HZ BOOKS

CSAI.cn 希赛® IT技术讲堂

自动化软件测试

张 瑾 杜春晖 等编著

- 自动化软件测试是测试人员的福音。
- 全面解析了自动化软件测试的模型、策略与工具，并配以项目案例说明，引领读者进入项目实战的天堂。
- 系统地介绍了QTP、NUNIT、LoadRunner、Test Director、VSS、Nant等自动化软件测试及相关工具的使用。
- 随书光盘中带有书中全部范例的脚本和项目案例的源程序，供读者学习参考使用。
- 作者在希赛网社区（http://www.csai.cn）“书评在线”版块中为读者提供全方位学习指导。

机械工业出版社
China Machine Press

自动化软件测试

张 瑾　杜春晖　等编著

机械工业出版社
China Machine Press

本书主要介绍软件测试知识以及测试工具的使用方法。本书使用的配置环境是 Windows 2003 和 . Net Framework 2. 0，并逐步引领读者学习基础知识和各个工具的使用技法，最后将其贯穿并设计了符合软件企业特点的自动化测试流程。全书内容由浅入深，并辅以大量的实例说明，最后给出了一个完整的项目案例。

随书光盘中含有本书所有实例的脚本，以及项目案例的源程序，供读者学习参考使用。

本书供有一定软件测试基础的测试人员使用，也可作为软件测试职业培训教材使用，对于缺乏软件测试知识和经验的爱好者来说，可以迅速对软件测试拥有一个全面清晰的认识，并积累实战经验。

图书在版编目(CIP)数据

自动化软件测试/张瑾，杜春晖等编著. —北京：机械工业出版社，2008. 1
(希赛 IT 技术讲堂)

ISBN 978-7-111-23182-0

Ⅰ. 自…　Ⅱ. ①张…　②杜…　Ⅲ. 软件－测试　Ⅳ. TP311. 5

中国版本图书馆 CIP 数据核字(2008)第 206221 号

机械工业出版社(北京市西城区百万庄大街 22 号　邮政编码　100037)
责任编辑：李南丰
北京牛山世兴印刷厂印刷 · 新华书店北京发行所发行
2008 年 1 月第 1 版第 1 次印刷
184mm × 260mm · 17. 75 印张
标准书号：ISBN 978-7-111-23182-0
　　ISBN 978-7-89482-516-2(光盘)
定价：39. 00 元(附光盘)

凡购本书，如有倒页、脱页、缺页，由本社发行部调换
本社购书热线：(010)68326294

编写委员会

组编：希赛顾问团

顾问：张友生

主审：邓子云

编委（按姓名拼音排序）：

边　伟	陈亿春	崔海波	方海光	冯向科
葛志春	郭　莹	赫　斌	黄　婧	黄少年
李　刚	刘　毅	陆秉炜	刘志成	娄嘉鹏
聂艳明	阮国明	孙鸿飞	施　游	唐　俊
唐天广	王　军	吴吉义	吴伟敏	薛大龙
王红安	王　冀	王　勇	谢　顺	杨　森
张晓燕	张立东	朱小平		

前　言

现在许多软件企业开始重视软件测试，越来越多的软件技术人员开始投身测试行业。在一些大型软件公司里，软件测试甚至比开发投入的资源还要多。目前，国内的软件测试人员大多数还停留在简单、重复的黑盒手动测试阶段，软件测试人员也常常觉得自己所从事的工作和计算机技术关系不大，只要了解产品的业务逻辑就可以完成工作。这也导致很多测试人员对本身的工作失去兴趣，同时也给企业管理者造成了“测试工作是没有技术含量的工作”的误解。

近年来自动化测试技术逐渐进入软件测试人员的视野。通过对比，人们逐渐发现：软件测试和软件开发一样具有挑战性、有技术含量。开发人员有自己的开发工具，软件测试人员同样也有像 QTP、LoadRunner、Rational Robot 等测试工具；开发人员有例如 Java、C#等语言来编写代码，测试人员也有测试专用的语法来编写脚本、调试脚本；开发人员生产出来的产品可以为企业直接创造效益，测试人员通过质量手段防止更多的缺陷遗留给客户。因此软件测试再也不是一般非专业人员所能够胜任的，而是和软件开发一样具有技术含量、前景美好的职业。

随着 CMMI、ISO 等质量体系在国内的推广，软件企业逐渐对软件质量有了正确的认识。软件的缺陷是在生产过程中产生的，软件测试人员只能被动地进行检查、避免缺陷落入客户手中，而不能在真正意义上避免缺陷的产生，要想提高产品的质量就要通过全员的培训和过程改进来实现。一旦企业的管理人员真正认识到这一点，那么软件测试人员就不再是替罪羊，软件测试人员辛苦的劳动也就会得到真正的认可。

1. 本书的知识体系

学习 CART 全面的软件自动化回归测试流程最好要有一定的开发基础，另外最好对软件工程中的质量体系、配置管理、度量管理有所了解和认识。本书的知识体系结构如图 1 所示。本书以循序渐进的方式从理论知识讲起，然后介绍各个工具的使用方法，最后将其融会贯通于项目之中。

2. 章节内容介绍

本书分为三篇。第一篇基础知识分为 6 章，分别讲述软件工程和 CART 的基础知识。优秀的软件测试人员不能只了解测试技术，应该对软件工程的各个部分都有所认识，在大学学习的软件工程理论基本都比较浅显，如果读者没有经历过 CMMI 或 ISO 的专业培训，那么可以通过阅读本书的第 1、第 5、第 6 章的内容来增加了解。

第 1 章引导读者了解软件质量的基础知识，通过理解各位质量大师的观点来领悟软件测试的含义。在此基础上再对软件测试的几种常用方法进行讲解，使读者对软件质量的各个方面有所认识。

第 2 章讲述 CART 全面自动化回归测试流程的理论。通过与 TDD 开发模型的比较，使读者了解 CART 是将日构建技术和自动化回归技术相结合，采用白盒与黑盒互补的方式对产品进行彻底的自动化测试。

第 3 章具体讲述日构建的策略，使读者加强对构建过程的日常性和重要性的理解，为日构建过程在企业中的推广奠定理论依据。

第4章的内容是回归测试的策略，“回归测试”这个名词对软件测试人员应该都不陌生，但为什么要进行回归测试，进行回归测试有什么好处，很多测试人员的理解都不够深入，本章对其进行了全面的讲解。

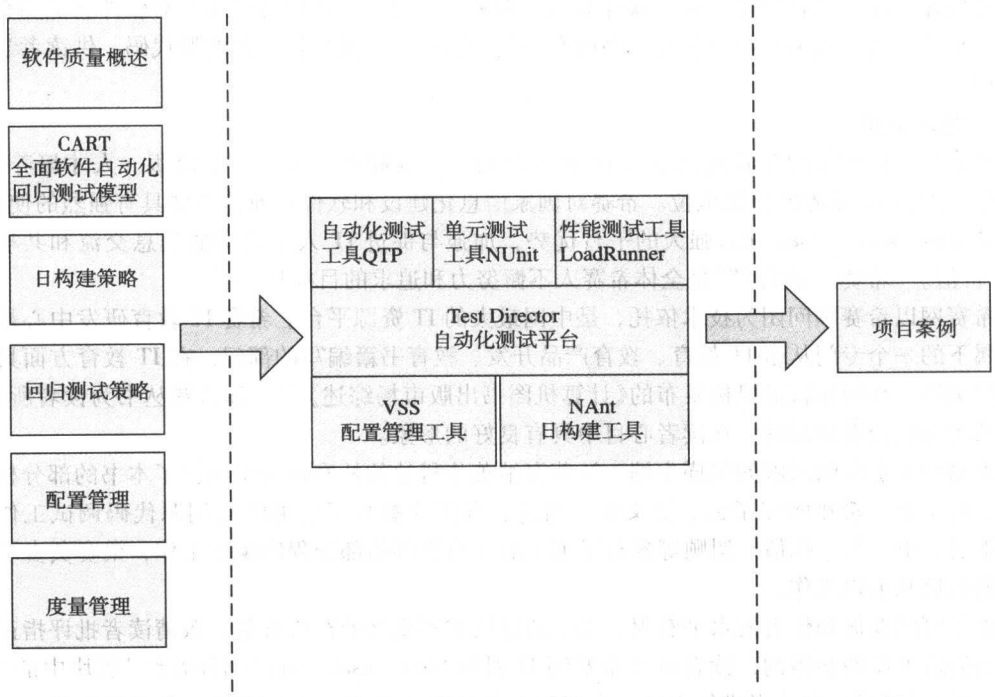

图1　本书的知识体系结构

第5章讲解与配置管理相关的知识，配置管理是软件工程中的基础环节，也是CART全面自动化回归测试流程的基础，日构建过程其实就是配置管理范畴的一个部分。更好地理解配置管理可以使软件测试人员的日常工作条理清晰，和开发人员的沟通更加准确。

第6章的内容是度量管理，SQC软件测试人员的一个重要发展方向就是SQA软件质量保证工程师，软件测试人员要学会从测试结果中进行分析，灵活使用因果分析和帕累托理论来找到问题的根源，选择重要的问题优先解决，从源头减少缺陷的产生，从源头提高产品的质量。

根据以上理论知识，本书为读者选取了一套CART全面自动化回归测试流程所使用到的工具进行逐一讲解，使读者可以迅速掌握。第二篇为工具篇，分为3章，共介绍了7种测试工具和测试管理平台。

根据第5章配置管理的理论，在第7章选取了3种配置管理工具介绍给读者。SubVersion和TortoiseSVN的组合可以对公司范围内的文档进行管理。通过对NAnt工具的语法和指令的详解，将第3章日构建策略理论应用到实际的工作中。

第8章介绍了Test Director的配置和使用，读者可以通过对该平台的学习来设计一套符合本公司特点的测试管理流程，并且根据第6章的内容设计并收集测试度量的信息。

第9章以QTP、LoadRunner、NUnit三个测试工具为例，详细讲述了在测试领域内的回归测试、性能测试、单元测试的做法和应用，使读者对其增加更多的了解。

通过第一篇理论知识和第二篇工具使用技巧的学习，第三篇项目案例按照CART全面回归测试流程的要求，将以上知识和工具进行贯穿。以项目实例为主线进行讲解，使读者对其功效进行全面了解。

本书以微软 .Net 开发环境为基础，选取了相应的测试工具和日构建工具。在 Java 环境下进行开发和测试的读者可以举一反三，选取相应的 Ant、JUnit 等工具来实现该自动化测试流程。

本书内容由浅入深，并辅以大量的实例说明，可以作为软件测试人员的参考用书，也可以作为软件职业培训的教材使用。缺乏软件测试知识和经验的读者可以通过迅速对软件测试拥有一个全面清晰的认识。随书光盘中含有本书所有实例的脚本，以及项目案例的源代码，供读者学习参考使用。

3. 技术支持

希赛是中国领先的互联网技术和 IT 教育公司，在互联网服务、图书出版、人才培养方面，希赛始终保持 IT 业界的领先地位。希赛对国家信息化建设和软件产业化发展具有强烈的使命感，利用希赛网(www. csai. cn)强大的平台优势，加强与促进 IT 人士之间的信息交流和共享，实现 IT 价值。“希赛，影响 IT”是全体希赛人不懈努力和追求的目标！

希赛网以希赛顾问团为技术依托，是中国最大的 IT 资源平台。希赛 IT 教育研发中心是希赛公司属下的一个专门从事 IT 教育、教育产品开发、教育书籍编写的部门，在 IT 教育方面具有极高的权威性。在国家权威机构发布的《计算机图书出版市场综述》中，称希赛丛书为读者所称道，希赛的图书已经形成品牌，在读者心目中具有良好的形象。

本书由希赛顾问团顾问张瑾主编。江苏海纳英华科技的杜春晖先生承担了本书的部分校验和技术支持工作。希赛网邓子云、扶文奇、周进、肖佳等参与了全书的实例源代码调试工作，王冀、王勇、史小琴、陈倩、谢顺等参与了书中的项目案例的部分程序编制工作，梁赛负责了部分章节的校稿和编辑工作。

由于时间仓促和作者的水平有限，书中的错误和不妥之处在所难免，敬请读者批评指正。有关本书的意见反馈和咨询，读者可在希赛网 IT 社区(bbs. csai. cn)“书评在线”版块中的“机械工业出版社”栏目中与作者进行交流。本书配套光盘中的内容，读者可以在希赛网下载中心(data. csai. cn)下载。

4. 致谢

感谢广州德捷科技的领导和各位同仁，特别是测试团队全体成员；感谢机械工业出版社的陈冀康编辑，他承担了大量的策划与编辑工作；感谢希赛公司的邓子云先生和梁赛女士，他们给本书的编写提出了许多修改意见；借此还向我的夫人蔡觅致敬，她一直默默地支持我书稿的创作工作。正是因为这么多人的大力支持和辛勤汗水，本书才得以出版。

张　瑾

2007 年 8 月于广州

目　录

第一篇

基础知识篇

第1章 软件质量概述

核心内容及学习重点：

- 了解各位质量大师的理论。
- 区分质量控制和质量保证。
- 理解质量的成本由哪些组成部分。
- 明确质量的分工及责任。
- 掌握质量的分析工具及方法。
- 掌握测试的常用方法。

为实现国家产业结构的调整，我国正从劳动密集型产业向知识密集型产业转移。国内软件企业经历了无数坎坷后，正以每年百余家的数量通过各级 CMMI 认证，逐渐由过去的作坊式开发过渡到如今的工厂式规模运作，软件工程的管理也越来越规范。

软件测试是软件工程的重要组成部分，又是软件开发不可或缺的环节。作为国内软件企业，要想在同行业树立自己的品牌，在国际软件外包业务中取得更多的市场份额，仅靠以往的人力资源优势已经不太现实，提高软件质量才是软件企业的唯一出路。

随着时代的发展，各企业的信息化程度越来越高，一旦软件在使用过程中出现问题，对客户的影响将无法估计。因此，越来越多的客户在软件无法正常使用的情况下，会向软件企业提出质疑，表示不满，甚至提出索赔。大浪淘沙，很多企业就这样悄然退出了历史的舞台。

如今软件企业已不能一味追求利润，而需要将更多的资源和精力倾注在软件质量上。为什么现在中国有那么多的软件企业每年投入大量资源和精力在 CMMI 认证上，因为 CMMI 可以给企业带来规范化的管理和过程，有了好的过程，才能产生好的结果。

那什么是软件的质量呢？它就是使企业开发出来的产品具备满足明确或隐含需求能力的所有特性的总和，也就是说要满足“需求规格说明书”上提出的或者隐含的所有需求，使项目相关人员对其认可。

质量和等级又是不同的概念。质量是项目范围内的所有交付成果对直接或间接需求的满足程度。等级是一种具有相同使用功能，不同质量要求的实体的类别或级别。例如，酒店都具有餐饮、住宿和娱乐功能，但其星级的标准是对其服务需求的划分，因此不同星级酒店之间的服务质量是没有可比性的。所以质量低通常是问题，但等级低就可能不是问题。

读者还需要理解质量和范围的概念，并不是多为客户做些功能，就意味着企业的产品质量得到提高。恰恰相反，在项目管理中常称这种行为叫镀金(Gold Plating)。镀金是一种错误。它使得项目的范围进行了蔓延，特别是软件开发中，越多的功能也就意味着存在越多的隐患，反而增加了出错的几率，降低了产品的质量。

软件产品的质量目标可以从以下几方面考虑：

1)产品或者服务是否满足使用性。

2)产品或服务是否达到设计的目标。

3)产品或服务是否符合需求的需要。

4)产品或服务是否满足客户的期望。

为了让读者对“质量”这个词加深了解，下面通过一些著名质量大师的经典理论从不同侧面认识质量。

1.1　软件质量的理论

软件测试人员几乎天天都在与质量问题打交道，但到底如何来提高质量呢？很多人会说当然是通过测试了，接下来看看各位质量大师是怎样阐述的吧。

1.1.1　戴明理论

W·爱德华·戴明(W. Edwards Deming)博士于 1900 年 10 月 4 日生于美国爱荷华州，1928 年取得耶鲁大学的物理博士学位。戴明博士从 1950 年去日本讲学，并在日本持续了长达 40 多年的质量管理指导工作。他被日本企业称为“质量之神”。1980 年 6 月 24 日美国 NBC 广播公司播放了举世闻名的“If Japan Can，Why Can't We”的纪录片，使得戴明博士一夜成名，并赢得了“第三次工业革命之父”的美誉。

戴明的重要成就之一就是“PDCA 戴明环”理论，如图 1-1 所示。它将管理过程分为 4 个阶段，并且形成封闭的环路，以达到持续改进的效果。

1)Plan(计划)：制订提高质量的改进计划。

2)Do(执行)：执行计划。

3)Check(检查)：通过检查来判断是否达到期望的结果。

4)Action(纠正)：实施纠正行动。

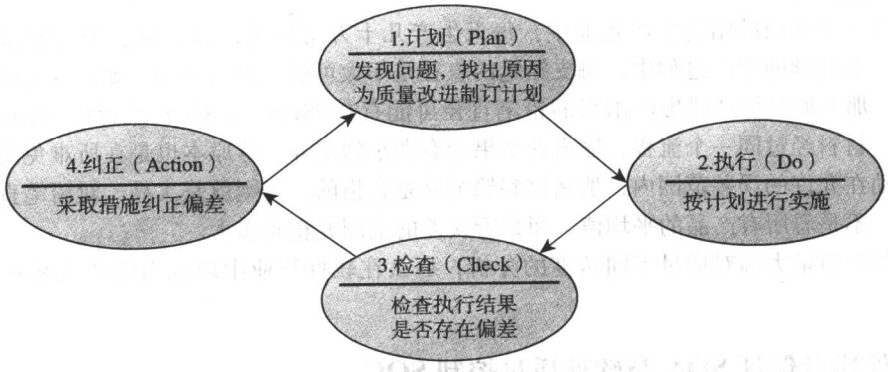

图 1-1　PDCA 戴明环

1.1.2　朱兰理论

有品质领域“首席建筑师”之称的约瑟夫·M·朱兰是继戴明之后对日本质量革命具有重大影响的大师。他最早把帕累托原理引入质量管理。1951 年出版的《朱兰质量手册》(Juran's Quality Handbook)被人们称为“质量管理领域中的圣经”。

朱兰认为至少 80% 的质量问题应该由管理负责，20% 的质量问题才是技术问题。他的重要成就就是被世人称颂的“朱兰三步曲”：

质量计划——为建立有能力满足质量标准的工作程序，质量计划是必须的。

质量控制——为了掌握何时应该采取措施纠正质量问题就必须开展质量控制。

质量改进——质量改进有利于发现更好的管理工作方式。

1.1.3　Crosby 理论

飞利浦·克劳士比(Philip Crosby)被誉为当代“伟大的管理思想家”、“零缺陷之父”、“世界质量先生”。他终身致力于“质量管理”哲学的发展和应用。1979 年出版的惊世巨著《质量是免费的》(Quality is Free)建议“组织向零缺陷努力”。

飞利浦·克劳士比认为质量就是符合要求的，而不是最好的；质量系统是用来预防的，而不是用来检验的；质量的标准是零缺陷，而不是差不多就好(Close enough is good enough)。

1.1.4　田口宏一理论

田口宏一因提出“坚固设计方法”而闻名。他认为“质量是设计出来的，而不是检查出来的”。这个观点在软件工程中也非常实用，软件产品的质量不能完全依靠测试人员，而是要从源头的需求调研、软件设计抓起。

田口宏一还提出了“质量最好是通过减少与目标的偏差来获得，产品应该对不可控的因素具有免疫力”的观点，另外，他认为应通过损失函数(Loss Function)来对产品质量进行衡量。

1.1.5　6∑理论

20 世纪 70 年代，摩托罗拉在统计学原理上建立了 6∑理论，该理论是以追求完美无暇为最终目标的管理理论，在非常多的大型企业中得到运用。西格马“∑”是一个希腊字符，代表“标准差”，在商业活动中，它代表流程与完美的差距。6∑的值是 99.999 66%，也就是说百万个产品中有三四个是次品。

例如在一家普通的铅笔生产企业中，每天生产几十万支铅笔，其中有一个重要参数就是铅笔的直径。在连续的生产过程中，每支铅笔的直径是不太可能一模一样的。如果规定的标准要求是 0.8cm，那么通过生产线生产出来的铅笔直径可能有 0.78cm、0.81cm 或者 0.79cm 等几种不同的数值。材料经过同一个流程，输出的结果会有微小的差异，这原本也是在所难免的。如果铅笔直径波动在允许的误差范围内，那么这些铅笔就是合格的，否则就是次品。对铅笔直径的控制可以看出，不是看所有产品的平均值，而是看误差的范围究竟有多大。

通过各位质量大师对质量不同角度的分析，那么在软件行业中应该由哪些人来从事质量工作呢?

1.2　软件质量保证 SQA 与软件质量控制 SQC

软件质量工作大体上可以分为软件质量控制和软件质量保证。

软件质量控制(Software Quality Control，SQC)通常是指软件测试，简单讲就是防止缺陷落到客户手中。SQC 用来监控项目的工作成果，以决定其是否符合各种质量标准，并识别消除各种缺陷。SQC 属于检查职能，是一个技术工种。

软件质量保证(Software Quality Assure，SQA)是为了确保软件开发的流程按照事先定义的规范开展，从而使软件质量得到提高。通常是以软件质量控制的结果作为其工作的重要输入，SQA 要保证在质量体系中实施全部的计划和活动，以保证质量得到提高。它是以非技术手段，从源头防止缺陷的产生，并贯穿项目的始终，属于管理职能。

SQA 人员在开发过程中往往起到监督和管理的作用。SQA 人员一般需要有丰富的技术和管理经验，不同能力的 SQA 可以在质量管理过程中扮演不同角色。有的可以扮演警察，只负责按照规范进行检查，及时地发现问题；有的可以扮演医生，发现问题解决问题；有的还可以扮演教

练，发现问题解决问题，并且指导他人如何避免问题的再次发生。

因此软件质量控制(SQC)只能被动性地修复缺陷，而不能防止缺陷的产生。软件质量的提高是靠软件质量保证人员来实现的，这样也就映证了田口宏一的理论："质量是设计出来的，而不是检查出来的"。

很多人都认为软件的价值是开发人员创造的，其他人员都不是利润的直接创造者，这其实是个误解，SQC 和 SQA 人员也一样在创造价值。

1.3　软件质量的成本

软件质量的成本是指为了获取产品或服务的质量所付出的所有努力的总成本。

可以按照成本的性质进行分类：

1)一致性成本就是"未雨绸缪"的成本。例如，软件测试的成本、代码走查的成本、同行审评的成本等。

2)非一致性成本就是"亡羊补牢"的成本。例如，返工、对投诉的处理、报废、缺陷等引发的成本。

缺陷成本还可以分为：

1)内部缺陷成本——缺陷、返工、设计错误等引发的成本。

2)外部缺陷成本——投诉处理、产品招回、保修服务等引发的成本。

SQA 和 SQC 是专职的质量人员，那么是不是软件的质量都应该由他们来负责呢，接下来要对质量的责任进行详细的分工和描述。

1.4　软件质量的责任分工

在软件开发过程中，承包方会经常遇到客户各种各样的投诉，总的来说这些都是和质量相关的。这就带来一个很敏感的问题，质量究竟应该由谁负责呢?

在项目中项目经理是团队的领导者，如果出了问题，客户比较喜欢找的人就是项目经理，因此他承担了非常大的压力。但项目经理往往也觉得委屈，明明有质量经理对测试进行把关，为什么受伤的总是我呢?

公司的高层经理同样会找来质量经理进行询问，让其提出改进方案。质量经理经过分析认为单元测试、功能测试、集成测试等手段都已使用，而且测试人员工作都十分努力，因此对高层经理就测试方式的质疑表示不能理解。质量经理与高层经理谈话回来后，找到负责该模块的测试人员进行询问，测试人员也十分委屈，自己辛苦工作并且经常加班，已经很尽力了，可自己的辛勤劳动并没有得到认可。仔细想想这些问题的发生也不能全部怨测试人员，开发人员没有给相关的文档，而导致对某些功能点测试得不够全面。

其实以上这个例子在软件项目过程中是经常发生的，企业经常会提出"质量问题，人人有责"的口号，但"人人有责"就意味着职责不明确，"人人有责"就变成了"人人不负责"。因此需要对质量的责任进行明确定义。

管理层应该对组织级的质量负责。应该对企业中长期的质量方针和质量目标进行定义，并组织员工学习理解并逐步落实。

项目经理应该对项目的质量负全面和首要的责任，因为项目经理是项目的领导者。作为项目和客户沟通的渠道，这个责任是在所难免的。

开发、设计、测试等人员应该对所做工作的质量负最终的责任，因为他们是产品的制造者。

质量经理应该对项目的测试流程和测试方法，以及测试用例和测试脚本的正确性和执行情况进行检验，指导测试人员工作，保证所有测试环节都得到全面执行。

软件工程过程组(Software Engineer Process Group，SEPG)应该对建立、改进相关开发规范和流程负有责任，保证管理流程以及各部门之间沟通协作的顺畅。

1.5　软件质量的分析工具

通过以上对软件质量的分工和成本进行分析，那么质量人员应该如何发现并解决质量问题呢？这里向读者介绍几种工具和方法。

1.5.1　因果图

因果图又称鱼骨图或石川图，它侧重表示潜在问题之间的各种相关因素，如图 1-2 所示。通过因果图可以刺激思维和讨论，以便对问题进行深入讨论，寻找潜在根源。

主要步骤：

1)确定讨论问题。

2)挑选人员进行讨论。

3)确定主要原因，画出问题框和主干箭线。

4)确定下一级主要原因。

5)确定各缺陷原因。

6)画出整个因果图。

7)分析原因及纠正行动。

8)确定纠正行动。

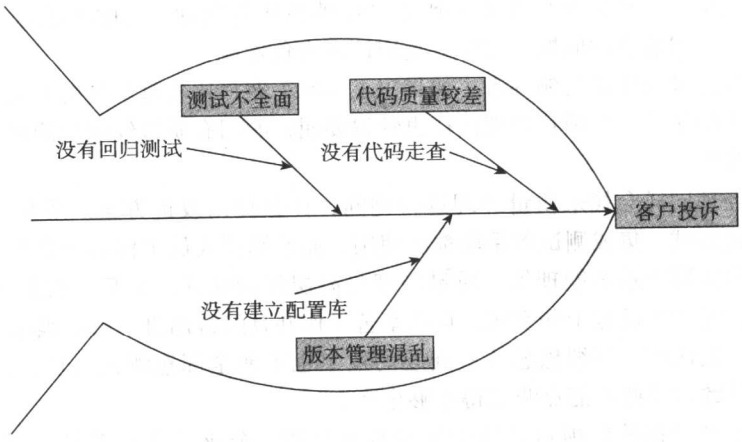

图 1-2　因果图

1.5.2　帕累托图

帕累托图是一种柱状图，如图 1-3 所示。帕累托图对问题发生的频率或影响按照从大到小的方式进行排列，从而确定解决问题的先后顺序。

帕累托图是根据帕累托法则得来的。帕累托法则也称为 80/20 法则，它是一种统计规律。例如：社会上 80% 的财富掌握在 20% 的人手中；大约 80% 的问题是因为 20% 的原因造成的。

通过帕累托图，管理人员可以很快找到那 20% 的主要原因并进行解决，使系统的质量得到迅速提升。

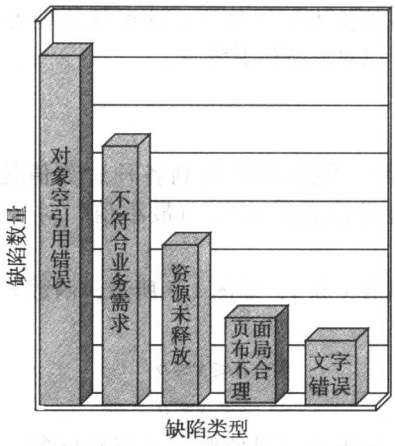

图 1-3　帕累托图

1.5.3　控制图

控制图可以对开发过程中的质量参数进行采样，科学地分析其是随机波动还是异常波动，从而对开发过程中的异常趋势提出预警，以便项目管理人员及时采取行动，消除异常，从而达到提高和控制质量的目的。

对于开发环节中的某一特性，例如，对缺陷严重程度进行跟踪时，允许其出现级别较高的缺陷。只要在控制界限内的过程结果都是可以接受的，如果过程结果超出了控制界限，那通常称其为失控。

在图 1-4 中，如果有连续的 7 个或 7 个以上的点分布在中心线的同一侧，或者出现同向变化的趋势，即使它们都处于控制界限内，但也意味着其出现了一定的问题或者受到了外界因素的干扰，应将其视为失控状态。一般来说，当一个过程处于受控状态下，则不应该调整这个过程，但可以继续改进此过程。

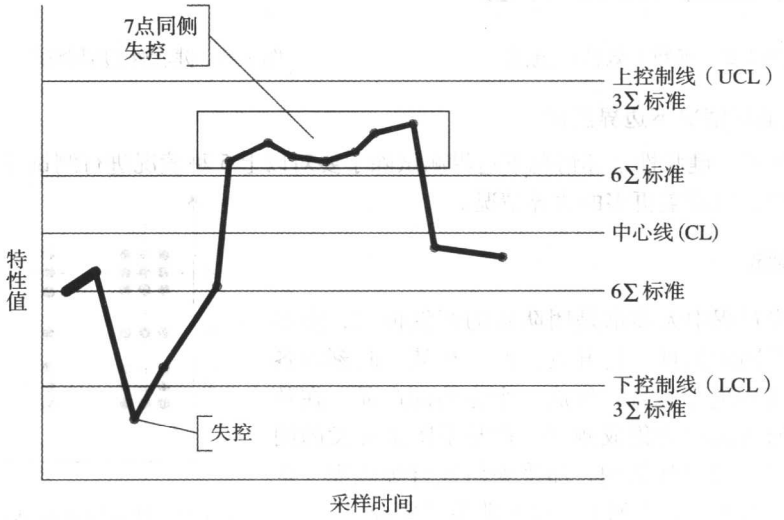

图 1-4　控制图

1.6　常用测试方法

在软件测试方法中有非常多的技术可以使用，这里从原理上介绍几种常见的测试方法供读者理解和学习，以便测试人员可以更好领悟其含义。

1.6.1　边界值测试

变量和函数都有自己的边界，能够准确找到各种边界值是软件测试人员所需要具备的基本功。对于变量来说它的边界值是最大值和最小值，对于函数来说它的边界就是一个区域。

例如：函数 F 有 2 个输入函数 x 和 y，如图 1-5 所示，函数 F 的有效值范围就是 2 个输入参数 x 和 y 的有效值所组成的区间。

$$a \leqslant x \leqslant b$$

$$c \leqslant y \leqslant d$$

基于对有效值区间的理解再来思考边界值测试就容易很多。边界值测试可以分为健壮性边界测试和健壮性最坏情况下边界测试。

1. 健壮性边界测试

如图 1-6 所示，健壮性边界测试就是要取最大值、最小值、正常值、略大值、略小值进行测试，以验证在这 5 种情况下是否正确。

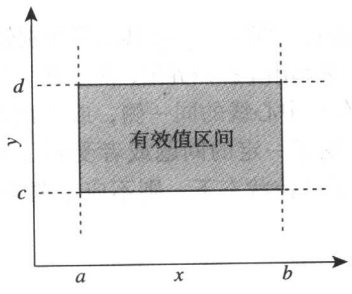

图 1-5　函数有效值的范围

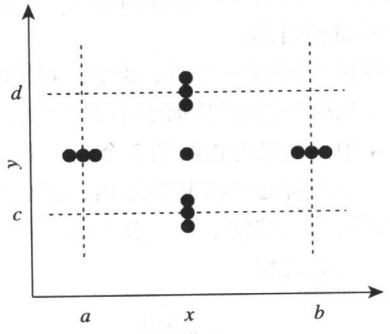

图 1-6　健壮性边界测试

2. 健壮性最坏情况下边界测试

如图 1-7 所示，健壮性最坏情况下边界测试除了要对以上 5 种境况进行测试外，还要对其进行笛卡儿积计算，以覆盖更多的边界情况。

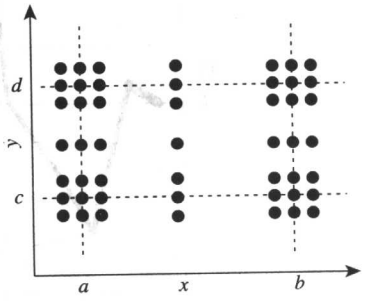

图 1-7　健壮性最坏情况下边界测试

1.6.2　集成测试

在软件开发过程中大多都是团队式的开发模式，由不同的小组负责不同的组件并行开发，然后在某一时刻将各个小组的工作成果进行整合，组成一个完整的产品。在整合过程中进行的测试称为集成测试。在基于团队开发的同时，常采用分层式的设计架构，那将如何进行集成呢？常用的方法有自上而下、自下而上以及三明治式集成。