

高等院校规划教材
软件工程系列

软件测试与维护基础教程

黄武 洪玫 杨秋辉 余静 编著



机械工业出版社
CHINA MACHINE PRESS



高等院校规划教材·软件工程系列

软件测试与维护基础教程

黄武 洪玫 杨秋辉 余静 编著



机械工业出版社

本书内容丰富完整,包括软件测试基础(软件测试的历史、原则、基本模型、流程等),软件测试技术(软件评审、白盒测试及黑盒测试技术等),软件测试级别(单元测试、集成测试、系统测试、确认测试及回归测试等),软件测试管理(构建测试环境、测试计划、测试设计、测试执行及测试评估等),软件测试工具(测试自动化及测试工具),软件测试应用(配置测试、兼容性测试、本地化测试、网站测试、安全性测试及面向对象测试等)及软件维护等七大部分 25 个章节的内容。

本书不仅内容丰富翔实,而且参考了众多软件测试的国际标准,力求做到概念和原理解释的精确。为了让读者易于理解,该书注重循序渐进的知识讲解方法,注重背景知识介绍及列举大量的实际测试案例来讲解测试知识。

本书可作为高等院校计算机(软件)学院或软件培训机构的教材使用,面向的读者对象包括:高校学生,专业软件培训机构学生,软件公司的测试人员,学习软件测试知识的入门者等。

图书在版编目(CIP)数据

软件测试与维护基础教程/黄武等编著. —北京:机械工业出版社, 2011. 12
高等院校规划教材·软件工程系列
ISBN 978-7-111-36402-3

I. ①软… II. ①黄… III. ①软件-测试-高等学校-教材 ②软件维护-高等学校-教材 IV. ①TP311.5

中国版本图书馆 CIP 数据核字(2011)第 230393 号

机械工业出版社(北京市百万庄大街 22 号 邮政编码 100037)
责任编辑:张宝珠
责任印制:李妍

北京振兴源印务有限公司印刷

2012 年 1 月第 1 版·第 1 次印刷
184mm×260mm·20.5 印张·501 千字
0001-3000 册
标准书号:ISBN 978-7-111-36402-3
定价:39.00 元

凡购本书,如有缺页、倒页、脱页,由本社发行部调换

电话服务

网络服务

社服务中心:(010)88361066

门户网:<http://www.cmpbook.com>

销售一部:(010)68326294

教材网:<http://www.cmpedu.com>

销售二部:(010)88379649

读者购书热线:(010)88379203

封面无防伪标均为盗版

出版说明

计算机技术的发展极大地促进了现代科学技术的发展，明显地加快了社会发展的进程。因此，各国都非常重视计算机教育。

近年来，随着我国信息化建设的全面推进和高等教育的蓬勃发展，高等院校的计算机教育模式也在不断改革，计算机学科的课程体系和教学内容趋于更加科学和合理，计算机教材建设逐渐成熟。在“十五”期间，机械工业出版社组织出版了大量计算机教材，包括“21世纪高等院校计算机教材系列”、“21世纪重点大学规划教材”、“高等院校计算机科学与技术‘十五’规划教材”、“21世纪高等院校应用型规划教材”等，均取得了可喜成果，其中多个品种的教材被评为国家级、省部级的精品教材。

为了满足计算机教育的需求，机械工业出版社策划开发了“高等院校规划教材”。这套教材是在总结我社以往计算机教材出版经验的基础上策划的，同时借鉴了其他出版社同类教材的优点，对我社已有的计算机教材资源进行整合，旨在大幅提高教材质量。我们邀请多所高校的计算机专家、教师及教务部门针对此次计算机教材建设进行了充分的研讨，达成了许多共识，并由此形成了“高等院校规划教材”的体系架构与编写原则，以保证本套教材与各高等院校的办学层次、学科设置和人才培养模式等相匹配，满足其计算机教学的需要。

本套教材包括计算机科学与技术、软件工程、网络工程、信息管理与信息系统、计算机应用技术以及计算机基础教育等系列。其中，计算机科学与技术系列、软件工程系列、网络工程系列和信息管理与信息系统系列是针对高校相应专业方向的课程设置而组织编写的，体系完整，讲解透彻；计算机应用技术系列是针对计算机应用类课程而组织编写的，着重培养学生利用计算机技术解决实际问题的能力；计算机基础教育系列是为大学公共基础课层面的计算机基础教学而设计的，采用通俗易懂的方法讲解计算机的基础理论、常用技术及应用。

本套教材的内容源自致力于教学与科研一线的骨干教师与资深专家的实践经验和研究成果，融合了先进的教学理念，涵盖了计算机领域的核心理论和最新的应用技术，真正在教材体系、内容和方法上做到了创新。同时本套教材根据实际需要配有电子教案、实验指导或多媒体光盘等教学资源，实现了教材的“立体化”配套。本套教材将随着计算机技术的进步和计算机应用领域的扩展而及时改版，并及时吸纳新兴课程和特色课程的教材。我们将努力把这套教材打造成为国家级或省部级精品教材，为高等院校的计算机教育提供更好的服务。

对于本套教材的组织出版工作，希望计算机教育界的专家和老师们能提出宝贵的意见和建议。衷心感谢计算机教育工作者和广大读者的支持与帮助！

机械工业出版社

前 言

今天，我们还有什么方面离得开计算机，离得开软件呢？无论在办公室还是在家里，我们通过网络查收 E-mail，看新闻，听音乐，在线聊天，上微博；我们开车需要电子导航系统；我们的手机，除了通信之外，还运行各种各样的软件，2011 年 6 月苹果公司宣布其 iPhone App Store 应用软件下载次数超过 140 亿次……

中国软件行业的发展速度可以用惊人来形容，据赛迪网 (http://news.ccidnet.com/art/945/20100602/2076169_1.html) 消息：中国计算机报社承办的 2010 第十四届中国国际软件博览会于 2010 年 6 月 2 日在北京展览馆召开，工业和信息化部部长李毅中在演讲中表示：我国软件服务业 2009 年业务收入为 9970 亿元，接近一万亿元，是 2000 年的 16 倍；软件出口当年是 196 亿美元，是 2000 年的 49 倍。而且他还透露，工信部将出台进一步鼓励软件产业发展的政策。

从 2000 年开始，中国政府就出台《鼓励软件产业和集成电路产业发展的若干政策》文件，给予软件企业较大幅度的税收优惠，将企业 17% 的增值税降为 3%，促使中国软件产业保持每年 38% 的迅猛增长速度。

尽管中国的软件行业发展是惊人的，但是依然存在很多问题，比如中国很少能开发出世界范围内通用的基础性软件，如操作系统、应用于科学计算的 MATLAB、应用于虚拟仪器的 LabView 等软件，而且即使开发出了同类产品，在性能和质量上也无法和国外大公司开发的产品相比较。

为什么会出现这种情况呢？一方面是由于中国软件公司的规模一般比不上国外软件公司，比如，截至 2010 年 6 月 30 日，微软公司的全球员工大约有 8.8 万人（经历 35 年），Google 公司的员工也到达了 2.2 万人（经历 12 年），而中国最大的软件公司东软的员工为 1.7 万人（经历 19 年），由此可以看出，中国的软件公司和国外相比在规模上还存在不小的差距。我们知道，中国经济近 20 年来的强劲发展，与中国企业的规模化经营是分不开的，只有达到规模化经营，才能在开发投入、产品质量提升、成本下降等方面起到推动作用。除了规模之外，人力资源结构不合理也是制约国内软件业发展的一个关键因素。软件开发需要各种不同层次的人才：商业分析人员、项目领导人员、知识过程专业技术人员、数据库分析人员、系统分析人员、网络分析设计人员、质量保证专业技术人员以及编程人员、文档和培训人员、测试工程师等。但是，中国很多软件公司的人才结构都不合理，很多公司仅重视软件的开发，而对于前期的设计和后期的测试则较为轻视，这种结构不合理必然会导致开发不出高质量的优秀软件。

软件测试作为软件开发过程中的重要一环，长期以来一直滞后于中国软件产业的发展。这一方面是由于小规模的企业不懂得或不重视软件测试，另一方面则是由于国内软件测试专业人员的缺失。据国内专业的软件测试培训机构调查，2010 年前，国内 120 万软件从业人员中，真正能担当软件测试职位的不超过 5 万人，软件测试人才缺口高达 30 多万人，同时还存在持续上升趋势，因此对软件测试人员的培训已经成为各个公司、高等院校等机构和单位的重要责任。

本书作者在高校承担软件测试课程讲解多年，发觉市面上鲜有一本能够致力于引领软件测试人员从无到有地全面掌握软件测试知识的书籍。很多软件测试书籍仅致力于阐述某种软件测试技术，比如黑盒测试；而有些书籍则详略不明，不能使人纵观软件测试的全貌，于是萌生了写一本软件测试书籍的想法，一方面可以作为高校及其他软件培训机构进行软件测试培训的教材使用；另一方面也为那些对软件测试感兴趣的各类人员提供指导。

本书是一本入门级的软件测试书籍，虽然是入门级的书籍，但内容非常丰富、完整，从软件测试的历史、各种测试的基本概念、各种测试模型与技术、软件测试管理的整个流程以及软件测试自动化及工具等方面对软件测试进行了全面的介绍，并且引入了与软件测试相近的软件维护的部分内容，系统地阐明了软件测试的全貌。可以说，该书基本上包含了软件测试所有主要方面的内容，既是一把打开软件测试大门的金钥匙，又是进入更高级软件测试的重要基石。

本书具有以下 5 个方面的特征。

1. 该书内容非常完整、丰富，涵盖了从测试概念、测试原则、测试技术、测试管理以及测试应用等知识，便于学生全面地掌握软件测试的整个知识体系；
2. 参考了众多的软件测试国际标准，比如 IEEE Std610，IEEE Std829，IEEE 1028 等，力求做到概念和原理讲解的精准；
3. 与国际软件测试认证委员会（ISTQB）软件测试初级认证大纲的要求相一致，便于学生学到的知识与国际接轨；
4. 该书力求循序渐进地讲解知识，不仅要学生学习到软件测试的知识，更重要的是通过背景的讲解、例子的讲解让学生理解和领会到学习的知识，并融会贯通；
5. 该书首次将部分软件维护的知识加入到软件测试之中，满足了全国示范性软件学院开设软件测试与维护课程的教材需要。

本书可作为高等院校计算机学院或软件学院软件测试相关课程教材，也可作为软件测试培训机构的教材，还可作为软件测试入门者自学的教材。总之，该书将引领致力于软件测试的人员进入到一个如此丰富的软件测试领域，为将来的软件测试工作提供帮助。

本书中的例子均采用 C 语言编写，由于是程序片断，且不涉及过多的编程细节，由于 C 语言语法本身的通用性，即使只学习过 Java 语言的程序员也可以轻松理解例子所表达的含义。

本书中，我们并没有严格区分程序与软件、缺陷与错误、过程与流程、测试案例与测试用例等术语，如果没有特别说明，则视为相同。

本书第 1~7 章和第 13~17 章由黄武编写，第 18、22 和 23 章由洪玫编写，第 19~21 章和第 24、25 章由杨秋辉编写，第 8~12 章由余静编写。

非常感谢整个编写小组成员的辛勤努力工作，也感谢帮助过本书编写的其他同志，最后还要感谢我的妻子刘玫和儿子黄瀚钦对我工作的支持。

由于本人知识水平有限，本书中难免存在错误或不当的地方，欢迎各位读者批评指正。

黄武
2011. 11

目 录

出版说明	
前言	
第一部分 软件测试概述	
第 1 章 软件测试的历史	2
1.1 最早的计算机程序员和最早发现的计算机 Bug	2
1.2 软件测试的发展历程	3
1.2.1 面向调试的测试 (1956 年前)	3
1.2.2 面向证明的测试 (1957 ~ 1978)	4
1.2.3 面向查错的测试 (1979 ~ 1982)	5
1.2.4 面向评估的测试 (1983 ~ 1987)	5
1.2.5 面向预防的测试 (1988 ~ 2000)	6
1.3 小结	7
习题 1	7
第 2 章 软件测试在软件工程中的地位	8
2.1 为什么引入软件工程	8
2.2 软件测试在软件工程中的位置	9
2.3 软件测试在软件工程知识体系中的内容	10
2.3.1 软件测试基础	11
2.3.2 软件测试级别	11
2.3.3 软件测试技术	12
2.3.4 测试相关的度量	12
2.3.5 测试过程	13
2.4 小结	13
习题 2	14
第 3 章 软件测试基础	15
3.1 为什么要引入软件测试	15
3.1.1 引发软件错误的原因	15
3.1.2 软件故障造成的危害	19
3.1.3 引入软件测试的真正原因	20
3.2 什么是软件测试	21
3.2.1 软件测试的概念	22
3.2.2 软件测试的分类	23
3.2.3 软件测试的成熟度	23
3.2.4 软件测试活动及其目标	23
3.3 软件测试相关的术语	24
3.4 软件测试的基本原则	26
3.5 软件测试的心理学	30
3.5.1 确立正确的测试目标	30
3.5.2 自行测试和独立测试之间的平衡	31
3.5.3 测试员和程序员的不同心理	32
3.5.4 良好的沟通在测试中起到积极的作用	32
3.6 软件测试的经济学	32
3.6.1 制定的测试工作量要恰当	32
3.6.2 尽早的发现缺陷, 尽早地修复缺陷	33
3.7 软件测试的基本模型	34
3.7.1 软件测试的 V 模型	34
3.7.2 软件测试的 W 模型	35
3.7.3 软件测试的 H 模型	36
3.7.4 软件测试的 X 模型	36
3.8 软件测试与软件质量	37
3.8.1 软件质量的定义和常见的软件质量模型	37
3.8.2 软件测试和软件质量是不同层级的概念	38
3.8.3 软件测试有利于提高软件质量	38
3.9 软件测试员应该具备的基本素质	39
3.10 小结	39
习题 3	40

第二部分 开始软件测试

第4章 建立软件测试系统	42
4.1 最简单的软件测试过程	42
4.2 完整的软件测试系统	44
4.3 完整的软件测试流程	46
4.3.1 一种实用的软件测试流程	46
4.3.2 Perry的7步软件测试流程	47
4.4 小结	50
习题4	51

第三部分 软件测试基本技术

第5章 静态测试	54
5.1 静态测试概述	54
5.1.1 为什么需要静态测试	54
5.1.2 静态测试的重要性	55
5.2 评审	56
5.2.1 评审成功的因素	57
5.2.2 评审的基本术语	57
5.2.3 评审的软件产品	57
5.2.4 评审的分类	59
5.2.5 非正式评审	59
5.3 正式评审	60
5.3.1 正式评审的最小可接受条件	60
5.3.2 正式评审的角色	61
5.3.3 正式评审的基本过程	62
5.4 管理评审	63
5.4.1 管理评审的目的和内容	63
5.4.2 管理评审的团队	63
5.4.3 管理评审过程	63
5.4.4 输出	64
5.4.5 举例	65
5.5 技术评审	66
5.5.1 技术评审的目的和内容	66
5.5.2 技术评审团队	66
5.5.3 技术评审会议	66
5.6 审查	67
5.6.1 审查的目的和内容	67
5.6.2 审查团队	67

5.6.3 审查的前提条件	68
5.6.4 审查会议过程	69
5.6.5 审查输出	71
5.6.6 数据收集	71
5.6.7 审查的检查表	71
5.6.8 审查的注意事项	72
5.7 走查	72
5.7.1 走查的目的和内容	72
5.7.2 走查团队	73
5.7.3 走查会议	73
5.7.4 走查与审查	73
5.8 审计	74
5.8.1 审计的目的和内容	74
5.8.2 审计团队	74
5.8.3 审计会议过程	75
5.9 5种正式评审的比较	76
5.10 小结	77
习题5	77
第6章 白盒测试(基于结构的测试)	78
6.1 白盒测试概述	78
6.1.1 白盒测试定义	78
6.1.2 为什么要进行白盒测试	78
6.1.3 白盒测试的分类	80
6.1.4 白盒测试的适应范围	80
6.1.5 白盒测试过程	81
6.1.6 白盒测试的优缺点	82
6.2 基于数据流的白盒测试	82
6.3 基于控制流的白盒测试	85
6.3.1 控制流图	85
6.3.2 基于控制流的几种白盒测试方式	89
6.3.3 几种基于控制流的白盒测试技术	91
6.3.4 基本路径测试(McCabe圈覆盖)	98
6.3.5 测试覆盖准则	103
6.4 其他白盒测试技术	106
6.4.1 程序插桩	106

6.4.2	域测试	106	7.8	基于经验的测试技术	131
6.4.3	符号测试	106	7.8.1	基于经验测试技术概述	131
6.4.4	Z 路径测试	106	7.8.2	错误推测	132
6.4.5	程序变异	107	7.8.3	探索性测试	133
6.5	小结	107	7.8.4	基于经验测试技术的总结	134
	习题 6	107	7.9	小结	134
第 7 章	黑盒测试 (基于规格说明的测试)	109		习题 7	135
7.1	黑盒测试概述	109	第四部分 软件测试级别		
7.1.1	黑盒测试定义	109	第 8 章	单元测试	138
7.1.2	黑盒测试和白盒测试之间的关系	109	8.1	单元测试概述	138
7.1.3	黑盒测试方法的有效性 & 方法选择的指导方针	113	8.2	单元测试现状和作用	139
7.1.4	黑盒测试的优缺点	113	8.3	单元测试的方法	140
7.2	等价划分	114	8.3.1	白盒方法的单元测试	140
7.2.1	为什么要引入等价划分	114	8.3.2	黑盒方法的单元测试	141
7.2.2	等价划分的概念	114	8.3.3	测试驱动模块与桩模块	142
7.2.3	等价划分的依据	115	8.4	单元测试工具 JUnit 简介	142
7.3	边界值测试	117	8.4.1	JUnit 框架组成	143
7.3.1	边界值测试概述	117	8.4.2	利用 JUnit 进行单元测试的步骤	146
7.3.2	边界值测试的步骤	118	8.4.3	Eclipse 中 JUnit 的使用	148
7.3.3	数值型变量的边界测试	118	8.5	小结	149
7.3.4	字符串变量的边界测试	119		习题 8	149
7.3.5	特殊等价类的测试	120	第 9 章	集成测试	150
7.4	决策表测试	121	9.1	集成测试概述	150
7.4.1	决策表概述	121	9.2	结构化软件的集成测试	151
7.4.2	决策表测试方法	122	9.2.1	自顶向下集成测试	152
7.5	因果图	124	9.2.2	自底向上集成测试	153
7.5.1	因果图概述	124	9.2.3	核心系统先行集成测试	154
7.5.2	因果图测试方法	125	9.2.4	高频集成测试	154
7.6	场景或用例测试	127	9.3	小结	155
7.6.1	采用场景或用例测试方法的原因	127		习题 9	155
7.6.2	按照场景来设计测试用例	128	第 10 章	系统测试	156
7.7	状态转换图测试	129	10.1	系统测试概述	156
7.7.1	状态转换图概述	129	10.2	功能性测试	158
7.7.2	利用状态转换图进行测试案例的设计	130	10.2.1	正常功能测试	158
			10.2.2	健壮性测试	158
			10.3	非功能性测试	159
			10.3.1	大数据量测试	159

10.3.2 负荷测试	160	13.9 小结	184
10.3.3 可用性测试	160	习题 13	184
10.3.4 文档测试	162	第 14 章 测试计划	185
10.4 小结	163	14.1 测试计划概述	185
习题 10	163	14.1.1 为什么要引入测试计划	185
第 11 章 确认测试	164	14.1.2 测试计划的目的	185
11.1 确认测试概述	164	14.2 测试计划的相关术语	185
11.2 α 测试	165	14.3 测试计划的多样性	186
11.3 β 测试	166	14.4 测试计划的注意事项	186
11.4 小结	166	14.5 测试计划的内容	186
习题 11	167	14.5.1 测试计划标识符	187
第 12 章 回归测试	168	14.5.2 简介	187
12.1 回归测试概述	168	14.5.3 测试项	187
12.2 回归测试的几种策略	168	14.5.4 被测测试的特征	188
12.2.1 测试用例库的维护	169	14.5.5 不被测试的特征	189
12.2.2 回归测试集的选择	169	14.5.6 测试策略(方法)	189
12.2.3 回归测试的基本过程	170	14.5.7 测试出口准则	189
12.3 回归测试与自动化	170	14.5.8 挂起准则以及重启要求	189
12.4 小结	171	14.5.9 测试交付文档	190
习题 12	172	14.5.10 测试任务	190
		14.5.11 测试团队的责任	190
		14.5.12 环境需求	191
		14.5.13 进度	192
		14.5.14 风险和应变计划	192
		14.5.15 批准	192
		14.6 小结	192
		习题 14	193
		第 15 章 测试设计	194
		15.1 测试设计概述	194
		15.2 测试设计与测试计划之间的	
		关系	194
		15.3 测试设计的目标	196
		15.4 测试设计说明	196
		15.4.1 目的	196
		15.4.2 内容	196
		15.5 测试案例说明	197
		15.5.1 目的	197
		15.5.2 内容	198
		15.6 测试过程说明	199

第五部分 软件测试管理

第 13 章 构建测试环境(测试	
准备)	174
13.1 测试环境概述	174
13.2 测试环境要素	174
13.3 测试的组织结构	175
13.3.1 测试组织的独立性	175
13.3.2 测试组织的几种结构	175
13.3.3 与测试工作相关的人员	177
13.4 构建标准的测试流程	177
13.5 构建标准的测试文档	179
13.6 测试人员的培训	180
13.6.1 测试员能力的评估	180
13.6.2 CSTE CBOK 公共知识体系	181
13.7 测试工具	182
13.8 测试的其他管理	183
13.8.1 配置管理	183
13.8.2 风险管理	183

第一部分 软件测试概述

在日常生活、工作中，测试工作无处不在，当我们搬入新房时，我们会对照房屋开发商给出的图样测量房屋的面积是否如他们宣传那样；当我们购买了新的汽车，要发动起来跑一跑，检查一下灯光、音响等；当我们买了一部新的手机，首先用它来给朋友打个电话，看看能否正常通话，通话的音质如何等。实际上，这些日常生活中的各种测试与我们的软件测试有很多相似之处，或者说这些测试属于软件测试的一个阶段（确认测试）。对于上述的任何一项检查失败，我们都会抱怨制造商，为什么生产如此低劣质量的商品！是的，我们的测试是可以发现商品缺陷的。

软件测试来源于我们对日常生活中各种商品的测试，如图 1 所示，只不过软件测试只与软件相关，从某种角度来说它和对一件具体事物（如汽车）进行测试是相似的，换句话讲，我们可以用日常生活中的测试经验进行类比以促进软件测试的学习、理解和发展。

除了经常对日常生活用品进行测试之外，我们也经常自觉不自觉地对日常使用的各种软件进行测试，比如我们常常听到某人抱怨程序运行得慢，网速慢（性能测试），软件死机（可靠性测试），说某个浏览器不能打开某些网页（兼容性测试）等等。

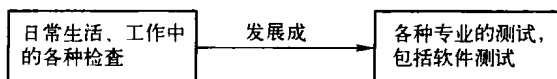


图 1 软件测试并不神秘，其来源于日常生活中的各种检查

的确，各种软件测试的工作就在身边，是否这就说明我们已经学会了软件测试呢？当然不是，要成为一个优秀的软件测试人员，需要系统地、理论地和规范地学习软件测试的概念、技术及流程，才能使经过正规测试的软件产品将错误降到最低，从而减少软件应用在用户那里的抱怨。

下面就从软件测试的历史、软件缺陷对于软件的影响等方面开始认识软件测试。

第 1 章 软件测试的历史

1.1 最早的计算机程序员和最早发现的计算机 Bug

与人类很多发明，比如中国的四大发明的历史相比，编写软件以及软件测试的历史是非常短的。软件测试的历史是与计算机技术的发展、软件的编写以及软件缺陷的发现分不开的，从 19 世纪 40 年代中期电子计算机发明至今，软件测试仅仅有 60 多年的历史。

如果我们编写的程序从来都没有出现过与我们想象不一致的地方，软件测试也就永远没有用武之地。计算机软件的编写和软件缺陷的发现与人类历史上两位伟大的女性是分不开的。

1842 年英国数学家、哲学家和发明家卡洛斯·巴比奇 (Charles Babbage 1791 ~ 1871) 发明了他的分析计算机，作为巴比奇合作伙伴的阿达·洛甫雷斯 (Ada Lovelace 1815 ~ 1852) 于 1842 年根据巴比奇的要求在分析机上翻译了意大利数学家 Menabrea 的文章。在阿达留下的笔记字母 G 一章中记录了我为巴贝奇分析机设计的一个求解伯努利数 (Bernoulli number) 的算法，她甚至还建立了循环和子程序的概念。由于她在程序设计上的开创性工作，Ada Lovelace 被称为世界上第一位程序员，后来美国军方开发的 Ada 编程语言正是为了纪念这位伟大的女性。

图 1-1 所示为世界上最早的计算机和程序员。

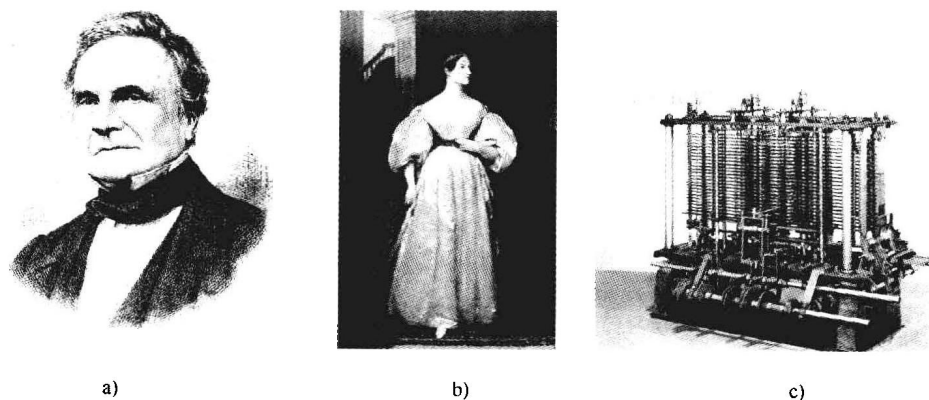


图 1-1 世界上最早的计算机和程序员 (来源于 Wikipedia 网站)

a) Charles Babbage b) Ada Lovelace c) 巴比奇分析机

由于巴比奇计算机从来没有真正运转起来，Ada 的程序也没有执行过。在其后的 100 多年时间中，都没有出现过真正可以实用的计算机及其软件，因此软件缺陷也没有出现过，当然也就没有软件测试。

直到 1946 年，当美国计算机科学家格蕾丝·霍珀 (Grace Hopper 1906 ~ 1992) 在哈

佛大学最早发明的电子计算机之一 Mark II 上工作时，机器突然停止工作，查找原因结果是一只飞蛾掉在了机器的继电器上，清除该只飞蛾后 MarkII 机器工作正常，她在工作日志上粘贴了这只蛾子，并写下术语 Bug。从此以后，计算机的缺陷统统被称为 Bug，而清除缺陷的工作则被称为 Debug，这就是最初软件测试的来源。Hopper 不仅提出了 Bug 的概念，她还是世界上第一位设计编译器的计算机科学家，并且提出了独立于机器的编程语言的概念性构想，最终导致早期的现代编程语言 COBOL 的诞生（1959 年），这是计算机科学界的另一位伟大的女性。

图 1-2 为 Hopper 和她发现的第一个计算机 Bug 的笔记。

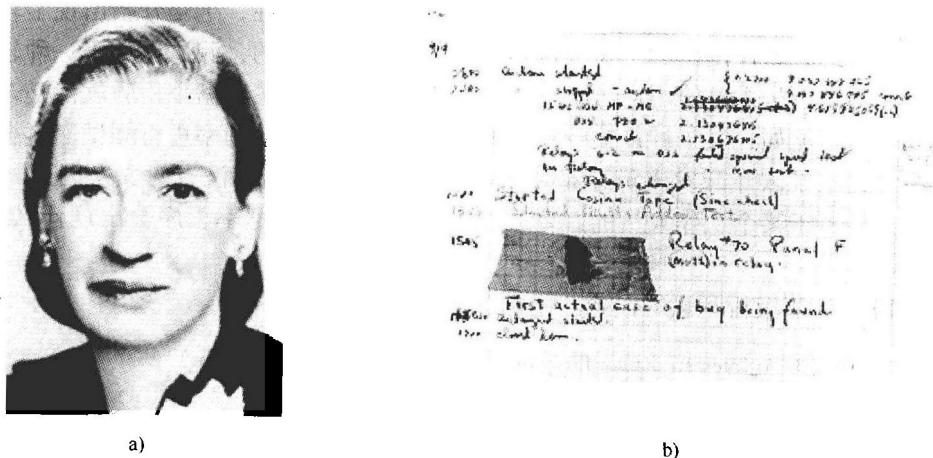


图 1-2 Grace. Hopper 和她发现的第一个计算机 Bug（来源于 Wikipedia 网站）

a) Grace. Hopper b) Hopper 发现第一个 Bug 的笔记

1.2 软件测试的发展历程

1988 年，Dave Gelperin 和 William C. Hetzel 将软件测试的历史按照时间分为了 5 个阶段，这与 Boris Beizer 在《软件测试技术》一书中将软件测试成熟度分为 5 个等级是一致的，参见表 1-1。

表 1-1 软件测试的历史年代划分和目的

序号	年代	目的
1	1956 年之前	面向调试的测试，测试和调试没有清晰的区分
2	1957 ~ 1978	面向证明的测试，证明程序可以运行
3	1979 ~ 1982	面向查错的测试，证明程序会出现错误
4	1982 ~ 1987	面向评估的测试，仅对软件的风险进行评估
5	1988 ~ 2000	面向预防的测试，防止软件不要出现错误

1.2.1 面向调试的测试（1956 年前）

在 1956 年之前，真正可用的高级编程语言还很少，当时所有可用的程序几乎都使用机

器语言和汇编语言来开发，发现软件错误和修改错误的是同样的编程人员。那个时候是不区分软件调试和测试的，既没有系统的测试理论出现，也没有专人从事发现软件错误的测试工作。软件设计人员仅在机器出现不正常状态时去找问题，然后修复它。

1.2.2 面向证明的测试 (1957 ~ 1978)

在这一阶段，软件测试和调试之间已经开始分家。从1957年开始，高级语言逐渐应用到计算机软件编程中，软件编程工作变得相对容易，但是由于开发软件本身复杂度的上升，又使得软件编写在时间、质量上变得不易控制，容易出现错误，直到爆发了软件危机。

这段时间，计算机科学家逐渐认识到软件检查工作的重要性，于是开始出现了软件测试的理论。但是，此时软件测试的主要工作是证明软件可以使用的，测试还处于一种软件开发的辅助状态，测试人员主要使用计算机软件可以正常运行的合法数据来检验软件是否能够正常地完成其工作。比如，对于计算银行利息的软件，只使用正的利率进行测试而不会使用负的利率，因为无论是软件开发人员还是使用人员都会说负的利率没有意义，因此不需要验证。但是，如果由于某个工作人员的好奇或者错误输入了负的利率程序会出现什么状况呢？参见例1-1。

【例1-1】 计算银行利息的函数，不加保护的程序。

```
double Get_Bank_Interest(double fPrincipal, double fBank_Rate, int iYear)
{
    double fTotal = fPrincipal × (1 + fBank_Rate)iYear;
    return(fTotal - fPrincipal);
}
```

表1-2 所列为1957 ~ 1978年对于软件测试有影响的事件。

表1-2 1957 ~ 1978年对于软件测试有影响的事件

序号	时间	事件
1	1961.1	Herbert Leeds 和 Jerry Weinberg 编写的《计算机编程基础》一书中描述有软件测试
2	1967	Herm Schiller 创造了第一个软件代码覆盖监视器，被称为 Memmap，它用于支持 IBM 360/370 汇编语言
3	1968	第一次提出了软件工程和结构化编程的概念
4	1969	Richard Bender 和 Earl Pottorf 创建了第一个静态和动态的使用数据流分析，以提高测试覆盖率的工具，它把 Memmap 中基于代码的语句和分支测试覆盖率提高了25%
5	1973	Elmendorf 和 William R. 介绍了软件功能测试中的因果图，Elmendorf 还是创立了基于边界分析的等价类测试的第一人，该理论基础奠定了动态黑盒测试的基础
6	1975	Hamlet R. G. 发表了基于编译器的系统测试，为编译器检查软件缺陷奠定了基础
7	1976	Fagan 和 Michael 发表了文章《设计和代码审查以减少程序开发中的错误》，提出了代码审查的过程
8	1977	基于需求的测试被引入

1.2.3 面向查错的测试 (1979 ~ 1982)

经过了面向证明测试的漫长时代后, 软件人员设计和开发程序的逻辑越来越严密, 不仅要考虑程序正常状态下的运行情况, 也要考虑程序在各种错误操作和数据下的承受能力, 从这个意义上说, 软件测试促进了程序质量的提高, 参见例 1-2。

【例 1-2】 计算银行利息的函数, 加保护的程序。

```
double Get_Bank_Interest(double fPrincipal, double fBank_Rate, int iYear)
{
    if(fPrincipal < 0.0)                /// 对本金加以保护, 必须大于或等于 0
    {
        return(0);
    }
    if(fBank_Rate < 0.0)              /// 对利率加以保护, 必须大于或等于 0
    {
        return(0);
    }
    if(iYear <= 0)                    /// 对年加以保护, 必须大于 0
    {
        return(0);
    }
    double fTotal = fPrincipal * (1 + fBank_Rate)iYear;
    return(fTotal - fPrincipal);
}
```

但是这一阶段对于软件测试的理解并不太成熟, 往往过分强调找到软件中的错误, 这是对软件测试目的的迷惑。实际上, 对于不同程序其错误的影响会有比较大的差异, 对于少量人员使用的小型程序而言, 并非一定要花巨大代价找出所有软件错误, 这属于测试经济学的范畴。

表 1-3 所列为 1979 ~ 1982 年对于软件测试有影响的事件。

表 1-3 1979 ~ 1982 年对于软件测试有影响的事件

序号	时间	事件
1	1979	Philip Crosby 发表他的著作《质量是免费的》
2	1979	Glenford J. Myers 发表了他对软件测试有影响的著作《软件测试的艺术》, 对软件测试的原则进行了深入阐述

1.2.4 面向评估的测试 (1983 ~ 1987)

1983 年, Boris Beizer 发表了他关于软件测试的代表作《软件测试技术》一书, 在这本书中他将软件测试的成熟度划分为 5 个等级, 其中第 4 个等级即面向评估的测试, 测试的目的不是为了证明软件的对错, 而是将可观察到的软件缺陷减少到一个可以接受的程度, 因此需要我们测试各种输入数据的有效和无效组合。在这种测试下, 测试员为程序员提供软件