第一图书网, tushu007.com

<<软件测试技术概论>>

图书基本信息

书名：<<软件测试技术概论>>

13位ISBN编号：9787302081777

10位ISBN编号：7302081778

出版时间：2004-4

出版时间：清华大学出版社

作者：古乐 编

页数：450

版权说明：本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：http://www.tushu007.com

内容概要

本书是一本比较全面地介绍软件测试方法的书籍，先介绍测试技术的发展历史和现状；然后，作为测试的一个基础，介绍了白盒测试、黑盒测试以及测试覆盖率等几个重要概念，并充分分析了业界在这几个概念方面的研究成果；之后从全流程测试的角度按动态测试和静态测试两个方面，分别介绍了单元测试、集成测试、系统测试、验证和确认过程、需求测试和设计测试等内容。
作为静态测试的一个重要手段，本书还对同行评审的概念和方法进行了阐述。
最后，作者总结了测试的基本原则和一些好的实践经验。
　本书可以作为大学本科高年级学生或研究生教材，也可以作为本科低年级学生学习的参考书。
对于软件工程师和测试工程师来说，本书是一本很好的指导书。

第一图书网, tushu007.com

作者简介

古乐，编著《软件测试技术概论》。

书籍目录

第1章 概述1.1 回顾测试的发展1.2 什么是软件测试1.2.1 IEEE的定义1.2.2 测试在软件开发中的角色1.3 为什么要进行软件测试1.4 测试的目的1.4.1 证明1.4.2 检测1.4.3 预防1.5 业界的软件测试现状1.6 软件测试中的误区1.7 本章小结第2章 白盒测试和黑盒测试2.1 白盒测试2.1.1 什么是白盒测试2.1.2 为什么要进行白盒测试2.1.3 白盒测试的常用技术2.1.4 一个白盒测试的例子2.2 黑盒测试2.2.1 什么是黑盒测试2.2.2 为什么要进行黑盒测试2.2.3 黑盒测试的常用技术2.2.4 一个黑盒测试的例子2.3 白盒测试和黑盒测试的比较2.3.1 白盒测试的优缺点2.3.2 黑盒测试的优缺点2.3.3 灰盒测试2.4 本章小结第3章 测试覆盖率3.1 覆盖率概念3.2 常见的逻辑覆盖3.2.1 语句覆盖3.2.2 判定覆盖3.2.3 条件覆盖3.2.4 判定条件覆盖3.2.5 路径覆盖3.2.6 逻辑覆盖小结3.3 功能覆盖率3.4 面向对象的覆盖率3.4.1 继承上下文覆盖3.4.2 基于状态的上下文覆盖3.4.3 基于线程的上下文覆盖3.5 其他覆盖率3.5.1 函数覆盖3.5.2 指令块覆盖3.5.3 判定路径覆盖3.5.4 更改条件判定覆盖3.5.5 分支条件组合覆盖3.5.6 过程到过程路径覆盖3.5.7 Z路径覆盖3.5.8 ESTCA覆盖3.5.9 LCSAJ覆盖3.6 如何使用覆盖率3.6.1 基本原则3.6.2 一个选择建议3.6.3 使用最少测试用例来达到覆盖3.7 本章小结第4章 程序分析技术4.1 文本视角4.2 句法视角4.3 控制流视角4.4 数据流视角4.5 计算流视角4.5.1 故障插入4.5.2 变体分析4.5.3 敏感性分析4.6 功能视角4.7 本章小结第5章 测试分析技术5.1 面向规格的测试5.1.1 测试独立于规格技术5.1.2 测试依赖于规格技术5.2 面向实现的测试5.2.1 面向结构的测试5.2.2 面向影响的测试5.2.3 面向传递的测试5.3 面向错误的测试5.3.1 基于错误的测试5.3.2 基于故障的测试5.3.3 基于风险的测试5.3.4 可能的正确性5.4 混合测试技术5.5 本章小结第6章 单元测试6.1 什么是单元测试6.1.1 单元测试的目的6.1.2 单元测试和集成测试的区别6.1.3 单元测试和系统测试的区别6.2 为什么要进行单元测试6.3 单元测试环境6.4 单元测试策略6.4.1 【策略一】由顶向下的单元测试策略6.4.2 【策略二】由低到上的单元测试策略6.4.3 【策略三】孤立测试6.4.4 综合测试6.5 单元测试分析6.5.1 模块接口6.5.2 局部数据结构6.5.3 独立路径6.5.4 出错处理6.5.5 边界条件6.6 单元测试用例设计思路6.6.1 为系统运行设计用例6.6.2 为正向测试设计用例6.6.3 为逆向测试设计用例6.6.4 为满足特殊需求设计用例6.6.5 为代码覆盖设计用例6.6.6 为覆盖率指标完成设计用例6.7 单元测试过程6.7.1 完成测试计划6.7.2 获取测试集6.7.3 度量测试单元6.8 单元测试工具介绍6.9 单元测试应坚持的原则6.10 我们的问题6.11 本章小结第7章 集成测试7.1 什么是集成测试7.1.1 集成测试与系统测试的区别7.1.2 集成测试关注的重点7.1.3 集成测试和开发的关系7.1.4 集成测试的层次7.2 集成测试策略7.2.1 大爆炸集成7.2.2 自顶向下的集成7.2.3 自底向上的集成7.2.4 三明治集成7.2.5 修改过的三明治集成7.2.6 基干集成7.2.7 分层集成7.2.8 基于功能的集成7.2.9 高频集成7.2.10 基于进度的集成7.2.11 基于风险的集成7.2.12 基于事件（消息）的集成7.2.13 基于使用的集成7.2.14 客户/服务器的集成7.2.15 分布式集成7.3 集成测试分析7.3.1 体系结构分析7.3.2 模块分析7.3.3 接口分析7.3.4 风险分析7.3.5 可测试性分析7.3.6 集成测试策略分析7.3.7 常见的集成测试故障7.4 集成测试用例设计思路7.4.1 为系统运行设计用例7.4.2 为正向测试设计用例7.4.3 为逆向测试设计用例7.4.4 为满足特殊需求设计用例7.4.5 为高覆盖设计用例7.4.6 测试用例补充7.4.7 注意事项7.5 集成测试过程7.5.1 计划阶段7.5.2 设计阶段7.5.3 实现阶段7.5.4 执行阶段7.6 集成测试环境集成测试7.7 集成测试工具介绍7.8 集成测试应坚持的原则7.9 本章小结第8章 系统测试8.1 系统测试概念8.2 系统测试方法8.2.1 功能测试8.2.2 协议一致性测试8.2.3 性能测试8.2.4 压力测试8.2.5 容量测试8.2.6 安全性测试8.2.7 恢复性测试8.2.8 备份测试8.2.9 GUI测试8.2.10 健壮性测试8.2.11 兼容性测试8.2.12 可用性测试8.2.13 可安装性测试8.2.14 文档测试8.2.15 在线帮助测试8.2.16 数据转换测试8.3 系统测试过程8.3.1 完成系统测试计划8.3.2 完成系统测试用例8.3.3 评审/审批系统测试计划8.3.4 执行系统测试8.4 本章小结第9章 可靠性与可靠性测试9.1 基本概念9.1.1 什么是软件可靠性9.1.2 错误、缺陷、故障和失效9.1.3 软件可靠性指标9.1.4 软件和硬件可靠性区别9.2 可靠性指标分配9.3 可靠性预计9.3.1 计数法9.3.2 应力法9.4 可靠性分析方法9.4.1 FMEA9.4.2 CA9.4.3 FTA9.4.4 ETA9.4.5 SCA9.5 软件可靠性测试9.5.1 可靠性测试流程9.5.2 可靠性模型介绍9.5.3 一个可靠性数据分析例子9.6 软件可靠性工程9.7 可靠性标准和可靠性工具9.7.1 可靠性标准9.7.2 可靠性工具9.8 本章小结第10章 其他专项性测试10.1 可接受性测试10.2 Alpha测试10.3 Beta测试10.4 标杆测试10.5 配置测试10.6 外场测试10.7 SQL测试10.8 2000年测试10.9 回归测试10.10 本章小结第11章 软件质量透视11.1 质量的定义11.2 质量的预防和检测11.3 如何提高软件产品的质量11.3.1 流程对质量的贡献11.3.2 流程与技术11.3.3 全面质量管理11.3.4 关注测试11.3.5 组织、流程和人11.4 质量标准11.5 本章小

结 第12章 软件验证和确认12.1 基本概念 12.2 软件验证和确认计划12.2.1 SVVP步骤12.2.2 SVVP的7个主题12.3 验证和确认任务分析12.3.1 关键性分析12.3.2 可跟踪性分析12.3.3 评估12.3.4 接口分析12.3.5 测试12.4 生命周期各阶段活动12.4.1 管理阶段的验证和确认12.4.2 概念阶段的验证和确认12.4.3 需求阶段的验证和确认12.4.4 设计阶段的验证和确认12.4.5 实现阶段的验证和确认12.4.6 测试阶段的验证和确认12.4.7 安装和校验阶段的验证和确认12.4.8 运行和维护阶段的验证和确认12.4.9 验证和确认任务总结12.5 验证和确认的报告12.5.1 标准要求的报告12.5.2 标准可选报告12.6 本章小结 第13章 软件质量保证13.1 基本概念13.1.1 目标13.1.2 执行的承诺13.1.3 执行的能力13.1.4 执行的活动13.1.5 度量分析13.1.6 验证实现13.2 SQA实施过程13.2.1 建立SQA组织13.2.2 选择SQA任务13.2.3 产生/维护SQA计划13.2.4 实施SQA计划13.2.5 产生/维护SQA规程13.2.6 标识SQA培训13.2.7 标识/选择SQA工具13.2.8 改进项目SQA过程13.3 本章小结第14章 需求测试14.1 需求测试概述14.1.1 什么是需求14.1.2 测试需求14.2 通过评审来测试需求14.2.1 需求评审中的常见风险14.2.2 需求评审检查表14.3 通过用例设计来测试需求14.4 需求建模测试14.4.1 统一建模语言14.4.2 消息顺序图（MSC）14.4.3 分析建模工具介绍14.4.4 需求的形式化描述14.5 基于原型的测试14.5.1 原型的目的14.5.2 原型的种类14.5.3 原型的测试方法14.6 本章小结第15章 设计测试15.1 设计测试概述15.1.1 什么是设计15.1.2 软件构架设计15.1.3 概要设计和详细设计15.2 设计的评审15.2.1 设计查检表15.2.2 构架设计评审方法15.2.3 软件构架评价最佳工业实践15.3 SDL及相关测试15.3.1 SDL介绍15.3.2 SDL基本概念15.3.3 SDL结构15.3.4 SDL测试15.4 本章小结第16章 同行评审16.1 基本概念16.2 同行评审的一般过程16.2.1 计划阶段16.2.2 实施被选择的同行评审过程16.2.3 同行评审过程度量 16.2.4 同行评审的评审/审计16.3 走读16.3.1 过程目标16.3.2 特定的角色和职责16.3.3 输入16.3.4 入口标准16.3.5 过程16.3.6 出口标准16.3.7 输出16.4 技术评审16.4.1 过程目标16.4.2 特定的角色和职责16.4.3 输入16.4.4 入口标准16.4.5 过程16.4.6 出口标准16.4.7 输出16.5 正规检视16.5.1 正规检视小组16.5.2 正规检视过程16.5.3 正规检视常用表格16.6 本章小结第17章 测试经验总结17.1 软件测试的10大原则17.1.1 原则1：测试是一个持续进行的过程，而不是一个阶段17.1.2 原则2：测试必须被计划、被控制，并且被提供时间和资源17.1.3 原则3：测试应当分级别17.1.4 原则4：测试应当有重点17.1.5 原则5：测试不是为了证明程序的正确性，而是为了证明程序不能工作17.1.6 原则6：测试是不可能穷尽的，当测试出口条件满足时就可以停止测试17.1.7 原则7：测试是开发的朋友，不是开发的敌人17.1.8 原则8：测试人员应公正地测试，如实地记录和报告缺陷17.1.9 原则9：测试自动化能解决一部分问题，但不是全部17.1.10 原则10：测试不能仅仅包括功能性的验证，还应当包含性能、可靠性、可维护性、安全性等方面的验证17.2 软件测试的10个最佳实践17.2.1 实践1：尽早地、频繁地进行测试是降低项目成本，提高质量的一个好方法17.2.2 实践2：尽早产生一个综合的主测试计划17.2.3 实践3：对质量要求较高或大型复杂的产品成立独立的测试组17.2.4 实践4：在每个开发阶段，使用测试和评价的结果作为是否可以通过的标准17.2.5 实践5：开发和维护一个测试需求和目标的风险优先级列表17.2.6 实践6：把测试件作为产品的一部分等同管理，使用相同的评价标准和过程17.2.7 实践7：提供集成化的测试工具和测试基础支持17.2.8 实践8：加强测试度量工作和缺陷分析工作，不断地改进测试17.2.9 实践9：加强测试的培训并且为测试人员提供技能发展的通道17.2.10 实践10：加强沟通和交流，让项目组内所有人员都了解测试的工作及其重要性17.3 本章小结附录A 常见测试术语附录B 测试技术分类附录C 常见的编码错误附录D 经典测试网站附录E 参考资料

编辑推荐

《软件测试技术概论》的重点在于介绍软件测试的一些基本概念和方法，涵盖了业界出现的大部分测试领域内的知识，是一本比较全面的测试方法介绍书籍。
《软件测试技术概论》综合考虑了测试的方方面面，内容涵盖了动态测试到静态测试，白盒测试到黑盒测试，单元测试到系统测试等多个测试领域。
对于近几年才出现的一些测试方法也进行了阐述和比较。
全书内容涉及面广，理论与实践巧妙结合，强调测试分析和测试设计的重要性，引入过程概念，强调全面质量管理，提出需求测试和设计测试的思想，值得一读。
《软件测试技术概论》可以作为大学本科高年级学生或研究生教材，也可以作为本科低年级学生学习的参考书。
对于软件工程师和测试工程师来说，《软件测试技术概论》是一本很好的指导书。

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:http://www.tushu007.com