软件的概要设计	61
3.1 软件设计的概念和原则	61
3.2 概要设计概述	69
3.3 概要设计中常用的图形工具	73
3.4 结构化设计方法	77
3.5 概要设计文档与复审	84
第 4 章 软件的详细设计	86
4.1 详细设计概述	86
4.2 详细设计的描述工具	87
4.3 人机界面设计	92
4.4 详细设计文档与复审	94

第 5 章 软件编码与实现	98
5.1 程序设计语言	98
5.2 编码风格	105
5.3 程序效率	113
第 6 章 软件测试、调试与维护	118
6.1 软件测试概述	118
6.2 软件测试方法	123
6.3 软件测试策略	126
6.4 软件调试与排错	140
6.5 软件维护与再工程	144
第 7 章 面向对象方法学研究	159
7.1 面向对象方法学概述	159
7.2 面向对象的分析	166
7.3 面向对象的设计	179
7.4 面向对象的实现	191
7.5 统一建模语言(UML)	196
第 8 章 软件工程标准与开发环境	213
8.1 软件工程标准	213
8.2 软件文档	220
8.3 软件开发工具	223
8.4 软件开发环境	226
8.5 计算机辅助软件工程(CASE)	228
第 9 章 软件项目进度与成本管理	232
9.1 软件项目进度管理	232
9.2 软件项目成本管理	254
第 10 章 软件项目质量与风险管理	294
10.1 软件项目质量管理	294
10.2 软件项目风险管理	319
第 11 章 人力资源管理与团队建设	346
11.1 人力资源管理	346

11.2 软件项目团队建设.....	356
第 12 章 软件项目收尾与验收管理	378
12.1 软件项目收尾.....	378
12.2 软件项目验收.....	381
12.3 项目移交与清算.....	392
12.4 软件项目后评价.....	393
参考文献.....	399

第 1 章 绪 论

1.1 软件与软件危机

1.1.1 软件

1. 软件的概念与特征

有些人认为软件就是程序,实际上,认为软件就是程序,开发软件就是编写程序的观点是错误的。

软件(Software)是能够实现预定功能和性能的可执行的计算机程序和使程序正常执行所需的数据,加上描述软件开发过程及其管理、程序的操作和使用的有关文档。可见,软件更为全面的定义应当包括程序、数据、相关文档的完整集合。

可以将软件简单地定义为:

软件=程序+数据+文档(见图 1-1)。

其中,“程序”是按事先设计的预定功能和性能要求编写的指令序列;“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、维护和使用有关的技术数据和图文材料。

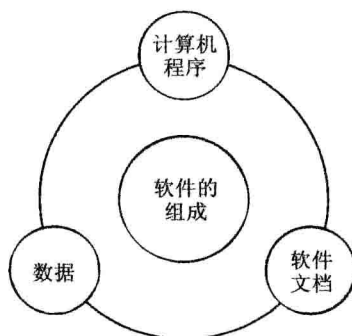


图 1-1 软件的组成

作为一种特殊的产品,软件具有下列独特的性质。了解它们能够帮助全面、正确地理解计算机软件的概念,更好地利用软件的特点进行软件的开发、管理和维护工作。

- (1) 软件是一种逻辑实体,具有抽象性。它不是具体的物理实体,没有明显的物理形态。
- (2) 软件不是物理实体,不会磨损和老化,但需要随着时间的推移进行升级或淘汰。
- (3) 软件主要的是研制,生产是简单的拷贝。
- (4) 软件成本相当昂贵,其开发仍然依靠少数软件工作者,以手工方式完成开发为主。
- (5) 软件开发周期长,质量不易控制。
- (6) 软件具有复杂性,有时来自它所反映的实际问题的复杂性,有时来自它本身的结构复杂性和算法复杂性,有时甚至来自社会因素的影响。
- (7) 软件的开发和运行常受到计算机系统的限制,即受环境影响大。
- (8) 软件维护不同于硬件的维修,常常会产生新的问题。

2. 软件的分类

随着软件技术的不断发展,支持人们日常学习、工作的软件产品的种类和数量都已经很多,这使得软件工程正面临着持续挑战。当前的计算机软件分为七大类。

(1) 系统软件

系统软件是指控制和协调计算机及外部设备,支持应用软件开发和运行的系统;是无须用户干预的各种程序的集合,其主要功能是调度、监控和维护计算机系统,并负责管理计算机系统中各种独立的硬件,使它们协调工作。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。

系统软件是计算机系统必不可少的一个组成部分。系统软件使得计算机使用者和其他软件将计算机当做一个整体,而不需要顾及底层的硬件是如何工作的。系统软件的工作通常伴随着频繁地与硬件交往、大量地为用户服务、资源的共享与复杂的进程管理,以及复杂数据结构的处理等。

(2) 应用软件

应用软件是用户可以使用的各种程序设计语言,以及用各种程序设计语言编制的应用程序的集合。应用软件分为应用软件包和用户程序,其中的应用软件包是利用计算机解决某类问题而设计的程序的集合,供多用户使用。

现在几乎所有的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,应用范围最为广泛,是直接服务于用户的软件。

(3) Web 应用软件

“Web 应用”涵盖了众多的应用程序产品,最简单的是仅用文本和有限的图形表达信息的一组超文本链接文件。

Web 应用软件不仅为用户提供了标准特性、计算功能和内容信息,而且还和企业数据库和商务应用程序相结合。但由于企业与企业之间电子商务模式(Business To Business, B2B)的发展,该网络应用的计算环境会越来越复杂。

(4) 工程和科学软件

工程和科学软件的特征是“数值分析”算法,工程和科学软件带着数字处理算法的标签,覆盖了从天文到地理、从汽车压力分析到航空航天轨道动力学、从分子生物学到自动制造的各个领域。

不过,如今的工程和科学软件已经不仅仅局限于传统的数值算法。计算机辅助设计(Computer Aided Design, CAD)、系统仿真和其他的交互性应用程序已经呈现出实时和系统软件的

特征。

(5) 嵌入式软件

嵌入式软件就是嵌入在硬件中的操作系统和开发工具软件,其在产业中的关联关系为:芯片设计制造→嵌入式系统软件→嵌入式电子设备开发、制造。

智能产品在几乎每一个消费市场或工业市场上都是必不可少的,嵌入式软件驻留在只读内存中,用于控制这些智能产品。嵌入式软件能够执行很有限但专职的功能,例如微波炉的按钮控制,也或者是提供比较强大的功能及控制能力,如汽车中的数字控制,包括燃料控制、仪表板显示、刹车系统等。

(6) 产品线软件

产品的设计是为多个不同用户使用提供特定功能,关注有限的特定市场(例如库存产品)或者大众消费品市场,如:文字处理软件、电子制表软件、计算机绘图软件、多媒体软件、娱乐软件、数据库管理、个人及公司财务应用等。

(7) 人工智能软件

人工智能软件利用非传统数值计算方法去解决复杂问题,这些问题不能通过计算或直接分析得到答案。该领域的应用包括:机器人、模式识别、专家系统、定理证明和博弈等。

人工神经网络(Artificial Neural Networks, ANN)是人工智能软件的一个新分支,最近它得到了很大发展。神经网络仿真人脑的处理机构(生物神经系统的功能),有可能导致一个全新类型的软件登场,它不仅能够识别复杂的模式,而且还能从过去的经验中自行学习进步。

3. 软件的发展历程

在计算机技术发展的初期,人们主要的注意力和兴奋点集中在计算机硬件研制与开发方面,20世纪60年代以前,人们对于软件的认识仅仅只是停留在程序设计阶段。到了计算机硬件技术进入按照“摩尔定律”发展的时期,人们才感觉到软件的重要性,软件工程的思想逐步建立,软件的发展进入了兴旺时期。软件的发展大致有如下四个阶段。

(1) 第一阶段:程序设计阶段

20世纪60年代中期以前是软件发展的第一个阶段。这一阶段的特点是程序员针对具体机型编制程序,编制程序没有规范,编写的程序规模小,强调编写程序的个人技巧,程序设计的效率低。

(2) 第二阶段:程序系统阶段

第二阶段跨越了从20世纪60年代中期到70年代中期的十余年。这个阶段的一个特点是软件产品的使用和“软件作坊”的出现。由于受“软件危机”的困扰,人们开始运用结构化的思想和方法开发软件,建立了“程序=程序语言+数据结构+算法”的概念,强调了程序的可读性、可理解性。这一阶段是“软件工程”的萌芽阶段。软件被开发,使得它们可以在很宽的范围中应用。

(3) 第三阶段:软件工程阶段

第三阶段始于20世纪70年代中期并跨越了整整十年。随着大型、超大型软件开发需求的旺盛,作坊式的软件生产方式完全不能满足发展的需要,软件开发的工程化思想、理论、方法、工具得到迅速发展,“软件工程”学科逐步形成。这一阶段的主要特点是强调软件生产的“工程化”“产品化”“标准化”,以达到软件产品的“可理解”“可阅读”“易维护”“可重用”“低成本”的要求。在十年间,计算机真正成为大众化的东西。

(4) 第四阶段:软件产业阶段

第四阶段是从20世纪80年代中期至今。这一阶段不再是着重于单台计算机和计算机程

序,而是面向计算机和软件的综合影响。软件工程的开发方法缓解和部分克服了软件危机,但是并不能完全克服软件危机,软件产品的开发依然周期长、成本高、难以维护。人们开始研究新的理论、新的方法、自动工具以及智能生产过程,以求达到软件产品生产的产业化、规模化,以期从根本上解决软件危机。这一阶段也可以称为“现代软件工程”阶段,一些新技术开始涌现。近年来出现的“构件技术”“面向应用的开发方法”“云计算架构”“软件流水线装配生产”等,都是这一阶段发展的轨迹。

计算机硬件的迅猛发展、计算机应用的更加深入、广泛,不断地对软件提出新的、更多的要求,软件及其开发技术的发展依然是“路漫漫其修远兮”。

1.1.2 软件危机

1. 软件危机

软件是限制计算机系统发展的关键因素,20 世纪六七十年代,西方计算机科学家把软件开发和维护过程中遇到的一系列严重问题统称为“软件危机(Software Crisis)”。“软件危机”是一种现象,是落后的软件生产方式无法满足迅速增长的计算机软件需求,从而导致软件开发与维护过程中出现一系列严重问题的现象。从总体上看,软件危机就是“落后的软件生产方式与现实需求之间的矛盾”。

2. 软件危机的表现

软件危机主要表现为软件开发中的问题和软件维护中的问题两个方面。

(1) 软件开发中的问题

① 开发的成本与进度估计不准

软件开发成本难以控制,进度不可预计。软件开发过程中用户需求的改变、开发方法的落后以及有效开发过程管理的缺乏,都会使开发进度跟不上,不能按时完成开发任务,最终导致开发成本和进度与原先的估计相差很大。很可能实际成本比估计成本高出一个数量级,而实际进度却比计划进度延迟几个月甚至几年。

② 开发的软件与用户需求不符

软件开发没有对用户需求进行深入的理解,甚至在没有搞清楚待解决问题的情况下就急于编写程序;在开发过程中不按照软件开发的过程规范来进行;软件开发者与用户之间缺乏有效的沟通,交流不充分,不能适应用户需求的变化。这些情况都会导致开发出的软件产品不符合用户的实际需要。

③ 开发的软件质量和可靠性很差

软件产品的质量在软件完成前很难明显看出,质量保证是贯穿于软件开发的整个过程的,如果在软件开发过程中没有很好地运用软件质量的测试和保证技术进行审查、复查和测试等,都可能导致软件发生质量问题。

④ 软件文档不全,难以使用、维护

在软件开发过程中应当同时产生一整套文档资料,它应该是“最新式的”(即和程序代码完全一致的)。这些文档资料可以作为软件开发组织的管理人员用来管理和评价软件开发工程进展状况的依据,也可以作为软件开发人员在软件开发过程中准确地交流信息的通信工具,还可以为

软件维护人员提供必要的帮助。缺乏必要的文档资料或者文档资料不合格,都会给软件开发和维护带来诸多困难和问题。

⑤调试时间太长

软件调试往往会占全部开发工作量的40%,而这其中30%为编程错误,70%为规范及设计错误。

⑥软件越来越复杂,开发生产率很低

软件开发生产率提高的速度,远远跟不上计算机应用迅速普及深入的趋势,形成软件产品“供不应求”的局面,使人类不能充分利用现代计算机硬件提供的巨大潜力。

(2)软件维护中的问题

①软件维护的难度大

在进行维护的同时,很可能是解决旧问题的过程带来新的错误的产生。

②软件维护的费用高

软件维护的费用往往会占到全部费用的40%。

3. 软件危机的产生原因

软件危机的产生有两个方面的原因:客观原因和主观原因。

(1)客观原因

软件本身的逻辑部件复杂,规模庞大。由于软件是逻辑部件,因此,开发者能力的不同造成软件质量、性能的差异;现实问题的复杂性、感知接受的复杂性、理性表达的复杂性、交流沟通的复杂性等构成了软件的复杂性;用户需求不明或者需求不断变化导致软件生产跟不上需求变化;硬件发展太快引起软件需求的剧增,等等。这些都是引起软件危机产生的客观原因。

(2)主观原因

开发过程不科学、不规范。具体表现为:软件开发范型不规范(模型)、软件设计方法不规范(方法)、软件开发支持不规范(工具)、软件开发管理不规范(过程)。另外,还有软件维护方法不科学、软件维护手段落后、软件维护工具缺乏等问题。

4. 软件危机的解决途径

解决软件危机主要从主观方面找问题,找到克服主观因素的办法。其解决主要有两个途径,分别是组织管理和技术措施。组织管理主要是采用系统工程项目管理方法,而技术措施主要是从软件开发技术与方法和软件工具两方面解决。

一般而言,我们可以从四个方面入手:(1)正确地认识软件,充分理解到软件是程序、数据、文档等的完整集合。(2)要理解软件开发不应该是一种组织良好、管理严密、各类人员协同配合、共同完成的工程项目。(3)推广软件开发成功的技术和方法,并且探索研究更好的技术和方法。(4)开发并使用更好的软件工具从而提高软件开发的效率。

1.2 软件工程及其基本原理

1.2.1 软件工程

软件工程形成于克服“软件危机”的过程中,在短短几十年的时间里,软件工程在理论和实践方面都取得了长足的进步。软件工程是一门指导计算机软件系统开发和维护的工程学科,涉及计算机科学、工程科学、管理科学、数学等多门学科。

1. 软件工程的定义

随着软件技术的发展,软件工程的定义也在不断完善,其基本思想仍是强调在软件开发过程中利用工程化准则。很多学者、组织机构对于软件工程都给出了自己的定义。

巴利·波姆(Barry Boehm)对软件工程的定义为:运用现代科学技术知识来设计并构造计算机程序及为开发、运行和维护这些程序所必需的相关文件资料。

1983年,电气和电子工程学会(简称IEEE)在《IEEE 软件工程标准术语》中对软件工程所下的定义为: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、相关的文档资料及在计算机上运行时所必需的数据。1990年,IEEE在软件工程技术术语汇编中,又将对软件工程的定义更改为:软件工程是将系统化的、严格约束的、可量化的方法应用于软件的开发、运行和维护,即对软件的工程化应用。

国家标准 GB/T 11457—1995《软件工程术语》对软件工程的定义为:软件工程是软件开发、运行、维护和引退的系统方法。因而,软件工程是指导计算机软件开发和维护的工程学科。软件工程实际是:采用工程的概念、原理、技术和方法来计划、开发、维护与管理软件,把经过实践检验正确的管理技术和最佳的技术方法相结合,以经济的手段获得在计算机上运行的可靠软件的一系列方法,即工程原理+技术方法+管理技术。

《计算机科学技术百科全书》中对软件工程的定义是:软件工程是应用计算机科学、数学、工程科学、管理科学的原理开发软件的工程。它借鉴传统工程的原则、方法来提高质量和降低成本。其中,计算机科学、数学用于构建模型与算法,工程科学用于制定规范、设计范型、评估成本和确定权衡;管理科学用于计划、资源、质量、成本等管理(见图 1-2)。

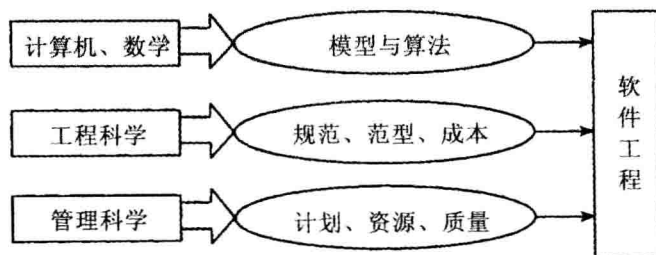


图 1-2 软件工程知识体系

综合上述的观点,软件工程是研究和应用如何以系统性的、规范化的、可量化的过程化方法去开发和维护软件的一门工程学科。它采用工程的概念、原理、技术和方法,把经过时间考验且被证明是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,用于开发和维护软件。软件工程涉及程序设计语言、数据库、软件开发工具、系统平台、标准、设计模式等方面。

软件开发方法的发展过程可分为程序设计时代、程序系统时代、软件工程学时代这三个阶段,如表 1-1 所示。

表 1-1 软件开发方法的 3 个发展阶段

	程序设计时代	程序系统时代	软件工程学时代
名称	程序	软件	软件产品
生产方式	个人	作坊式项目小组	软件组织
质量	与个人水平相关	与小集团水平先关	生产管理可靠性评价和质量控制
设计对象	以硬件为中心	以硬件、软件为中心	以软件为中心
开发工具	无	无系统工具且为个人所有	软件生成器等,为组织所公有
维护	无	由开发者进行维护,且在设计中不重视设计维护问题	设计制作时均考虑维护问题,维护占成本主要部分,近年已达到 80% 以上
设计方法	没有系统的方法	自顶向下的方法	结构化程序设计及自顶向下和自底向上结合的方法

表 1-1 从名称、生产方式、质量、设计对象、开发工具、维护、设计方法等方面介绍了三个阶段各自的特点。从根本上说,软件工程学时代摆脱了软件“个体式”或“作坊式”的生产方法,把软件作为一种社会产品,并进行批量生产。软件工程学研究这种生产过程的有关基础理论、方法论和工具系统。

2. 软件工程的研究内容

软件工程学是一个跨学科、跨领域、范围广泛的新兴学科,是计算机软件理论研究的前沿和核心之一,是当前十分活跃的一个研究方向。软件工程学研究的内容包含三个方面:软件工程理论、软件工程方法学、软件工程管理。

(1) 软件工程理论

软件工程理论包括程序正确性证明、软件可靠性理论、成本估计模型、软件开发模等。

(2) 软件工程方法学

软件工程方法学是研究软件开发方法的学科,包括“方法”“工具”和“过程”三个方面。

①方法:完成软件开发各项任务的技术方法。回答“技术上如何做”的问题。

②工具:为软件工程的过程与方法的应用提供自动或半自动的软件支撑环境和软件工具。

③过程:为了获得高质量软件而需要执行的一系列任务的框架(开发模型)和工作步骤。回答“怎么做才能做好”的问题。

从另一方面来说,软件工程又是一种层次化的技术,如图 1-3 所示。因为任何工程方法都必须以质量控制为基础,因此质量控制是整个软件工程的基础。保证软件开发质量的前提条件是对软件工程中的各个过程进行有效的管理,为此必须为软件过程规定一系列的关键过程域,以此

作为软件项目管理控制的基础,通过人员组织管理、项目计划管理、质量管理等环节来保证软件开发按时按质量完成。软件工程中的“方法”需要完成一系列的任务,如需求分析、开发模型、设计、编码、测试和支持等。利用“工具”适当的辅助,开发人员可以既快又好地做好软件开发工作,这些工具称为计算机辅助软件工程(CASE)工具。所以,一般将“过程”“方法”和“工具”称为软件工程的三要素。这也是现代软件工程的研究内容。



图 1-3 软件工程的层次

(3) 软件工程管理

所谓管理就是通过计划、组织和控制等一系列活动,合理地配置和使用各种资源,以达到既定目标的过程。管理技术包括:软件质量管理、软件项目管理、软件经济等。

3. 软件工程的原则

软件工程的原则是软件开发过程中必须遵循的原则,它们是围绕工程设计、工程支持以及工程管理展开的。软件工程有四项基本原则。

(1) 选取适宜的开发范型

该原则与系统设计有关,能够帮助控制易变的需求。在系统设计中,软件需求、硬件需求以及其他因素之间是相互制约、相互影响的,需要进行权衡。因此,必须认识需求定义的易变性,采用适宜的开发范型予以控制,以保证软件产品满足用户的要求。

(2) 采用合适的技术方法

在软件设计中,通常要考虑软件的模块化、抽象与信息隐蔽、局部化、一致性以及适应性等特征。合适的设计方法有助于这些特征的实现,以达到软件工程的目标。

(3) 提供高质量的工程支持

在软件工程中,软件工具与环境对软件过程的支持颇为重要,软件工程项目的质量与开销直接取决于对软件工程所提供的支撑质量和效用。

(4) 重视开发过程的管理

软件工程的管理,包括有效利用可用的资源、生产满足目标的软件产品、提高软件组织的生产能力等问题。因此,仅当软件过程得以有效管理时,才能实现有效的软件工程。

1.2.2 软件工程的基本原理

作为一项工程,软件工程有许多相关的准则与基本原理。自从 1968 年第一次提出软件工程的概念以来,全世界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与基

本原理。

美国著名的软件工程专家 Barry Boehm 对这 100 多条准则进行了归纳,并总结了美国天合公司(TRW)多年的软件开发经验,于 1983 年提出了软件工程的七条基本原理,他认为这七条原理是保证软件产品质量和开发效率最小且相当完备的集合。

1. 建立完善的计划

统计表明,在不成功的软件项目中有 50%左右是由于计划不周造成的,可见把建立完善的计划作为第一条基本原理是吸取了前人的教训而提出来的。

一个软件从定义、开发、使用、维护,直至被废弃,需要经历一个漫长的时期,这个时期即为软件的生存周期。在软件开发与维护的生命周期中,需要完成许多性质各异的工作。这条基本原理意味着,应该把软件生命周期划分成若干个阶段,并相应地制定出切实可行的计划,从而严格按照计划对软件的开发与维护工作进行管理。

Boehm 认为,在软件的整个生命周期中应该制定并严格执行六类计划,不同层次的管理人员都必须严格按照计划各尽其职地管理软件开发与维护工作,绝不能受用户或上级人员的影响而擅自背离预定计划。这六项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、运行维护计划。

2. 坚持进行阶段评审

在每个阶段都进行严格的评审,以便尽早发现在软件开发过程中所犯的错误,这是软件工程中另一条必须遵循的基本原理。

Boehm 当时就已经认识到,软件的质量保证工作不能等到编码阶段结束之后再进行。原因在于:第一,大部分错误是在编码之前造成的,例如,根据 Boehm 等人的统计,设计错误占软件错误的 63%,编码仅占 37%;第二,错误发现与改正得越晚,所需付出的代价也越高。

3. 实行严格的产品控制

在软件开发过程中,任何一项需求的改变往往需要付出较高的代价,所以是不应随意改变需求的。但是在软件开发过程中改变需求又是难以避免的,根据外部环境的变化,相应地改变用户需求是一种客观需要。显然不能硬性禁止用户提出改变需求的要求,依靠科学的产品控制技术来顺应这种要求才是明智之举。

为了保持软件各个配置成分的一致性,当需求改变时,必须实行严格的产品控制,其中主要是实行基准配置管理。基准配置又称基线配置,它们是经过阶段评审后的软件配置成分(各个阶段产生的文档或程序代码)。基准配置管理也称为变动控制。一切有关修改软件的建议,特别是涉及对基准配置的修改建议,都必须按照严格的规程进行评审,获得批准以后才能实施修改。绝对不能谁想修改软件就随意进行修改。

4. 采用现代程序设计技术

自软件工程的观念提出之后,人们一直把主要精力用于研究各种新的程序设计技术。20 世纪 60 年代末提出的结构化程序设计技术,已经成为绝大多数人公认的先进的程序设计技术。以后各种结构化分析(SA)与结构化设计(SD)技术又进一步发展出来。而面向对象技术的出现真正给软件开发带来了翻天覆地的变化。可见,采用先进的技术既可提高软件开发的效率,又可提高软件维护的效率。