



# 软件缺陷管理设计与软件质量

Software Defect Management Design and Software Quality

田志全 王洁



## 一、前言

近年来,软件质量问题受到高度重视,很多开发组织在开展软件工程化的工作,作为软件质量管理的重要组成部分,软件缺陷管理也逐渐得到认识和重视,正在成为软件开发管理过程的一个亮点。

本文结合地面测控系统软件缺陷管理,阐述了地面测控系统软件缺陷管理应用模式的分析与设计,分析了地面测控系统软件缺陷管理模型的体系结构,介绍了建立软件缺陷库的基本方法。最后,对测控系统(本文特指地面测控系统)软件缺陷管理的实践应用情况进行了总结。

## 二、软件缺陷的基本概念

### (一) 软件缺陷的定义

狭义地讲,软件缺陷是软件设计过程中造成的错误。广义地讲,软件缺陷是影响客户正常使用的任何问题,就是说,缺陷不仅仅是编程中出现的问题,还包括了客户功能规范等方面的问题。本文将软件中的缺陷(Defect或Bug)定义为存在于软件(包括文档、数据、程序)之中的那些不希望或不可接受的偏差,其结果是软件运行于某

一特定条件时出现软件故障。

### (二) 软件缺陷的属性

根据在地面测控系统软件开发过程中发现的缺陷特点,缺陷属性页设计可参照图1所示。

图1 描述缺陷的属性页

具体地缺陷属性的描述,我们定义了软件缺陷的10个属性,见表1所示。

### 三、软件缺陷管理系统的应 用模式设计

地面测控系统软件缺陷管理系统

表1 缺陷属性

属性	解释
缺陷ID	标记缺陷的一组唯一的符号
缺陷提交人	缺陷的提交人
缺陷提交时间	提交缺陷的时间,包含日期和时间信息
缺陷状态	缺陷所处的状态
项目信息	缺陷涉及的项目信息,包括项目名称、项目负责人、项目版本
模块信息	缺陷涉及的模块信息,包括模块名称、缺陷处理负责人、模块版本
缺陷严重性	缺陷引起的故障对软件的影响程度,包括致命错误、严重错误、轻微错误、建议
缺陷错误分类	根据Thayer软件错误分类方法[4]将缺陷分类,包括计算错误、逻辑错误、操作系统和支持软件错误、配置错误、接口错误、用户需求改变、文档错误等16种缺陷类型
缺陷优先级	缺陷必须被修复的紧急程度,包括必须修复、立即修复、应该修复、考虑修复
缺陷的描述	描述缺陷的复现步骤、缺陷表现、修改建议等

采用C/S的应用模式,见图2所示。企业内部已经建立的网络为我们搭建了硬件基础平台,借助软件缺陷管理系统,每个软件人员都可以坐在自己的电脑旁边与他人进行沟通和协同工作。软件缺陷管理是利用网络数据库技术搭建的一个软件人员之间的交流平台,同时完成软件缺陷从发现到最终解决全过程的跟踪管理。这也是全面质量管理的重要方法。

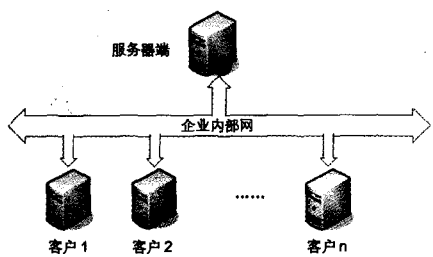


图2 软件缺陷管理系统应用模式示意图

### 四、软件缺陷管理模型的体系结构分析

缺陷管理系统是按照标准的三层结构设计的，包括数据层、功能层（业务层）、表示层，见图3所示。

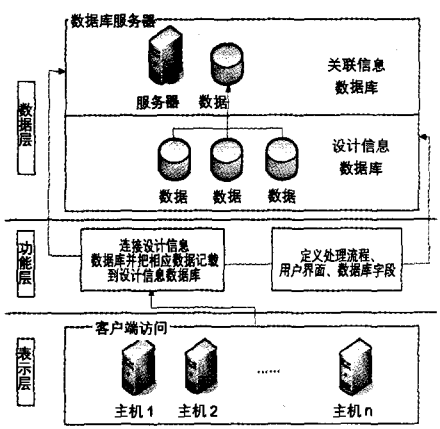


图3 缺陷管理系统的体系结构示意图

下面具体分析缺陷管理系统的体系结构。

#### (一) 数据层

缺陷管理系统的基础是网络数据库，缺陷管理系统相关的所有内容都储存在数据库中。

缺陷管理应用时，所有信息都是被存放在后台的数据库中，通过表示层来对数据库中的数据元素进行操作。由于每个人软件测试的经验背景不同，很难控制报告的缺陷质量，也不利于保持软件缺陷报告的一致性。测控系统项目型号经常要面临项目期限的压力，研发的产品复杂度高等情况，单纯采用文档和口头通知的形式来进行缺陷管理，将造成测试项目实施过程中，

软件测试缺陷的交流效率低下，缺陷的流程管理难以实施控制。由于目前的测控系统软件模块复用程度很高，同样的模块大量同时应用在不同的项目型号中，采用共享的公共数据库进行缺陷管理，便于其他项目的人员对类似缺陷的查询，降低了发现缺陷、解决缺陷的时间成本和人力成本，从而在根本上解决了由于知识、技能和经验不足造成的损失。

缺陷管理数据库的数据层分为两个数据库（关联信息数据库DevCon和设计信息数据库DevDB），关联信息数据库用来存放各种缺省定义的处理流程、用户界面、数据库字段等信息；在设计信息数据库中保存的是某个具体的设计进行个性化配置后的信息和提交的所有缺陷，以及同缺陷处理相关的所有信息，例如用户和组的信息。

#### (二) 功能层

功能层是缺陷管理系统中最重要的部分，主要实现的功能可以分为以下三种：

##### 1) 数据库的建立

在使用缺陷数据库之前，必须创建至少一个关联信息数据库，也就是说，所有自定义的设计都附属于特定的关联信息数据库。关联信息数据库实际是一个连接设计信息数据库并把相应数据记载到设计信息数据库的行为。如果想在客户端进行缺陷管理操作，则必须建立与设计相对应的设计信息数据库。

##### 2) 定义设计

定义新的设计是一个继承的过程，必须选择一个原有的设计模板，继承原有设计模板所有的字段设置、状态和操作、窗体等内容。可以建立一个

配置好的标准设计模板（不用连接数据库），所有其它的设计都从此标准模板中继承，然后进行简单的修改就可以使用了。

#### 3) 设置用户和组

在缺陷管理系统的设计和应用过程中，用户和组是很重要的两个因素。根据地面测控软件开发团队成员的分工和作用，我们规划设计了管理组、测试组和开发组，将软件组成员添加到各个组，某一个用户可以隶属于多个组。

#### (三) 表示层

缺陷管理系统的客户端是实际进行变更流程操作的访问程序。在客户端，可以建立查询表、统计图表等。客户端通过建立的查询表才可以看到提交的记录。

### 五、软件缺陷管理的流程设计

软件缺陷管理贯穿于软件缺陷的生命周期。缺陷生命周期是指从发现缺陷直到完成缺陷处理的全过程。

#### (一) 角色的定义

根据在缺陷管理流程中的作用的不同，设计定义了4种角色：管理人员、软件总体负责人（即软件项目负责人，下同）、缺陷处理负责人、测试人员。某个软件人员可以分担多个角色，同一个人不同的缺陷中可能会承担不同的角色。

管理人员：在软件缺陷管理中拥有最高管理权限，可以提交、分派、处理、验证、关闭和删除缺陷。

项目负责人：就是软件总体负责

表2 状态转移矩阵

To \ From	已提交	已分配	正在处理	待验证	已关闭	重复问题	暂不处理
已提交							
已分配		分配					分配
正在处理		启动处理		验证不通过	重新打开		
待验证			结束处理				
已关闭	关闭	关闭		验证通过	关闭		
重复问题	重复问题		重复问题	重复问题	重复问题		
暂不处理	暂不处理		暂不处理	暂不处理			



人，任何一个缺陷都是属于某个项目的，它负责指派该软件缺陷具体由哪个人负责处理，同时也负责缺陷验证工作。

**缺陷处理负责人：**一般是软件模块负责人，当缺陷不是某个具体模块问题时，此角色负责处理该软件缺陷。

**测试人员：**一般是软件测试负责人，负责软件缺陷的发现提交和测试验证工作。

### (二) 缺陷状态的转移

在缺陷跟踪管理的过程中，在不同的阶段，通过将缺陷记录修改为不同的状态来进行标记。共设计划分7个缺陷状态：分别为“已提交”、“已分配”、“正在处理”、“待验证”、“关闭”、“重复问题”、“暂不处理”。改变状态的操作共10个，包括：“提交”、“分配”、“启动处理”、“结束处理(等待验证)”、“关闭”、“验证通过”、“验证不通过”、“重新打开”、“重复问题”、“暂不处理”。所有提交的缺陷，都是以记录的形式保存在数据库相应表内。在下面的表2中，上方和左边的灰色区域是缺陷的状态(上方From表示的是源状态，

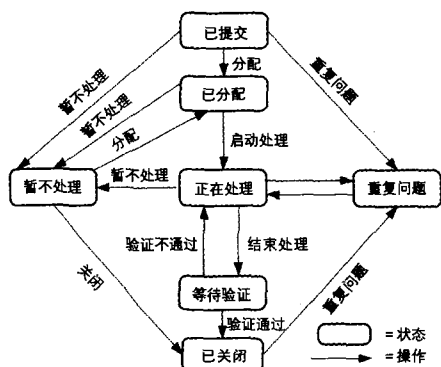


图4 软件缺陷状态转移模型示意图

左侧To表示的是目标状态)，白色区域内则是进行状态转换所需要进行的操作。

软件缺陷状态转移模型如图4所示，从图中可以看出每个软件缺陷开始

于缺陷的发现，发现的缺陷被提交后就进入“已提交”状态，这是所有缺陷的初始状态。当软件总体人员看到缺陷，并安排具体人员处理此缺陷，此时缺陷进入“已分配”状态。当相应的缺陷处理负责人看到分配给自己处理的缺陷，并开始修改时，会将缺陷状态设置为“正在处理”状态，在缺陷处理结束并提交回归测试时，将缺陷的状态置为“待验证”状

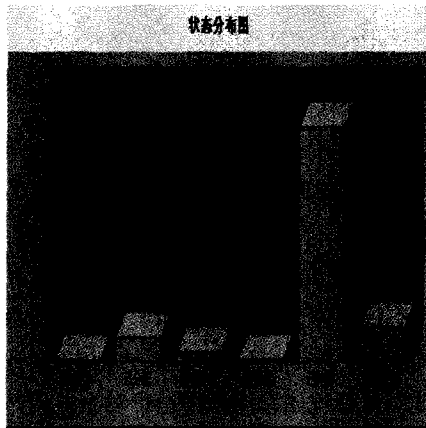


图5 缺陷状态分布图

态。如果回归测试通过，则通过“验证通过”操作将缺陷“关闭”，否则“验证不通过”，缺陷重新回到正在处理状态，由缺陷处理负责人重新检查处理。

### 六、软件缺陷管理的应用情况

地面测控系统软件缺陷管理系统自从2004年开始应用以来，已有数十个项目纳入软件缺陷管理系统，提交的软件缺陷达600多条，为查询和报告提供了较大的方便。软件项目相关人员可以随时了解项目的缺陷情况，提高了软件缺陷处理的效率。从缺陷状态分布图(见图5)可以查看缺陷库中所有缺陷状态的统计情况。其中，绝大多数缺陷已经被解决，处于关闭状态。

从图6可以看出缺陷库中所有缺陷的严重性分布，其中每一类严重性又按照优先级再进行分类。

地面测控系统软件缺陷管理系统的应用，实现了软件缺陷的全过程跟

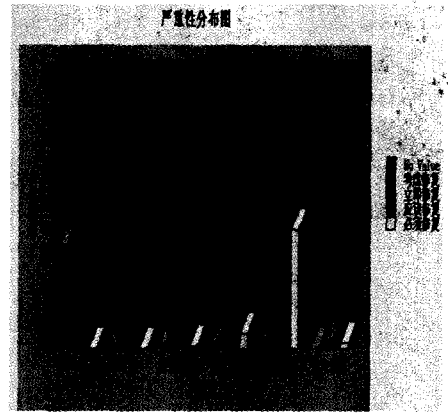


图6 缺陷严重性分布图

踪管理，便于软件缺陷的及时解决，为软件开发团队人员之间提供了信息和技术交流的平台，降低了软件开发的总体成本，提高了软件测试的效率，缩短了软件开发周期。总之，随着软件缺陷管理系统的不断完善，随着测控系统软件开发人员开发能力的提高，缺陷管理系统将会发挥越来越重要的作用，使用更容易、界面更友好、功能更完善的缺陷管理系统将广泛应用到地面测控系统中。

(作者单位：中国航天时代电子公司第七〇四研究所)

