

内 容 简 介

本书从软件工程实际出发,详细介绍了有关软件质量管理、控制、保障及度量的程序、方法和技术。考虑到软件可靠性是软件质量中最重要的因素,本书对软件可靠性模型及其应用、软件可靠性测试、软件可靠性数据收集和处理进行了深入的论述。

全书内容具有理论联系实际、系统、简洁的特点;在软件开发、使用和维护方面,具有较强的工程适用性。

本书保留了第一版的基本内容,继承了第一版的特点和风格。本书对第一版中一些过于浓缩或隐含的内容作了必要的说明,并且加进了以 CMM 和 CMMI 为基础的软件组织或软件开发项目的过程管理等内容,使得本书作为软件工程的一个工程应用读本,具有更好的完整性和工程适应性。

本书可供广大软件项目管理人员、开发人员、系统分析人员及软件客户参考,特别适用于其中的质量管理和质量保障人员;本书也可作为软件学院、软件工程专业的高年级学生、研究生的教材或参考书。

图书在版编目(CIP)数据

软件质量工程/洪伦耀,董卫卫编著. —2 版. —西安:西安电子科技大学出版社,2008.1

ISBN 978-7-5606-1358-1

I. 软… II. ①洪 ②董… III. 软件质量-质量管理 IV. TP311.5

中国版本图书馆 CIP 数据核字(2007)第 182909 号

策 划 戚文艳

责任编辑 阎 彬 戚文艳

出版发行 西安电子科技大学出版社(西安市太白南路 2 号)

电 话 (029)88242885 88201467 邮 编 710071

http://www.xduph.com

E-mail: xdupfxb@pub.xaonline.com

经 销 新华书店

印刷单位 西安文化彩印厂

版 次 2008 年 1 月第 2 版 2008 年 1 月第 2 次印刷

开 本 787 毫米×1092 毫米 1/16 印张 14.625

字 数 335 千字

印 数 4001~8000 册

定 价 22.00 元

ISBN 978-7-5606-1358-1/TP·0721

XDUP 1629012-2

*** 如有印装问题可调换 ***

本社图书封面为激光防伪覆膜,谨防盗版。

第二版前言

2004年在写本书第一版的时候，作者当时的想法是：尽可能地简练，不超过30万字！在这种思想指导下，本书的第一版浓缩、简练有余，以至于许多读者，包括评阅本书的专家、教授以及出版社审读本书的社领导，都觉得本书（第一版）的内容叙述以及行文方式有些像论文报告。作者在当时写的过程中有时也感觉到有些地方应该再阐述得更详细一些，有些内容应该写进去。但是，在上述思想指导下，另一个“我”又会说：算了吧，简练些。这本书就这样定稿、出版了。对这样的一本书，有些读者也发出了另一种声音，他们说：这样的书就应该这样写，从某种意义上说，它起到了“菜单”的作用，它告诉了读者在这个餐桌上（指软件质量工程方面）有多少菜，这些菜各是什么味道，列出了做这些菜的基本方法、步骤，如果你要实实在在地做出这些菜来，那就要找出详细菜谱，而具体的菜谱书中也给出来了（指参考资料），这就够了。这种说法，给多少因前面的批评使作者有些遗憾的心情，带来了些许安慰。

本书这次修订，作者慎重地考虑了上述两方面的意见和要求，作出如下决策：

（1）不改变本书的基本内容和风格。

（2）对有些过于隐含的内容、非得要读者很费劲地深入研究才能搞清楚的内容，特别是有关数据和结论方面的内容，要加以阐述和点明。

（3）有些软件工程中重要的内容要加进去，对这些内容不再吝惜笔墨。

在这个原则指导下，作者在第二版中作了近30处较大的修改和补充，主要是补充。一般的修改和补充这里就不细说了，现将其中补充较多的几处内容，按第一版的章节顺序，作一简单的介绍。

在第2章“软件质量管理”中，考虑到软件产品的验收仍是软件产品开发及其寿命期的重要组成部分，而且这一部分所涉及的问题不但多而且集中，涉及的组织部门和人员不但广而且复杂，无论是软件开发单位还是软件用户单位都极其重视软件的验收工作，因此，在“系统联试阶段”这一节之后，增加了“软件的验收和交付”一节。在这一节中，对软件验收中可能涉及的问题，诸如验收的前提、组织、过程、准则、评审等作了详细的论述。

第3章“软件质量控制”是本书的一个重点章节，内容比较多，篇幅比较长，读者在读完这一章后，可能对这一章的整体构思仍没有清晰的认识，而这一点对于理解和运用本章所介绍的软件质量控制方法、技术等是很重要的。因此，我们在本章的最后增加了一节“全面软件质量控制PDCA方法的总结与回顾”，这将有助于读者从整体上把握这一章的内容。

软件可靠性是软件质量的一个非常重要的指标，也是作者研究比较深入的领域。但当时在写本书的第一版时，考虑到我国当前深入研究和运用软件可靠性及其度量技术的软件企业还比较少，所以将这一部分内容大大压缩了。为了在一定程度上（只能在一定程度上）弥补这一缺陷，在第二版中，作者在第5章“软件可靠性度量”中进行的修改和补充有八处之多。

统观第一版全书，我们可以看到，在“不超过 30 万字”的思想束缚下，本书第一版的全部内容太过于专注软件质量本身的方方面面了，其中涉及到的管理，也只是为改善和提高软件质量所执行的具体技术和方法上的管理，对一个软件组织在开发软件产品中整体上的、全面的过程管理并没有涉及。而软件生产过程理论告诉我们，软件质量以及开发软件的效率(资金的投入和耗费的时间)往往取决于一个开发组织所具有的总体上的软件过程的管理水平和持续改进其过程的能力。在某种意义上可以这样说：一个软件组织的过程管理水平和能力将决定软件产品的一切！因此，在第二版中，在论述了与软件质量有关的方方面面后，我们又增加了一章“软件过程管理”，详细介绍了 CMM 并简单介绍了 CMMI。

本书第二版能这么快地与广大读者见面，是广大读者对本书关爱的结果。这种关爱还表现在第一版出版后的许多方面，在 2004~2007 年的这段日子里，一些读者对作者热情地谈了他们的读后感；一些读者善意地提出了他们的建议与希望；也有一些读者诚恳地提出了他们的批评意见。这一切，对作者是一种莫大的鼓舞和激励。对此，作者表示真挚的感谢！

本书第二版相对于第一版而言，虽然以四年以后更深思熟虑的眼光进行了仔细的审阅，作了进一步的修订和补充，但是不足之处仍然是在所难免的，恳请广大读者一如继往地提出建议和批评，作者将不胜感谢！

作 者

2007 年 11 月

第一版序

随着计算机与因特网的普及和社会信息化进程的加快，计算机软件的地位已发生了巨大的变化，它已成为支撑社会正常运行和发展的重要基础设施。软件变得越来越复杂，人们对软件的要求也越来越高，特别是对软件的可靠性、易用性、应变性和兼容性等各方面都提出了很高的要求。

软件的生产方式也发生了巨大的变化。开始时是手工业生产方式，软件的质量依赖于软件编制者个人的技艺和才智。后来，演变为工业化的规模生产，软件成为软件公司一批开发人员共同协作的产品。这时，软件产品的质量在很大程度上取决于软件公司实施的开发过程的优化程度和具体的管理水平。现在，又进一步扩展为软件生产的社会化。一个软件系统常常要由不同软件专业公司生产的软件构件产品集成而成。于是为保证软件的质量，又要求有相关的统一标准和协议，以及按照标准进行的测试和认证。

软件质量的控制和管理技术是一门实践的技术，它是根据软件开发的大量实践经验积累而来的。这其中有很多成功的经验，也不乏惨痛的失败教训。坦诚地讲，我们国家软件产业发展较晚，目前真正能称得上是有一定规模的，具有大型自主软件开发能力的软件公司还不多，还缺乏这种规模化开发软件的实践经验。因而，人们往往对软件质量管理的重要性和必要性认识不足，理解不深，同时也缺乏相应的量化的实际数据。大家知道，只有在大量数据的基础上才能做到软件开发过程的真正的优化，而数据要靠长期实践的积累。

洪伦耀教授和董云卫高级工程师联合编写的这本《软件质量工程》一书，全面、系统地介绍了国内外流行的软件开发过程中实施的质量管理、控制、度量 and 保障技术。其内容大都是根据他们多年来的实践经验，从国内外较新、较成熟而且比较实用的技术中精选而来的。本书的出版对于指导我国软件产业的实践，改善软件产品的质量，促进我国软件产业的发展肯定会起到非常重要的作用。同时我也期望，伴随着我国软件产业的形成和发展，我国科技人员能在吸收、消化国外的软件开发技术的基础上，通过自身大量软件开发的实践，不断地总结、提高和创新，进一步为我国软件开发技术的发展做出应有的贡献。

郝克刚

2004年1月4日于西安

第一版前言

自从第一台计算机诞生以来，信息技术革命无疑成为当今世界最快的技术变革。今天，计算机的应用已渗透到了人类现代生活的各个方面，人们对它的依赖程度与日俱增，因计算机系统故障而引起危机的可能性也逐渐增大。同时，在计算机的发展与应用中，存在着严重的软/硬件不均衡发展的现象，软件的质量、性能、生产率及耗费等，都无法与硬件同步。随着人们对软件需求的日益增大，软件产品的功能复杂性和结构复杂性越来越高，因软件系统失效造成计算机系统失败和瘫痪的重大事故屡见不鲜。例如：

(1) 在美国宇航局的“旅行者”计划中，天王星探测器就是因为深度太空网络软件发送信息迟缓和能力衰竭而处于危险之中。

(2) 软/硬件干扰问题是使一些航天飞机经常推迟发射时间的主要原因。

(3) 在美国国防部某项工程中，事先设计好的先进程序无一可用，从而导致 AFTI/F-16 的首航时间耽搁了一年。

关键软件系统的失败也影响了大量民用项目和科学工作。例如：

如果不是数据分析程序由于“越界溢出”而隐藏了异常数据，南极上空的臭氧空洞本来可以更早引起科学界的注意。

软件失效也可导致巨大的经济损失。例如：

1990年1月15日，美国一通信中转系统新投入使用的软件发生了错误，导致主干线远程网大规模崩溃；1991年夏季，美国一系列局域电话网由于软件问题而中断。这些严重事故使数以千计依靠电讯公司运营业务的公司损失了巨额资金。

出错的软件还会危及人们的生命安全。例如：

(1) 美国的 Therac-25 放射性治疗仪曾以其安全性著称于世，但有一次由于其软件出错而使其控制系统失灵，导致多名病人失去生命。

(2) 据专家们统计后指出，软件控制系统对飞行员在异常飞行期间发出的紧急查询反应不当，是发生飞行事故以致造成机毁人亡的主要原因。

……

这一切表明：软件中哪怕是很微小的差错，都可能导致生产、生活和其他各项活动中质量事故，如产品报废，商业停顿，甚至是人员伤亡。面对这样一些问题，在要求软件质量越来越高的情况下，应该怎么办呢？答案只有一个：因为软件产品是人类智力活动的产物，所以软件的质量问题，也只有用人类的智力活动来解决！

本书就是考虑到目前人们对高质量软件的迫切需要而撰写的，它集中讨论了与软件产品质量密切相关的质量管理、质量控制、质量度量和质量保障等问题。本书的内容实际上就是人们为了解决软件的质量问题，在总结长期的开发实践经验的基础上逐渐形成的智力成果。另外，从软件产品在质量范畴内区别于硬件产品的特殊性来说，撰写本书还有如下的理由：

(1) 软件质量是在开发过程中构建(需求分析、设计、编码和集成等)入软件产品之中的,开发过程完成之后是加不进去的。

(2) 软件产品是由许多人通过集约式作业方式,在绞尽脑汁的智力活动中开发出来的,而人的思维活动是非常易变的,集约在一起的众多人的思维更难保持协调一致。

(3) 导致软件产品质量低下的差错和缺陷,是随着软件开发的进展而向后面阶段传播的。值得注意的是,这些差错和缺陷的危害程度,随着从它发生到被纠正的时间间隔的增加而成倍地增长。

(4) 通过智力活动开发出来的产品,它的质量也只有通过智力活动来确认和检验,而不能像硬件产品那样,可以用有关的测量仪器来检验。

相应地,针对质量方面的上述特殊性,软件的质量管理、质量控制、质量度量和质量保障要解决的主要问题有:

(1) 如何建立必要的标准、规范、条例及程序,使开发人员和开发过程保持以下三个方面的一致性:

▲ 横向的,人与人之间的一致性;

▲ 纵向的,一个开发阶段与另一个开发阶段的一致性;

▲ 产品(含阶段性产品)的输入与输出的一致性。

(2) 如何将一个软件产品的整个开发过程分解为若干个可实现并可管理的部分;对每个部分,应用相应的、有效的技术和方法,展开软件的开发活动并对其阶段性产品的质量进行验证和确认,保证其高质量;最终组合成整个软件产品,并验证和确认其质量是否达到要求。

(3) 如何尽可能地避免在开发中产生差错;如何尽可能多地检测出并纠正已经产生的差错;当差错不可避免地残留在软件中时,如何容差。

(4) 当差错不可避免地引入到软件中时,如何保证自它们引入时起,尽可能早地发现并纠正它们。

(5) 软件这个无形的智力产品,它的质量究竟如何定义,又如何度量。

(6) 如何控制软件的开发过程、开发资源和产品,使得产品质量、工期和费用实现最佳的折中。

进行质量管理、质量控制、质量度量和质量保障的目的就在于:在开发过程中,在有限的资源限制下和规定的时间内,把软件产品的质量构建入软件产品之中。当读者在阅读这本书的时候,抓住上述问题,就可以更好地理解书中各章节的内容。同时,在读完了书中各章节之后,对上述问题也就会有更实质性的把握。

本书作者洪伦耀教授,在软件可靠性和软件质量方面,有着近二十年的研究和教学经验;作者董云卫高级工程师,曾就职于西安协同软件股份有限公司、西安软件工程技术研究中心等单位,一直从事软件项目的组织、管理和软件的实际开发工作以及软件工程理论与应用的研究。

2003年,本书的两位作者先后就职于国家863软件专业孵化器西安基地(西安软件工程技术研究中心),参加863计划相关课题的研究工作。西安软件工程技术研究中心是西安市科学技术局批准设立的独立的科学研究与技术开发机构。它围绕西安软件产业的发展和国家863软件专业孵化器的建设,专门从事技术开发及测试评估、质量咨询等软件工程技

术服务。在此期间，课题组对软件质量管理、控制、度量以及软件质量保障的各种技术和方法的深入探讨，使作者长期积累的有关软件质量管理的研究与教学经验以及软件开发经验得到了系统升华，拓展了作者的视野，并使作者坚信：科学的软件质量管理、控制、度量和保障的方法和手段是可以形成一种面向软件产业发展的服务能力的，一旦这种能力与软件产业结合并广为产业界所接受，它将对软件产业的发展起到巨大的推动作用。这也就是作者写作本书的用意所在。

本书是面向软件项目开发领导部门、管理机构、质量组织、开发人员和软件项目的购置者、使用者和维护者的工程实用性读物。同时，本书也面向培养上述人才的大专院校教师、高年级学生、研究生。因此，本书不追求冗长、复杂的理论叙述，而着重于“做什么”、“为什么做”、“怎样做”、“结果是什么”等这样一些实用性软件工程的内容。

本书共分7章。第1章作为基础，叙述了软件与硬件之间的区别及质量、软件质量等一般性的概念，并叙述了造成软件质量低下的根源——软件中的差错。第2章、第3章和第4章分别介绍了软件质量管理、质量控制和质量度量。由于软件可靠性是软件质量中最重要的质量因素，因此第5章专门对软件可靠性、软件可靠性模型及其应用进行了专门的论述。由于上述各质量活动不能无序、无计划自动进行，而是要在强有力的监督和保障之下，落实到开发过程的每一个阶段和每一个环节，因此，第6章安排了这样的内容——软件质量保障。第7章对近期软件工程和软件可靠性研究的某些问题做了简单的提示。

作者特别要感谢863计划计算机软硬件技术主题专家组和西安市科学技术局领导富有远见的支持；还要感谢西安软件工程技术研究中心楼文晓主任和西安协同软件股份有限公司前任总经理苏新雷先生，正是有了他们对软件质量在软件产品开发过程中及软件产业发展中重要地位的深层次理解，才使我们有机会尝试、应用和实践软件质量管理、控制、度量和保障的理论和方法；更要感谢郝克刚教授在写作过程中给予作者的指导，他还在百忙之中给本书撰写了序言。武汉大学软件工程研究所的徐仁佐教授，中国航天总公司204研究所的宋晓秋(博士)研究员，空军工程大学的王兴亮教授以及西安软件工程技术研究中心、西安协同数码软件股份有限公司的同仁对本书的编写和出版给予了热情的关心、帮助和支持，在此一并表示感谢。

参加本书编写、审校、资料收集和其他辅助工作的还有：凌惠娟、孙训峰、姬伟锋、王海滨、杜宏、安宣、李水清等。

由于作者知识水平和工作经验有限，书中难免有错误和不当之处，敬请读者指正。

作 者

2004年1月

目 录

第 1 章 引言	1
1.1 概述	1
1.2 质量	3
1.3 软件质量	3
1.4 造成软件质量低下的根源——软件中的差错	4
1.5 影响软件质量的主要因素	6
1.6 本章小结	7
思考题	8
参考文献	8
第 2 章 软件质量管理	9
2.1 概述	9
2.2 系统需求分析阶段	10
2.3 软件需求分析阶段	12
2.4 概要设计阶段	15
2.5 详细设计阶段	17
2.6 软件实现阶段	20
2.7 软件集成测试阶段	23
2.8 确认测试阶段	28
2.9 系统联试阶段	30
2.10 软件的验收和交付	31
2.11 软件的更新、存档和复制	36
2.12 本章小结	37
思考题	37
参考文献	38
第 3 章 软件质量控制	39
3.1 引言	39
3.1.1 软件质量控制的基本概念	39
3.1.2 有关软件质量控制的几个问题	41
3.1.3 软件质量控制的一般方法	43
3.2 全面软件质量控制	45
3.2.1 全面软件质量控制模型及其基本要素	45
3.2.2 全面软件质量控制参数	46
3.2.3 全面软件质量控制的实施	48
3.3 软件质量控制技术	50
3.3.1 软件质量控制技术的特征	50
3.3.2 质量控制问题与质量控制技术	54
3.3.3 软件质量控制技术的选择	60

3.4 全面软件质量控制的应用——质量控制计划的编制	61
3.4.1 概述	61
3.4.2 确定软件质量需求	63
3.4.3 确认限制条件	64
3.4.4 识别风险	65
3.4.5 质量控制技术选择的提炼	67
3.4.6 检查点的计划安排	67
3.5 全面软件质量控制计划的修改	69
3.6 全面软件质量控制 PDCA 方法的总结与回顾	70
3.7 本章小结	72
思考题	73
参考文献	74
第 4 章 软件质量度量	75
4.1 概述	75
4.1.1 软件质量度量问题及质量观点	75
4.1.2 与软件质量度量有关的定义	77
4.2 软件质量度量体系结构	79
4.2.1 软件质量剖面	79
4.2.2 软件质量度量总体框架	80
4.2.3 质量因素、度量标准及度量细则	81
4.2.4 质量因素、度量标准及度量细则的汇总	85
4.3 软件质量因素及度量标准的定义	88
4.3.1 质量因素的定义	88
4.3.2 度量标准的定义	89
4.4 软件质量度量过程模型	90
4.5 软件质量度量方法	92
4.5.1 软件质量度量和评估与产品开发过程的关系	92
4.5.2 选择和确定质量因素	94
4.5.3 选择和确定度量标准	100
4.5.4 选择和确定度量细则	102
4.6 软件质量度量评分过程举例	103
4.6.1 软件需求分析阶段的完备性度量	103
4.6.2 软件需求分析阶段的一致性度量	105
4.7 软件质量度量评审	106
4.7.1 软件质量度量得分的评审	106
4.7.2 软件质量评估报告	107
4.8 本章小结	108
思考题	109
参考文献	110
第 5 章 软件可靠性度量	111
5.1 引言	111
5.1.1 软件可靠性工程及软件可靠性	111
5.1.2 软件的差错、故障和失效	115

5.1.3	软件失效过程及其表示	117
5.1.4	软件失效率和硬件失效率	117
5.2	软件可靠性度量和软件可靠性建模基础	118
5.2.1	基本概念	118
5.2.2	某些重要的软件可靠性度量	119
5.2.3	软件可靠性工程中某些有用的分布	121
5.2.4	参数估计	122
5.3	软件可靠性数据收集	123
5.3.1	差错数据的收集	123
5.3.2	过程数据的收集	125
5.3.3	产品数据的收集	126
5.4	软件系统的运行剖面	129
5.4.1	软件系统运行剖面的基本概念及其在软件可靠性工程中的应用	130
5.4.2	软件系统运行剖面与软件可靠性的关系	131
5.4.3	运行剖面的建立	132
5.5	软件可靠性模型及其应用	133
5.5.1	软件可靠性模型的历史简述及一般情况	134
5.5.2	软件可靠性模型参数	136
5.5.3	软件可靠性模型及其应用	137
5.5.4	软件可靠性模型应用举例	140
5.5.5	小结	142
5.6	软件可靠性测试	143
5.6.1	软件测试的类型	143
5.6.2	软件可靠性测试	144
5.6.3	软件可靠性测试的具体实施过程	147
5.7	软件故障数据的趋向性分析	149
5.7.1	趋向性分析的含义及其作用	149
5.7.2	软件故障数据工程分析模型	150
5.7.3	趋向性分析	152
5.7.4	趋向性分析结果的应用	153
5.8	本章小结	153
	思考题	155
	参考文献	155
第6章	软件质量保障	157
6.1	概述	157
6.1.1	软件质量保障的定义	157
6.1.2	软件质量保障的目的及所涉及的有关问题	158
6.1.3	软件质量保障的活动过程、活动的主要内容及所用技术	159
6.1.4	软件质量保障的三个主要活动领域	160
6.2	软件质量保障的质量体系和质量成本	161
6.2.1	软件质量保障的质量体系	161
6.2.2	软件质量成本	162
6.3	软件质量保障计划	164

6.4 软件质量保障的精髓——软件配置管理	168
6.4.1 软件配置管理的基本概念	168
6.4.2 软件配置管理实施过程中几个值得注意的问题	170
6.4.3 软件配置管理过程中的活动	172
6.4.4 软件配置管理计划	173
6.4.5 计算机辅助的软件配置管理系统	175
6.5 软件质量保障中的评审和检查	176
6.5.1 评审和检查的作用与目标	176
6.5.2 检查	177
6.5.3 评审	179
6.5.4 评审问题清单	183
6.6 软件质量保障组织	184
6.7 本章小结	186
思考题	186
参考文献	187
第7章 软件过程管理——CMM 介绍	188
7.1 概述	188
7.1.1 关于 ISO 9001	189
7.1.2 关于 CMM	189
7.2 与软件能力成熟度有关的定义	190
7.3 软件能力成熟度等级	191
7.4 软件能力成熟度等级的说明示例	192
7.5 软件能力成熟度、可视度、度量之间的关系	195
7.6 软件能力成熟度等级的结构组成	195
7.7 软件能力成熟度等级的评估方法	198
7.8 软件能力成熟度集成模型(CMMI)简介	201
7.8.1 概述	201
7.8.2 CMMI 的评估	203
7.8.3 我国软件能力评估标准的制定与实施	203
思考题	204
参考文献	205
第8章 软件工程近期的发展	206
8.1 软件工程技术	207
8.2 软件分析	208
8.2.1 软件故障树分析(SFTA)	208
8.2.2 失效模式效应及危害度分析	213
8.2.3 软件故障树分析和软件故障模式效应及危害度分析的应用	214
8.3 软件工程的自动化	214
8.4 软件统计质量控制	216
8.5 软件可靠性预计的两个值得关注的研究领域	216
8.6 建立一个“经验教训”库	218
思考题	219
参考文献	219

第 1 章 引 言

1.1 概 述

当今世界，计算机应用已经遍布于人们现代生活的各个方面。随着计算机的应用范围变得越来越广泛，计算机的软件系统也变得越来越复杂和越来越庞大。与硬件系统的飞速发展形成鲜明对照的是，计算机软件技术却没有得到相应的快速发展，这既与计算机硬件的发展不相适应，也远远满足不了人们对软件的需求，特别是对软件质量方面的需求。

对 Pascal 等语言和数据结构有着卓越贡献的 1984 年图灵奖得主 Niklaus Wirth 曾这样说明当前软件和硬件的情况：“我们已不再被运行缓慢的硬件所局限，但却受困于我们自己的智能。”他这里所说的“智能”，显然是指通过人的大脑所开发出来的软件。软件和硬件当前存在这种情况的内在原因就是，软件和硬件之间存在着本质区别。

首先，就软件自身的特点来说，软件和硬件之间的区别在于：

- ▲ 软件并非有形存在；
- ▲ 软件包含数据和逻辑；
- ▲ 就当前的普遍情况来说，几乎不存在软件质量的度量，特别是定量度量；
- ▲ 软件容易更改，但也更容易引入错误；
- ▲ 软件更改后（包括引入的错误），其影响会迅速传播；
- ▲ 从软件试图实现的复杂功能来说，软件有比硬件高得多的复杂性。

其次，从形式上来说，软、硬件之间的区别更加明显。硬件有齐全的设计图纸、技术资料；硬件是由机器生产的，有实物可以对照。硬件早就在工厂进行了工业化生产：设计人员设计出图纸、生产程序；工人按图纸和规定的程序生产出产品；检验人员用设备和仪器检查产品是否满足要求。与硬件不同的是，软件是人的大脑的智力活动的产物，是人的逻辑思维的产品。一方面，由于人的智力的不完善和不一致，以及人的大脑对复杂逻辑系统的理解和控制的局限性，在软件开发过程中，势必会产生大量错误；另一方面，对软件错误的检查和纠正，对软件合格或不合格的度量，也只能且必须通过人的大脑的智力活动才能完成。

由于软件有区别于硬件的这些特点，在当今系统硬件变得越来越可靠的情况下，软件已经成为导致系统失败和停机的主要因素；同时，由于软件系统的规模和复杂性的增长，使得软件的开发过程变得很难控制和管理，软件的开发周期延长，成本增加，软件质量和可靠性降低，其结果就形成了世界软件行业所共同面临的“软件危机”。人们在展望 21 世纪计算机科学的发展方向和策略时，把提高软件质量放在优先于提高软件功能和性能的地

位。世界上许多著名的计算机科学家一致认为：21 世纪的高技术市场将与高质量的软件紧密相关，高质量软件的开发技术和保障技术是打开 21 世纪高技术市场大门的钥匙。

软件危机是指在软件开发中所遇到的一系列严重而又难以解决的问题。这些问题集中地反映在软件的质量方面：怎样开发高质量的软件？怎样维护现有的、容量又在不断增加的软件？怎样做才能满足软件不断增长的质量需求？

面对“软件危机”，为了使软件开发中的质量、工期和费用这三大要素得到最佳的实现，计算机专家和软件工程师们经过长期的研究和实践，逐渐形成了一种有效的工程方法，即软件开发中的每一个活动(预算、计划、分析、详细说明、设计、实现、测试和维护)都必须是可理解的，并且是可控制的，也就是说，软件必须在一整套程序、标准、规范、方法、技术和工具的规范下，以受控的和科学的方法来实现。人们把这种工程方法称为“软件工程”。实践证明，使用正确的方法，即软件工程方法，可以明显地提高软件质量和生产效率。

但是，我国不少软件企业的软件生产方式，现在还处于计算机时代的早期阶段，其特点是“自编、自导、自演”。在很多的软件鉴定会和验收会上，所谓“测试”只是对预先确定的测试用例进行一下演示就算通过，起不到应有的作用。这将把软件中大量的剩余差错留给软件的现场安装调试，甚至留给用户，从而使安装调试阶段的投入(人力、时间和资金)超过开发阶段的总投入，使得用户花在软件维护方面的费用超过了购买该产品时的总费用。不少用户(单位)由于所使用的软件质量不高，不好使用，或由于负担不起高昂的维护费用，而只能过早地使软件报废。

许多人存在着不合理的误解和偏见，使得软件工程方法得不到广泛而有效的使用。不少人认为，在软件开发中所涉及到的具体的质量管理、质量控制和质量保障工作，是繁琐的、费时费力的，而且需要投入大量的资金、人力和设备；他们更多地着眼于开发计划的完成以及资金和人力的节省。因而他们不能、也不愿按软件工程标准及方法去做。在我国，许多有相当规模的软件企业，对交付的软件没有确定的质量认证，更给不出定量的软件可靠性这一重要的质量指标；软件测试在许多软件企业实际上还是空白(名义上也有几个人进行测试工作)。对比来说，美国的某些软件工厂的测试人员竟超过开发总人数的 30%；测试工作量超过总开发工作量的 40%~50%，高质量的软件甚至高达 80%。这种差别是何等的悬殊！

但是，客观规律是不以人的意志为转移的，软件工程的实践毋庸置疑地证明了以下几点：

(1) 软件产品的质量是在开发中构造入产品的，是后天加不进去的。那么，对于形成产品的开发过程，如果不加以科学的管理和控制，没有质量保障目标和计划，那么软件产品是得不到所要求的质量的。

(2) 在软件开发期间，不采用软件工程方法及行之有效的软件工程标准、程序和技术，从表面上看是省时省力的，但是由此可能产生的大量软件差错和缺陷，将需要用更多的时间、人力和资金来修正，真正符合质量要求的产品的释放时间反而会更加推迟。这种教训的实例是举不胜举的。

(3) 在软件质量管理、控制和质量保障方面的投入，一般只占项目总经费的百分之几，但这种投入从总体效果(差错和缺陷减少、问题解决得早而及时、无返工、产品质量提高等)上看，往往能得到高于投入 10 倍或更高的回报。

如果没有理由反对软件生产的必要性和昂贵性，那就必须认真地采用任何有助于改进软件产品质量和可靠性的方法和技术，并使之成为软件开发和支持活动不可或缺的一部分，

因为软件质量不高和软件不可靠将会让人们付出更大的代价，甚至带来灾难性的后果。

软件质量问题是本书讨论的中心问题。本书的目的是运用软件工程的观点和方法，讨论什么是软件质量，如何提高我们所开发的软件产品的质量以及如何度量一个软件产品的质量等问题。

1.2 质 量

一个产品或一种服务所具有的满足规定的或隐含的需求能力的各种特征和特征的总和，称为这个产品或这种服务的质量。

一个产品的质量具有如下的性质：

(1) 一个产品质量的好与坏不是绝对的。对同一个产品而言，不同的用户、不同的使用环境，就有不同的质量需求。

(2) 构成质量的诸因素往往不是互相独立的，有的具有正向性的关系，而有的则互相冲突。对于具有正向性关系的那些质量因素，有一个提高了品质，另一些也会受益；而对于那些互相冲突的质量因素，则要进行折中的考虑。

(3) 构成质量的诸因素并不具有同等的重要性。一个产品的质量关系到该产品是否能够被用户所接受，当产品的质量受到代价、人力、时间等条件限制时，应该确认那些最重要的因素，而放宽次要的因素。

(4) 质量必须在一定的前提下衡量。在许多情况下，不能脱离实际情况和花费难以接受的代价来要求一个产品的高质量。

(5) 产生质量问题的根源是产品在设计、生产过程中存在缺陷和差错。

软件产品的质量也具有上述特性。认识这些特性，在软件质量管理和度量的研究中将起到重要的作用。

1.3 软件质量

软件质量没有一个统一的、惟一的定义，不同的系统也许有不同的定义。我们这里给出的定义如下：软件质量是软件产品满足客户(买主、使用者和维护者)需求的一组可度量的特性。从上述这个简明的定义中可以看出：

(1) 软件质量是软件产品的属性。

(2) 定义软件产品质量的特性可以有許多。

不同的客户对质量的需求是不同的。客户应该确定他们需求哪些质量特性，并且确定如何度量它们。表 1-1 给出了软件质量要求的一些例子。

表 1-1 软件质量要求举例

客 户	质 量 要 求
买主	成本，资金，使用效率
使用/操作者	功能性，可靠性，使用容易性
维护者	改变容易，可扩展

我们还可以用另外一种形式来更详细地说明软件的质量特性。为此，可把软件质量特性分成三组：产品运行、产品修改和产品转移，它们分别反映客户(买主、使用者/操作者、维护者)对软件产品质量要求的三种不同的倾向或观点。这三组软件质量特性可用图 1-1 表示。

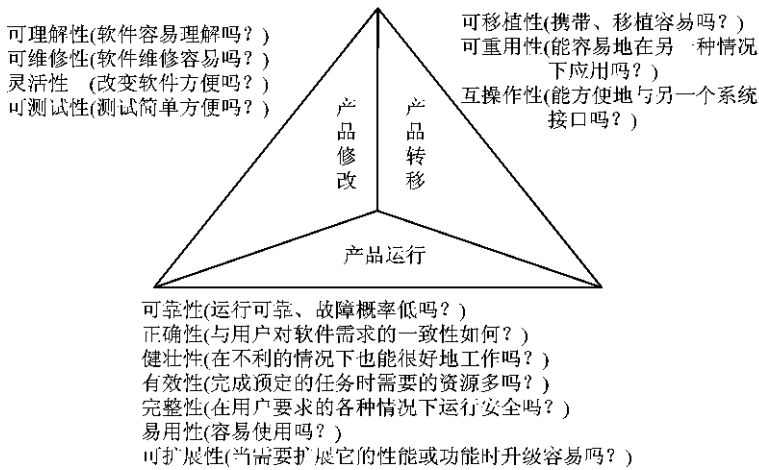


图 1-1 软件产品的质量特性图示

需要指出的是：系统不同，软件的质量特性也将不完全相同；而且对不同系统的同一个特性，是否满足要求的度量标准也不一样。

软件产品不像硬件产品那样，存在着由于机件的磨合(早期失效)以及材料的耗损(损耗失效)所带来的质量问题；同时，软件不像硬件那样是由机器生产的，它是由不同个体的人通过大脑的逻辑思维生产出来的。因此，软件开发中出现缺陷和差错是不可避免的，特别是在当今软件功能越来越复杂、软件规模越来越庞大的情况下更是如此。在某种程度上，我们可以这样说：如何改善软件产品质量的问题，实质上就是在软件的开发中如何避免错误、如何检测和纠正错误的问题；同样，度量一个软件产品的质量，主要就是确认软件产品中缺陷和差错存在状态的问题。

1.4 造成软件质量低下的根源——软件中的差错

软件工程的三大要素是软件质量、成本和进度。应注意的是，成本和进度是可以很容易定量度量的，对质量的定量度量显得非常困难，但却是十分重要的。缺乏软件质量的具体度量，就意味着当质量、成本和进度产生矛盾时，将牺牲质量。事实上，这正是软件产品存在质量问题的主要原因。正是由于这一原因，软件工程相对于它的三大要素而言，极端地侧重于软件的质量要素是完全必要的。

软件质量的高低主要是以在已释放的软件产品中残留差错的多少来衡量的。软件中的差错是造成软件质量低下的根源。因此，软件工程相对于软件质量来说，其核心内容是：避免差错、纠正差错、容忍差错(容差)和度量差错。由此不难看出，软件中的差错是软件工程关注的焦点。

我们下面将对软件中的差错进行分析，这将有助于我们对软件工程方法和软件工程过程的理解。据统计：

(1) 在软件寿命期中各阶段的差错分布为：

- ▲ 需求分析和设计阶段占 64%；
- ▲ 编码阶段占 36%。

(2) 差错总数中，差错类型的分布为：

- ▲ 文档占 2%；
- ▲ 计算占 5%；
- ▲ 人的差错占 5%；
- ▲ 环境占 5%；
- ▲ 接口占 6%；
- ▲ 数据占 6%；
- ▲ 逻辑占 28%；
- ▲ 需求转化占 36%；
- ▲ 其他占 7%。

(3) 在软件寿命期各阶段中可能发现差错的分布为：

- ▲ 需求分析阶段占 9%；
- ▲ 设计占 2%；
- ▲ 编码占 7%；
- ▲ 软件集成占 15%；
- ▲ 系统集成占 48%；
- ▲ 现场试验占 13%；
- ▲ 运行占 6%。

(4) 纠正一个软件差错的最小费用：据国外统计，纠正一个软件差错的费用将会随着软件生存期阶段的进展而扩大 10 倍以上(倍增因子大约为 14)，如图 1-2 所示。

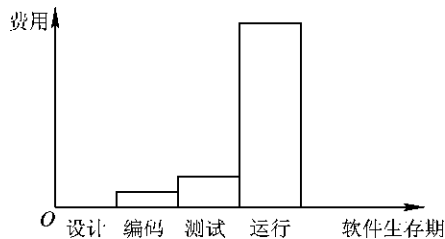


图 1-2 纠正一个软件差错的最小费用

由上面的有关差错统计数据可以看到：软件中约 60% 的差错是在需求分析和设计阶段引入的，而在此阶段一般只能查出 10% 左右，大部分差错都只能在软件生存期的后几个阶段发现；同时，随着软件进展到生存期的后几个阶段，纠错所需的代价将成 10 倍地增加。可想而知，软件的差错，特别是早期差错，不仅在功能方面产生的影响是巨大的，甚至是灾难性的，而且在经济上造成的损失也是极其重大的。值得指出的是：上述数据是由那些软件工程搞得很好的先进国家统计出来的，他们移交软件产品的每 1000 行可执行源代码

的差错数为 3~7 个, 而我国则高于 15 个。因此我们更应该运用软件工程的方法和过程, 建立强有力的软件质量管理组织, 加强软件生存期中各阶段、特别是早期阶段的质量管理, 生产出高质量的软件。

1.5 影响软件质量的主要因素

从总体上说, 影响软件产品质量的要素有三个: 开发软件产品的组织、开发过程以及开发过程中所使用的方法和技术。有人称这三个要素构成了软件产品质量的铁三角, 缺一不可。这就是说, 在适当的成本与进度条件下, 一个有能力的开发组织通过规范化的开发过程, 使用先进的、有效的开发工具、技术和方法并使它们得以正确的实施, 才能在软件开发过程中更少地引入差错、更多且更早地发现并纠正已引入的差错, 从而使软件产品的质量得到保证。我们不难想象, 影响软件质量的因素, 不是单一的, 而是综合性的。

从具体的开发角度来说, 有很多开发因素决定着软件产品的质量, 即决定着引入软件中的差错的多少。美国罗姆空军开发中心(RADC)和空军系统控制中心(AFSC)做了大量工作, 以确定这些因素的存在、影响及其与软件质量的关系。软件工程方法就是在不断地克服那些对软件质量有不利影响的因素、增强对软件质量产生有利影响的因素的过程中发展起来的。因此, 了解影响软件质量的主要因素, 对我们理解后面将要叙述的软件工程方法的执行和实施将是十分有益的。

影响软件质量的因素有以下 15 种, 简述如下。

(1) 开发方法和工具。影响软件质量的开发方法和工具主要有:

- ▲ 结构化设计、编码、测试和维护;
- ▲ 伪码和流程图技术, 它是设计工具, 也是编码工具;
- ▲ 设计、编码、测试、维护工具以及需求分析跟踪码产生工具;
- ▲ 进程和状态报告及差错跟踪;
- ▲ 设计和编码排查;
- ▲ 正式评审。

(2) 开发人员的训练因素。它主要指开发人员的全部经验以及结构化方法的经验, 它们将对软件质量产生有利的影响。

(3) 软件开发的组织形式。组织机构、指导方针以及使用的标准将影响软件的质量。

(4) 文档的提供。源码、技术资料以及开发计划等将影响软件的质量。具体文档指:

- ▲ 包含在软件编码中的文档, 即模块名、模块在层次树中的位置及模块功能;
- ▲ 软件以外的文档, 即配置管理计划、质量保证计划、软件开发计划及软件测试计划。

(5) 复杂性。结构和功能的复杂性将影响软件的质量。

(6) 环境。终端用户的环境以及对环境建模的难易程度将影响软件的质量。具体的环境因素有:

- ▲ 软件、硬件与人之间的接口;
- ▲ 用户的训练;
- ▲ 输入数据的确认。