

<<软件缺陷模式与测试>>

图书基本信息

书名：<<软件缺陷模式与测试>>

13位ISBN编号：9787030317261

10位ISBN编号：7030317262

出版时间：2011-7

出版时间：科学出版社

作者：宫云战 等著

页数：268

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>

<<软件缺陷模式与测试>>

内容概要

基于缺陷模式的软件测试是21世纪初发展起来的一种新型软件测试技术，是高可信、大型及基础软件测试必备的方法之一，有强烈的工程需求，它以缺陷检测效率高、缺陷定位准确、自动化程度高、易学易用、与其他软件测试技术具有很好的互补性等特点，目前已逐步成为国际上主流的软件测试技术。

《软件缺陷模式与测试》全面论述了基于缺陷模式软件测试的一般方法，包括软件缺陷的综合论述、面向C/C++/Java的软件缺陷模式的分类、各种软件缺陷模式的定义、基于缺陷模式的软件测试原理、提高测试精度的区间运算技术、敏感路径分析技术、函数间分析技术等。

《软件缺陷模式与测试》是软件测试领域的专业书籍，可供从事软件测试技术工作的研究人员学习和参考。

本书主要由北京邮电大学宫云战、杨朝红、金大海、肖庆和王雅文撰写。

<<软件缺陷模式与测试>>

书籍目录

前言

第1章 软件缺陷与缺陷模式

- 1.1 软件缺陷的概念
- 1.2 软件缺陷的来源
- 1.3 软件缺陷的严重性和优先级
- 1.4 软件缺陷的发现、排除及效率
- 1.5 软件缺陷数据库
- 1.6 软件缺陷管理
- 1.7 软件缺陷预测
 - 1.7.1 撒播模型
 - 1.7.2 基于软件规模和复杂性的测量模型
 - 1.7.3 基于白盒测试的覆盖率进行预测
 - 1.7.4 基于软件研制的质量控制过程进行预测
 - 1.7.5 基于测试时错误发生的时刻进行预测
- 1.8 软件缺陷预防
 - 1.8.1 了解缺陷
 - 1.8.2 缺陷查找技术
- 1.9 软件缺陷的对数正态分布
- 1.10 软件代码缺陷模式
- 1.11 C缺陷
- 1.12 基于缺陷模式的软件测试的意义

第2章 故障模式

- 2.1 Java故障模式
 - 2.1.1 空指针使用
 - 2.1.2 数组越界
 - 2.1.3 资源泄漏
 - 2.1.4 非法计算
 - 2.1.5 死循环
 - 2.1.6 并发
- 2.2 C/C++故障模式
 - 2.2.1 内存泄漏
 - 2.2.2 数组越界
 - 2.2.3 使用未初始化变量
 - 2.2.4 空指针使用
 - 2.2.5 非法计算
 - 2.2.6 死循环
 - 2.2.7 悬挂指针

第3章 安全漏洞模式

- 3.1 Java安全漏洞模式
 - 3.1.1 未验证的输入
 - 3.1.2 滥用API
 - 3.1.3 安全特性
 - 3.1.4 竞争条件
 - 3.1.5 不合理的异常处理
 - 3.1.6 低质量代码

<<软件缺陷模式与测试>>

3.1.7 封装不当

3.2 C/C++安全漏洞模式

3.2.1 缓冲区溢出

3.2.2 被污染的数据

3.2.3 竞争条件

3.2.4 风险操作

第4章 疑问及规则模式

4.1 疑问模式

4.1.1 性能相关

4.1.2 冗余代码

4.1.3 不良代码

4.2 规则模式

4.2.1 声明定义类

4.2.2 版面书写类

4.2.3 分支控制类

4.2.4 指针使用类

4.2.5 跳转控制类

4.2.6 运算处理类

4.2.7 过程调用类

4.2.8 语句使用类

4.2.9 调用返回类

4.2.10 循环控制类

4.2.11 类型转换类

4.2.12 初始化类

4.2.13 比较判断类

4.2.14 名称、符号与变量使用类

第5章 基于缺陷模式的测试技术

5.1 基于缺陷模式的测试技术概述

5.1.1 发展概况

5.1.2 基于缺陷模式的软件测试指标分析

5.2 缺陷测试系统

5.2.1 缺陷测试系统的结构

5.2.2 缺陷测试系统的特点

5.3 缺陷测试系统的缺陷模式描述

5.4 缺陷测试系统的缺陷模式检测

5.4.1 抽象语法树分析

5.4.2 控制流分析

5.4.3 符号表构建

5.4.4 数据流分析

5.5 使用缺陷测试系统进行缺陷检测

第6章 区间运算技术

6.1 经典的区间代数

6.1.1 区间和区间运算

6.1.2 区间向量和区间函数

6.2 扩展的区间运算

6.2.1 数值型区间集代数

6.2.2 非数值型区间代数

<<软件缺陷模式与测试>>

6.2.3 条件表达式中的区间计算

6.2.4 基于区间运算的变量值范围分析

6.3 变量的相关性分析

6.3.1 变量间关联关系分类

6.3.2 符号分析

6.4 区间运算在程序分析中的应用

6.4.1 检测矛盾节点

6.4.2 检测不可达路径

6.4.3 提高缺陷检测效率

第7章 路径敏感分析技术

7.1 数据流分析

7.1.1 控制流图

7.1.2 数据流分析概述

7.1.3 四种典型的数据流问题

7.2 数据流分析的理论基础

7.2.1 格理论

7.2.2 不动点理论

7.2.3 数据流分析的最大和最小不动点解

7.2.4 数据流解的含义

7.3 基于数据流的缺陷检测

7.3.1 缺陷模式状态机实例

7.3.2 基于传统数据流的缺陷检测

7.4 路径敏感的缺陷检测技术

7.4.1 静态缺陷检测中的误报

7.4.2 路径信息抽象

7.4.3 路径敏感的缺陷检测算法

第8章 函数间分析技术

8.1 问题描述

8.1.1 函数约束对测试的影响

8.1.2 函数副作用对测试的影响

8.1.3 当前研究状况

8.2 函数约束信息

8.2.1 约束信息描述

8.2.2 约束信息在静态测试中的应用

8.2.3 实验环境及结果

8.3 函数后置信息

8.3.1 后置信息描述

8.3.2 生成算法

8.3.3 应用实例

8.3.4 实验结果

参考文献

<<软件缺陷模式与测试>>

章节摘录

版权页：插图：缺陷的严重性和优先级是含义不同但相互联系密切的两个概念。

它们都从不同的侧面描述了软件缺陷对软件质量和最终用户的影响程度和处理方式。

一般地，严重性程度高的软件缺陷具有较高的优先级。

严重性高说明缺陷对软件造成的质量危害性大，需要优先处理；而严重性低的缺陷可能只是软件不太尽善尽美，可以稍后处理。

但是，严重性和优先级并不总是一一对应的。

有时严重性高的软件缺陷优先级不一定高，甚至不需要处理，而一些严重性低的缺陷却需要及时处理，具有较高的优先级。

一方面，修正软件缺陷不是一件纯技术问题，有时需要综合考虑市场发布和质量风险等因素。

例如，若某个严重的软件缺陷只在非常极端的条件下产生，则没有必要马上解决。

另外，若修正一个软件缺陷，则需要重新修改软件的整体架构，这可能会产生更多潜在的缺陷，而且软件由于市场的压力必须尽快发布，此时即使缺陷的严重性很高，是否需要修正，也需要全盘考虑。

另一方面，如果软件缺陷的严重性很低，如界面单词拼写错误，此时可以稍后修正。

但是如果是软件名称或公司名称的拼写错误，则必须尽快修正，因为这关系到软件 and 公司的市场形象。

对于比较规范的软件测试，一般使用软件缺陷管理数据库进行缺陷报告和处理，这需要在测试项目开始前对全体测试人员和开发人员进行培训，对缺陷严重性和优先级的表示和划分方法进行统一规定和遵守。

<<软件缺陷模式与测试>>

编辑推荐

《软件缺陷模式与测试》由科学出版社出版。

<<软件缺陷模式与测试>>

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>