

卷号	
卷内编号	
密级	

# 软件缺陷管理指南

ALAN ZHOU

Version 1.0

分类：  
<指南>  
使用部门：  
<测试人员、项目组>

# 目 录

1. 简介.....	1
2. 如何收集缺陷.....	1
2.1 缺陷类型.....	1
2.2 了解缺陷.....	1
3. 如何分析和统计缺陷.....	2
3.1 缺陷严重程度.....	2
3.2 解决优先级 ( PRIORITY ) .....	3
3.3 缺陷状态 ( STATUS ) .....	3
3.3.1 评审缺陷分析.....	3
3.4 缺陷分类适用范围.....	5
4. 缺陷管理工具.....	6
5. 参考文献.....	6
6. 附录.....	6

# 软件缺陷管理指南

## 1. 简介

本文档的目的是指导如何管理同行评审、软件测试中发现的缺陷，即通过收集缺陷、分析和统计缺陷、排除缺陷以及预防缺陷等步骤达到有效地减少软件产品的缺陷数。

## 2. 如何收集缺陷

缺陷既指程序中存在的错误，例如语法错误、拼写错误或者是一个不正确的程序语句，缺陷也指可能出现在设计中，甚至在需求、规格说明或其他的文档中的种种错误。为了对缺陷进行管理，首先应对缺陷进行分类，通过对缺陷进行分类，可以迅速找出哪一类缺陷的问题最大，然后集中精力预防和排除这一类缺陷。而这正是缺陷管理的关键，一旦这几类缺陷得到控制，再进一步找到新的容易引起问题的几类缺陷上。

### 2.1 缺陷类型

缺陷类型编号	缺陷类型	描述
10	F-功能	如逻辑，指针，循环，递归，功能等缺陷
20	G-语法	拼写、标点符号、打字
30	A- 赋值	如声明、重复命名，作用域
40	I- 接口	与其他组件、模块或设备驱动程序、调用参数、控制块或参数列表相互影响的缺陷
50	B- 联编打包	由于配置库、变更管理或版本控制引起的错误
60	D-文档	需求、设计类文档
70	U-用户接口	人机交互特性：屏幕格式，确认用户输入，功能有效性
80	P-性能	不满足系统可测量的属性值，如：执行时间，事务处理速率等
90	N-标准	不符合各种标准的要求，如编码标准、设计符号等
100	E-环境	设计、编译、其他支持系统问题

### 2.2 了解缺陷

缺陷管理的第一步是了解缺陷，为此，必须首先收集缺陷数据，然后才能了解这些缺陷，并且找出如何预防它们，同时也能领会到如何更好地发现，修复甚至预防仍在引入的缺陷。可以按照以下步骤收集关于缺陷的数据：

- 为测试和同行评审中发现的每一个缺陷做一个记录
- 对每个缺陷要记录足够详细的信息，以便以后能更好地了解这个缺陷
- 分析这些数据以找出主要哪些缺陷类型引起大部分的问题
- 设计出发现和修复这些缺陷的方法（缺陷排除）

通常为了收集缺陷数据，可以采用缺陷记录日志来登记所发现的每一个缺陷

日期	编号	状态	类型	缺陷来源	排除阶段	修改时间	修复缺陷
	描述：						
	描述：						
	描述：						

对于缺陷记录日志中的描述应该足够清楚，以便今后可以看出该缺陷的起因。修复缺陷一栏说明此缺陷是由于修复其他缺陷而引入的。引入阶段表示该缺陷的来源，缺陷的来源可以分为以下几类：

缺陷来源	描述
Requirement	由于需求的问题引起的缺陷
Architecture	由于构架的问题引起的缺陷
Design	由于设计的问题引起的缺陷
Code	由于编码的问题引起的缺陷
Test	由于测试的问题引起的缺陷
Integration	由于集成的问题引起的缺陷

排除阶段表示发现和修复这个缺陷的阶段，通常分为如下：

排除阶段	描述
Requirement	在需求阶段发现的缺陷
Architecture	在构架阶段发现的缺陷
Design	在设计阶段发现的缺陷
Code	在编码阶段发现的缺陷
Test	在测试阶段发现的缺陷

### 3. 如何分析和统计缺陷

为了更好地分析缺陷，需要对缺陷在严重程度、优先级以及状态上加以区分。

#### 3.1 缺陷严重程度

#	缺陷严重等级	描述
1	严重缺陷 (Critical)	不能执行正常工作功能或重要功能。或者危及人身安全
2	较大缺陷 (Major)	严重地影响系统要求或基本功能的实现，且没有办法更正。 (重新安装或重新启动该软件不属于更正办法)

3	较小缺陷 (Minor)	严重地影响系统要求或基本功能的实现，但存在合理的更正办法。(重新安装或重新启动该软件不属于更正办法)
4	轻微缺陷 (Cosmetic)	使操作者不方便或遇到麻烦，但它不影响执行工作功能或重要功能。
5	其他缺陷 (Other)	其它错误

### 3.2 解决优先级 (Priority)

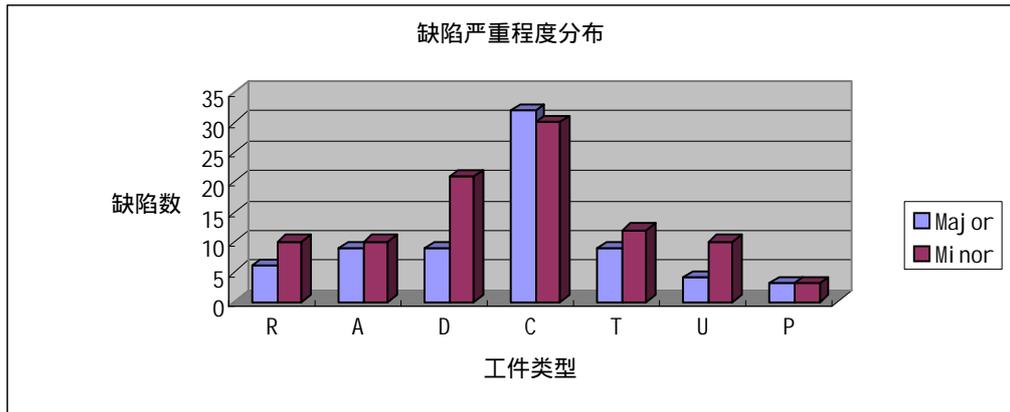
#	解决优先级	描述
1	立即解决 (Resolve Immediately)	缺陷必须被立即解决。
2	正常排队 (Normal Queue)	缺陷需要正常排队等待修复或列入软件发布清单。
3	不紧急 (Not Urgent)	缺陷可以在方便时被纠正。

### 3.3 缺陷状态 (Status)

缺陷状态	描述
已提交 (Submitted)	已提交的缺陷
打开 (Open)	确认“提交的缺陷”，等待处理
已拒绝 (Rejected)	拒绝“提交的缺陷”，不需要修复或不是缺陷
已解决 (Resolved)	缺陷被修复
已关闭 (Closed)	确认被修复的缺陷，将其关闭

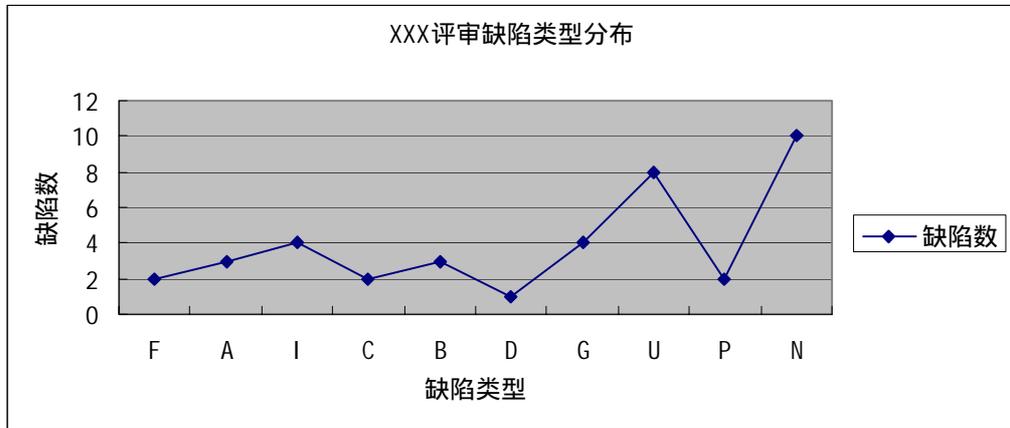
#### 3.3.1 评审缺陷分析

- 缺陷分布情况统计分析
  - 按照缺陷严重程度及工件类型分布



按照缺陷严重程度及工件类型分布可以统计整个项目生命周期中所有同行评审的缺陷分布，也可以统计某一阶段所有同行评审的缺陷分布。

— 按照缺陷类型分布



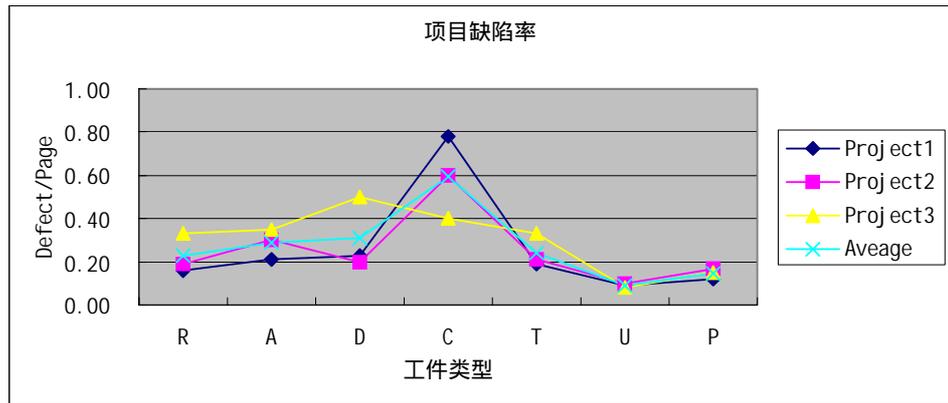
按照缺陷类型统计分布图，可以是某一次评审的缺陷统计，也可以是某一类型工件评审的缺陷统计，可以是某一阶段所有同行评审的缺陷统计，也可以是整个项目周期内所有同行评审的缺陷统计。建议以某一类型工件和某一阶段来进行统计分布。

➤ 缺陷率分析

— 单个项目的缺陷率

工件类型	缺陷数	Page/KLOC	Defect/Page (KLOC)
R	16	102	0.16
A	19	90	0.21
D	30	130	0.23
C	62	80	0.78
T	21	110	0.19
U	14	150	0.09
P	6	50	0.12

— 多个项目的缺陷率



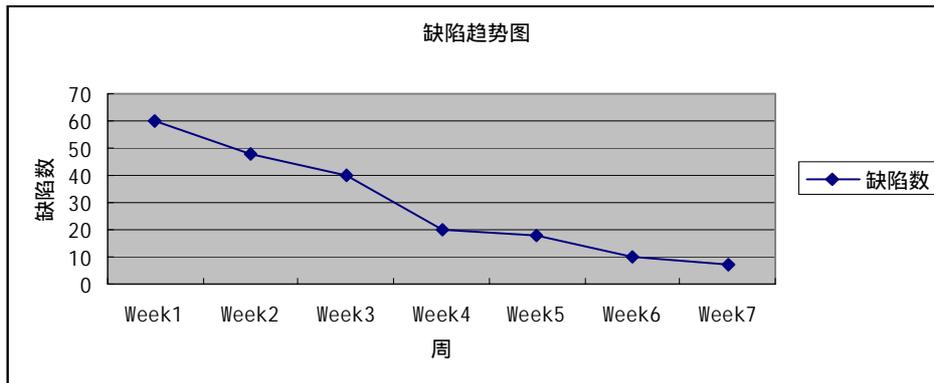
➤ 控制图

— 缺陷率控制图

Upper Control Limit (UCL) = Mean + L \* Standard Deviation

Lower Control Limit (LCL) = Mean - L \* Standard Deviation

➤ 缺陷趋势图



3.4 缺陷分类适用范围

缺陷属性针对文档和代码具有不同的适用性，下表列出了各自适用范围

缺陷属性	软件测试	同行评审
缺陷标识 (Identifier)		
缺陷类型 (Type)		
缺陷严重程度 (Severity)		
解决优先级 (Priority)		
缺陷状态 (Status)		
缺陷起源 (Origin)		
缺陷原因 (Cause)		

: 需要记录

: 可以不考虑/可以记录

□ : 不考虑

#### 4. 缺陷管理工具

可以采用 Rational ClearQuest 作为缺陷管理的工具。

#### 5. 参考文献

[QSDMM 1A]

*QUALITY STANDARDS DEFECT MEASUREMENT MANUAL* United Kingdom Software Metrics Association October 2000

#### 6. 附录

无