

基于 DSH 架构的软件缺陷跟踪管理系统

马军霞, 张志锋

(郑州轻工业学院 软件学院, 河南 郑州 450001)

摘要:针对传统软件缺陷跟踪管理系统中用户体验差、业务效率低等问题,设计并实现了一个基于 DSH 架构的软件缺陷跟踪管理系统.系统采用 Dorado 展现中间件和 Spring, Hibernate 等 Java Web 主流技术,基于 B/S 模式构建,实现了项目管理、用户管理、缺陷跟踪管理、缺陷数据统计分析、报表输出等功能.应用实践表明,本系统改善了页面图形显示和用户体验,可扩展性和可维护性高,适用于中小规模软件开发过程中的缺陷跟踪管理.

关键词: DSH; Dorado; 软件缺陷; 缺陷跟踪; 管理系统

中图分类号: TP311 **文献标志码:** A **DOI:** 10.3969/j.issn.2095-476X.2014.04.021

Software defect tracking management system based on DSH architecture

MA Jun-xia, ZHANG Zhi-feng

(College of Software Engineering, Zhengzhou University of Light Industry, Zhengzhou 450001, China)

Abstract: Aiming at the problems of traditional software defect tracking and management system such as poor user experience and low business efficiency etc., a software defect tracking management system based on DSH architecture was designed and realized. The system was constructed by B/S mode, using Dorado presentation middleware and mainstream Java Web technology such as Spring, Hibernate, etc. The system realized the functions of project management, user management, defect tracking management, defect data statistical analysis, report output and so on. The application practice has indicated that it improved the page graphics display and user experience, had good scalability and maintainability, and was suitable for defect tracking of small and medium scale software development.

Key words: DSH; Dorado; software defect; defect tracking; management system

0 引言

随着计算机的普及和信息技术的发展,软件产品的质量越来越引起人们的重视.软件产品具有不可见性、灵活性以及高度复杂性等特征^[1],软件缺陷是软件的固有属性.软件项目管理的重要任务之一就是软件缺陷跟踪管理.使用工具是软件缺陷管理的基本理念之一,软件缺陷管理工具能够帮助软件组织有效跟踪和管理软件缺陷状态的变更,分析

软件缺陷数据,为相关决策提供数据支持,利用缺陷信息建立组织过程能力基线 PCB (process capability baseline),并实现量化过程管理,通过缺陷预防实现项目开发过程的持续优化,从而提高软件开发与测试的效率,提升最终软件产品的质量.因此,设计开发适用的软件缺陷跟踪系统对于软件组织实现有效的项目管理是必要的^[2-6].

在现有的缺陷跟踪管理系统、集成了缺陷管理功能的软件测试管理系统中,成熟的商业化产品功

能比较强大,比较注重与其他测试工具的整合.系统设计较为复杂,配置也有一定的复杂度,通常包含较多客户并不需要的功能,系统订制和管理人员也需要经过专业培训,经济成本较高.开源免费工具对客户的支持相对较差,有些操作界面不够友好,甚至本身存在缺陷.调查显示^[7],目前公司使用的缺陷管理工具中,商业化产品 HP Quality Center (QC,即 TD,现已升级更名为 ALM)和 Jira (Atlassian 公司)所占比例最高,其次是 Bugzilla, Bugfree 等开源工具.值得注意的是,还有 18% 的公司自行研发个性化的缺陷管理工具.

现有系统大多采用 B/S 模式开发,以传统 Web 方式提供服务,用户界面图形显示和交互性较差,页面通常比较简单,功能单一,需要通过页面刷新实现页面转换、完成业务要求,影响了系统的可用性,无法满足越来越复杂的用户体验需求^[8].通过邮件提醒缺陷信息的变更使相关人员比较被动,有可能无法及时跟进缺陷处理工作.

本文拟采用主流的 B/S 模式设计开发一个基于 DSH (Dorado + Spring + Hibernate) 架构的软件缺陷跟踪管理系统,即用 Dorado 展现中间件代替传统 SSH 架构中的 Struts 实现业务展现层,以改善页面设计和用户体验,提高页面转换效率和交互性,并尝试改进系统中软件缺陷状态更新提醒的实时性,进一步满足软件缺陷管理实际需求.

1 系统分析与设计

1.1 业务流程

本系统涉及的用户角色包括开发人员、测试人员和项目经理等软件开发相关人员.一个软件缺陷被发现、报告到这个缺陷被修复、验证直至最后关闭的完整过程称为缺陷的生命周期.本系统定义了提交、激活、挂起、已解决、拒绝、关闭 6 个缺陷状态:在软件开发过程中,测试人员发现缺陷后录入并提交缺陷;项目经理审查确认后,将缺陷置为激活状态,并将缺陷分配给相关开发人员修复;开发人员修复缺陷后交由测试人员验证,确认修复无误后关闭缺陷.此外,缺陷审查时若被认为暂时不需要修复或不具备修复条件,将被置为挂起状态;若认为不是缺陷,将置为拒绝状态.缺陷在不同角色用户之间的流转通过权限控制实现,并通过在公告板实时发布消息的方式通知相关用户.用户登录后将首先看到与自己相关的消息,以便及时了解并处理软

件缺陷.

1.2 功能分析

根据软件缺陷跟踪管理的实际需求,系统应具备以下功能:

1) 实现必要的权限管理.对于不同角色用户应设置不同权限,从而使不同权限用户可以对缺陷执行不同操作,以保证缺陷处理流程的完整性与系统数据的安全性.

2) 以项目为单位来组织缺陷数据的记录、存储,以满足多项目测试管理的需求.

3) 以特定的格式记录软件测试中所发现的缺陷,可上传附件辅助说明缺陷信息,并对缺陷状态的改变进行有效的跟踪、控制和管理;提供有效的缺陷变更提醒手段,使团队成员能得到即时反馈,以确保每个被发现的缺陷都能够及时得到处理.

4) 具备一定的统计分析功能.对系统中已经发现的缺陷信息进行统计分析,以监控软件开发过程质量,评估软件开发的阶段性成果,并为开发过程中的相关决策提供可靠依据.

5) 提供数据的导入、导出功能,以满足缺陷数据迁移的需要.

基于以上分析,将系统分为 4 大模块,系统的功能模块划分如图 1 所示.

1.3 系统架构

经分析比较,系统采用多层 DSH 架构、MySQL 数据库系统来完成系统开发,以保证系统的安全性、可移植性、可扩展性及快速开发,同时实现布局复杂、功能强大、操作流畅的页面效果.

系统基于 B/S 模式搭建,以用户浏览器为客户端,服务器端包括应用服务器和数据库服务器.系统以 Spring 为核心容器搭建框架基础并管理业务层,以 Hibernate 实现业务数据持久层,Dorado 作为 Web 展现层框架负责用户界面与操作逻辑.系统架构设计如图 2 所示.

2 系统实现与应用

2.1 系统实现

本系统开发环境为 Windows 操作系统,开发工具使用 MyEclipse 10, Dorado 5.4, 应用服务器采用 Tomcat 6.0, 后台数据库采用 MySQL 5.1. 在系统架构中, Dorado 负责用户界面与操作逻辑,业务逻辑由 Spring, Hibernate 框架负责.

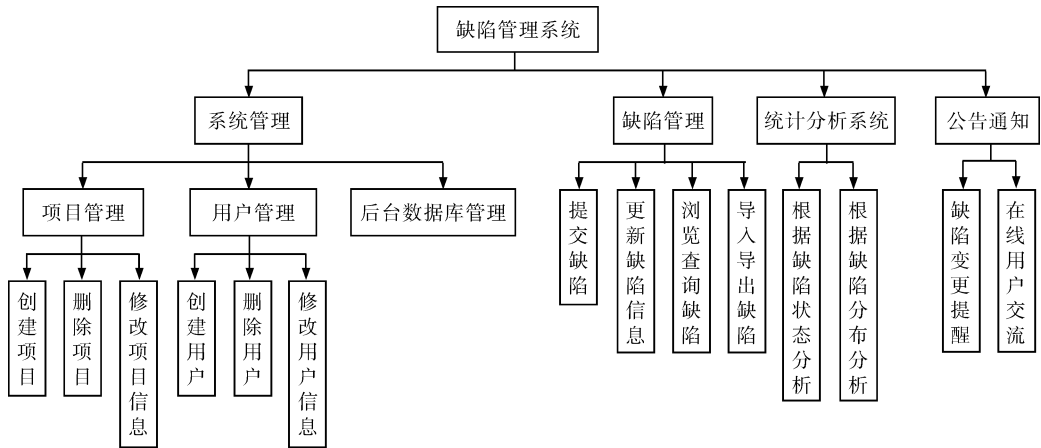


图 1 系统功能模块层次

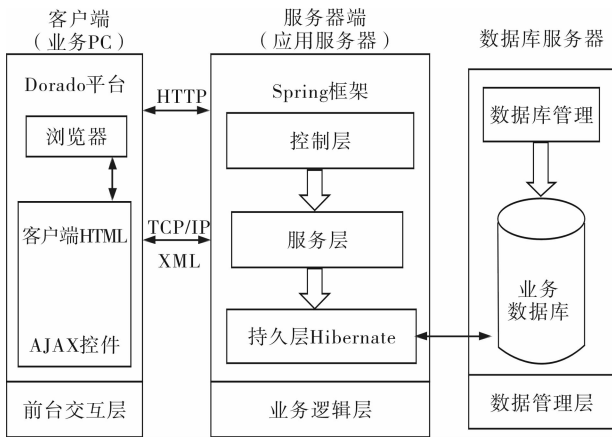


图 2 系统架构

1) 展现层. 系统页面的基本业务布局和展现都基于 Dorado 提供的常用页面要素和 AJAX 控件实现, 包括不可见类型组件和可见类型组件, 表现形式为 HTML + JavaScript 页面. Dorado 的核心是 XML 表述的 View 文件, Dorado 引擎对 View 文件进行解释后生成 HTML 文件并在浏览器运行. 用户界面的交互控制也通过 Dorado 处理. 通过编写前台 JavaScript 来实现界面要素逻辑的控制. 通过 AJAX 方式将需要交互的信息同步到后台. Web 客户端展现层实现用户与系统之间的交互, 将请求发送给应用逻辑层, 调用核心业务逻辑服务. 用户使用浏览器作为客户端, 通过 Web 应用服务器进行缺陷数据的录入、查询、更新等操作, 同时浏览器接收、验证用户的输入并显示从 Web 应用服务器返回的操作结果.

2) 业务层. 系统采用 Spring 作为业务类的核心管理容器, 完成业务逻辑服务组件的定义以及用户界面和数据管理层的信息加工. Dorado 通过 Marmot

应用框架实现与 Spring 的集成. 系统服务器端的核心业务逻辑服务包含控制层、服务层, 用于接收用户界面发来的请求并将请求结果返回给用户界面, 向数据管理层发送用户的数据操作请求并调用数据管理层的相关服务. 控制层响应客户端用户界面提交的服务申请, 并将其发送给服务层应用. 控制层采用 Dorado 的 Provide 和 Resolver 为用户页面提供数据, 并将页面数据集合 DataSet 提供给服务层, 从而实现控制层与页面的交互. 控制层没有任何业务逻辑代码, 不负责具体的事务处理. 服务层的业务处理异常信息也由控制层接收并处理, 返回友好的错误提示信息. 服务层封装并实现核心的业务逻辑服务, 负责完成具体的事务处理. 服务层接收用户界面发来的请求, 完成事务处理后将处理结果返回给用户界面, 并根据需要发送数据操作请求并调用数据管理层的服务.

3) 持久层. 系统采用 Hibernate 实现持久层, Marmot 应用框架通过 Spring 实现表示层与数据持久层的沟通, 使 Hibernate 完成数据的持久化, 主要表现为 XML 形式的 Hibernate Mapping 文件以及对应的实体类. 在 Dorado 的 View 中通过添加属性的方式来指定操作的实体, 形成统一通用的数据持久代码类, 完成数据提供和数据持久化.

2.2 关键技术

得益于 Dorado 在 Web 表现层开发方面的优势, 本系统中相关业务功能可以在一个页面上实现, 只需要少量页面就能承载系统所有功能的实现. 例如, 将项目列表、缺陷列表、当前选中缺陷的详细信息、实时更新的公告板、登录用户基本信息、

当前页面相关功能按钮等都设计到缺陷信息页面中,充分利用了 Dorado 对复杂页面的支持能力. Dorado 能够以秒级效率轻松处理多数据源的复杂页面. 复杂页面的实现使得用户可以在 1 个页面中完成同一业务功能的所有相关操作,而不必在多个页面中跳转,提高了用户业务操作效率.

在系统开发过程中,利用 Dorado 提供的页面要素和 AJAX 控件,只需通过轻点鼠标添加组件并配置相关参数,必要时添加少量代码,就可以像搭积木一样轻松地设计实现出美观大方的页面视图,从而减少了系统代码的书写量,提高了系统开发效率. Dorado 提供的页面元素功能强大. 如数据表格 DataTable 的全编辑特性使得表格中的任意单元格都可以编辑,单击相应的表头栏即可实现数据排序. 该组件还支持表格上部锁定、左部列锁定、锁定列数、可见列数自由调整、多表头组合、鼠标调整列宽与列顺序等特性. 这些特性在本系统的缺陷管理、用户管理、项目管理等主要功能页面中得到了充分体现,从而带给用户与众不同的操作体验.

此外,由于 Dorado 对 Ajax 技术的全面支持,可以实现页面局部刷新,使用户在复杂页面中执行的单一操作不会引起整个页面全部刷新,减少了客户端和服务端之间的数据传输,从而减少了系统响应时间,提高了系统性能. 本系统中,用户可以在缺陷列表中修改某个缺陷属性值,页面其他部分不会提交刷新;始终显示在页面前端的公告板内容会随着在线用户的操作实时更新,这些数据提交、内容更新操作也是局部实现的.

通过 Dorado 与 Spring, Hibernate 集成的多层架构,在保证系统可扩展性、可维护性的基础上改善了系统页面交互效果和用户体验,提高了用户业务效率和系统开发效率.

2.3 系统应用

本系统已应用于“缺陷跟踪与软件测试项目管理”课程的实践教学及多个大学生创新项目开发管理过程. 应用实践表明,本系统能够在不同 Windows 操作系统下稳定运行,兼容不同浏览器版本,能够帮助软件测试专业学生在实践中进一步理解、掌握软件缺陷跟踪管理基本流程和基本理念,能够满足

小规模软件开发管理中的软件缺陷管理需求.

3 结论

本文在分析现有同类系统的基础上,设计并实现了一个软件缺陷跟踪管理系统. 系统开发采用 Dorado 展现中间件及 Spring, Hibernate 等 Java Web 主流技术,基于 B/S 模式提供软件缺陷跟踪相关服务,通过系统公告板实现了缺陷状态更新的实时提醒以及用户间的即时在线通信功能,规避了邮件通知方式的信息延迟和被动性. 以 Dorado 作为 Web 展现层框架的 DSH 架构能够有效改善系统的用户体验与交互效率,降低网络流量、提高系统性能,同时提升开发效率. 使用本系统能够帮助软件开发团队有效地收集、跟踪、反馈软件系统在开发、测试过程中的缺陷,系统界面友好,操作便捷,性能稳定,可扩展性和可维护性高,基本满足了中小规模软件在开发过程中的缺陷跟踪管理之需求.

参考文献:

- [1] 于波,姜艳. 软件质量管理实践——软件缺陷预防、清除、管理实用方法[M]. 北京:电子工业出版社,2008: 1-2,6,298.
- [2] 连进,朱晓燕. 软件缺陷管理系统的研究[J]. 江汉大学学报:自然科学版,2008,36(2):54.
- [3] 罗霄,侯红. 基于开源的软件缺陷管理工具的改进策略[J]. 计算机工程,2009,35(1):65.
- [4] 张向宏,张翠然. 基于 CMMI4 的软件缺陷管理体系研究及应用[J]. 山西财经大学学报,2011,33(3):291.
- [5] 邹燕飞,罗鸿伟. 基于 Struts + Spring + Hibernate 缺陷管理系统实现[J]. 计算机技术与发展,2012,22(2):146.
- [6] 孙莉,艾磊. 基于 JIRA 的软件缺陷管理及二次开发研究[J]. 计算机应用与软件,2013,30(4):171.
- [7] 51Testing 软件测试网. 2012 年中国软件测试从业人员调查报告[EB/OL]. (2013-05-07)[2013-11-13]. <http://www.51testing.com/html/55/n-845155.html>.
- [8] 包一磊. Web 表现层——跑完龙套唱主角[J]. 程序员,2006(10):50.