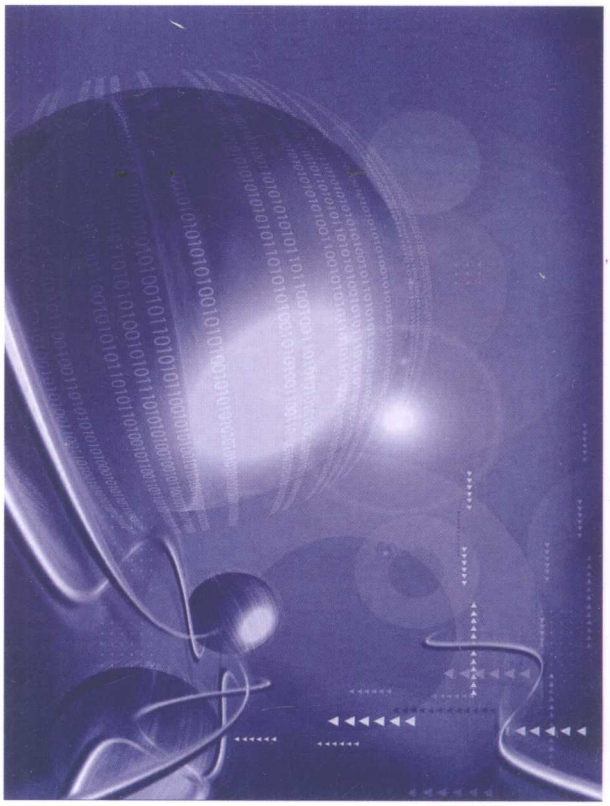
高等学校计算机应用规划教材

软件测试实验指导教程

- 软件与软件危机
- 软件缺陷管理
- 软件测试管理
- 程序理解工具
- 代码静态分析工具
- xUnit单元测试框架
- 单元覆盖测试
- Java GUI基础类库应用测试
- Web页面测试
- Gtk+用户界面测试
- 单元性能测试
- Web应用性能测试
- 软件综合评测工具Eastt



实验所需资源
使用案例

蔡建平　编著

清华大学出版社

高等学校计算机应用规划教材

软件测试实验指导教程

蔡建平　编著

清华大学出版社
北　京

内 容 简 介

软件测试是软件工程的一个重要分支，它对测试人员的专业知识要求极全、专业技术要求极强、专业能力要求极高，而目前企业对测试人员的要求是要有较丰富的测试经验及较强的测试工具应用能力。本书作为《软件测试大学教程》配套的实验教材，通过覆盖软件评测的各个环节和知识点，以主流的开源软件测试工具应用为基础，以实战能力培养为目的，为高等院校不同学历教育的软件工程专业和计算机相关专业开设软件测试课程提供了全方位的、并且是可行或可用的实践教学方案和实践教学平台，以及配套的实践教学案例。

全书共12章，分为管理、静态分析、单元测试、GUI测试、性能测试及软件综合评测共6大部分。主要内容包括：软件缺陷管理、软件测试管理、程序理解、代码静态分析、xUnit单元测试框架、单元覆盖测试、Java GUI基础类库应用测试、Web页面测试、Gtk+用户界面测试、单元性能测试、Web应用性能测试以及软件综合评测工具等。

掌握软件测试技术、构建软件测试环境、编写软件测试用例、开展软件测试工作并有效进行软件测试管理，无论是对于软件管理人员、开发人员、质量保证人员还是测试人员，都具有较强的现实意义。本书针对软件测试的实验内容全面，实验方案完整，实践环境建设可行，实验步骤及过程讲解清晰，实验案例丰富实用，可作为高等院校不同学历教育的软件工程及计算机相关专业的“软件测试实验课程”教材(如本科生、研究生，甚至高职生或高专生等)，也可作为软件测试实战培训教材，同时本书也是软件开发或管理人员、测试或质量保证人员非常好的自学参考书。

图书在版编目(CIP)数据

软件测试实验指导教程/蔡建平 编著. —北京：清华大学出版社，2009.11

(高等学校计算机应用规划教材)

ISBN 978-7-302-21434-2

Ⅰ. 软… Ⅱ. 蔡… Ⅲ. 软件—测试—高等学校—教材 Ⅳ. TP311.5

中国版本图书馆CIP数据核字(2009)第204169号

责任编辑：王　军　李维杰
装帧设计：康　博
责任校对：胡雁翎
责任印制：李红英
出版发行：清华大学出版社　　**地　　址**：北京清华大学学研大厦A座
http://www.tup.com.cn　　**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
印 刷 者：北京密云胶印厂
装 订 者：北京市密云县京文制本装订厂
经　　销：全国新华书店
开　　本：185×260　**印　张**：25　**字　数**：577千字
附DVD光盘1张
版　　次：2009年11月第1版　　**印　　次**：2009年11月第1次印刷
印　　数：1～4000
定　　价：40.00元

序

软件工程是为了解决计算机软件危机而提出来的新专业，它是一门研究如何用系统化、规范化、数量化等工程原则和方法去进行软件开发和维护的学科。如：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、提高软件生产率水平的目标。

软件工程作为一门迅速兴起的独立学科，国家教育部十分重视软件工程专业的发展。2001 年教育部和原国家计委联合下文成立了 35 所示范性软件学院，2006 年又成立了高等学校软件工程专业教学指导分委员会。软件学院的首要任务就是根据现代软件工程人才的培养要求，不断发展和改革软件工程专业教育，以满足软件工程专业快速发展的需要，培养高质量的、适应社会经济发展需要的软件人才。

北京工业大学软件学院作为第一批国家示范性软件学院，积极面向 IT 产业，努力为北京市服务。作为国家和北京市重要的软件人才培养基地，成立 8 年来，学院在软件人才培养上，积极探索、大胆改革、努力创新，成功地进行学科交叉、专业拓展，为培养应用型、交叉型及复合型软件人才做了大量的工作，取得了可喜的成绩。学院在软件工程学科和专业建设上，走内涵式建设与发展的软件人才培养的办学之路。特别是近几年来，学院结合北京市及学校大力开展的质量工程，以教育部特色专业建设、国家人才培养模式创新实验区建设以及北京市软件工程实践教学示范中心建设为契机，将课程建设、教材建设(包括实验教材)作为学院的一项重要工作来抓，并对精品课程建设以及配套的教材建设进行规划。如结合学院的师资队伍情况，以软件工程核心课程建设为突破口，开展软件测试的精品课程建设，并与清华大学出版社合作，陆续推出软件测试系列教材(《软件测试大学教程》、《软件测试实验指导教程》，《嵌入式软件测试教程》等)。蔡建平教授为《软件测试大学教程》配套编著的《软件测试实验指导教程》就是在这种背景下完成的。

由于在软件开发过程中，不论采用什么技术和方法，只要存在着对人的依赖，软件出错就是不可避免的。采用新的语言、先进的开发技术、现代的开发方法、完善的开发过程，可以减少错误的引入，但不可能完全杜绝软件中的错误，而这些错误必须通过测试来发现。

软件测试是软件开发的重要部分，是软件工程学科中一门工程性极强的课程，对实验或实践的教学要求特别高。由于软件测试内容覆盖面广，软件测试种类或测试类型繁多，因而对支持实践教学的软件测试工具要求也很高。如果依靠商用软件测试工具开展软件测试的实践教学，要么面太窄，要么投入太高，并且很难对在这方面取得的实践教学成果进行推广普及。针对这些问题，蔡建平教授以主流的开源软件测试工具作为软件测试实践教学的基础，并在《软件测试实验指导教程》中全面、系统地介绍了软件测试实践教学的方

法、步骤和案例，很好地解决了软件测试实践教学面临的教材问题、案例问题和实践环境建设问题等。

蔡建平教授长年从事软件工程、软件测试以及软件质量保证的研究、实践和教学，并为编写此书做了长期的准备。蔡建平教授以他多年在软件测试领域开展工作的经验和对软件测试能力培养如何满足 IT 企业要求的了解，设计了软件测试独特的实践教学方法，并深受学生认可和欢迎。作为蔡建平教授多年在软件测试实践教学上的经验和成果总结，《软件测试实验指导教程》的出版发行，将有益于国内软件测试人员和软件工程相关专业本科生及研究生的学习与实践能力培养，有益于推动我国高等院校软件测试实践教学方法研究的进一步发展，同时对我国软件测试业的发展和软件测试紧缺人才的培养起到积极的促进作用。

北京工业大学 教授、副校长

教育部高等学校软件工程专业教学指导分委员会 副主任

侯义斌

前　言

软件测试是一门对工程实践要求极高，对学生动手能力要求极强的软件工程核心课程。目前许多高校不同学历教育的计算机专业或软件工程专业均开设了这门课程，并配套有大量学时的实验课程或额外配套的课程设计实践课程。

事实上，如何开展软件测试的实验或实践教学，不同的学校，不同的授课老师或实验指导老师都有着各自的高招或各自的体系，但我想这些学校或这些教师也都有着大致同样的感受或痛苦。这是因为软件测试贯穿软件工程整个软件生命周期，涉及各种软件开发技术、应用技术以及测试技术，覆盖软件各种应用领域，需要用到多种测试技术、方法和测试类型。而要建设能够让学生全面参与这些实验或实践的软件测试实验室，特别是全面配置商用软件测试工具，几乎是不可能的。首先不要说进行一次性的投入以满足数十位学生同时上机进行实验或实践的要求，就是经常性的升级维护，就会让条件优越的高校接受不了，更不要说大多数条件并不优越的高校了。

因此，本人在多年从事软件工程、软件测试以及软件质量保证的研究和多年讲授软件测试课程、指导学生进行软件测试实践的经验和体会的基础上，对软件测试实践教学进行了重大改革和创新：以主流的开源软件测试工具应用为基础，以实战能力培养为目的，有组织地开展软件测试的实践教学活动。经过多年的软件测试课程设计活动组织、实践指导和成绩考核，结果表明：学生对这种形式的实践教学反映非常好，学生参与实验和实践的热情非常高，学生经过 2 到 3 周的实践后，收获非常大、能力提高非常快，学生为实践收集或设计的案例非常丰富，特别是部分学生撰写的实践总结报告水平也非常高。为此，本人萌生了总结这几年实践教学的经验、案例和成果，编写软件测试实践教材的想法，并期望通过该教材将本人在软件测试实践教学上的经验和成果进行推广，使广大的教师和学生受益。该想法得到了学校、学院以及清华大学出版社的大力支持，并促成了本书的编写和完稿。

目前国内为软件测试理论课程配套的实验教程很少，即使有也大多是以商用软件测试工具的使用介绍为主，正如前面所述，很难用于实践教学。本书是前期出版的《软件测试大学教程》配套的实验教程，是作者在总结多年软件测试工作和软件测试理论教学及实践教学的经验、有关技术以及测试案例积累的基础上编写的。本书充分考虑了国内大多数院校办学条件不足，实验教学经费有限，无法全方位引进商用软件测试工具，无法开展软件测试实验室建设的实际情况，对国内外主流的开源软件测试工具进行全面分析、研究和优选，并经过几轮实践教学的检验，来设计本书的实验教学重点和实践能力要求。本书的实验内容之广，涉及的软件测试知识之多，以及开源软件测试工具介绍之全面，无论是对于学生学习，教师进行实验指导还是培训机构开展实战训练都是不可多得的实验教材。

本书以现代 IT 企业软件测试需求为背景，以主流的软件测试技术和方法为基础，以

当前软件测试通常应用为典型实例，全面介绍了支持各种软件测试类型的开源软件测试工具的主要功能、应用流程及实际案例。与国内常见的软件测试实践教材重点讲授一般商用软件测试工具的方法不同，本书特色在于：

- 本书实验或实践内容完整、全面，涉及到测试管理、缺陷管理、代码分析、单元测试、系统测试、性能测试以及软件综合评测等重要内容，所选择的开源测试软件实践内容覆盖软件测试的各个测试阶段和各种测试类型，保证了国内许多院校在办学条件不足，实验教学经费有限，无法全方位引进商用软件测试工具的情况下，仍然能够开设软件测试实践类课程。
- 本书既对软件测试基本知识以及相关的测试方法和技术进行一般性的总结或介绍，又对支撑这些方法和技术应用的开源软件测试工具进行了全面介绍，并特别突出了软件测试工具在实际测试项目中的运用，能够使学生有效地巩固所学的软件测试知识，掌握软件测试方法和技术，以提高他们的软件测试实战能力。
- 本书重点给出了主流的软件测试工具如何建立测试环境，如何用于实际软件项目的测试。这种举一反三、抛砖引玉的内容设计，对高校软件测试实践类课程的开设、培训机构软件测试的实战培训以及开发人员和测试人员自学是非常有现实意义的。
- 本书特别强调了软件综合评测的意义，并以南京大学研制的 EASTT 工具应用为实例，全面介绍了软件评测的思想和过程，有助于学生全面掌握软件评测的方法和技术，提高他们实际的软件评测能力。
- 本书内容全面、条理清晰、结构严谨、可用性强，对重点、难点阐述透彻，使其即符合现代软件测试技术发展的潮流，又具有相对的稳定性，还易于剪裁或扩充，能满足各类机构软件测试实践教学的需要以及各类软件测试人员学习和实践的需要。

本书的完成得益于许多学生的积极参与。首先，我要在这里特别感谢我指导的研究生乔丽平、路翠等同学，本书的撰写和完稿花费了他们很多心血，如资料的收集、许多章节的起草、部分内容的组织、实验的完成以及图表的制作等。没有他们的积极参与，本书的完稿时间可能会受到影响。另外，本书很多内容来自于软件工程专业 03 至 06 级学生完成的软件测试课程设计实践总结报告，如 03 级的安文怡、李征和刘欣宇等，04 级的孙建和刘茜等，05 级的杨天放、时永欣、赵京超和周丰等，06 级的黄飞和霍晓珍等，以及限于篇幅无法列举的其他学生，在此也一并向他们表示感谢。

本书相当部分的内容是对互联网资料进行收集、整理和改编的结果，包括软件测试专业网站、软件测试专家或工程师的博客以及软件测试人员在网上论坛的经验之谈。由于本书的所有知识点和实验内容是建立在开源软件(如开源软件测试工具)的基础上，而开源软件最大的特点就是广大开源软件爱好者的无私奉献。因此，本人在这里表示对他们的敬意和感谢。本人用到的大部分素材是有意义的，尽管他们中的很多人可能很平常——不是学者或专家，但他们从实际工作或项目中总结出的经验、体会或感想，对软件测试实践教学是很有帮助的。本人在用这些素材时，只是对它们中存在的原则性错误、语法错误和文字错误等进行了简单的修正和改编。当然，本人也要在这里表示一点遗憾，很多网上资料由于转载或引用频繁找不到原创处，在参考文献中无法标注。

最后，我要感谢我的家人，本书的撰写全部是利用这个暑期的时间完成的(“闭门造书”)，没有家人的支持和照顾，本书也是很难完成的。

尽管本书是在本人长达二十几年软件工程、软件测试和软件质量保证实践经验和教学经验的基础上借鉴前人的成果，调研当前 IT 企业对软件测试人才能力的要求，并经过几轮软件测试实践教学的探索和总结编写的。但由于软件测试覆盖面太大，涉及领域太多，开源测试工具种类繁多、应用复杂，加之时间紧、水平有限，一定有许多不周到、不准确或存在错误之处，恳请广大读者提出批评和建议，并争取再版时修正。

北京工业大学软件学院　蔡建平

2009 年 8 月 20 日 于北京工业大学

目　录

第Ⅰ部分　管　理　篇

第Ⅱ部分　静态分析篇

第III部分　单元测试篇

第IV部分　图形用户界面测试篇

第Ⅰ部分　管　理　篇

软件工程除了技术外，最重要的思想之一就是管理。软件测试作为软件工程的一个重要分支，其目标是保证软件生命周期中每个阶段的活动结果是正确的，即现代软件测试思想——全生命周期软件测试思想。软件测试管理是软件测试质量的重要保证手段，它要解决的问题是如何确保软件测试技术能使软件项目在软件生命周期内得到顺利实施，并产生预期的效果。

事实上，随着技术的发展，软件系统的规模急剧增大，采用国际协作的模式，由位于世界上不同国家不同城市的多个团队联合开发软件系统，已经成为目前软件开发的主要趋势。与之相适应，测试也需要物理上分布的多个团队共同参与。由于各个团队承担不同的任务，并有着不同的项目管理模式，为保证整个系统能得到一致、有效的质量控制，测试管理至关重要。测试管理有助于系统、规范地管理各种测试资源和测试活动，以提高测试的效率和质量。按照管理对象的不同，软件测试管理大致分为软件测试团队组织管理、软件测试计划管理、软件缺陷(错误)跟踪管理以及软件测试资源管理这 4 大部分。

软件测试团队组织管理，通俗地讲就是测试团队应该如何组建、人员应该如何分工与管理以及绩效应该如何考核等。

软件测试计划管理，通俗地讲就是安排好测试流程。这部分内容具体涵盖软件测试策划、软件测试技术剪裁、测试进度管理、测试成本管理等几个部分。其中：软件测试策划工作主要是指在具体测试活动实施之前做好策划工作，如起草测试大纲和测试计划；软件测试技术剪裁工作主要是指测试团队应根据软件项目的具体情况剪裁出所要实施的测试技术；测试进度管理工作主要是指排出各项测试的时间进度及人员安排，如有变动则应做相应调整；测试成本管理工作主要是开列出测试活动中会涉及到的资源需求。

软件测试团队组织管理和软件测试计划管理属于软件项目管理的范畴，可根据软件测试的特点和 GB/T 15532-2008 计算机软件测试规范以及 GB/T 9386-2008 计算机软件测试文档编制规范来开展相关的管理工作。

软件缺陷(错误)跟踪管理，通俗地讲就是确保发现的缺陷(错误)已经被开发团队纠正或处理过，并且没有引入新的缺陷(错误)。具体来讲，当测试团队通过各种途径发现了文档或代码中的缺陷或错误以后，并不是交一份测试报告就草草了事，而是在递交报告以后继续督促开发团队及时关闭已知缺陷或错误(当然，如有必要，应对这些缺陷、错误做严重程度排序，以便开发团队能视轻重缓急安排处理顺序)。当开发团队关闭了测试报告中的缺陷(错误)以后，测试团队还需验证开发团队在关闭过程中有没有引入新的错误。通常，这个过程称为回归测试。回归测试如发现问题，则继续报告给开发团队，按上述流程循环，直

至回归测试最终通过。

软件测试资源管理，通俗地讲就是努力建设好测试团队的测试资源库，并对测试团队成员进行技能培训以帮助他们使用好这个测试资源库。软件测试资源库所包括的内容是测试团队在长期实践过程中逐步积累起来的经验教训、测试技巧、测试工具、规格文档以及一些经过少量修改便能推广至通用的测试脚本程序。测试资源管理工作做得越好，测试团队在实际测试过程中就越能少走弯路，测试团队内部的知识交流和传递就越充分，测试脚本或规格文档的重复开发工作也就越能被有效地避免。软件测试资源管理工作包括两部分：一个是建设，另一个是培训。建设工作大抵是收集各类测试文档、测试工具、测试脚本，也包括收集整理测试人员的会议发言、总结报告、技术心得等等。培训工作大抵是通过技术讲座、正式或非正式团队会议、印发学习资料等形式进行。

软件测试管理工具是软件测试管理最重要的保证手段，对于大型系统测试来说，测试管理工具可以帮助组织测试资产、监督项目状态、集成自动化测试工具以及度量测试效果，能够为所有这些参与者提供一个交流和协作的平台，是项目管理中必不可少的。近年来，测试工具的应用越来越普遍，很多工具都能提供一定程度的测试管理功能。当然，商用软件测试管理工具不论从能力还是从成熟度来说都是有较强竞争力的，在条件许可的情况下应该作为首选。但是，开源软件测试管理工具发展越来越迅速，功能越来越强大，应用越来越普及，是进行软件测试管理实践教学的最佳解决方案之一。本部分以开源为基础，重点讲解有关的软件测试管理工具。

第1章　软件缺陷管理

软件开发是引入软件错误或软件缺陷的过程，软件测试则是发现软件错误或软件缺陷的过程。对于大型软件来说，错误数目是非常可观的，必须借助工具才能对所发现的这些错误进行有效的管理，为软件缺陷或错误的消除或者软件质量的评价及软件开发的决策提供依据。

1.1　软件缺陷管理概念

在当今社会中，软件测试变得越来越重要，软件缺陷管理的重要意义自然是不言而喻。首先我们了解一下什么是软件缺陷，软件缺陷是指系统或系统部件中那些导致系统或系统部件不能实现其功能的缺陷，具体而言就是指在程序或文档中存在各种不希望出现的问题，是对软件产品预期属性的偏离现象，如语法错误、拼写错误、标点错误，或者是一个不正确的、冗余的程序语句或有缺陷的程序段等。缺陷可能出现在程序中、设计中，甚至出现在需求规格说明中或其他文档中。

软件缺陷是软件“与生俱来”的特征。不管是小程序还是大型软件系统，无一例外地都存在缺陷，而且无法完全避免。这些软件缺陷，有的容易表现出来，有的隐藏很深难以发现，有的对使用影响轻微，有的会造成财产甚至生命的巨大损失。因此，软件缺陷是影响软件质量的重要和关键因素之一，发现与排除软件缺陷是软件生命周期中的重要工作之一。每一个软件组织都知道必须妥善处理软件中的缺陷，因为这是关系到软件组织生存、发展的质量根本。

影响软件缺陷数目的因素有很多。在不同的软件阶段，软件的缺陷密度是不同的。从宏观上看，包括管理水平、技术水平、测试水平等；从微观上看，包括软件规模、软件复杂性、软件类型、测试工具、测试自动化程度、测试支撑环境、开发成本等。初始的软件缺陷密度一般是靠经验来估计的。

软件的缺陷是多种多样的，从理论上看，软件中的任何一个部分都可能会产生缺陷，而这些缺陷的来源不外乎下列四个方面：疏忽造成的错误、不理解造成的错误、二义性造成的错误、遗漏造成的错误。其中前三类缺陷主要存在于软件开发的前期阶段，如需求分析阶段、设计阶段、编码阶段。

总的来说，缺陷是错误、故障和失效的根源。缺陷的引入、发现、修复和关闭贯穿于整个软件开发过程当中，要获得高质量的软件，就必须有效地管理缺陷，对缺陷进行预防、检测并主动清除。虽然缺陷无法完全避免，但缺陷管理可以帮助开发者尽可能避免缺陷。使开发者能把精力投入到最可能出现缺陷的地方，这样才能有效地提高软件质量。

对于缺陷的管理，必须建立一个比较完整的缺陷信息，如果缺陷信息不完整，就难以对已有缺陷进行分析处理，就无法科学地评估软件的质量并发现软件产品和软件过程的待改进之处，不利于相关人员日后作为经验教训的积累和查询一个完整的缺陷信息。缺陷信息应该包括：缺陷的状态、严重性、所在模块、软件版本、类型、详细描述、阶段、提交者、测试者、测试日期和时间等。

1.1.1　缺陷描述与分类

缺陷管理的关键是在仔细分析缺陷信息的基础上准确地描述缺陷，并对缺陷进行分类。

软件缺陷的描述是否准确、简单、专业，对软件测试报告有着重要的影响，如果对软件缺陷的描述含糊不清，描述术语晦涩难懂，则可能误导测试人员，从而无法得到准确的测试结果。因此对软件缺陷的描述需要做到：单一准确，可以再现，完整统一，短小精练，特定条件，补充完善，不做评价。

- 单一准确是指：每个报告只针对一个软件缺陷。在一个报告中报告多个软件缺陷常常会导致只有部分缺陷能引起注意和修复，而不能使所有缺陷得到彻底的修正。
- 可以再现是指：要提供缺陷的精确操作步骤，使开发人员容易看懂，可以自己再现这个缺陷。一般来讲，开发人员只有再现了缺陷，才能正确地修复它。
- 完整统一是指：提供完整、前后统一的软件缺陷的步骤和信息。
- 短小精练是指：通过使用关键词，既可以使软件缺陷的标题描述短小简练，又能准确解释缺陷产生时的现象。
- 特定条件是指：许多软件功能在通常情况下没有问题，但是在某特定条件下会存在缺陷，所以软件缺陷描述不要忽视这些看似细微但又是非常有必要的特定条件，要能够提供帮助开发人员找到原因的线索，如“搜索功能在没有找到结果返回时跳转页面不对”。
- 补充完善是指：从发现 bug 的那一刻起，测试人员的责任就是保证能正确地报告出 bug，并且得到应有的重视，继续监视其修复的全过程。
- 不做评价是指：软件缺陷描述不要带有个人观点，不要对开发人员进行评价。软件缺陷报告是针对产品、针对问题本身，将事实或现象客观地描述出来就可以了，不需要有任何评价或议论。

软件缺陷描述中最重要的一点是要完整地描述出软件缺陷的属性。软件缺陷的属性包括缺陷标识、缺陷类型、缺陷严重程度、缺陷产生可能性、缺陷优先级、缺陷状态、缺陷起源、缺陷来源、缺陷原因。其中缺陷严重程度主要分为：致命，严重，一般，较小。致

命是指系统任何一个主要功能完全丧失，用户数据受到破坏，系统崩溃、悬挂、死机或者危及人身安全；严重是指系统的主要功能部分丧失，数据不能保存，系统的次要功能完全丧失，系统所提供的功能或服务受到明显的影响；一般是指系统的次要功能没有完全实现，但不影响用户的正常使用，较小是指使操作者不方便或遇到麻烦，但并不影响功能过程的操作和执行，如不影响产品理解的个别错别字、文字排列不整齐等一些小问题。

软件缺陷的来源一般包括：需求说明书、设计文档、系统集成接口、数据流(库)、程序代码。需求说明书错误、不准确、不清楚会引发缺陷；设计文档描述不准确，与需求说明书描述不一致会引发缺陷；系统模块参数不匹配、开发组之间缺乏协调会引发缺陷；数据字典、数据库中的错误会引发缺陷；在编写代码中所触发的问题也会引发缺陷。而软件缺陷的根源在于：测试策略，过程、工具和方法，团队，缺乏组织和沟通，硬件，软件，工作环境等。

软件缺陷分类是对软件缺陷进行有效管理的基础。根据软件缺陷的自然属性，我们可以将软件缺陷分为功能、用户界面、文档、软件包、性能及系统/模块接口缺陷。其中，功能缺陷是指影响了各种系统功能，是逻辑缺陷；用户界面缺陷是指影响了用户界面、人机交互特性，包括屏幕格式、用户输入灵活性、结果输出格式等方面的缺陷；文档缺陷是指影响了发布和维护，包括注释、用户手册、设计文档方面的缺陷；软件包缺陷是指由于软件配置库、变更管理或版本控制而引发错误所导致的缺陷；性能缺陷是指不满足系统可测量的属性值，如执行时间、事务处理速率等；系统/模块接口缺陷是指因与其他组件、模块或设备驱动程序、调用参数、控制块或参数列表等不匹配、冲突而引起的缺陷。

软件缺陷优先级一般分为：立即解决、高优先级、正常排队、低优先级。而软件缺陷的状态一般又分为：激活或打开、已修正或修复、关闭或非激活、重新打开、推迟、保留。

通过分类，测试人员可以迅速找出哪一类问题最大，然后集中精力预防和排除这类缺陷。把精力集中到最容易引起问题的几类缺陷上，一旦这几类缺陷得到控制，就可以进一步找到新的容易引起问题的几类缺陷上。及早发现并修复缺陷，将会极大地促进软件生产。

1.1.2　缺陷管理流程

软件缺陷管理是对缺陷进行登记、处理的过程。通过字典定义、输入界面的约束，来达到一致的缺陷编写要求，实现缺陷信息处理的规范化。对于缺陷登记、缺陷修复登记、缺陷回归测试登记形成的缺陷信息，通过软件自动统计，可以迅速获得任意时间段内、任意系统模块、任意缺陷级别、任意缺陷状态等各种组合的缺陷信息及缺陷统计数据，实现查询、统计的自动化。

使用缺陷管理工具收集缺陷数据，并将测试结果数据与在测试过程中总结的 Excel 模板相结合，可以制作出各种缺陷分析图表。通过分析缺陷生成趋势图来指导软件发布时机，根据缺陷趋势曲线来确定测试过程是否结束，这是最常用并且较为有效的一种方式。例如，在发现缺陷数量未收敛时发布软件，显然风险会很大。当然，使用图表时还应结合实际。

例如在曲线平坦时，考虑是否有效开展了测试工作；在曲线上升时，缺陷的严重性是否很低等。

通过关于严重性等级的柱状图可以分析被测系统的总体状况，从而预测项目风险或解释测试结果。通过导致缺陷产生的原因分布图，可以将测试注意力集中到引起最严重、最频繁问题的领域，从而消耗最少的资源改进过程，以取得最显著的成果。

缺陷管理流程需要结合缺陷生命周期，缺陷生命周期是指从发现缺陷到完成缺陷处理的过程。根据对国内外著名 IT 公司缺陷管理流程的研究，总结出一般的软件缺陷管理流程，如图 1-1 所示：

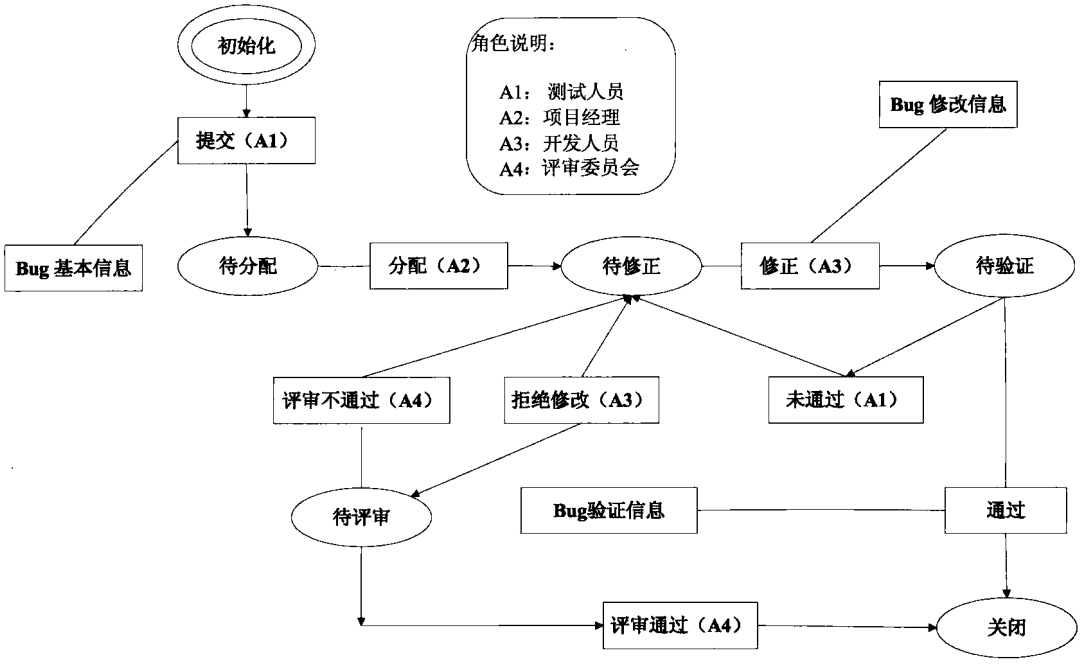


图 1-1　软件缺陷的一般管理流程

缺陷管理流程中有以下 4 个角色：

- 测试人员(A1)：进行测试的人员，缺陷的发现者。
- 项目经理(A2)：对整个项目负责，对产品质量负责的人员。
- 开发人员(A3)：执行开发任务的人员，完成实际的设计和编码工作及缺陷的修复工作。
- 评审委员会(A4)：对缺陷进行最终确认，当项目成员对缺陷无法达成一致意见时，行使仲裁权力。

缺陷管理流程中包含以下 6 种缺陷状态：

- 初始化：缺陷的初始状态。
- 待分配：缺陷等待分配给相关开发人员处理。
- 待修正：缺陷等待开发人员修正。
- 待验证：开发人员已完成修正，等待测试人员验证。