

21世纪高等学校规划教材 | 软件工程

软件测试

韩利凯 主编

高寅生 袁溪 副主编



清华大学出版社

21世纪高等学校规划教材 | 软

软件测试

韩利凯 主编

高寅生 袁溪 副主编



清华大学出版社

内 容 简 介

本书针对高校计算机相关专业软件测试课程的需要而编写,系统地介绍了软件测试的基础知识与应用技术。本书内容包括软件测试的基本概念和基本知识、软件测试计划、软件测试的基本技术、软件测试过程、测试用例设计、测试报告与测试评测、软件测试项目管理、面向对象软件测试、软件测试自动化以及一个实际软件项目的测试案例,通过该案例的学习,以加深读者对软件测试技术和软件测试过程的理解,加强理论知识的实践性。本书还对目前比较流行的测试工具软件做了介绍。

本书内容全面、深入浅出、理论和实践相结合,通过本书的学习读者能够较好地掌握软件测试的基本知识和基本技术。本书可作为高校计算机专业的软件测试课程的教材,也可作为软件测试培训班的教材或者软件测试人员的自学参考书。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

软件测试/韩利凯主编. —北京:清华大学出版社,2013

21 世纪高等学校规划教材·软件工程

ISBN 978-7-302-32840-7

I. ①软… II. ①韩… III. ①软件—测试—高等学校—教材 IV. ①TP311.5

中国版本图书馆 CIP 数据核字(2013)第 136596 号

责任编辑:郑寅堃 王冰飞

封面设计:傅瑞学

责任校对:梁毅

责任印制:沈露

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦 A 座 邮 编:100084

社总机:010-62770175 邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质 量 反 馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

课 件 下 载: <http://www.tup.com.cn>, 010-62795954

印 装 者:北京市清华园胶印厂

经 销:全国新华书店

开 本:185mm×260mm 印 张:12.75 字 数:320 千字

版 次:2013 年 8 月第 1 版 印 次:2013 年 8 月第 1 次印刷

印 数:1~2000

定 价:25.00 元

出版说明

随着我国改革开放的进一步深化,高等教育也得到了快速发展,各地高校紧密结合地方经济建设发展需要,科学运用市场调节机制,加大了使用信息科学等现代科学技术提升、改造传统学科专业的投入力度,通过教育改革合理调整和配置了教育资源,优化了传统学科专业,积极为地方经济建设输送人才,为我国经济社会的快速、健康和可持续发展以及高等教育自身的改革发展做出了巨大贡献。但是,高等教育质量还需要进一步提高以适应经济社会发展的需要,不少高校的专业设置和结构不尽合理,教师队伍整体素质亟待提高,人才培养模式、教学内容和教学方法需要进一步转变,学生的实践能力和创新精神亟待加强。

教育部一直十分重视高等教育质量工作。2007年1月,教育部下发了《关于实施高等学校本科教学质量与教学改革工程的意见》,计划实施“高等学校本科教学质量与教学改革工程(简称‘质量工程’)”,通过专业结构调整、课程教材建设、实践教学改革、教学团队建设等多项内容,进一步深化高等学校教学改革,提高人才培养的能力和水平,更好地满足经济社会发展对高素质人才的需要。在贯彻和落实教育部“质量工程”的过程中,各地高校发挥师资力量强、办学经验丰富、教学资源充裕等优势,对其特色专业及特色课程(群)加以规划、整理和总结,更新教学内容、改革课程体系,建设了一大批内容新、体系新、方法新、手段新的特色课程。在此基础上,经教育部相关教学指导委员会专家的指导和建议,清华大学出版社在多个领域精选各高校的特色课程,分别规划出版系列教材,以配合“质量工程”的实施,满足各高校教学质量和教学改革的需要。

为了深入贯彻落实教育部《关于加强高等学校本科教学工作,提高教学质量的若干意见》精神,紧密配合教育部已经启动的“高等学校教学质量与教学改革工程精品课程建设工作”,在有关专家、教授的倡议和有关部门的大力支持下,我们组织并成立了“清华大学出版社教材编审委员会”(以下简称“编委会”),旨在配合教育部制定精品课程教材的出版规划,讨论并实施精品课程教材的编写与出版工作。“编委会”成员皆来自全国各类高等学校教学与科研第一线的骨干教师,其中许多教师为各校相关院、系主管教学的院长或系主任。

按照教育部的要求,“编委会”一致认为,精品课程的建设工作从开始就要坚持高标准、严要求,处于一个比较高的起点上;精品课程教材应该能够反映各高校教学改革与课程建设的需要,要有特色风格、有创新性(新体系、新内容、新手段、新思路,教材的内容体系有较高的科学创新、技术创新和理念创新的含量)、先进性(对原有的学科体系有实质性的改革和发展,顺应并符合21世纪教学发展的规律,代表并引领课程发展的趋势和方向)、示范性(教材所体现的课程体系具有较广泛的辐射性和示范性)和一定的前瞻性。教材由个人申报或各校推荐(通过所在高校的“编委会”成员推荐),经“编委会”认真评审,最后由清华大学出版

社审定出版。

目前,针对计算机类和电子信息类相关专业成立了两个“编委会”,即“清华大学出版社计算机教材编审委员会”和“清华大学出版社电子信息教材编审委员会”。推出的特色精品教材包括:

(1) 21 世纪高等学校规划教材·计算机应用——高等学校各类专业,特别是非计算机专业的计算机应用类教材。

(2) 21 世纪高等学校规划教材·计算机科学与技术——高等学校计算机相关专业的教材。

(3) 21 世纪高等学校规划教材·电子信息——高等学校电子信息相关专业的教材。

(4) 21 世纪高等学校规划教材·软件工程——高等学校软件工程相关专业的教材。

(5) 21 世纪高等学校规划教材·信息管理与信息系统。

(6) 21 世纪高等学校规划教材·财经管理与应用。

(7) 21 世纪高等学校规划教材·电子商务。

(8) 21 世纪高等学校规划教材·物联网。

清华大学出版社经过三十多年的努力,在教材尤其是计算机和电子信息类专业教材出版方面树立了权威品牌,为我国的高等教育事业做出了重要贡献。清华版教材形成了技术准确、内容严谨的独特风格,这种风格将延续并反映在特色精品教材的建设中。

清华大学出版社教材编审委员会

联系人:魏江江

E-mail: weijj@tup.tsinghua.edu.cn



前言

随着软件规模和复杂性的大幅度提升,软件质量可靠性的问题变得日益突出。软件测试是保证软件质量的关键技术之一,同时也是软件开发过程中的一个重要环节,其理论知识和技术工具都在飞速发展。几乎每个大中型 IT 企业的软件产品在发布前都需要大量的质量控制、测试和文档工作,而这些工作必须依靠拥有娴熟技术的专业软件人才来完成。软件测试人员就是这样的一个企业重头角色。但是目前在国内大多数软件企业中,测试人员的数量不到开发人员数量的五分之一,远远落后于国外先进水平。由此可见,我国对软件测试人员的需求量越来越大。

在上述社会现状下,高校计算机相关专业已广泛开设软件测试课程,由此,对软件测试课程教材的需求也与日俱增,特别是理论和实践相结合的优质教材。

本书的编写特色是:(1)理论和实践相结合,在理论上拓展实践技能。本书前半部分详细介绍软件测试理论知识,后半部分通过一个实际软件项目的测试案例将理论知识加以应用,以提高读者的实践能力。(2)注重知识的多元性。全书贯穿软件工程、软件质量管理和软件项目管理等知识,让读者体会到软件测试在整个软件工程生命周期中的地位和作用。(3)例题丰富。在软件测试基本技术、软件测试用例设计等章节有详细的例题分析,有助于读者对理论知识的深刻理解。

本书按照软件测试流程编写。第 1 章主要介绍软件测试的基本概念和相关理论;第 2 章详细讲解软件测试计划,强调在开始测试工作之前,详细、全面的计划是很重要的;第 3~5 章详细介绍软件测试的核心技术和测试过程,以及测试用例的设计方法,还重点介绍白盒测试法、黑盒测试法、单元测试、集成测试、系统测试、回归测试等重点知识和相关技能,同时针对测试用例的设计给出若干例题,让读者全面理解软件测试的实际测试方法;第 6 章、第 7 章具体介绍软件测试项目跟踪和管理的相关知识和技术,以培养读者软件测试文档的编写、缺陷的报告和分析以及问题跟踪系统等各方面的技能;第 8 章介绍面向对象软件的测试方法,因为面向对象软件具有自身的特殊性,所以单独对它进行讲解;第 9 章重点介绍软件测试自动化的引入、发展和优缺点,让读者能正确地认识软件测试自动化,并介绍目前较为流行的一些测试工具,包括 LoadRunner、WinRunner、TestDirector、JUnit 等;第 10 章通过一个实际软件项目的测试案例,把软件测试的流程系统地展现在读者面前,加深读者对软件测试技术和软件测试过程的理解,使其学会理论和实践的结合。

本书由韩利凯主编,高寅生、袁溪任副主编。参与编写的人员有韩利凯、袁溪、孙少波、袁佳乐、任强、吴燕妮。编写安排如下:孙少波编写第 1 章、第 7 章;袁佳乐编写第 2 章、第 5 章;韩利凯编写第 3 章;任强编写第 4 章;袁溪编写第 6 章、第 9 章、第 10 章;吴燕妮编写第 8 章。本书在编写过程中得到了多方面的帮助、指导和支持,在此向他们表示由衷的感谢。

由于编者水平有限,书中难免有疏漏和不妥之处,敬请广大读者提出宝贵的意见和建议。

编 者

2013 年 5 月于西安文理学院

目 录

第 1 章 软件测试概述	1
1.1 软件、软件危机和软件工程	1
1.1.1 软件	1
1.1.2 软件危机	2
1.1.3 软件工程	4
1.2 软件缺陷与软件故障	5
1.2.1 相关概念	5
1.2.2 软件缺陷	6
1.2.3 软件故障	8
1.3 软件质量与质量模型	9
1.3.1 软件质量的定义	9
1.3.2 软件质量的三种模型	10
1.3.3 软件质量的度量	11
1.4 软件测试	12
1.4.1 软件测试原则	13
1.4.2 软件测试目标	13
1.4.3 软件测试的具体内容	13
1.4.4 软件测试的主要方法	14
1.4.5 软件测试人员的基本素质	18
1.5 本章小结	20
习题 1	20
第 2 章 软件测试计划	21
2.1 软件测试计划的作用	21
2.2 软件测试计划的原则	22
2.3 如何制定软件测试计划	24
2.4 制定测试计划时面对的问题	25
2.5 衡量测试计划的标准	26
2.6 制定测试计划	27
2.6.1 确定测试范围	27
2.6.2 选择测试方法	29
2.6.3 测试标准	29

2.6.4	自动化测试工具的选择	31
2.6.5	测试软件的编写	32
2.6.6	合理减少测试的工作量	32
2.6.7	制定测试计划	33
2.6.8	编写系统测试计划文档	34
2.7	本章小结	35
习题 2		35
第 3 章	软件测试的基本技术	36
3.1	软件测试技术的分类	36
3.1.1	从是否需要执行被测软件的角度分类	36
3.1.2	从软件测试用例设计方法的角度分类	36
3.1.3	从软件测试的策略和过程的角度分类	37
3.2	静态测试和动态测试	37
3.2.1	静态测试	37
3.2.2	动态测试	38
3.3	黑盒测试方法	38
3.3.1	黑盒测试方法概述	38
3.3.2	等价类划分法	39
3.3.3	边界值分析法	42
3.3.4	决策表法	45
3.3.5	因果图法概述	48
3.3.6	黑盒测试方法的选择	51
3.4	白盒测试	52
3.4.1	逻辑覆盖测试	52
3.4.2	路径分析测试	57
3.5	本章小结	61
习题 3		61
第 4 章	软件测试过程	62
4.1	软件测试过程概述	62
4.2	单元测试	64
4.2.1	单元测试的主要任务	64
4.2.2	单元测试的执行过程	65
4.3	集成测试	66
4.3.1	集成测试的主要任务	67
4.3.2	集成测试的方法	67
4.3.3	集成测试方法的对比	70
4.4	确认测试	71

4.4.1 进行有效性测试(功能测试)	72
4.4.2 软件配置复查	72
4.5 系统测试	72
4.6 验收测试	75
4.7 回归测试	75
4.8 本章小结	77
习题 4	78
第 5 章 测试用例设计	79
5.1 测试用例的基本概念	79
5.2 测试用例的设计	80
5.2.1 测试用例设计说明	80
5.2.2 测试用例的编写标准	81
5.2.3 测试用例设计考虑的因素	82
5.2.4 测试用例设计的基本原则	82
5.2.5 测试用例的分类	83
5.3 测试用例设计实例	84
5.4 测试用例的执行与跟踪	89
5.4.1 执行测试用例	89
5.4.2 跟踪测试用例	89
5.4.3 维护测试用例	91
5.5 测试用例管理	92
5.6 本章小结	95
习题 5	95
第 6 章 测试报告与测试评测	97
6.1 软件缺陷和软件缺陷种类	97
6.1.1 软件缺陷案例	97
6.1.2 软件缺陷的含义	98
6.1.3 软件缺陷的种类	99
6.1.4 软件缺陷的严重性等级	102
6.2 软件缺陷的生命周期	103
6.3 分离和再现软件缺陷	104
6.4 正确面对软件缺陷	105
6.5 报告软件缺陷	107
6.5.1 报告软件缺陷的基本原则	107
6.5.2 有效地报告软件缺陷带来的好处	108
6.5.3 IEEE 软件缺陷报告模板	108
6.6 软件缺陷的跟踪管理	110

6.6.1	软件缺陷跟踪管理的目标	110
6.6.2	软件缺陷的描述	110
6.6.3	软件缺陷管理的一般流程	111
6.6.4	软件缺陷数据统计	112
6.6.5	软件缺陷跟踪管理系统	112
6.6.6	手工报告和跟踪软件缺陷	114
6.7	软件测试的评测	114
6.7.1	覆盖评测	115
6.7.2	质量评测	115
6.7.3	性能评测	116
6.8	测试总结报告	117
6.9	本章小结	118
习题 6		119
第 7 章	软件测试项目管理	120
7.1	软件测试项目管理概述	120
7.2	软件测试文档	121
7.2.1	测试文档的作用	121
7.2.2	测试文档的类型	121
7.2.3	主要软件测试文档	122
7.3	软件测试的组织与人员管理	124
7.3.1	测试的过程及组织	124
7.3.2	测试方法的应用	125
7.3.3	测试的人员组织	125
7.3.4	软件测试文件	126
7.4	软件测试过程管理	127
7.4.1	软件测试过程模型	127
7.4.2	软件测试过程管理	129
7.4.3	软件测试过程管理理念	130
7.4.4	软件测试过程管理实践	131
7.4.5	软件测试过程可持续改进	132
7.5	软件测试的配置管理	133
7.5.1	进行测试配置管理的必要性	133
7.5.2	测试配置管理的方法和内容	133
7.5.3	测试配置管理的应用	136
7.5.4	软件测试的测试版本控制	137
7.5.5	测试版本控制的概念	137
7.6	软件测试风险管理	141
7.6.1	风险管理	141

7.6.2 风险识别	141
7.6.3 风险评估	142
7.6.4 风险应对	142
7.7 软件测试的成本管理	143
7.8 本章小结	145
习题 7	145
第 8 章 面向对象软件测试	146
8.1 面向对象软件的特点及其对测试的影响	146
8.2 面向对象软件测试的不同层次及其特点	148
8.3 面向对象软件测试模型	150
8.3.1 面向对象分析的测试	151
8.3.2 面向对象设计的测试	153
8.3.3 面向对象编程的测试	154
8.4 本章小结	154
习题 8	154
第 9 章 软件测试自动化	155
9.1 软件测试自动化基础	155
9.1.1 软件测试自动化的含义	155
9.1.2 手工测试和自动化测试的比较	156
9.1.3 软件测试自动化的局限性	156
9.2 软件测试自动化的作用和优势	157
9.3 软件测试自动化的引入	162
9.3.1 软件测试自动化的正确认识	162
9.3.2 对企业自身现状的评估分析	162
9.3.3 软件测试自动化的引入条件	163
9.4 软件测试自动化的实施	164
9.4.1 软件测试自动化的流程框架	164
9.4.2 软件测试自动化的实施过程	165
9.5 软件测试工具分类	167
9.6 几种常用软件测试工具	167
9.6.1 性能测试工具 LoadRunner	167
9.6.2 功能测试工具 WinRunner	169
9.6.3 白盒测试工具 JUnit	171
9.6.4 测试管理工具 TestDirector	171
9.6.5 专用测试工具 WAST	172
9.7 本章小结	173
习题 9	173

第 10 章 测试实践——一个实际软件项目的测试案例	174
10.1 被测试项目介绍	174
10.1.1 被测系统概述	174
10.1.2 用户注册、登录和注销模块介绍	178
10.2 测试计划	180
10.2.1 概述	180
10.2.2 定义	180
10.2.3 质量风险摘要	181
10.2.4 测试进度计划	181
10.2.5 进入标准	182
10.2.6 退出标准	182
10.2.7 测试配置和环境	182
10.2.8 关键参与者	182
10.3 测试过程概述	183
10.3.1 单元测试	183
10.3.2 集成测试	183
10.3.3 系统测试	184
10.3.4 验收测试	184
10.4 测试用例设计	184
10.5 缺陷报告	186
10.6 测试结果总结分析	186
10.6.1 测试总结报告	186
10.6.2 测试用例分析	187
10.7 软件测试自动化工具	188
10.8 文档测试	190
10.9 本章小结	190
习题 10	191
参考文献	192

第1章

软件测试概述

软件的质量就是软件的生命,人们在长期的开发过程中积累了大量经验并形成了许多行之有效的方法,以期保证软件的质量。但是借助这些方法,我们也只能尽量减少软件中的错误和不足,却不能完全避免错误。软件测试是最有效的排除和防止软件缺陷与故障的手段,并由此促进了软件测试理论与技术的快速发展。新的测试理论、测试方法、测试技术手段正在不断涌现。

本章主要介绍软件测试的一些基础知识:软件危机和软件工程、软件缺陷与软件故障、软件质量与质量模型、软件测试、软件测试人员的基本素质等相关知识。

1.1 软件、软件危机和软件工程

1.1.1 软件

软件是一系列按照特定顺序组织的计算机数据和指令的集合。一般来讲软件被划分为编程语言、系统软件、应用软件和介于后两者之间的中间件。软件并不只包括可以在计算机(这里的计算机是指广义的计算机)上运行的计算机程序,与这些计算机程序相关的文档一般也被认为是软件的一部分。简单地说软件就是程序加文档的集合体。

1. 系统软件

系统软件是为计算机使用提供最基本的功能的软件,它负责管理计算机系统中各种独立的硬件,使得它们可以协调工作。系统软件使得计算机使用者和其他软件可以将计算机当作一个整体而不需要顾及底层每个硬件是如何工作的。系统软件可分为操作系统和支撑软件,其中操作系统是最基本的软件。

(1) 操作系统是管理计算机硬件与软件资源的程序,它是计算机系统的内核与基石。操作系统身负诸如管理与配置内存、决定系统资源供需的优先次序、控制输入与输出设备、操作网络与管理文件系统等基本事务,并提供一个让使用者与系统交互的操作接口。

(2) 支撑软件顾名思义是支撑各种软件开发与维护的软件,又称为软件开发环境(SDE)。它主要包括环境数据库、各种接口软件和工具组。著名的软件开发环境有 IBM 公司的 Web Sphere 等,包括一系列基本的工具(例如编译器、数据库管理、存储器格式化、文件系统管理、用户身份验证、驱动管理、网络连接等方面的工具)。

2. 应用软件

系统软件并不针对某一特定应用领域,而应用软件则相反,不同的应用软件根据用户和所服务的领域提供不同的功能。

简而言之,应用软件是为了某种特定的用途而被开发的软件。它可以是一个特定的程序,比如一个图像浏览器;也可以是一组功能联系紧密,可以互相协作的程序的集合,比如微软的 Office 软件;还可以是一个由众多独立程序组成的庞大的软件系统,比如数据库管理系统。

3. 手机软件

顾名思义,手机软件就是可以安装在手机上的软件,用于完善原始系统的不足并提供个性化。随着科技的发展,现在手机的功能也越来越多,越来越强大,不像过去那么简单死板,目前手机甚至发展到了可以和掌上电脑相媲美。手机软件与计算机软件一样,要考虑你购买这一款手机所安装的系统来决定下载相对应的软件。目前主流手机系统有以下几种: Windows Phone、Symbian、Linux、Research in Motion、Windows Mobile、iPhone iOS、Android。

1.1.2 软件危机

20 世纪 60 年代以前,由于计算机刚刚投入实际使用,软件设计往往只是为了一个特定的应用而在指定的计算机上设计和编制,采用密切依赖于计算机的机器代码或汇编语言,软件的规模比较小,文档资料通常也不存在,很少使用系统化的开发方法,设计软件往往等同于编制程序,基本上是个人设计、个人使用、个人操作、自给自足的私人化的软件生产方式。

20 世纪 60 年代中期,大容量、高速度计算机的出现,使计算机的应用范围迅速扩大,软件开发急剧增长。高级语言开始出现,操作系统的发展引起了计算机应用方式的变化,大量数据处理导致第一代数据库管理系统的诞生。软件系统的规模越来越大,复杂程度越来越高,软件可靠性问题也越来越突出。原来的个人设计、个人使用的方式不再能满足要求,迫切需要改变软件生产方式,提高软件生产率。

1968 年北大西洋公约组织的计算机科学家在联邦德国召开国际会议,第一次讨论软件的危机问题。所谓软件危机(software crisis),泛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。

1. 软件危机的主要表现

软件危机的主要表现如下。

(1) 软件开发费用和进度失控。费用超支、进度拖延的情况屡屡发生。有时为了赶进度或压低成本不得不采取一些权宜之计,这样又往往严重损害了软件产品的质量。

(2) 软件的可靠性差。尽管耗费了大量的人力物力,而系统的正确性却越来越难以保证,出错率大大增加,由于软件错误而造成的损失十分惊人。

(3) 生产出来的软件难以维护。很多程序缺乏相应的文档资料,程序中的错误难以定位,难以改正,有时改正了已有的错误又引入新的错误。随着软件的社会拥有量越来越大,维护占用了大量人力、物力和财力。进入 20 世纪 80 年代以来,尽管软件工程研究与实践取得了可喜的成就,软件技术水平有了长足的进展,但是软件生产水平依然远远落后于硬件生

产水平的发展速度。

(4) 用户对“已完成”的系统不满意现象经常发生。一方面,许多用户在软件开发的初期不能准确完整地向开发人员表达他们的需求;另一方面,软件开发人员常常在对用户需求还没有正确全面认识的情况下,就急于编写程序。

(5) 软件成本在计算机系统总成本中所占的比例居高不下,且逐年上升。由于微电子技术的进步和硬件生产自动化程度不断提高,硬件成本逐年下降,性能和产量迅速提高。然而软件开发需要大量人力,软件成本随着软件规模和数量的剧增而持续上升。美、日两国的统计数字表明,1985年度软件成本甚至占到计算机系统总成本的将近90%。

(6) 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需要,软件产品供不应求的状况使得人类不能充分利用现代计算机硬件所能提供的巨大潜力。

2. 产生软件危机的原因

研究表明,产生软件危机的原因主要有以下几个方面。

1) 与软件本身的特点有关

软件不同于硬件,它是计算机系统逻辑部件:软件样品即是产品,试制过程也就是生产过程;软件不会因使用时间过长而“老化”或“用坏”;软件具有可运行的行为特性,在写出程序代码并在计算机上试运行之前,软件开发过程的进展情况较难衡量,软件质量也较难评价,因此管理和控制软件开发过程十分困难;软件质量不是根据大量制造的相同实体的质量来度量的,而是与每一个组成部分的不同实体的质量紧密相关,因此,在运行时所出现的软件错误几乎都是在开发时期就存在而一直未被发现的,改正这类错误通常意味着改正或修改原来的设计,这就在客观上使得软件维护远比硬件维护困难;软件是一种信息产品,具有可延展性,属于柔性生产,与通用性强的硬件相比,软件更具有多样化的特点,更加接近人们的应用问题。

2) 用户需求不明确

在软件开发过程中,用户需求不明确问题主要体现在4个方面:

- 在软件开发出来之前,用户自己也不清楚软件开发的具体需求;
- 用户对软件开发需求的描述不精确,可能有遗漏、有二义性,甚至有错误;
- 在软件开发过程中,用户还提出修改软件开发功能、界面、支撑环境等方面的要求;
- 软件开发人员对用户需求的理解与用户本来愿望有差异。

3) 缺乏正确的理论指导

缺乏有力的方法学和工具方面的支持。由于软件开发不同于大多数其他工业产品,其开发过程是复杂的逻辑思维过程,其产品在很大程度上依赖于开发人员的智力投入。由于过分地依靠程序设计人员在软件开发过程中的技巧和创造性,加剧软件开发产品的个性化,也是发生软件危机的一个重要原因。

4) 软件开发规模越来越大

随着软件开发应用范围的增广,软件开发规模愈来愈大。大型软件开发项目需要组织一定的人力共同完成,而多数管理人员缺乏开发大型软件开发系统的经验,多数软件开发人员又缺乏管理方面的经验。各类人员的信息交流不及时、不准确、有时还会产生误解。软件开发项目开发人员不能有效地、独立自主地处理大型软件开发的全部关系和各个分支,因此

容易产生疏漏和错误。

5) 软件开发复杂度越来越高

软件不仅仅在规模上快速地发展扩大,而且其复杂性也急剧地增加。软件开发产品的特殊性和人类智力的局限性,导致人们无力处理“复杂问题”。所谓“复杂问题”的概念是相对的,一旦人们采用先进的组织形式、开发方法和工具提高了软件开发效率和能力,新的、更大的、更复杂的问题又摆在人们的面前。

1.1.3 软件工程

在1968年的北约计算机科学家国际会议上,同时提出了“软件工程”一词,正式宣告软件工程学的诞生,作为一个新兴的工程学科,软件工程主要研究软件生产的客观规律性,建立与系统化软件生产有关的概念、原则、方法、技术和工具,指导和支持软件系统的生产活动,以期达到降低软件生产成本、改进软件产品质量、提高软件生产率水平的目标。软件工程从硬件工程和其他人类工程中吸收了许多成功的经验,明确提出了软件生命周期的模型,发展了许多软件开发与维护技术和方法,并应用于软件工程实践,取得良好的效果。

在不断演进的软件开发过程中人们开始研制和使用软件工具,用以辅助进行软件项目管理与技术生产,人们还将软件生命周期各阶段使用的软件工具有机地集合成为一个整体,形成能够连续支持软件开发与维护全过程的集成化软件支援环境,以期从管理和技术两方面解决软件危机问题。

软件工程学科诞生后,人们为其给出了许多不同的定义。最早的定义是由 F. L. Bauer 给出的,即“软件工程是为了经济地获得能够在实际机器上高效运行的、可靠的软件而建立和应用一系列坚实的软件工程原则”。美国梅隆卡耐基大学软件工程研究所(SEI)给出的定义则是:软件工程是以工程的形式应用计算机科学和数学原理,从而经济有效地解决软件问题。但目前普遍使用的软件工程定义是由 IEEE 给出的,即软件工程是将系统性的、规范化的、可量化的方法应用于软件的开发、运行和维护。

综合来看,软件工程概念实际存在两层含义,从狭义概念看,软件工程着重体现在软件过程中所采用的工程方法和管理体系,例如,引入成本核算、质量管理和项目管理等,即将软件产品开发看作是一项工程项目所需要的系统工程和管理学。从广义概念看,软件工程涵盖了软件生命周期中所有的工程方法、技术和工具,包括需求工程、设计、编程、测试和维护的全部内容,即完成一个软件产品所必备的思想、理论、方法、技术和工具。

软件工程学科包含为完成软件需求、设计、构建、测试和维护所需的知识、方法和工具。对软件工程的理解不应局限在理论之上,要更注重实践,软件工程能够帮助软件组织协调团队、运用有限的资源,遵守已定义的软件工程规范,通过一系列可复用的、有效的方法,在规定的时间内达到预先设定的目标。针对软件工程的实施,无论是采用什么样的方法和工具,先进的软件工程思想始终是最重要的。只有在正确的工程思想指导下,才能制定正确的技术路线,才能正确地运用方法和工具达到软件工程或项目管理的既定目标。

软件工程是一门交叉性的工程学科,它将计算机科学、数学、工程学和管理学等基本原理论应用于软件的开发与维护中,其重点在于大型软件的分析与评价、规格说明、设计和演化,同时涉及管理、质量、创新、标准、个人技能、团队协作和专业实践等。从这个意义上看,软件工程可以看作由下列3部分组成。

- 计算机科学和数学,用于构造软件的模型与算法。
- 工程科学,用于制定规范、设计范型、评估成本以及确定权衡等。
- 管理科学,用于计划、资源、质量、成本等管理。

例如,计算机辅助软件工程(Computer Aided Software Engineering, CASE)就是一组工具和方法的集合,可以辅助软件生命周期各阶段进行的软件开发活动。CASE 吸收了CAD(计算机辅助设计)、软件工程、操作系统、数据库、网络和许多其他计算机领域的原理和技术。这个例子也体现了软件工程是学科交叉的、集成和综合的领域。

如果从知识领域看,软件工程学科是以软件方法和技术为核心,涉及计算机的硬件体系、系统基础平台等相关领域,同时还涉及一些应用领域和通用的管理学科、组织行为学科。例如,通过应用领域的知识帮助我们理解用户的需求,从而可以根据需求来设计软件的功能。在软件工程中必然要涉及组织中应用系统的部署和配置所面临的实际问题,同时又必须不断促进知识的更新和理论的创新。为了解决实际问题,需要在理论和应用上获得最佳平衡。

1.2 软件缺陷与软件故障

1.2.1 相关概念

质量、进度和成本是软件项目关注的3大要素,它们相互依赖、相互制约,软件项目管理就是要在这3个目标的建立、跟踪和实现方面达成最佳均衡。软件质量要求的高低,会影响到成本,又会影响到进度。一般来说,不论是顾客还是承制方,对软件质量的既定目标都不会妥协,宁可追加成本或延迟进度,也不会迁就质量目标的降低,对质量都采用“一票否决制”,因此,软件质量在三大目标中往往是位于第一位的。不论是业界传统的狭义软件质量观,还是以ISO/IEC 9126系列标准代表的广义软件质量观,软件缺陷的概念都在软件质量范畴中处于举足轻重的地位。在狭义软件质量观中,软件缺陷的概念绝对处于核心地位,它衍生出一系列软件质量指标,是考察软件质量的唯一依据。在当今业界大力实施的软件能力成熟度模型(Capability Maturity Model for Software, CMM)或能力成熟度模型集成(Capability Maturity Model Integration, CMMI)中,软件缺陷度量仍然是一个核心度量,尤其是攀登高成熟度级别不可或缺的基石。总之,软件缺陷度量的体系结构是各种软件过程改进模型的基础设施,是实施与评估软件质量活动的先决条件。建立完善的软件缺陷度量的体系结构对软件质量管理的策划和实施、软件缺陷的跟踪和管理、软件质量目标的设定和实现、软件过程能力基线的创建和优化都具有重大的现实意义。

历史上,曾经对与软件缺陷相关的一些概念搞得很混淆,各家众说纷纭,使得从业人员无所适从。下面对这些概念进行区分,它们是建立软件缺陷度量的体系结构的先决条件。

(1) 错误(error): 即人为错误,指软件开发人员在开发软件的过程中无意间犯下的技术错误,正是这些错误导致软件工作产品的缺陷。

(2) 缺陷(defect): 指软件工作产品中不满足指定要求的成分,它是静态的,如果不将其消除,它将永远存在。在业界,人们常用另外一个词“bug”指代缺陷,早期美国海军在调试软件时曾发生一个臭虫(bug)引发系统不能正常工作的情况,此典故流传下来致使人们