软件质量保证与软件测试技术

RUANJIAN ZHILIANG BAOZHENG YU RUANJIAN CESHI JISHU

马海云 张少刚 著

国防工业出版社
National Defense Industry Press

软件质量保证与软件测试技术

马海云　张少刚　著

国防工业出版社

内容简介

软件质量保证与软件测试技术在过去的几十年中一直是软件开发的重要课题，基于数学知识的软件质量与软件测试方面的研究更是一个全新的研究领域。

本书收集了作者在这方面的多篇论文，也收集了这一领域知名专家的研究成果，并对这些成果进行再探索，形成了自己的见解，内容包括：绪论，主要包括软件危机、软件工程的基本概念，软件质量、软件测试技术的国内外研究现状及发展趋势；软件生命周期及软件开发过程的研究现状；软件质量保证方法分析；软件质量管理；软件测试的基本概念及测试技术探索，主要包括软件可靠性测试的基本概念及常用方法；蒙特卡罗方法和马尔可夫链；蒙特卡罗方法和马尔可夫链模型在软件可靠性测试中的应用；测试策略问题的讨论；网络安全技术的背景与探索。

本书结构合理，集中讲述了在软件质量保证与软件测试技术方面的探索成果，适合从事软件质量保证与软件测试的技术人员与有关院校师生学习参考。

图书在版编目(CIP)数据

软件质量保证与软件测试技术/马海云，张少刚著. —
北京：国防工业出版社，2011.6
ISBN 978-7-118-07507-6

Ⅰ.①软… Ⅱ.①马…②张… Ⅲ.①软件质量—
质量管理②软件—测试 Ⅳ.①TP311.5

中国版本图书馆 CIP 数据核字(2011)第 107737 号

※

国防工业出版社出版发行
（北京市海淀区紫竹院南路 23 号 邮政编码 100048）
腾飞印务有限公司印刷
新华书店经售

*

开本 787×1092 1/16 印张 19½ 字数 446 千字
2011 年 6 月第 1 版第 1 次印刷 印数 1—4000 册 定价 38.00 元

（本书如有印装错误，我社负责调换）

国防书店：(010)68428422 发行邮购：(010)68414474
发行传真：(010)68411535 发行业务：(010)68472764

前　言

随着计算机应用日益普及和深化，计算机软件的数量以惊人的速度急剧膨胀，软件在现代社会中的确是必不可少的，软件不能出错。软件测试在软件生存周期中占有重要地位，而且它直接影响着软件的质量，是保证软件可靠性的重要方法之一。目前，简化测试难度、提高测试准确性的软件测试方法和保证软件质量的研究已经成为新的发展方向。因此，软件质量与软件测试技术方面的知识市场需求很大。就目前软件工程发展的状况而言，软件质量保证与软件测试技术又是相对于其他研究方向而言较为薄弱的一个方面，目前该领域还处于百家争鸣的阶段，尚未形成一套较为成熟与完善的理论，尤其是单一学科封闭式研究多，真正意义上的多学科交叉与综合集成研究只有国外在进行。

本书既有软件质量领域的理论与实践，又有软件测试领域的理论与实践，从整个软件生命周期的角度，把软件质量保证与软件测试结合起来，实现软件质量的提升。第1章 绪论，本章主要包括软件的基本概念及发展，软件危机、软件工程的基本概念，软件质量、软件测试技术的国内外研究现状及发展趋势。第2章 软件生命周期与软件过程的研究现状，本章介绍软件工程中两个重要概念：软件生命周期与软件过程。在早期的软件工程中，这两个概念常不加区分地使用，以后演变为既相互联系又有重要差异的不同概念，本章对它们及相关概念加以说明。第3章 软件质量保证方法分析。第4章 软件质量管理，本章对传统质量保证模型进行了总结分析并进行了改进，在软件生命过程、开发软件的估算、影响软件质量等概念的量化方法方面作了分解分析，对现存的几种软件质量度量标准进行了分析与比较。第5章 软件测试的基本概念及测试技术探索，本章介绍常用软件测试方法和当今软件测试的前沿理论及常用的测试工具。第6章 蒙特卡罗方法及马尔可夫链，本章介绍了蒙特卡罗方法及马尔可夫链的基本概念，对蒙特卡罗方法和马尔可夫链在统计分析方面的应用进行了研究。第7章 蒙特卡罗方法和马尔可夫链模型在软件可靠性测试中的应用，本章提出了基于马尔可夫链模型的软件可靠性测试结果评判准则，并把该准则应用于对软件测试结果的分析中；还提出了基于蒙特卡罗方法的软件可靠性测试模型，并结合评判准则把该测试模型应用于案例中；同时探索出几种测试用例的生成方法和几种软件系统可靠性的优化策略，并通过实例证明其有效性和实用性。第8章 测试策略问题的讨论，本章详细地探讨了测试策略的问题，考虑了达到主要测试目标最可能需要的步骤，以一种有序的和有效的方法来发现并纠正错误。第9章 网络安全技术的背景与探索，本章主要介绍与网络安全有关的网络安全概念、网络黑客与网络防御、数据加密、防火墙技术等，并在网络入侵检测方面作了些探索。

本书对于刚进入信息技术领域的软件质量保证人员和软件测试人员具有理论的指导意义和实践的借鉴意义。对于有一定工作经验的人士来说，本书知识面广，也是一本提升境界、扩展思路的宝典。

本书第1、6、7、8、9章以及书稿篇目设计和统稿由马海云老师完成；第2、3、4、5章由张少刚老师完成。

本书得到天水师范学院科研处，教务处资助。

由于作者水平有限，书中不足之处恳请同行、专家及读者批评指正。

作 者

2011年5月

目　录

第1章 绪 论

随着计算机应用日益普及和深化，计算机软件的数量以惊人的速度急剧膨胀，而且现代软件的规模往往十分庞大，包含数百万行代码，耗资几十亿美元，花费几千人一年的劳动开发出来的软件产品，现在已经屡见不鲜了。例如，Windows 3.1 约有250 万行代码，而现在被广泛使用的 Windows XP 的开发历时三年，代码约有 4000 万行，耗资 50 亿美元，仅产品促销就花费了 2.5 亿美元。计算机软件已经成为一种驱动力。它是进行商业决策的引擎；它是现代科学研究和工程问题解决的基础；它也是区分现代产品和服务的关键因素。它在各种类型的系统中应用，如交通、医药、通信、军事、产业化过程、娱乐、办公……难以穷举。软件在现代社会中的确是必不可少的。软件将成为从基础教育到基因工程的所有各领域新进展的驱动器。软件不能出错。为了降低软件开发的成本，提高软件的开发效率，20 世纪 60 年代末诞生了一门新的工程学科——软件工程学。

1.1 软件的发展

软件担任着双重角色。它是一种产品，同时又是开发和运行产品的载体。作为一种产品，它表达了由计算机硬件体现的计算潜能。软件是一个信息转换器——产生、管理、获取、修改、显示或转换信息，这些信息可以很简单，如一个单个的位(bit)，或很复杂，如多媒体仿真信息。作为开发运行产品的载体，软件是计算机控制(操作系统)的基础、信息通信(网络)的基础，也是创建和控制其他程序(软件工具和环境)的基础。

早期阶段，了解了很多关于计算机系统的实现，但对于计算机系统工程几乎一无所知。但是那个时期开发的许多计算机系统，不少一直到今天还在使用，并继续发挥着巨大的作用。计算机系统发展的第二阶段跨越了从 20 世纪 60 年代中期到 70 年代末期的十余年。多道程序设计、多用户系统引入了人机交互的新概念。交互技术打开了计算机应用的新世界，以及硬件和软件配合的新层次。实时系统能够从多个源收集、分析和转换数据，从而使得进程的控制和输出的产生以毫秒而不是分钟来进行。在线存储的发展导致了第一代数据库管理系统的出现。

第二阶段还有一个特点就是软件产品的使用和“软件作坊”的出现。软件被开发，使得它们可以在很宽的范围中应用。主机和计算机上的程序能够有数百甚至上千的用户，开始开发各类软件包。表 1 - 1 为计算机发展的四个阶段。

表 1-1　计算机发展的四个阶段

早期阶段	第二阶段	第三阶段	第四阶段
• 面向批处理	• 多用户	• 分布式系统	• 强大的桌面系统
• 有限的分布	• 实时	• 嵌入“智能”	• 面向对象技术
• 自定义软件	• 数据库	• 低成本硬件	• 专家系统
• 软件产品	• 消费者的影响	• 人工神经网络	• 并行计算和网络计算机

随着计算机系统的增多，计算机软件库开始扩展。内部开发的项目产生了上万行的源程序，从外面购买的软件产品加上几千行新代码就可以了。这时，当发现错误时需要纠正所有这些程序（所有这些源代码）；当用户需求发生变化时需要修改；当硬件环境更新时需要适应。这些活动统称为软件维护。在软件维护上所花费的精力开始以惊人的速度消耗资源，并且许多程序的个人化特性使得它们根本不能维护，出现了“软件危机”。

计算机系统发展的第三阶段始于20世纪70年代中期并跨越了整整十年。分布式系统——多台计算机，每一台都在同时执行某些功能，并与其他计算机通信——极大地提高了计算机系统的复杂性。广域网和局域网、高带宽数字通信以及对“即时”数据访问需求的增加都对软件开发者提出了更高的要求。然而，软件仍然继续应用于工业界和学术界，个人应用很少。第三阶段的主要特点是微处理器的出现和广泛应用。微处理器孕育了一系列的智能产品——从汽车到微波炉，从工业机器人到血液诊断设备——但哪一个也没有个人计算机那么重要，在不到十年时间里，计算机真正成为大众化的东西。

计算机系统发展的第四个阶段已经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。由复杂的操作系统控制的强大的桌面机，广域和局域网络，配合以先进的软件应用已成为标准。计算机体系结构迅速地从集中的主机环境转变为分布的用户/服务器（C/S）环境。世界范围的信息网提供了一个基本结构，必须考虑“信息高速公路”和“网际空间连通”的问题。事实上，因特网可以看做是能够被单个用户访问的“软件”。

软件产业在世界经济中不再是无足轻重的。由产业巨子如微软做的一个决定可能会带来成百上千亿美元的风险。随着第四阶段的进展，一些新技术开始涌现。面向对象技术在许多领域中迅速取代了传统软件开发方法。虽然关于“第五代”计算机的预言仍是一个未知数，但是软件开发的“第四代技术”确实改变了软件界开发计算机程序的方式。专家系统和人工智能软件终于从实验室里走了出来，进入了实际应用，解决了现实世界中的大量问题。结合模糊逻辑应用的人工神经网络软件揭示了模式识别和类似人的信息处理能力的可能性。虚拟现实和多媒体系统使得与最终用户的通信可以采用完全不同的方法。“遗传算法”则提供了可以驻留于大型并行生物计算机上的软件的潜在可能性。

20世纪后半叶，硬件性能的极大提高，计算机体系结构的不断变化，内存和硬盘容量的快速增加，以及大量输入/输出设备的多种选择，均促进了更为成熟和更为复杂的基于计算机的软件系统的出现。如果一个系统是成功的，那么这种成熟性和复杂性能够产生出奇迹般的结果，但是它们也给建造这些复杂系统的人员带来很多的问题。在计算机系统的整个发展过程中一直存在着一系列软件相关的问题，而且这些问题还会继续恶化。

（1）硬件的发展一直超过软件，使得建造的软件难以发挥硬件的所有潜能。

(2) 建造新程序的能力远远不能满足人们对新程序的需求,同时开发新程序的速度也不能满足商业和市场的要求。

(3) 计算机的普遍使用已使得社会越来越依赖于可靠的软件。如果软件失败,会造成巨大的经济损失,甚至有可能给人类带来灾难。

(4) 一直在不断努力建造具有高可靠性和高质量的计算机软件。

(5) 拙劣的设计和资源的缺乏使得难以支持和增强已有软件。

为了解决这些问题,整个产业界开始采用软件工程实践。

在计算机发展的早期,计算机系统是采用面向硬件的管理方法来开发的。项目管理者着重于硬件,因为它是系统开发中最大的预算项。为了控制硬件成本,管理者建立了规范的控制和技术的标准:要求在真正开始建造系统之前,进行详尽的分析和设计,度量过程,以发现哪里还可以进一步改进,坚持质量控制和质量保证,设立规程,以管理变化。简言之,管理者应用的控制、方法和工具,可以称为硬件工程,但软件工程概念出现比较晚。

在早期,几乎没有规范化的软件开发方法。程序员往往从试验和错误中积累经验。现在,计算机系统开发成本的分配发生了变化,硬件/软件成本变化趋势如图 1 - 1 所示。

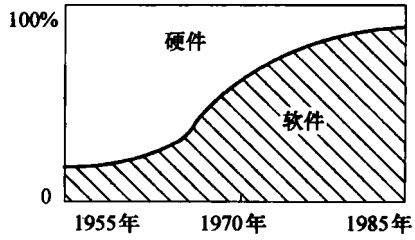

图 1 - 1 硬件/软件成本变化趋势

在近 20 年里,管理者和很多开发人员在不断地探索中,一直思考以下问题:

(1) 为什么需要那么长时间才能结束开发?

(2) 为什么成本如此之高?

(3) 为什么不能在把软件交给用户之前就发现所有的错误?

(4) 为什么在软件开发过程中难以度量其进展?

在回答以上问题的过程中,产生了软件工程学科。

1.2 软件的基本概念

计算机系统由软件和硬件两大部分构成。软件的定义是随着计算机技术的发展而逐步完善的。在 20 世纪 50 年代,人们认为软件就等于程序;60 年代人们认识到软件的开发文档在软件中的作用,提出软件等于程序加文档,但这里的文档仅是指软件开发过程中所涉及的分析、设计、实现、测试、维护等,不包括管理文档;到了 70 年代人们又给软件的定义中加入了数据。因此,软件是计算机系统中与硬件相互依存的一部分,它包括如下内容:

(1) 在运行中能提供所希望的功能与性能的程序;

(2) 使程序能够正确运行的数据及其结构;

(3) 描述软件研制过程和方法所用的文档。

1.2.1 软件的特点

从广义来说,软件与硬件之间是有差别的,了解并理解这种差别对理解软件工程是非常重要的。

1）软件角色的双重性

软件作为一种产品具有双重性,一方面它是一个产品,利用它来表现计算机硬件的计算潜能,无论它是在主机中,还是驻留在设备(如手机)中,软件就是一个信息转换器,可以产生、管理、获取、修改、显示或传送信息。而另一方面它又是产品交付使用的载体,它可以控制计算机(如操作系统),可以实现计算机之间的通信,又可以创建其他程序与控制。

2）软件是被开发或设计的,而不是传统意义上的被制造

一般意义上的产品包括硬件产品总要经过分析、设计、制造、测试等过程,也就是说,要经过一个从无形的设想到一个有形的产品的过程。但软件仅仅是一个逻辑上的产品而不是有形的系统元件,软件是通过人的智力劳动设计开发出来的,而不是制造出来的。而且软件一旦被开发出来,就可以进行大量的复制,因此,其研制成本要远远大于生产成本。这也意味着软件的开发不能像制造产品那样进行管理。

3）软件不会“磨损”,但会退化

一般情况下,有形的硬件产品在使用过程中总会要磨损的。在使用初期,往往磨损比较严重(这实际上是磨合),而经过了一段不长时间的磨合后,将进入相对的稳定期。由于任何硬件产品总有一定的生命周期,随着时间的流逝,由于硬件受到种种不同的损害,硬件的磨损才真正开始,这也意味着硬件的寿命快要到了。硬件的磨损与时间之间的关系可以用图 1-2 所示的“浴缸曲线”来表达。

但对于软件来说,由于软件并不是一种有形的产品,因此也就不存在“磨损”问题,理想情况下,软件的故障曲线如图 1-3 所示。在软件的运行初期,由于未知的错误会引起程序在其生命初期有较高的故障率,然而当修正了这些错误而且也没有引入新的错误后,软件将进入一种比较理想的平稳运行期。这说明软件是不会“磨损”的。但在实际情况中,软件尽管不会“磨损”,可是会退化,如图 1-3 中的实际曲线那样。这是因为软件在其生命周期中会经历多次修改,每次修改都会引入新的错误,而对这些错误又要进行新的修改,使得软件的故障曲线呈现一种锯齿形,导致最后的故障率慢慢升高了,即:软件产生了退化,而这种退化缘于修改。

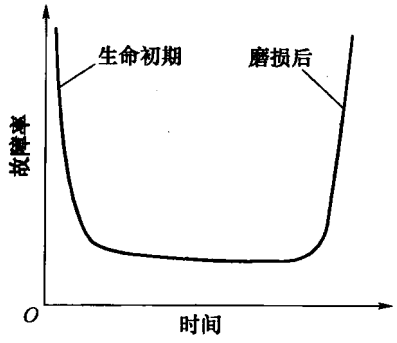

图 1-2 硬件故障率曲线

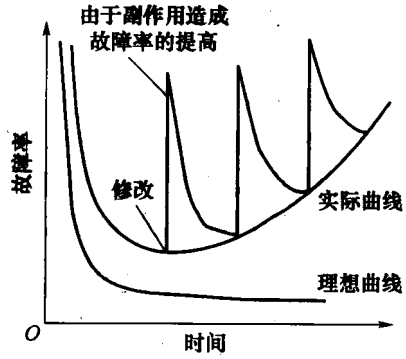

图 1-3 软件故障率曲线

4）绝大多数软件都是定制的且是手工的

在硬件制造业，构件的复用是非常自然的。但由于软件本身的特殊性，构件复用才刚刚起步。理想情况下软件构件应该被设计成能够被复用于不同的程序，尽管今天的面向对象技术、构件技术已经使软件的复用逐渐成为流行，但这种复用还不能做到像硬件产品那样拿来即用，还需要进行必要的定制（构件之间的组合、接口的设计、功能的修改与扩充等），而且软件开发中构件的使用比例也是有限的。整个软件产品的设计基本上还是依赖于人们的智力与手工劳动。

5）开发过程的复杂与费用的昂贵

现代软件的体系结构越来越复杂，规模越来越庞大，所涉及的学科也越来越多，导致了软件的开发过程也异常复杂。靠一个人单枪匹马开发一套软件的时代已经一去不复返了，软件的开发需要一个分工明确、层次合理、组织严密的团队才能完成，显然软件的开发成本也会越来越昂贵。

1.2.2 软件的分类

软件的应用非常广泛，几乎渗透到各行各业。因此，要给出一个科学的、统一的、严格的计算机软件分类标准是不现实也是不可能的，但可以从不同的角度对软件进行适当的分类。常用的分类方法及意义见表1-2。

表1-2 软件的分类

分类序号	分类方法	对应类别	典型应用与特征
1	按功能分类	（1）系统软件	与计算机硬件的接口并为其他程序服务，如操作系统、驱动程序等
		（2）支撑软件	用于开发软件的工具性软件，如开发平台、数据库管理系统、种种工具软件等
		（3）应用软件	为解决某一领域而开发的软件，如商业软件、嵌入式软件、个人计算机软件、Web软件、人工智能等
2	按版权分类	（1）商业软件	版权受法律保护、经授权方可使用且必须购买的软件
		（2）共享软件	与商业软件类似，但可“先尝后买”，其获取途径主要是通过因特网
		（3）自由（免费）软件	无须支付许可证费用便可得到和使用的软件，发行渠道类似于共享软件
		（4）公有领域软件	没有版权，任何人均可以使用而且可以获得源代码的软件
3	按工作方式分类	（1）实时软件	用于及时处理实时发生的事件的软件，如控制、订票系统等
		（2）分时软件	多个联机用户同时使用计算机的软件
		（3）交互式软件	能够实现人机通信的软件
		（4）批处理软件	多个作业或多批数据一次运行，顺序处理的软件
4	按销售方式分类	（1）订制软件	受某个特定的用户委托，在合同的约束下而开发的软件
		（2）产品软件	由软件开发机构开发可以为众多用户服务的，并直接提供给市场的软件