实用软件评测技术

主　编　翟天喜

副主编　南　宁

国防科技大学出版社

实用软件评测技术

主 编 翟天喜
副主编 南 宁
编 委 叶建芳 袁肃蓉 黄万民 叶建威 刘 新
安郁虹 葛文学 罗泽湘 王兰波 林 琳

国防科技大学出版社

图书在版编目（CIP）数据

实用软件评测技术 / 翟天喜主编.—长沙：国防科技大学出版社，2007.7

ISBN 978-7-81099-338-8

Ⅰ. 实... Ⅱ. 翟... Ⅲ. 软件—测试 Ⅳ. TP311.5

中国版本图书馆 CIP 数据核字（2007）第 077083 号

实用软件评测技术

翟天喜 主编

国防科技大学出版社出版发行

电话：（0731）4572640 邮政编码：410073

http://www.gfkdcbs.com

责任编辑：徐飞 责任校对：唐卫葳

广州市番禺时代文化印刷厂印刷

（如发现印装质量问题，请与印刷厂联系调换）

各地新华书店经销

开本：787 × 1092 1/16 印张：18 字数：437 千字

2007 年 9 月第 1 版 2007 年 9 月第 1 次印刷

ISBN 978-7-81099-338-8

定价：28.00 元

前 言

随着软件规模的大型化和软件开发的复杂化，软件的质量问题变得越来越突出。而软件测试是软件质量保证的重要手段，因此，软件测试的重要性越来越受到重视。软件测试是一项专业性很强的工作，测试工作者不仅需要具备相关的理论知识，还需要不断的实践，缺少这些知识和经验，测试工作是无法深入的。鉴于此，我们编写的这本书特别注重突出应用性和实践性，力求理论与实践相结合。

本书内容

本书面向软件测试和软件质量管理人员，分别从理论、实践、管理的角度介绍了软件测试与质量评估的精华与要点。在理论部分，介绍了软件测试基础、软件测试方法、软件测试类型、自动化测试以及性能测试等方面的内容。在实践部分，介绍了典型的软件测试工具、白盒和黑盒测试用例设计以及黑盒测试（功能测试）、自动化测试、性能测试等实际测试案例，并详细讲解了如何设计测试用例、如何用工具进行自动化测试，方便读者直接把握实践的要点。在管理部分，介绍了影响软件质量的因素、质量体系模型，以及从哪些方面着手控制软件的质量。附录部分还提供了实际的测试规范及最新的软件评测方面的标准，供读者参考。

读者对象

本书的主要读者对象是企业软件质量管理人员和软件测试人员，以及高校软件工程相关专业的师生。本书也可以作为软件学院的教学参考书。为了更好地理解本书内容，读者应了解基本的软件工程知识，为了做好软件测试工作，仅仅阅读书籍是远远不够的，读者还需要不断地学习这一领域的新知识和新技术。而且，软件测试强调实践性，读者可以把本书作为测试工作的指导和参考，在实际软件测试项目中加强实践，积极思考，善于总结，才能不断提高。

感谢

本书的编写得到了广东软件评测中心的大力支持，书中的许多实际案例以及专业化的测试工具，均由广东软件评测中心提供，在此表示衷心的感谢！同时，也感谢国防科技大学出版社所提供的合作机会，使该书可以早日和读者见面。

由于作者水平有限，书中不可避免会出现一些错误，恳请读者批评指正。

编 者

2007 年 8 月

目 录

第1章 软件工程与软件测试

1.1 软件的概念和特点

软件是计算机系统中与硬件相互依存的一部分，它是包括程序、数据及其相关文档的完整集合。其中，程序是按事先设计的功能和性能要求执行的指令序列；数据是使程序能正常操纵信息的数据结构；文档是与程序开发、维护和使用有关的图文材料。软件可分为应用软件和系统软件。应用软件用来实现信息处理功能所要求的过程，系统软件完成使应用软件能与其他系统元素交互的控制功能。一个基于计算机的系统可以用输入—处理—输出（IPO）模型来表示，软件元素渗透到这个模型的各个方面。需要注意的是软件不同于我们所熟悉的客观世界的物质，它是一个抽象的概念。

为了能全面、正确地理解计算机软件，必需了解软件的特点。经过总结，我们给出计算机软件的8个特点：

(1) 软件是一种逻辑实体，而不是具体的物理实体，因而它具有抽象性。这个特点使它和计算机硬件有着明显的差别。人们可以把它记录在介质上，但无法看到软件的具体形态，而必须通过观察、分析、思考、判断，去了解它的功能、性能及其他特性。

(2) 软件的生产与硬件不同。在软件的开发过程中没有明显的制造过程。软件是通过人们的智力活动，把知识与技术转化成信息的一种产品。一旦某一软件项目研制成功，以后就可以通过介质拷贝的方式大量、快速、廉价地复制。

(3) 在软件的运行和使用期间，没有硬件那样的机械磨损、老化等问题，但存在功能退化问题。在软件的生命周期中，为了使它能克服以前没有发现的故障，使它能够适应硬件、软件环境的变化以及用户新的要求，需要多次修改（维护）软件，而每次修改会不可避免地引入新的错误。因此，软件维护比硬件维护要复杂得多，与硬件的维修有着本质的差别。

(4) 软件的开发和运行常常受到计算机系统的限制，对计算机系统有着很强的依赖性。软件不能摆脱硬件单独活动。在开发和运行中必须以硬件提供的条件为依据。为了减轻这种依赖性，在软件开发中提出了软件移植的问题，并且把软件的可移植性作为衡量软件质量的因素之一。

(5) 软件的开发至今尚未摆脱手工制作的开发方式。对于软件人员来说，开发工作是一种高强度的脑力劳动，没有哪一位软件人员认为这是一项轻松的工作。

(6) 软件是复杂的。有人认为，人类能够创造的最复杂的产物是计算机软件。软件的复杂性一方面来自于它所反映的实际问题的复杂性；另一方面来自于程序逻辑结构的复杂性。软件开发，特别是应用软件的开发常常涉及到其他领域的专门知识，这对软件人员提出了很高的要求。

(7) 软件成本相当昂贵。软件的研制工作需要投入大量的、复杂的、高强度的脑力劳动，它的成本是比较高的。20世纪80年代以来，软件的开销大大超过硬件的开销。然而，并非所有在软件开发上的花费都能获得相应的成果。

(8) 相当多的软件工作涉及到社会因素。许多软件的开发和运行涉及到机构、体制及管理方式等问题，甚至涉及到人们的观念和心理。对于这些人的因素重视不够，常常是软件工作遇到

的问题之一。为此，我们需要加强对软件开发组织管理的研究。

1.2 软件危机

在20世纪60年代以后，随着计算机应用的普及，计算机软件的需求也随之增大。由于计算机软件本身具有的很强的“软”特性，使人们不好把握它。所以，当软件的生产进入到大规模开发的时代时，开发与应用之间的矛盾也日益尖锐。这些问题归纳起来有：

(1) 软件开发无计划性

由于缺乏软件开发的经验和有关软件开发数据的积累，使得开发工作的计划很难制定。主观盲目地制定计划，执行起来和实际情况有很大差距，致使常常突破经费预算，对于工作量估计不准，进度计划无法遵循，开发工作完成的期限一拖再拖。在这种情况下，软件开发的投资者和软件的用户对软件开发工作既不满意也不信任。

(2) 软件需求不充分

作为软件设计依据的需求，在开发的初期阶段提得不够明确，或是未能得到确切的表达。开发工作开始后，软件人员和用户又未能及时交换意见，造成开发后期矛盾的集中暴露，而此时问题既难分析，又难解决。

(3) 软件开发过程无规范

开发过程没有统一的、公认的方法论和规范指导，参加的人员各行其事。加之不重视文字资料工作，使设计和实现过程的资料很不完整，或者忽视了程序接口部分，发现问题修修补补，造成整个软件系统难于维护。

(4) 软件产品无评测手段

未能在测试阶段充分做好检测工作，提交用户的软件质量差，在运行中暴露出大量的问题。在应用领域不可靠的软件，轻者影响系统的正常工作，重者造成生命和财产的损失。

以上这些矛盾仅描绘了软件危机的某些侧面，如果这些障碍不能突破，软件的发展是没有出路的。

1.3 软件工程

为了解决软件发展中遇到的问题，人们对软件生产中的质量控制越来越重视，很多人致力于这方面的研究，试图从规范软件生产的各个环节入手，改善软件生产的质量，提高软件生产的效率。通过实践证明，这是一条行之有效的道路。那么，软件生产有哪些环节，应该如何改进这些环节呢？这就是软件工程所要研究的问题。

1.3.1 软件工程的定义

著名软件工程专家Boehm曾为软件工程下了定义：“运用现代科学技术知识来设计并构造计算机程序以及为开发、运行和维护这些程序所必需的相关文件资料。”这里对“设计”一词应有广义的理解，它应包括软件的需求分析和对软件进行修改时所进行的再设计活动。1983年IEEE给出的定义为：“软件工程是开发、运行、维护和修复软件的系统方法。”其中“软件”的定义为：计算机程序、方法、规则、相关的文档资料以及在计算机上运行时所必需的数据。

Fairly进一步认为，“软件工程学是为在成本限额以内按时完成开发和修改软件产品所需的系统生产和维护的技术和管理的学科”。软件工程学就是建立并使用完善的工程化原则，以较经济的手段获得能在实际机器上有效运行的可靠软件的一系列方法。很多人对软件工程提出了更为完善的定义，但主要思想都是在强调在软件开发过程中需要应用工程化原则的重要性。

1.3.2 软件工程三要素

软件工程包括三个要素：方法、工具和过程。

软件工程方法为软件开发提出了“如何做”的技术。它包括多方面的任务，如项目计划与估算、软件系统需求分析、数据结构、系统总体结构的设计、算法的设计、编码、测试以及维护等。

软件工程方法常采用某种特殊的语言或图形的表达方法及一套质量保证标准。

由于软件生产过程的特殊性和复杂性，尤其是大型软件项目，需要设计软件工具为软件工程方法提供自动或半自动的软件支撑环境。

软件工程的过程则是将软件工程的方法和工具综合起来以达到合理、及时地进行计算机软件开发的目的。过程定义了方法的顺序，要求交付的文档资料，为保证质量和协调变化所需要的管理，及软件开发各个阶段完成的里程碑。

软件工程就是包含上述方法、工具及过程在内的一些步骤。

组织实施软件工程项目，从技术上和管理上采取了多项措施以后，最终希望得到项目的成功。成功的标志就是要达到以下目标：

(1) 付出较低的开发成本；

(2) 达到要求的软件功能；

(3) 取得较好的软件性能；

(4) 开发的软件易于移植；

(5) 需要较低的维护费用；

(6) 能及时完成开发工作，及时交付使用。

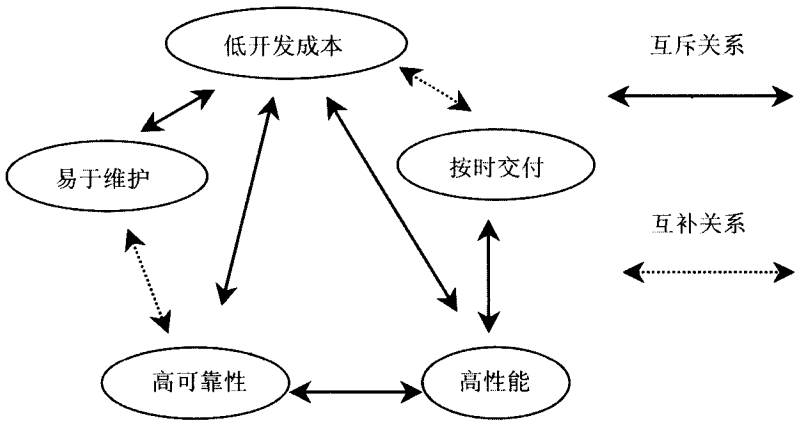

图1—1 软件工程目标之间的关系

在实际开发的具体项目中，让以上几个目标都达到理想的程度往往是非常困难的。而且上述目标之间很可能是互相冲突的，如图1-1所示。若只顾降低开发成本，很可能也降低了软件的可靠性；另一方面，如果过于追求软件的性能，可能造成开发出的软件对硬件有较高的依赖，从而直接影响到软件的可移植性。

1.4 软件质量保证

通常意义的质量保证(QA)是指为了保证产品和服务充分满足用户的需求而进行的有计划、有组织的活动，是从用户的角度来控制产品质量的。

所谓软件的质量保证，是指为了保证软件质量而对软件开发过程和被开发的软件产品进行的有计划、有系统的管理活动。它主要包括质量保证方针、质量保证标准、质量保证计划的制定；质量保证体系的建立和管理；明确各阶段的质量保证工作；各阶段的质量评审，审查软件过程对标准的符合性；质量信息的收集、分析与使用；重要质量问题的提出与分析；软件开发方法和软件测试方法的评价等。

为了在软件开发过程中保证软件的质量，软件的质量保证活动应贯穿整个软件生存周期的每个阶段。软件的质量保证措施主要有检查、评审和测试。由于人的认识不可能百分百地符合客观实际，因此在软件生存周期每一个阶段的工作中都可能发生错误，由于前一阶段的成果是后一阶段工作的基础，前一阶段的错误必然导致后一阶段工作的错误，出现错误积累，甚至产生扩大。另一方面，软件开发实践表明，越在早期发现错误，越容易改正，代价也越低。因此，前一阶段的工作应尽量不让错误进入下一阶段。如图1—2所示，软件质量保证的工作从项目一开始就应介入。

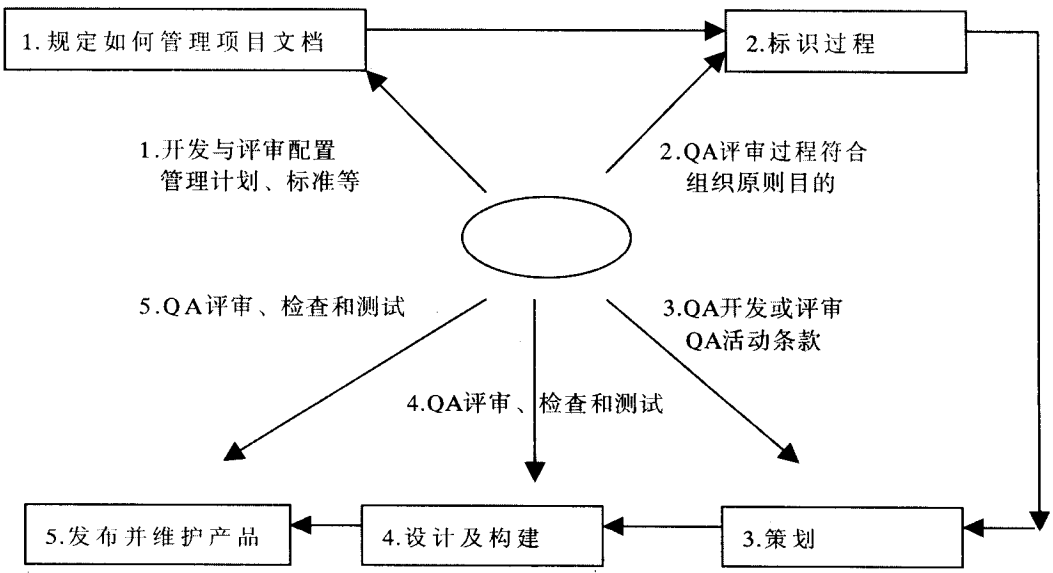

图1—2 质量保证活动

检查是指在软件生存周期的每一个阶段的工作中，由开发者及其同组成员来寻找项目的缺陷。质量检查并不是要等到每一个阶段结束时才执行惟一的一次，应该在每个实践环节都执行，实际上，自从项目的第一份文档被建立起来就应该开始使用这项技术。

质量检查的内容有二：一是分析、寻找缺陷和错误；二是做出建议，进行分析，以便“改差为好”、“好上加好”。

评审就是在软件生存周期每个阶段结束之前，都正式使用结束标准对该阶段生产出的软件产品配置成分进行严格的技术审查。

评审的目标包括以下几点：

（1）针对任意一种软件范型，发现软件在功能、逻辑和实现上的错误；

（2）验证经过评审的软件确实满足要求；

（3）保证软件是按照已经确实的保证表述的；

（4）使得软件能按一定的方式开发；

（5）使软件项目更容易管理。

软件评审是软件产品开发过程中的一种不可忽视的质量保证手段，它的实质是集中时间，发挥集中智慧，及时地提示隐藏的软件风险，即通过会议或展示、演示的形式将当前阶段生产出的软件产品配置成分及相关的资料、文档提交给软件项目人员、管理人员、用户和/或同行专家进行评价或审批。

测试是软件质量保证的关键元素，代表了规约、设计和编码的最终检查。软件产品最大的成本是检测软件错误、修正软件错误的成本。在整个软件开发中，测试工作量一般占30%~40%，甚至大于或等于50%。对某些特殊行业的软件(如飞机控制、核反应堆等)的测试所花费的时间往往是其他软件的3~5倍。

1.5 软件测试的定义

1972年6月，Bill Hetzel（代表论著《The Complete Guide to Software Testing》）在美国的北卡罗来纳（North Carolina）大学组织了首次以软件测试为主题的会议。

1973年，Bill Hetzel给软件测试下了一个这样的定义：“就是建立一种信心，认为程序能够按预期的设想运行（Establish confidence that a program does what it is supposed to do）。”

1983年，Bill Hetzel又将定义修订为：“评价一个程序和系统的特性或能力，并确定它是否达到预期的结果。软件测试就是以此为目的的任何行为（Any activities aimed at evaluating an attribute or capability of a program or system）。”在他的定义中的“设想”和“预期的结果”其实就是我们现在所说的用户需求或功能设计。他还把软件的质量定义为“符合要求”。

1979年，Glenford Myers的《The Art of Software Testing》是软件测试方面的经典。Myers定义及诠释的测试方法论已成为软件测试的基本模块。提出测试的目的是证伪。

1990年的IEEE/ANSI标准将软件测试进行了这样的定义：“就是在既定的状况条件下，运行一个系统或组建，观察记录结果，并对其某些方面进行评价的过程。The process of operating a system or component under specified conditions, observing or recording the results, and making an