



普通高等教育“十一五”国家级规划教材 计算机系列教材

软件测试技术教程

傅兵 编著

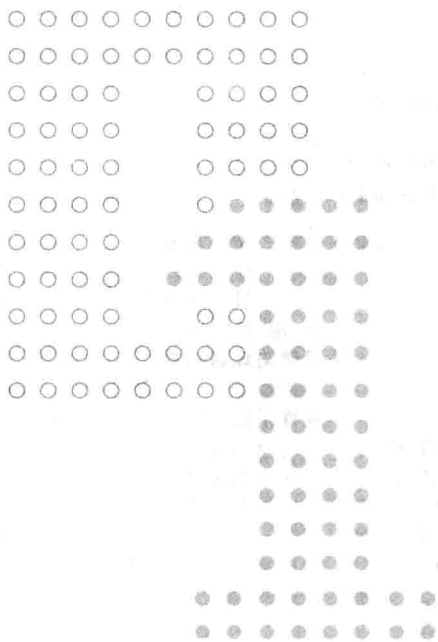
清华大学出版社



计算机系列教材

傅兵 编著

软件测试技术教程



清华大学出版社
北京

内 容 简 介

本书全面地介绍了软件测试的基本理论、基本技术和常用方法以及近几年新的软件测试技术和方法。首先,从软件测试背景和软件开发过程入手,介绍软件测试的基本知识,以及软件质量保证 SQA、软件能力成熟度模型 CMM 和能力成熟度整合模型 CMMI 等内容;然后,全面介绍黑盒测试、白盒测试、单元测试、集成测试、系统测试等测试技术和基本方法;最后,介绍目前比较流行的自动化软件测试工具,并介绍软件测试行业的现状和发展趋势以及如何成为合格的软件测试工程师。

本书的特点是测试技术介绍全面,实践和理论并重,本书的另一个特色是实例多。本书着重介绍软件测试及管理技术理论中最重要、最精华的部分以及它们之间的融会贯通。本书既有整体框架,又有重点理论和技术。通过本课程的学习,使学生了解并掌握软件测试技术的基本内容和软件质量保证的基础知识以及具体的软件测试技术的方法、技术和工具的使用,为全面掌握软件技术和软件项目管理打下坚实的基础。本教材注重培养读者的软件测试实践能力,适应软件企业的环境和业界标准,并和国际先进的软件开发理念和软件测试技术同步。

全书共分9章,第1章绪论,第2章软件测试基础,第3章黑盒测试,第4章白盒测试,第5章单元测试,第6章集成测试,第7章系统测试,第8章自动化测试及工具,第9章软件测试行业综述。每章均附有习题,并配有内容丰富的附录。

本书适合作为高校的计算机专业、软件工程或其他相关专业高年级本科生或研究生软件测试课程的教材,同时还可作为广大软件开发人员、软件测试人员和研究人员的参考书。

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

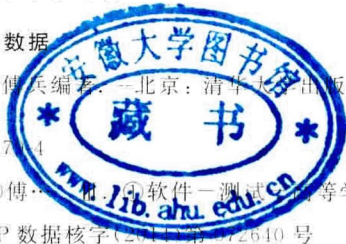
软件测试技术教程/傅兵编著. —北京:清华大学出版社,2014

计算机系列教材

ISBN 978-7-302-36171-4

I. ①软… II. ①傅… 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(2014)第 22640 号



责任编辑:自立军 徐跃进

封面设计:常雪影

责任校对:李建庄

责任印制:王静怡

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 刷 者:三河市君旺印装厂

装 订 者:三河市新茂装订有限公司

经 销:全国新华书店

开 本:185mm×260mm

印 张:18

字 数:449千字

版 次:2014年5月第1版

印 次:2014年5月第1次印刷

印 数:1~2000

定 价:34.50元

产品编号:057164-01

主 任：周立柱

副 主 任：王志英 李晓明

编委委员：（按姓氏笔画为序）

汤志忠 孙吉贵 杨 波

岳丽华 钱德沛 谢长生

蒋宗礼 廖明宏 樊晓桢

责任编辑：马瑛珺

随着信息时代的到来,人们对软件质量的要求越来越高;同时由于软件系统变得越来越复杂,如何提高软件质量是广大计算机技术人员所关注的,这使得软件开发人员和软件测试人员面临着巨大挑战。

基于这种情况,国内许多高校的计算机、软件工程和信息技术等相关专业纷纷开设软件测试课程以培养更多的软件测试人才。目前,市场上的软件测试教材良莠不齐,精品教材少,尤其是对软件测试技术介绍全面、深入的教材更少。为了适应当前教学和软件测试技术人员的需要,编者查阅了大量国内外有关软件测试方面的著作和文献,并结合自己多年的从业和教学经验编写了这本教材。

本书的特点是测试技术介绍全面,不但阐述了所有基本的软件测试技术,并附有许多软件测试实例,从而使读者更好地理解 and 掌握软件测试的基本理论,可以迅速地应用到实际测试工作中去。

全书共分9章。第1章绪论,本章主要介绍软件和软件危机、软件开发过程、软件缺陷以及著名软件缺陷案例,为学习本书的后续内容打好基础,做好准备。第2章软件测试基础,介绍软件测试的定义、软件测试的过程,软件测试的分类,软件测试的必要性,软件测试的原则等。第3章黑盒测试,介绍黑盒测试的基本概念,包括等价类划分、边界值分析法、因果图法、决策表法、正交实验设计法等。第4章白盒测试,介绍白盒测试的基本概念,详细阐述了语句覆盖、判定覆盖、条件覆盖、判定/条件覆盖、条件组合覆盖和路径覆盖等白盒测试的方法以及基于缺陷模式的测试技术等内容。第5章单元测试,主要讲解单元测试概述、对单元测试的误解、单元测试的必要性、单元测试环境和方法、单元测试策略、单元测试用例设计和单元测试过程。第6章集成测试,介绍集成测试概念、集成测试策略、测试用例设计和测试过程等。第7章系统测试,本章从性能测试、可靠性测试、安全性测试、恢复测试、备份测试、可用性测试、协议测试、文档测试、GUI测试、网站测试、 α 测试和 β 测试、回归测试等方面介绍系统测试。第8章自动化测试及工具,介绍自动化测试概述、自动化测试的实施、自动化测试工具的选择和比较。第9章软件测试行业综述,介绍了软件测试行业的现状和发展趋势,软件测试技术的发展方向以及软件测试人员职位和责任、对软件测试工程师的要求。本书的附录内容丰富,既有工具性的内容,如软件测试基本术语中英文词汇、正交表和IEEE模板;也有非常实用性的内容,如软件测试工程师面试题、软件测试工程师考试模拟试题及解析。

本书在编写过程中参阅了大量国内外同行的著作及文献,汲取了软件测试领域的最新知识。在此,对这些作者表示深深的感谢。同时,由于编者的水平有限、时间仓促,书中难免存在错误和不足之处,希望大家批评指正。

编者

2014年3月

F O R E W O R D

第 1 章 绪论 /1

1.1 软件和软件危机 /1

1.1.1 计算机软件 /1

1.1.2 软件危机 /2

1.2 软件开发 /4

1.2.1 软件开发过程 /5

1.2.2 软件开发过程模型 /6

1.3 软件缺陷 /9

1.3.1 软件缺陷概述 /9

1.3.2 软件缺陷的严重性和优先级 /11

1.3.3 软件缺陷分类 /13

1.3.4 预防和修复软件缺陷 /15

1.3.5 软件缺陷案例 /17

习题 /21

第 2 章 软件测试基础 /22

2.1 软件测试的含义 /22

2.1.1 软件测试的发展 /22

2.1.2 软件测试的基本原则 /23

2.1.3 软件测试与软件开发的关系 /25

2.2 软件测试模型 /26

2.3 软件测试过程 /30

2.4 软件测试基本理论 /33

2.4.1 软件测试用例设计 /33

2.4.2 软件测试方法 /36

2.4.3 软件测试的误区 /38

2.5 软件质量 /41

2.5.1 软件质量概述 /41

2.5.2 软件质量保证 /44

2.5.3 软件能力成熟度模型 /48

- 2.5.4 能力成熟度整合模型 /50
- 2.6 软件可靠性 /51
- 习题 /55

第3章 黑盒测试 /56

- 3.1 黑盒测试概述 /56
- 3.2 等价类划分法 /57
 - 3.2.1 划分等价类 /57
 - 3.2.2 设计测试用例 /58
 - 3.2.3 等价类划分法举例 /58
- 3.3 边界值分析法 /63
 - 3.3.1 边界值分析法的含义 /64
 - 3.3.2 边界值分析法原理 /65
 - 3.3.3 边界值分析法举例 /66
- 3.4 决策表法 /68
 - 3.4.1 决策表的含义 /68
 - 3.4.2 决策表法举例 /69
- 3.5 因果图分析法 /73
 - 3.5.1 因果图法的含义 /73
 - 3.5.2 因果图法的原理 /73
 - 3.5.3 因果图法举例 /75
- 3.6 正交实验设计法 /77
 - 3.6.1 正交实验设计法的含义 /77
 - 3.6.2 正交实验法举例 /79
- 3.7 黑盒测试方法比较 /84
- 习题 /85

第4章 白盒测试 /87

- 4.1 白盒测试概述 /87
 - 4.1.1 白盒测试含义 /87
 - 4.1.2 黑盒测试和白盒测试的比较 /88
 - 4.1.3 静态测试和动态测试 /90

4.1.4	程序流程图和控制流图	/91
4.2	逻辑覆盖测试	/92
4.3	白盒静态测试	/97
4.3.1	桌前检查	/97
4.3.2	代码审查	/97
4.3.3	代码走查	/98
4.3.4	代码评审和同行评审	/99
4.3.5	基于缺陷模式测试	/100
4.4	其他白盒测试方法	/109
4.4.1	程序插装测试	/109
4.4.2	程序变异测试	/110
4.4.3	循环语句测试	/111
4.5	白盒测试策略	/112
	习题	/113
第5章	单元测试	/114
5.1	单元测试概述	/114
5.1.1	单元测试的定义	/114
5.1.2	单元测试的目标	/115
5.1.3	单元测试的任务	/115
5.2	对单元测试的误解	/116
5.3	单元测试的必要性	/119
5.4	单元测试环境和方法	/120
5.4.1	驱动模块和桩模块的定义	/120
5.4.2	驱动模块和桩模块的使用条件	/121
5.5	单元测试策略	/122
5.6	单元测试用例设计	/123
5.7	单元测试过程和单元测试工具	/124
5.8	面向对象的单元测试	/126
	习题	/127

第 6 章 集成测试 /128

- 6.1 集成测试概述 /128
- 6.2 集成测试方案 /130
 - 6.2.1 大爆炸式集成测试 /130
 - 6.2.2 渐增式集成 /131
 - 6.2.3 几种集成测试比较 /134
 - 6.2.4 基于功能的集成测试 /135
 - 6.2.5 核心系统先行集成测试 /135
 - 6.2.6 客户/服务器集成测试 /136
 - 6.2.7 高频集成测试 /137
- 6.3 集成测试用例设计 /138
- 6.4 集成测试过程 /139
- 习题 /141

第 7 章 系统测试 /143

- 7.1 性能测试 /146
 - 7.1.1 性能测试概述 /146
 - 7.1.2 压力测试 /151
 - 7.1.3 容量测试 /153
 - 7.1.4 负载测试 /156
- 7.2 可靠性测试 /156
 - 7.2.1 可靠性测试方法 /156
 - 7.2.2 可靠性测试的数学模型 /157
- 7.3 安全性测试 /159
 - 7.3.1 安全性测试概述 /159
 - 7.3.2 安全性测试的主要内容 /160
 - 7.3.3 安全性测试方法 /161
- 7.4 恢复测试 /164
 - 7.4.1 恢复测试的含义 /164
 - 7.4.2 恢复测试的主要内容和步骤 /165
 - 7.4.3 恢复测试中一些要注意的地方 /166
- 7.5 备份测试 /167

7.6	可用性测试	/167
7.6.1	可用性测试概述	/167
7.6.2	可用性测试的发展	/168
7.6.3	可用性测试方法	/168
7.6.4	可用性测试的必备要素	/170
7.6.5	可用性测试时需要注意的问题	/172
7.7	协议测试	/172
7.8	文档测试	/174
7.8.1	文档测试的含义	/174
7.8.2	文档性测试方法	/177
7.9	GUI 软件测试	/178
7.9.1	GUI 测试概述	/179
7.9.2	GUI 软件测试方法	/180
7.9.3	GUI 测试的几个要素	/182
7.9.4	GUI 测试主要内容	/185
7.9.5	GUI 测试常见问题	/186
7.10	网站测试	/187
7.10.1	网站测试的含义	/187
7.10.2	网站测试的主要内容	/188
7.11	α 测试和 β 测试	/192
7.12	回归测试	/194
	习题	/196
第 8 章	自动化测试及工具	/198
8.1	自动化测试概述	/198
8.1.1	自动化测试的含义	/198
8.1.2	自动化测试的优点	/199
8.1.3	自动化测试的缺点	/200
8.1.4	自动化测试与手工测试的互补性	/201
8.2	实施自动化测试	/201
8.2.1	自动化测试的对象和范围	/201
8.2.2	自动功能测试的脚本开发	/202

8.3	自动化测试工具的选择与比较	/203
8.3.1	白盒测试工具	/204
8.3.2	黑盒测试工具	/205
8.3.3	测试管理工具	/206
8.3.4	常用自动化测试工具	/207
8.3.5	自动化测试工具 QTP	/212
8.3.6	自动化测试工具 AutoRunner	/216
	习题	/220
第9章	软件测试行业综述	/221
9.1	软件测试的发展和现状	/221
9.2	软件测试技术的发展方向	/224
9.3	软件测试外包	/226
9.4	对软件测试工程师的要求	/228
9.4.1	软件测试工作特点	/229
9.4.2	软件测试工程师	/232
9.5	软件测试工程师考试	/236
	习题	/238
附录A	基本术语中英文词汇	/239
附录B	正交表	/250
附录C	IEEE 模板	/259
附录D	软件测试工程师面试题及参考答案	/261
附录E	全国计算机等级考试四级软件测试工程师练习	习题 /269
	参考文献	/274

第 1 章 绪 论

随着 1946 年第一台电子计算机的诞生,计算机软件也开始走进人们的视野。由于软件的复杂度的增加,导致软件危机的产生,由于软件缺陷引起了很多著名的软件缺陷案例,在 20 世纪 70 年代逐渐形成了软件工程的概念。

本章主要介绍软件和软件危机、软件开发过程、软件缺陷以及著名软件缺陷案例,为学习本书的后续内容打好基础,做好准备。

1.1 软件和软件危机

计算机系统分为硬件系统和软件系统。软件是软件开发的基础,软件在现代社会占有重要的地位,软件产业已经成为信息社会的支柱产业之一。

1.1.1 计算机软件

1. 软件的概念

计算机软件(Computer Software)是指计算机系统中的程序、数据及其文档。程序是计算任务的处理对象和处理规则的描述。文档是为了便于了解程序所需的阐明性资料。程序必须装入机器内部才能工作,文档一般是给人阅读的,不一定装入机器。

软件是用户与硬件之间的接口界面。用户主要是通过软件与计算机进行交流。软件是计算机系统设计的重要依据。为了方便用户,为了使计算机系统具有较高的总体效用,在设计计算机系统时,必须通盘考虑软件与硬件的结合以及用户的要求和软件的要求。

2. 计算机软件分类

计算机软件总体分为系统软件和应用软件两大类。

系统软件是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将计算机视为一个整体而不需要顾及底层每个硬件是如何工作的。

系统软件包括各类操作系统,如 Windows、Linux、UNIX 等,还包括操作系统的补丁程序及硬件驱动程序,这些都是系统软件类。此外,系统软件还包括一系列基本的工具软件(比如编译器、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)。

应用软件是为了某种特定的用途而开发的软件。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,比如微软公司

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。应用软件可以进一步细分,如工具软件、游戏软件、管理软件等都属于应用软件类。

3. 几种软件的功能

1) 操作系统

操作系统是管理、控制和监督计算机软硬件资源协调运行的程序系统,由一系列具有不同控制和管理功能的程序组成,它是直接运行在计算机硬件上的、最基本的系统软件,是系统软件的核心。操作系统是计算机发展中的产物,其主要目的有两个:一是方便用户使用计算机,是用户和计算机的接口。比如用户输入一条简单的命令就能自动完成复杂的功能,这就是操作系统帮助的结果;二是统一管理计算机系统的全部资源,合理组织计算机工作流程,以便充分、合理地发挥计算机的效率。

2) 语言处理程序(翻译程序)

机器语言是计算机唯一能直接识别和执行的程序语言。如果要在计算机上运行高级语言程序就必须配备程序语言翻译程序(或简称翻译程序)。翻译程序本身是一组程序,不同的高级语言都有相应的翻译程序。

3) 服务程序

服务程序能够提供一些常用的服务,它们为用户开发程序和使用计算机提供了方便,像计算机上经常使用的诊断程序、调试程序、编辑程序均属此类。

4) 数据库管理系统

在信息社会里,社会和生产活动产生的信息很多,使人工管理难以应付,人们希望借助计算机对信息进行搜集、存储、处理和使用。数据库系统就是在这种需求背景下产生和发展的。数据库是指按照一定联系存储的数据集合,可为多种应用共享。数据库管理系统则是能够对数据库进行加工、管理的系统软件,其主要功能是建立、消除、维护数据库及对库中数据进行各种操作。数据库系统主要由数据库、数据库管理系统以及相应的应用程序组成。

5) 应用软件

为解决各类实际问题而设计的程序系统称为应用软件。从其服务对象的角度,又可分为通用软件和专用软件两类。通用软件通常是为解决某一类问题而设计的,而这类问题是很多人都要遇到和解决的。例如,文字处理、表格处理、电子演示等。专用软件则针对性较强,专用软件有些在市场上可以买到,有些特殊功能和需求的软件需要用户自己组织开发。

1.1.2 软件危机

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。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。

1. 软件危机的由来

20世纪60年代以前,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,软件危机开始爆发。

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,第一次讨论软件危机问题,并正式提出“软件工程”一词,从此一门新兴的为研究和克服软件危机应运而生的学科,软件工程诞生了。

2. 软件危机的主要表现

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,出错率很高,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) 用户对软件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) 软件成本逐年上升

由于硬件成本逐年下降,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。从美、日两国的统计数字表明,1985年度软件成本大约占总成本的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。

3. 软件危机产生的原因

如此多的软件危机表现,它们产生的原因是什么呢?软件工程研究结果表明,软件危机的原因主要有以下两个。

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而不是物理部件。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同种实体的质量来度量,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却一直未被发现的,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) 来自于软件开发人员

软件产品是人的思维结果,因此软件生产水平最终在相当程度上取决于软件人员的教育、训练和经验的积累;对于大型软件往往需要许多人合作开发,甚至要求软件开发人员深入应用领域的问题研究,这样就需要在用户与软件人员之间以及软件开发人员之间相互沟通,在此过程中难免发生理解的差异,从而导致后续错误的设计或实现,而要消除这些误解和错误往往需要付出巨大的代价;由于计算机技术和应用发展迅速,知识更新周期加快,软件开发人员经常处在变化之中,不仅需要适应硬件更新的变化,而且还要涉及日益扩大的应用领域问题研究;软件开发人员所进行的每一项软件开发几乎都必须调整自身的知识结构以适应新的问题求解的需要,而这种调整是人所固有的学习行为,难以用工具来代替。

4. 解决途径

软件工程诞生于20世纪60年代末期,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、提高软件生产率的目标。

在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。

1.2 软件开发

软件开发过程是软件工程重要部分,也是软件测试的基础。软件开发流程即软件设计思路和方法的一般过程,包括设计软件的功能和实现的算法和方法、软件的总体结构设

计和模块设计、编程和调试、程序联调和测试以及编写、提交程序,最后达到用户的满意。

1.2.1 软件开发过程

通常的软件开发过程包括需求分析、概要设计、详细设计、编写代码、软件测试、运行和维护六个阶段。

1. 需求分析

系统分析员向用户初步了解需求,然后列出要开发的系统的大功能模块,每个大功能模块有哪些小功能模块。系统分析员进一步了解和析需求,根据自己的经验和需求用相关的工具做出一份文档系统的功能需求文档。系统分析员向用户再次确认需求。

2. 概要设计

开发者需要对软件系统进行概要设计,即系统设计。概要设计需要对软件系统的设计进行考虑,包括系统的基本处理流程、系统的组织结构、模块划分、功能分配、接口设计、运行设计、数据结构设计和出错处理设计等,为软件的详细设计提供基础。

3. 详细设计

在概要设计的基础上,开发者需要进行软件系统的详细设计。在详细设计中,描述实现具体模块所涉及的主要算法、数据结构、类的层次结构及调用关系,需要说明软件系统各个层次中的每一个程序(每个模块或子程序)的设计考虑,以便进行编码和测试。应当保证软件的需求完全分配给整个软件。详细设计应当足够详细,能够根据详细设计报告进行编码。

4. 编写代码

在软件编码阶段,开发者根据《软件系统详细设计报告》中对数据结构、算法分析和模块实现等方面的设计要求,开始具体的编写程序工作,分别实现各模块的功能,从而实现对目标系统的功能、性能、接口、界面等方面的要求。

5. 软件测试

将测试编写好的系统交付给用户使用,用户使用后一个一个地确认每个功能。软件测试是项目研发中一个相当重要的步骤,对于一个大型软件,几个月到1年以上的外部测试都是正常的,因为永远都会有不可预料的问题存在。完成测试后,完成验收并完成最后的一些帮助文档,整体项目才算告一段落,当然日后少不了升级,修补等工作。

6. 运行和维护

在软件测试证明软件达到要求后,软件开发者应向用户提交开发的目标安装程序、数据库的数据字典、用户安装手册、用户使用指南、需求报告、设计报告、测试报告等双方合