

现代

软件工程模型

及方法探析

XIANDAI

**RUANJIAN GONGCHENG MOXING
JI FANGFA TANXI**

主 编 刘中华 郑毅平

副主编 冯永政 焦小刚 简显锐 邢立峰 赵 韬



中国水利水电出版社
www.waterpub.com.cn

现代

软件工程模型 及方法探析

XIANDAI
RUANJIAN GONGCHENG MOXING
JI FANGFA TANXI

主 编 刘中华 郑毅平
副主编 冯永政 焦小刚 简显锐 邢立峰 赵 韬



中国水利水电出版社
www.waterpub.com.cn

内 容 提 要

本书分 15 章,以软件工程生命周期为主线,深入浅出地对软件工程模型和方法进行了讨论。主要包括绪论,软件生命周期模型,软件可行性研究与需求分析,软件设计,面向对象分析与设计,软件编码与实现,软件测试,软件的调试、维护与再工程,软件复用与构件技术,软件项目管理,软件质量管理,软件配置管理,软件工程标准与环境,敏捷软件开发,软件工程新技术等。

本书内容丰富、取材先进、文字表述简单扼要,是一本比较适合软件开发爱好者的实用性强的学术著作类图书,同时对相关领域的研究人员而言也是一本颇为有益的参考书。

图书在版编目(CIP)数据

现代软件工程模型及方法探析/刘中华,郑毅平主
编.--北京:中国水利水电出版社,2014.11
ISBN 978-7-5170-2636-5
I. ①现… II. ①刘… ②郑… III. ①软件工程
IV. ①TP311.5
中国版本图书馆 CIP 数据核字(2014)第 244675 号

策划编辑:杨庆川 责任编辑:陈 洁 封面设计:崔 蕾

书 名	现代软件工程模型及方法探析
作 者	主 编 刘中华 郑毅平 副主编 冯永政 邢立峰 焦小刚 简显锐 赵 韬
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 24 印张 614 千字
版 次	2015 年 7 月第 1 版 2015 年 7 月第 1 次印刷
印 数	0001—3000 册
定 价	84.00 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

在飞速发展的 IT 领域,新的技术和应用层出不穷,信息技术和信息产业已直接影响到人类的生活和国家的实力。作为信息技术有力支撑的软件,在功能和应用范围上发生了很大的变化,其功能日益强大,应用领域日益扩展,这些变化对软件的开发模式和开发思想产生了巨大的影响。由于学术界和产业界的不懈努力,软件工程已经逐步发展成为一门成熟的专业学科。随着计算机系统的快速发展,应用领域对软件的需求及软件的维护和管理技术不断提高,人们逐渐认识到严格遵循软件工程的方法,可以大大提高软件的开发效率和成功率,减少软件开发和维护中的问题。

软件工程是一门集技术科学、人文科学与实验科学于一体的、交叉的应用科学。本书以软件工程生命周期为主线,比较全面的反映了软件工程的全貌,兼顾了传统的、实用的软件开发方法,又介绍了比较新的技术方法。该书内容全面,不仅对软件的分析、设计、开发、测试和维护的过程进行了详尽地讲解,还配以丰富的内容,对面向对象这种软件开发技术的使用方法进行了讨论。此外,考虑软件工程的“工程”特性,除了技术外,还对软件项目管理、软件质量管理和软件工程环境等知识进行了研究,让读者掌握软件工程技术的同时深刻领会软件工程管理的重要性。

本书有两个特点:一是内容新颖,反映了当前软件开发和管理的最新技术;二是实用性强,对实际的软件开发工作起一定的指导作用。本书在内容组织结构方面作了精心安排,共分为 13 章:第 1 章作为基础,论述了软件、软件危机、软件工程、软件开发方法等相关知识点;第 2 章对软件生命周期的定义、阶段划分及模型展开讨论;第 3 章介绍了软件的可行性研究与需求分析及方法等内容;第 4~8 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,其中,在第 5 章中也就当前软件中最流行的面向对象研究方法进行了描述;第 9 章软件复用与构件技术,对软件复用的现状及问题、构件复用、构件工程、领域工程、基于构件的软件开发等知识展开讨论;第 10 章主要讲述了软件项目管理的有关内容,包括进度计划与管理、成本估算、风险管理、组织和人员管理等多方面的内容;第 11 章对软件质量管理进行了研究,包括软件质量度量与评价、质量保证、技术评审、质量体系;第 12 章软件工程标准与环境中,对软件开发工具与开发环境进行了探析;第 13 章敏捷软件开发探析中,对敏捷宣言、敏捷方法的发展历程及动态、极限编程、Scrum 方法进行了讨论。

本书在编写过程中,阅读和借鉴了大量的国内外相关专家学者的研究成果,吸取了许多人的宝贵经验,在此向这些文献的作者表示衷心的感谢。由于现代软件工程是一门迅速发展的学科,相关技术发展日新月异,因此,在内容的编写和安排等方面难免存在不妥之处和错误,敬请读者批评指正,提出宝贵意见和建议。

编者

2014 年 8 月

目 录

前言

第 1 章 绪论	1
1.1 软件	1
1.2 软件危机	5
1.3 软件工程	7
1.4 软件开发方法	12
第 2 章 软件生命周期及模型	16
2.1 软件生命周期的定义	16
2.2 软件生命周期的阶段划分	16
2.3 软件生命周期模型	18
第 3 章 软件可行性研究与需求分析	30
3.1 问题定义与可行性研究	30
3.2 需求分析概述	35
3.3 获取需求的方法	43
3.4 结构化分析方法	45
3.5 快速原型分析方法	47
3.6 需求验证与评审	49
3.7 需求管理	51
第 4 章 软件设计	60
4.1 软件设计概述	60
4.2 软件概要设计	63
4.3 软件详细设计	71
4.4 用户界面设计	78
4.5 软件结构化设计方法	81
4.6 面向数据结构的设计方法	87
第 5 章 面向对象分析与设计	94
5.1 面向对象方法学概述	94
5.2 面向对象的分析	98

5.3	面向对象的设计	108
5.4	面向对象的实现	118
第 6 章	软件编码与实现	124
6.1	软件实现的目标及策略	124
6.2	程序设计语言	125
6.3	程序设计风格	131
6.4	程序效率	139
6.5	编程安全	141
第 7 章	软件测试	144
7.1	软件测试概述	144
7.2	软件测试方法和技术	149
7.3	软件测试的策略	161
7.4	面向对象的测试	173
7.5	测试用例的设计	175
第 8 章	软件的调试、维护与再工程	182
8.1	软件调试与排错	182
8.2	软件维护概述	186
8.3	软件维护的过程	189
8.4	软件的可维护性分析	192
8.5	逆向工程和再工程	200
第 9 章	软件复用与构件技术	204
9.1	软件复用概述	204
9.2	软件复用的现状及问题	208
9.3	构件复用与构件工程	210
9.4	领域工程	215
9.5	基于构件的软件开发	222
第 10 章	软件项目管理	233
10.1	软件项目管理概述	233
10.2	软件项目进度管理	238
10.3	软件项目成本估算	251
10.4	软件项目风险管理	259
10.5	软件项目组织和人员管理	273
10.6	软件项目收尾与验收管理	278

第 11 章	软件质量管理	288
11.1	质量管理概述	288
11.2	软件质量度量与评价	295
11.3	软件质量保证	306
11.4	软件技术评审	312
11.5	软件质量体系	314
第 12 章	软件工程标准与环境	319
12.1	软件工程标准	319
12.2	软件文档	326
12.3	软件开发工具	330
12.4	软件开发环境	333
12.5	计算机辅助软件工程(CASE)	338
第 13 章	敏捷软件开发探析	350
13.1	敏捷软件开发概述	350
13.2	极限编程方法	363
13.3	Scrum 方法	368
参考文献		375

第 1 章 绪 论

1.1 软 件

软件是信息化的核心,信息、物资和能源已经成为人类生存和发展的重要保障,信息技术的快速发展为人类社会带来了深刻的变革。软件产业关系到国家信息化和经济发展、文化与系统安全,体现了一个国家的综合实力。

1.1.1 软件的定义

软件是计算机系统运行的指令、数据和资料的集合,包括计算机程序、数据及其相关文档的完整集合,如图 1-1 所示。

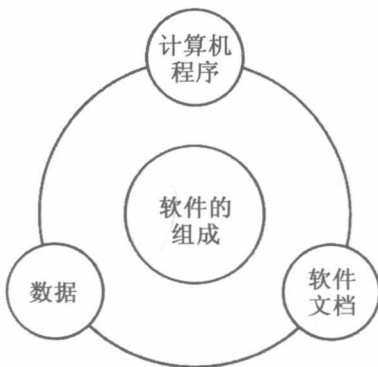


图 1-1 软件的组成

其中,计算机程序是人们为了完成特定的功能而编制的一组指令集;数据是使程序能处理的具有一定数据结构的信息;软件文档(Document)是与程序开发、维护和使用有关的图文材料,如软件开发计划书、需求规格说明书、设计说明书、测试分析报告和用户手册等。

国内外一些专家认为:软件包括程序及开发、使用、维护程序所需的文档,由应用程序、系统程序、面向用户的文档及面向开发者的文档构成,即软件=程序+文档。

软件(Software)更为全面准确的定义应当包括程序、数据、相关文档的完整集合和完善的售后服务,即软件=程序+数据+文档+服务。

有时软件也称为信息系统,它是指由一系列相互联系的部件(程序模块)组成,为实现某个目标对信息进行输入、处理、存储、输出、反馈和控制的集合,分为操作系统和应用系统等。一般实例介绍的信息系统主要是指应用系统,即应用软件。

1.1.2 软件的特点

与硬件产品相比,软件的特点可归纳如下。

①软件是一种逻辑实体,而不是具体的物理实体,因而它具有抽象性。这个特点使它与计算

机硬件或其他工程对象有着明显的差别。人们可以把它记录在介质上,但却无法看到软件的形态,必须通过观察、分析、思考、判断,去了解它的功能、性能及其他特性。因此,软件开发过程的进度难以衡量,质量难以评价,管理和控制相当困难。

②软件是复杂的。软件是用来解决实际问题的,而且是人开发出来的。人在认识实际问题时存在着两重复杂性:一是感性认识的复杂性,这是因为问题本身的复杂性以及人知识结构的缺陷导致的;二是理性认识的复杂性,经常存在“只可意会不可言传”的情景,即使表达出来,可能也已经和实际问题本身存在差异了。人在解决实际问题时,受开发方法和工具的限制,有时会出现程序逻辑结构和问题结构之间存在些许差异的情况,问题空间和解空间不一致会导致软件更加复杂。软件的复杂性主要来自于软件需求的复杂性,对软件需求的正确理解是保障软件质量的重要环节。

③软件的生产与硬件不同,既没有明显的制造过程;也不像硬件那样,一旦研制成功,就可以完全重复地制造,并在制造过程中进行质量控制,以保证产品的质量。软件是通过人们的智力活动,把知识与技术转化成信息的一种产品。虽然有些软件研制成功后,也可以大量地复制同一内容的副本,但很多软件产品是“定做”的,是高强度的脑力劳动成果,而且软件的质量控制重在软件开发。

④在软件的运行和使用期间,没有硬件那样的机械磨损,老化问题。任何机械、电子设备在运行和使用中,其失效率大都遵循如图 1-2(a)所示的 U 形曲线(即浴盆曲线)。而软件的情况与此不同,因为它不存在磨损和老化问题,而存在退化问题。为了适应硬件、系统环境以及需求的变化,必须要多次修改(维护)软件,如图 1-2(b)所示。然而软件的修改不可避免地会引入新的错误,导致软件的失效率升高,从而使得软件可靠性下降。当修改的成本变得难以接受时,软件就被抛弃。

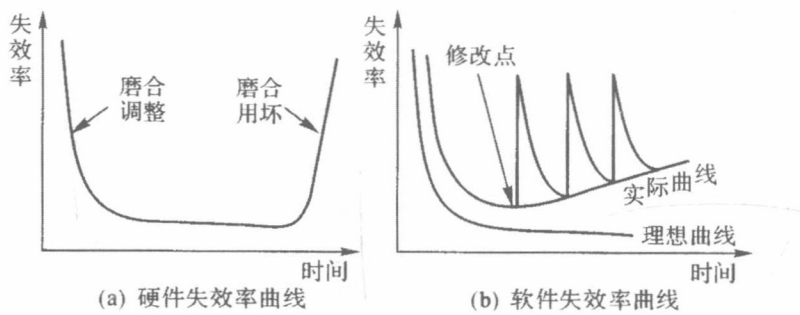


图 1-2 失效率

⑤软件对硬件和环境有着不同程度的依赖性,这导致了软件升级和移植的问题。随着计算机技术的发展,计算机硬件和支撑环境不断升级,为了适应运行环境的变化,软件也需要不断维护,并且维护成本通常比开发成本高许多。

⑥硬件的设计和建造可通过画电路图,做一些基本的分析以保证可以实现预定的功能,根据功能和接口要求选定并购买零件。而软件设计中几乎没有软件构件。大多数软件是新开发的,而不是通过已有的构件组装而来的。有可能在货架上买到的软件,它本身就是一个完整的软件,而不能作为构件再组装成新的程序。

⑦软件成本相当昂贵。软件的研制工作需要投入大量的、复杂的、高强度的脑力劳动,它的成本自 20 世纪 80 年代以来已大大超过硬件成本,硬件/软件成本比率的变化趋势如图 1-3

所示。

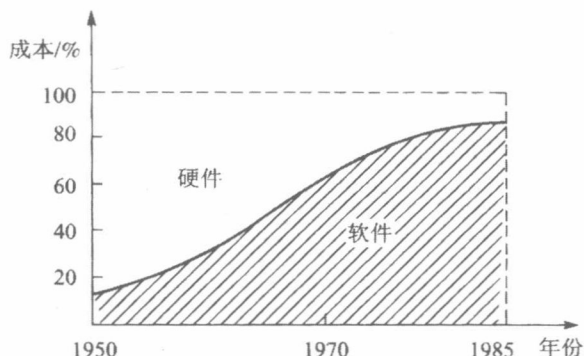


图 1-3 硬件/软件成本比率的变化趋势

⑧软件投入运行时还涉及许多社会因素。软件的实际需求来自于客户，最终投入客户环境使用时，其应用效果会受到客户组织结构、管理机制、企业文化的影响，如当软件无法适应客户的业务处理流程时，软件的实际应用效果就会受到限制；另外，软件的应用效果还受到最终用户的观念和心理因素的影响，这是因为最终用户必须放弃过去的工作习惯方式来适应软件新的业务处理方式，由于惯性的作用，软件开始投入运行时的效果经常不理想。因此，在软件开发和投入运行时必须考虑这些社会因素，这是和硬件应用完全不一样的。

以上特点使得软件开发进展情况较难衡量；软件开发质量难以评价；产品的生产管理、过程控制及质量保证都相当困难。

1.1.3 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。由于人们对软件关心的侧重点不同，对软件的分类也很难有一个科学、统一的标准。但对软件的类型进行必要的划分，根据不同类型的工程对象采用不同的开发和维护方法是很有价值的，因此，有必要从不同的角度讨论计算机软件的分类情况。

1. 按照软件功能分类

按照功能的不同，可以把软件划分为系统软件、支撑软件和应用软件。

①系统软件：是与计算机硬件结合最紧密的软件，它在计算机系统中必不可少，可以协调各个物理部件的工作，同时服务于其他上层软件。操作系统就是最典型的系统软件，它负责管理系统的资源，并为上层软件的运行提供了必备的接口和条件。

②支撑软件：是工具性软件，它一方面可以协调用户进行软件开发，另一方面还能对应用软件进行维护。我们常用的文本编辑器、绘图软件、数据库管理系统和 CASE 工具系统等都属于支撑软件。

③应用软件：是为特定的领域或服务开发的针对性较强的软件。它的种类极其繁多，应用范围最为广泛，是直接服务于用户的软件。比如，地理信息系统软件、航空售票软件、教务管理系统软件和信息管理系统等。

2. 按照软件规模分类

按照软件开发所需要的人力、时间及软件的规模大小，可以把软件划分成为微型、小型、中

型、大型和超大型 5 种类型,如表 1-1 所示。

表 1-1 软件规模的分类

类别	开发人员	研制期限	源程序行数
微型	1 名	1~4 周	小于 500 行
小型	1~2 名	1~6 月	500~5000 行
中型	2~5 名	1~2 年	5000~50000 行
大型	5~20 名	2~3 年	50000~100000 行
超大型	20 名以上	3 年以上	10 万行以上

现在微型软件和小型软件较少,绝大部分是大中型软件。随着软件产品规模的不断增大,类别指标也可能会变化。

3. 按照软件工作方式分类

按软件工作方式的不同,可将软件划分为实时处理软件、分时软件、交互式软件和批处理软件。

①实时处理软件:指在事件或数据产生时,立即给予处理,并及时反馈信号,控制需要监测和控制的过程的软件。这类软件的工作主要包括数据采集、分析、输出三部分,其处理时间是有严格限定的,如果在任何时间超出了这一限制,都将造成事故。

②分时软件:允许多个联机用户同时使用计算机。系统把处理器的时间轮流分配给各联机用户,使各用户都感到只是自己在使用计算机的软件。

③交互式软件:能实现人机通信的软件。这类软件接收用户给出的信息,但在时间上没有严格的限定,这种工作方式给予用户很大的灵活性。

④批处理软件:把一组输入作业或一批数据以成批处理的方式一次运行,按顺序逐个处理的软件,属于最传统的工作方式。

4. 按软件使用的频度分类

按使用的频度,可将软件分为使用频度低的软件,如用于人口普查、工业普查的软件,以及使用频度高的软件,如银行的财务管理软件等。

5. 按软件可靠性的要求分类

有些软件对可靠性的要求相对较低,软件在工作中偶尔出现故障也不会造成不良影响。但也有一些软件对可靠性要求非常高,一旦发生问题就可能造成严重的经济损失或人身伤害。因此,这类软件特别强调软件的质量。

6. 按照软件服务对象的范围分类

按软件服务对象的范围,可将软件分为面向部分客户的项目软件和面向市场的产品软件。

①项目软件:也称定制软件,是受某个特定客户(或少数客户)的委托,由软件开发机构在合同的约束下开发出来的软件。

②产品软件:是面向市场需求,由软件开发机构开发出来后直接提供给市场,或是为千百个用户服务的软件,如办公处理软件、财务处理软件和一些常用的工具软件等。

1.2 软件危机

软件危机是指在软件开发和维护过程中遇到的一系列严重问题。这些问题绝不仅是“不能正常运行的”软件才具有的,实际上几乎所有软件都不同程度地存在这些问题。概括地说,软件危机包含两方面的问题:①如何开发软件,怎样满足对软件的日益增长的需求;②如何维护数量不断膨胀的已有软件。

1.2.1 软件危机的主要表现

软件危机是指在计算机软件的开发和维护过程中所遇到的一系列严重问题。这些问题不仅是不能正常运行的软件才具有的,事实上,几乎所有软软件都不同程度地存在这些主要。具体包括两方面的问题:一个是如何满足日益增长的软件需求,另一个是如何对现有的软件进行维护。这两方面的危机主要有以下6种表现形式。

(1) 软件开发成本和进度估计不够准确

由于在软件开发过程中变数很多,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,往往造成开发进度一拖再拖,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。而赶进度和节约成本的做法又往往损害软件产品的质量,引起用户的不满。

(2) 用户对“已完成的”软件系统不满意

产生这种现象的原因主要有三个:一是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;二是在开发过程中不按照软件开发的过程规范来进行;三是软件开发与用户之间缺乏有效的沟通,交流不充分,闭门造车从而导致开发出的软件产品不符合用户的实际需要。

(3) 软件产品的质量没有保证

造成这种现象的原因主要是很长一段时间以来,对软件的可靠性和质量没有确切的定量概念和评价标准,在软件开发的过程中不能很好地运用软件质量的测试和保证技术。

(4) 软件产品维护困难

很多程序中的错误是非常难改正的,实际上不可能使这些程序适应新的硬件环境,也不能根据用户的需要在原有程序中增加一些新的功能。可重用的软件还是一个没有完全做到的、正在努力追求的目标,人们仍然在重复开发类似的或基本类似的软件。

(5) 软件缺少合格的文档资料

计算机软件不仅仅是程序,还应该有一整套文档资料。这些文档资料应该是在软件开发过程中产生出来的,而且应该是“最新式的”(即和程序代码完全一致的)。软件开发组织的管理人员可以根据这些文档资料来管理和评价软件开发工程的进展状况;软件开发人员可以利用它们作为通信工具,在软件开发过程中准确地交流信息;对于软件维护人员而言,这些文档资料更是必不可少的。缺乏必要的文档资料或者文档资料不合格,必然给软件开发和维护带来诸多困难和问题。

(6) 软件的开发效率无法满足需求

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。

以上列举的仅仅是软件危机的一些明显的表现,与软件开发和维护有关的问题远远不止这些。

1.2.2 产生软件危机的原因

软件危机的出现及其日益严重的趋势,充分暴露了软件产业在早期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的理解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) 软件项目缺乏有力的组织和管理

软件与一般程序不同,它的一个显著特点是规模庞大,而且程序复杂性将随着程序规模的增加而呈指数上升。为了在预定时间内开发出规模庞大的软件,必须由许多人分工合作,然而,如何保证每个人完成的工作合在一起确实能构成一个高质量的大型软件系统,更是一个极端复杂困难的问题,这不仅涉及许多技术问题,诸如分析方法、设计方法、形式说明方法、版本控制等,更重要的是必须有严格而科学的管理。

(2) 软件开发过程缺乏恰当的方法和工具

软件本身独有的特点确实给开发和维护带来一些客观困难,但是人们在开发和使用计算机系统的长期实践中,也确实积累和总结出了许多成功的经验。如果坚持不懈地使用经过实践考验证明是正确的方法,许多困难是完全可以克服的,过去也确实有一些成功的范例。但是,目前相当多的软件专业人员对软件开发和维护还有不少糊涂观念,在实践过程中或多或少地采用了错误的方法和技术,这可能是使软件问题发展成软件危机的主要原因。

(3) 软件开发人员与用户沟通不足

事实上,对用户要求没有完整准确的认识就匆忙着手编写程序是许多软件开发工程失败的主要原因之一。只有用户才真正了解他们自己的需要,但是许多用户在开始时并不能准确地叙述他们的需要,软件开发人员需要做大量深入细致的调查研究工作,反复多次地和用户交流信息,才能真正全面、准确、具体地了解用户的要求。对问题和目标的正确认识是解决任何问题的前提和出发点,软件开发同样也不例外。急于求成,仓促上阵,对用户要求没有正确认识就匆忙着手编写程序,这就如同不打好地基就盖高楼一样,最终必然垮台。事实上,越早开始写程序,完成它所需要用的时间往往越长。

1.2.3 软件危机的解决途径

解决软件危机的途径必须从软件开发的工程化方法入手,使用现代工程的概念、原理、技术和方法去指导软件的开发、管理和维护。必须充分认识到软件开发不是某种个体劳动的神秘技巧,而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。必须充分吸取和借鉴人类长期以来从事各种工程项目所积累的行之有效的原理、概念、技术和方法。特别要吸取几十年来人类从事计算机硬件研究和开发的经验教训。

同时应该推广使用在实践中总结出来的开发软件的成功的技术和方法,并且研究探索更有效的方法,尽快消除在计算机系统早期发展阶段形成的一些错误概念和做法。

研发更好的开发和使用的软件工具也非常必要。正如机械工具可以“放大”人类的体力一样,软件工具可以“放大”人类的智力。在软件开发的每个阶段都有许多繁琐重复的工作需要做,在适当的软件工具辅助下,开发人员可以把这类工作做得既快又好。

总之,为了解决软件危机,既要有技术措施(方法和工具),又要有必要的组织管理措施。现代软件工程正是从管理和技术两方面研究如何更好地开发和维护计算机软件的一门新兴学科。

1.3 软件工程

1.3.1 软件工程的定义

概括地说,软件工程是指导计算机软件开发和维护的一门工程学科。采用工程的概念、原理、技术和方法来开发与维护软件,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,以经济地开发出高质量的软件并有效地维护它,这就是软件工程。

人们曾经给软件工程下过多种定义,下面给出两个典型的定义。

1968年在第一届 NATO(North Atlantic Treaty Organization)会议上曾经给出了软件工程的一个早期定义:“软件工程就是为了经济地获得可靠的且能在实际机器上有效地运行的软件,而建立和使用完善的工程理念。”这个定义不仅指出了软件工程的目的是经济地开发出高质量的软件,而且强调了软件工程是一门工程学科,它应该建立并使用完善的工程原理。

1999年IEEE(Institute of Electrical and Electronics Engineers)进一步给出了一个更全面的定义:“软件工程是:①把系统的、规范的、可度量的途径应用于软件开发、运行和维护过程,也就是把工程应用于软件;②研究①中提到的途径。”

虽然软件工程的定义不尽相同,强调的重点也有差异,但是,人们普遍认为一个成功的软件项目应该具有以下几个特征。

- ①成本:付出的开发成本比较低。
- ②可见性:每个过程活动均能取得明确的结果告终,使过程的进展对外可见。
- ③可移植性:开发出来的软件产品容易移植。
- ④可靠性:不会出现过程错误,或发现在产品出现故障之前。
- ⑤健壮性:软件性能优良,不受意外发生问题的干扰。
- ⑥可维护性:过程可随软件机构需求的变更或随认定的过程改进而演进。
- ⑦速度:从给出规格说明起,就能较快地完成开发而交付。

软件工程包括方法、工具和过程三个基本要素。方法主要指提供如何进行软件开发的技术,包括项目规划、需求分析、数据结构、总体结构设计等多个方面。软件工具的作用是为方法的实施提供自动的或半自动的软件支撑环境。过程通过定义方法使用的顺序、所需交付的文档资料、用于质量管理和关系协调的管理等,将软件工程的方法和工具结合起来,其目的是进行合理、及时的开发。软件工程的三个要素并不是孤立的,而是相互联系在一起的。首先这些因素可以从不同侧面帮助开发者提升个人能力,提高协作的效率和水平;另外,这些因素相互支持和关联,形成了软件工程的技术体系。

1.3.2 软件工程的目标和原则

1. 软件工程的目标

软件工程是一门工程性学科,其目的是采用各种技术上和管理上的手段组织实施软件项目,成功地建造软件系统。项目成功的几个主要目标是:①付出较低的开发成本,在规定的时限内获

得功能、性能方面满足用户需求的软件；②开发的软件移植性较好；③易于维护且维护费用较低；④软件系统的可靠性高。

在实际开发的过程中，要同时满足上述几个目标是非常困难的。这些目标之间有些是互补关系，有些是互斥关系，如图 1-4 所示。因此在解决问题时，要根据具体情况，必要时牺牲某个目标以满足其他优先级更高的目标，只要保证总体目标满足要求，软件开发就是成功的。

可见，软件工程所追求的目标是：多、快、好、省。

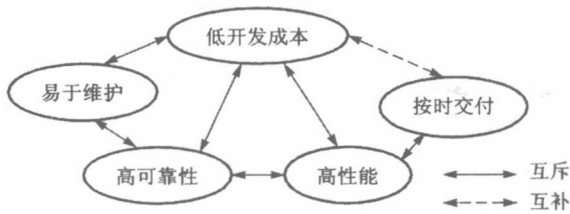


图 1-4 软件工程的目标

2. 软件工程的原则

若要满足上述这些目标，在软件开发过程中必须遵循以下软件工程原则。

(1) 模块化

模块是程序中逻辑上相对独立的成分，通过分解的手段，将复杂的问题从时间上或是从规模上划分成若干个较小的、相对独立的、容易求解的子问题，子问题间应有良好的接口定义，然后分别求解。例如，C 语言中的函数、C++ 语言中的类都是模块，模块化有助于抽象和信息隐藏，有助于表示复杂的系统。

(2) 抽象

抽象是事物最基本的特性和行为，通常采用分层抽象的方法。这种方法可以有效地控制软件开发的复杂性，有利于软件的可理解性和开发过程的管理。把问题(系统)由顶向下逐层分解，上层可以看作是下一层的抽象。

(3) 信息隐蔽

将模块中的软件设计决策封装起来的技术。信息隐蔽由 D. Parnas 提出。根据信息隐蔽的原则，系统中的模块设计成黑箱，与模块有关的数据信息尽量隐蔽在模块内部，与该模块无关的信息就不要放在里头。如果要修改，尽量控制在只修改一个模块就可以，模块之间的调用仅使用模块接口说明。

(4) 模块的高内聚和低耦合

模块划分时，要考虑将逻辑上相互关联的计算机资源集中到一个物理模块内，保证模块之间具有松散的耦合，模块内部具有较强的内聚。这有助于控制解的复杂性。

(5) 确定性

软件开发过程中所有概念的表达应是规范的、确定的和无二义性的。这有助于人们在进行交流时不会产生误解，保证整个开发工作能够协调一致、顺利地进行。

(6) 一致性

整个软件系统(包括文档和程序)的各个模块均使用一致的概念、符号和术语，程序内部接口一致性，软件与硬件接口一致性，系统规格说明与系统行为一致性，用于形式化规格说明的公理系统的一致性。实现一致性需要良好的软件设计工具、设计方法和编程风格的支持。

(7) 完备性

考虑管理和技术的完备性是为了能够在时限内实现系统所要求的功能,并保证软件质量。在软件开发和运行过程中必须进行严格的技术评审,以保证各阶段开发结果的有效性。

(8) 可验证性

开发大型软件系统需要对系统逐步分解,系统分解应该遵循易检查、易测试,以及易评审的原则,以便保证系统的正确性。

1.3.3 软件工程研究的内容

软件工程有方法、工具和过程三个要素。软件工程方法就是研究软件开发是“如何做”的;软件工具是研究支撑软件开发方法的工具,为方法的运用提供自动或者半自动的支撑环境,软件工具的集成环境,又称为计算机辅助软件工程(Computer Aided Software Engineering, CASE);软件工程过程则是指将软件工程方法与软件工具相结合,实现合理、及时地进行软件开发的目的,为开发高质量软件规定各项任务的工作步骤。

软件工程是一门边缘学科,涉及的学科多,研究的范围广。归结起来,软件工程研究的主要内容有软件开发方法和技术、软件开发工具及环境、软件管理技术、软件规范(国际规范)等方面,这里主要讲述软件开发技术和软件工程管理。

1. 软件开发技术

软件开发技术主要讨论软件开发的各种方法及工作模型。其中,开发方法包括面向过程的结构化开发方法、面向数据结构的开发方法和面向对象的开发方法;工作模型包括多方面的任务,如软件系统需求分析、总体设计,以及如何构建良好的软件结构、数据结构和算法设计等,同时讨论具体实现技术。软件工程工具为软件工程提供了支持,计算机辅助软件工程 CASE 为软件开发建立了良好的工程环境。

2. 软件工程管理

软件工程管理是指对软件工程的全过程进行控制管理,包括质量管理、软件工程经济学、成本估算、计划安排等内容,软件工程化与规范化使得各项工作有章可循,以保证软件生产率和软件质量的提高。软件工程标准可分为4个层次:国际标准、行业标准、企业标准和项目规范。

随着软件事业的发展,软件工程所研究的内容也在不断地发生变化。

1.3.4 软件工程的基本原理分析

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许多关于软件工程的准则或信条。美国著名的软件工程专家 Barry W. Boehm(柏瑞·贝姆教授)综合这些专家的意见,并总结了 TRW 公司(美国汤普森-拉莫-伍尔德里奇公司)多年开发软件的经验,于1983年提出了软件工程的7条基本原理。

1. 用分阶段的生命周期计划严格管理

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,然后严格按照计划对软件的开发与维护工作进行管理。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,绝不能受客户或上级人员的影响而擅自背离预定计划。其中,计划包括6类:项目概要计划、里程碑计划、项目控制计划、产品控制计划、验证计划和运行维护计划。

2. 实行严格的产品控制

变更需求是让开发人员很头痛的一件事。但实践告诉我们,需求的改动往往是不可避免的。这就要求我们采用科学的产品控制技术来顺应这种要求。其中主要是实行基准配置管理(又称为变更控制),即凡是修改软件的建议,尤其是涉及基本配置的修改建议,都必须按规定进行严格的评审,评审通过后才能实施。这里的“基准配置”指的是经过阶段评审后的软件配置成分,及各阶段产生的文档或程序代码等。当需求变更时,其他各个阶段的文档或代码都要随之相应变化,以保证软件的一致性。

3. 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,尽量明确地制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。

4. 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。事实上,高素质开发人员的工作效率比低素质开发人员的工作效率要高几倍到几十倍,开发工作中犯的错误也要少得多;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。

5. 采用现代程序设计技术

现代程序设计技术就是结构化技术,包括结构化分析、结构化设计、结构化编码和结构化测试。采用先进的技术不仅可以提高软件开发和维护的效率,而且可以提高软件产品的质量。

6. 坚持进行阶段评审

统计结果显示:大部分错误是设计错误,大约占 63%;错误发现得越晚,改正错误付出的代价就越大,相差大约两到三个数量级。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。

7. 承认不断改进软件工程实践的必要性

要积极主动地采纳新的软件技术,注意不断总结经验,改进开发的组织和过程,有效地通过过程质量的改进提高软件产品的质量。

上述 7 条原理是在面向过程的程序设计时代提出来的,但是在目前出现了面向对象程序设计的时代仍然有效。还有一条基本原理在软件的开发和管理中特别重要,需要补充进去,作为软件工程的第 8 条基本原理。

8. 二八定律

对软件项目进度和工作量的估计:一般人主观上认为已经完成了 80%,但实际上只完成了 20%。对程序中存在问题的估计:80%的问题存在于 20%的程序之中。对模块功能的估计:20%的模块实现了 80%的功能。对人力资源的估计:20%的人解决了软件中 80%的问题。对投入资金的估计:企业信息系统中 80%的问题,可以用 20%的资金来解决。在软件开发和管理的历史上有无数的案例都验证了二八定律。所以软件工程发展到今天,可以认为它的基本原理共有 8 条。

1.3.5 软件工程环境和管理

1. 软件工程环境

软件工程方法和软件工具是软件开发的两大支柱,它们之间密切相关。软件工程方法提出