EasyChair Preprint

№ 2211

ODMP: A Lightweight Online Software Defect Management Platform

Alexander N. K. Ottinah, Ya Pan and Ao Haiyang

December 21, 2019

ODMP：A Lightweight Online Software Defect Management Platform

Alexander N.K Ottinah
School of Computer Science and Technology
Southwest University Of Science and Technology
Mianyang, China
aottinah@gmail.com

Pan Ya
School of Computer Science and Technology
Southwest University Of Science and Technology
Mianyang, China
panya@swust.edu.cn

Ao Haiyang
School of Computer Science and Technology
Southwest University Of Science and Technology
Mianyang, China
oceanao@163.com

Abstract—Early detection, tracking, controlling and managing software errors effectively has become an important means to improve the quality of software and ensure the normal operation of the system. Commercial defect management software is more expensive, but some open-source defect managements like Bugzilla, are more or fewer deficiencies in meeting actual demand. it is found that these open source tools have certain redundancy functions with actual use, complicated processes, pictures can only be uploaded in the form of attachments, weak defect statistical functions, problems in data export and migration, etc.

This paper is to design and implement a lightweight online software defect management platform (ODMP) suitable for small software research and development teams; which uses SSM (Spring+SpringMVC+Mybatis) open-source framework for development, and achieves tracking and management of defects in different roles. Testers can submit defects, developers can view the tasks assigned to repair defects, and modify the defect status after completing the tasks assigned by the project team leader to repair defects, and also submit to the project team leader for defect confirmation. By providing visual defect statistics results, ODMP is a means that gives insight in the quality level of the code or documentation. To ensure that the process is uniform and standardized, this tool handles and controls defect effectively following an appropriate defect life cycle, which attains different states during its life span.

It shows by applying ODMP to some small development teams that the management of users, projects and defect levels is simplified, which can better meet the daily management needs of software testing team laboratories and can run steadily, basically meeting the needs of users.

Keywords— Software defects, Defect management, SSM framework

I. INTRODUCTION

Many domestic software research and development units have low awareness of defect management or are still in a relatively one-sided understanding. A common situation is that many software developers believe that software defect management is to completely avoid software defects in the software development process, but in fact this is only a one-sided understanding, software defect management is to find and predict software deficiency from the root cause, and taking appropriate measures to improve, in many cases cannot achieve complete evasion, but to improve the quality of software development.

The larger the scale of the software, the greater the probability of defects in research and development. Effectively tracking, controlling and managing software errors has become an important means to improve software quality and ensure the normal operation of the system. Only correcting these errors correctly, quickly, and accurately can eliminate software errors and ensure that the software to be released meets the goals of the requirements design. In the actual software testing process, each defect must be tested, confirmed, repaired, verified, etc., which is an important part of software testing. In some small teams, the developer often uses tools such as Bugzilla for software defect management. Bugzilla is an open-source defect tracking system that manages the entire lifecycle of bug submission, repair, shutdown, etc.; in software development. However, it is found that these open source tools have certain redundancy functions and actual use, and the process is cumbersome. therefore, in this context, the system's function being too large and difficult to meet the demand, it is especially important to develop a more practical software defect management system for the actual use requirements and business processes of the software test team laboratory.

Through this system, it will help small and medium-sized development teams to better collect, track, and feedback errors and problems in the testing and operation of software systems, and improve defect tracking efficiency. At the same time, simplify the management of users, projects, and defect levels, and better meet the daily management needs of software testing team laboratories.

II. METHOD AND SOLUTION

a) System Requirement

The ODMP system is used for a small team of 30-50 people in order to get efficient feedback of defect, the administrator is the Team leader, the project leader is the director of each project, and the developers and testers are members of each project.

The main functions of the defect management system are submitting defects, tracking defects and also helps in communication among members on a project. The system is divided into four roles: Administrator, Project leader, Tester, and Developer. These users have different functions according to their roles. Administrators manage the information in the system, including users, projects, defects,

and defect levels; create accounts for users, oversees the duties of a project leader. The project leader's main task is to assign defect, repair tasks and track the status of all defects of the project, can also view the statistical chart of defect. Testers submit and view defects. The developer repairs defect task assigned, view defects, and modify the defect status after completing the tasks assigned and submit the modify defect for confirmation and testing.

In addition, the project leader can also act as a tester and developer to complete the operations of submitting defects and repairing defects, as shown in figure 1 below.

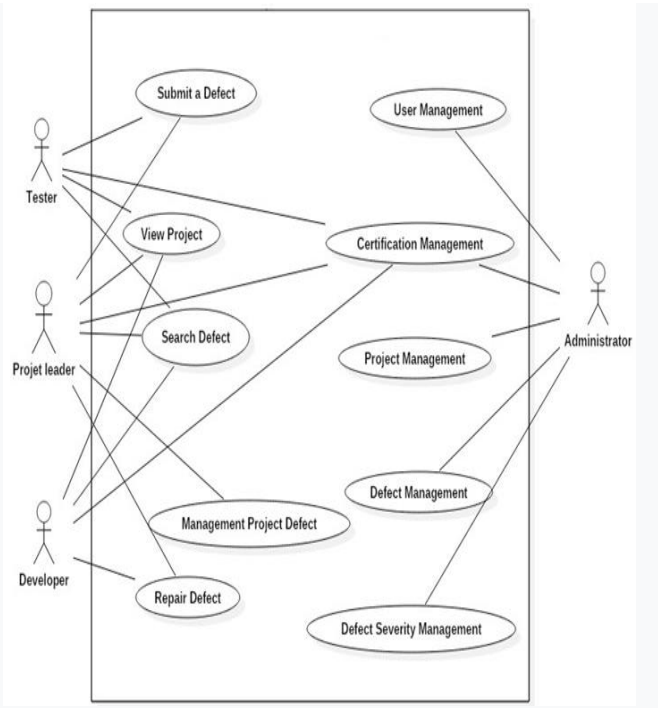

Figure 1. Use case diagram

b) System Architecture Design

The system is developed based on SSM (Spring+SpringMVC+Mybatis) framework and is a typical MVC framework model type, making the hierarchy of the system clearer. It is mainly divided into user interface layer (View), control layer (Controller layer), business logic processing layer (Service layer), and data persistence (Dao) layer. SpringMVC, an excellent MVC framework, is adopted as the control layer in SSM framework. Spring, as the core of SSM framework, runs through the whole middle layer and seamlessly integrates the control layer, business logic layer, data access layer, and entity objects. Application of data persistence layer Mybatis, as the ORM framework, maps the data of relational database into objects through SQL Maps, which makes it easy to operate the database in an object-oriented way. The whole SSM framework is built by annotation +XML, which is different from the traditional one, Compared with the SSH framework, it has fewer configuration files, simpler implementation, and easy development. The system architecture diagram is shown below in figure 2.

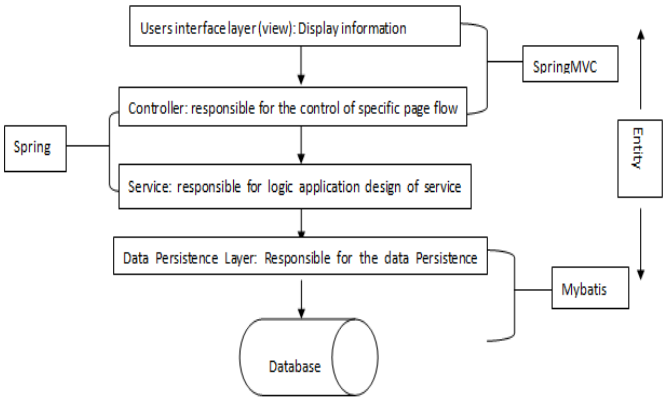

Figure 2. System Architecture Design

i. The user interface layer is mainly responsible for displaying information and calling the interface function corresponding to the control layer through JavaScript to obtain the required information; then the information is displayed in the browser in the form of JSP file.

ii. The control layer is responsible for controlling the process of specific business modules. In this layer, the interface of the business layer is called to control the business process. Different controllers are required for specific business processes;

iii. The Service layer is responsible for the logic design of service functions. First, the interface needs to be designed, and then the implementation class of the interface needs to be designed. The service implementation of the service layer needs to call the interface of the defined data persistence layer. Encapsulating the business logic of the business layer is beneficial to the independence and reusability of the common service logic.

iv. The data persistence layer mainly needs to design the interface function of the operation database, which is used for calling the service layer. The basic needs are to add, update, delete, query and other methods. Some special methods can be customized according to actual business needs.

The system is designed hierarchically, which is divided into four layers according to architecture analysis. In order to express the hierarchical design content in the following sections more clearly, the related classes of each layer and their functions are summarized as shown in Table 1.

Table 1 Summary of Class Names and Functions

Level	**Class name**	**Effect**
Dao	UserDao	User table database interface
	AdminuserDao	Administrator table database interface
	ProjectDao	Project Table Database Interface

	BugseverityDao	Defect Severity Table Database Interface
	BugDao	Defect table database interface
	BugfileDao	Defect attachment table database interface
Service	UserService	Service Interface Related to Students
	AdminUserService	Service Interface Related to Administrators
	ProjectService	Project-related service interfaces
	BugseverityService	Service Interface Related to Defect Severity Level
	BugService	Service Interface Related to Defects
	BugfileService	Service Interface Related to Defect Attachment
Controller	UserController	An execution processor related to the user authentication management operation
	UserManageController	An execution processor related to user management operation
	AdminuserController	An execution processor related to the administrator authentication management operation
	ProjectController	Execution processor related to project management operations
	BugseverityController	An execution processor associated with the defect severity level management operation
	bugController	An execution processor associated with the defect management operation
	BugfileController	An execution processor associated with the defective accessory management operation
Pojo	User	User entity object
	AdminUser	Administrator entity object
	Project	Project entity object
	Bugseverity	Defect level entity object
	Bug	Defect entity object
	Bugfile	Defect attachment entity object

c) Database Design

A relational database is used in ODMP software to organize the information collected and stored in related tables; MyBatis, as ORM framework, maps the data of relational database into objects through SQL Maps, which makes it easy to operate the database in an object-oriented way. The Related tables are connected to each other by some common attribute, which gives all members' easy access to view and repair defect base on your role and the type of information acquiring. Related tables are dependent on other tables for certain pieces of information as shown in figure 3.

Furthermore, we have listed some different entities that are going to comprise the database. With our library database, data collection details would include :

i. Users: Table id, Student ID, Name, Password, Role, Registration Time;
ii. Administrator: Table id, User Name, Password;
iii. Defect Level: Table id, Defect Level, Description;
iv. Project: Table id, Project Name, Project Leader id, Project Description, Version Number;
v. Defects: Table id, Title, Defect Level id, Item id, Submission Time, Submitter id, Handler id, Defect Status, Processing Result, Environment Configuration, Step Description, Comments;
vi. Defect attachment: Table id, Defect id, File Name, Upload Date.

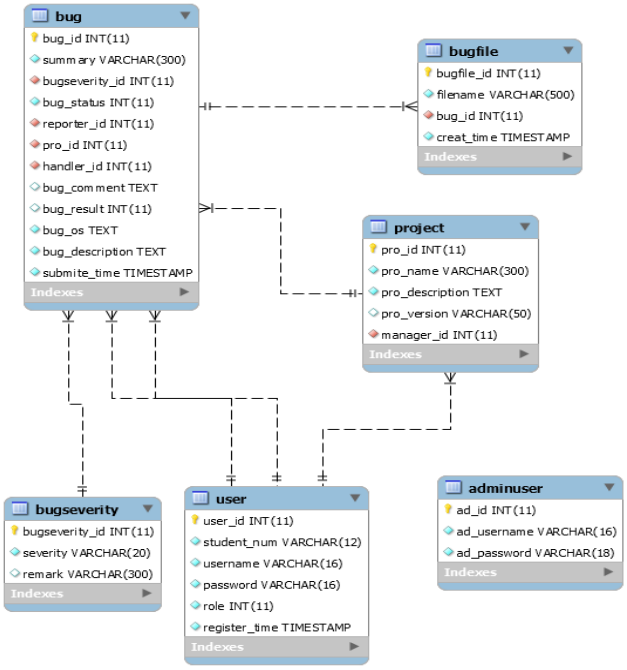

Figure 3. Relationship between the tables

d) Defect life cycle

The system needs to ensure that the final function is normal and the performance is stable. Most bugs arise from mistakes and errors made by people in either a program's source code or its design, or in frameworks and operating systems used by such programs, and a few are caused by compilers producing incorrect code. In order to handle the projects appropriately, we need to know how to deal with the development and release, but along with that we also need to know how to handle defects; and to control and handle the defect effectively; we need proper Defect life cycle.

Defect Life Cycle is a pattern in which a defect goes through during its life span and attains different states in the life cycle. It starts when defect is found by the tester and it goes through various stages during its lifetime and ends when a defect is closed by the tester, after ensuring it's not reproducible or rejected by a development team. The

number of stages that a defect goes through varies from project to project. The defect life cycle includes the followings stages: New, Assigned, Open, Fixed, Re-test and verified, Re-open and closed. The various stages are shown in Figure 4.

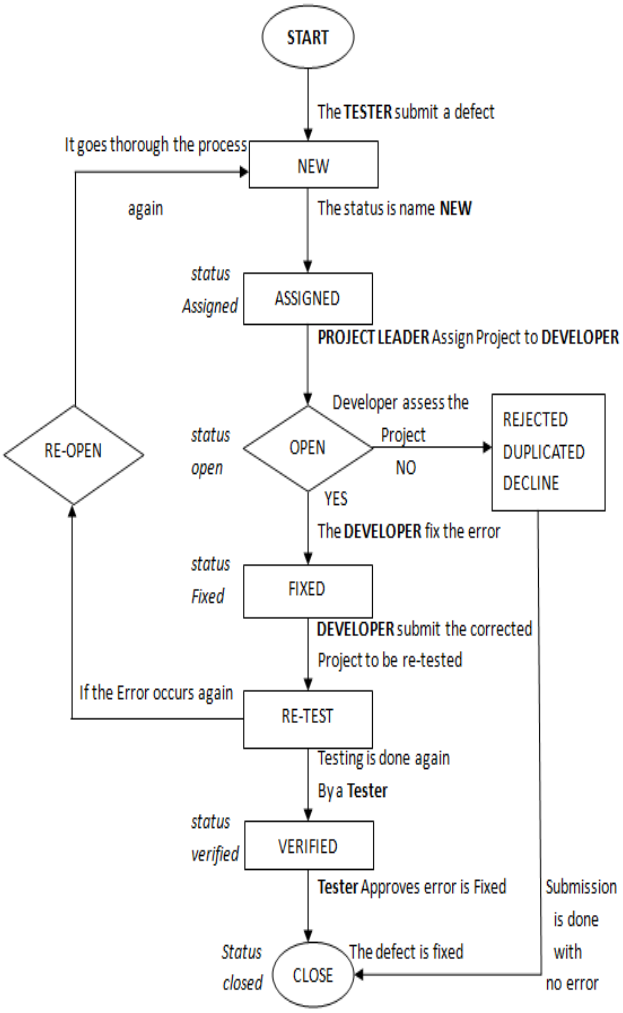

Figure 4 defect life cycle

III. RESULT

According to the design and implementation, some particular functions which have been shown in our result.

i. **Submit defect:** The tester clicks the submit defect button on the home page, enters defect title, defect level, affiliated items, defect status, environment configuration, step description, accessories and other information on the submit defect page, and clicks add to complete the submit defect function. After the tester submits the defect, the project leader of the project, then assign the task of repairing the defect to the developer and modify the defect status, change the defect process status to Assigned and add handlers to complete the operation of assigning the task of repairing the defect, in figure 5.

ii. **Repair defects:** After successfully logging in to the system, the user clicks on the left side to repair the defect. The system will display all defect records that need to be repaired by this developer in pages. Click on the processing defect icon on the line where the defect is located to enter the processing defect page. On this page, you need to modify the process status of this defect and add processing results and comments. Users include developers and project leaders as shown in figure 5

iii. **Statistic:** The project leader or administrator can click on the chart statistics icon in the row where the project is located to view the chart information of the project by defect level and the author respectively. The visualization of charts is completed by E-charts plug-in. The specific process is shown in Figure 5.

iv. **View defect:** Users can view defect details in the view project module, manage project module and search defect module. Users can click on the defect title to view all information about the defect, including attachments. The process of viewing the project module is similar to that of managing the project module. The specific process is shown in Figure 5

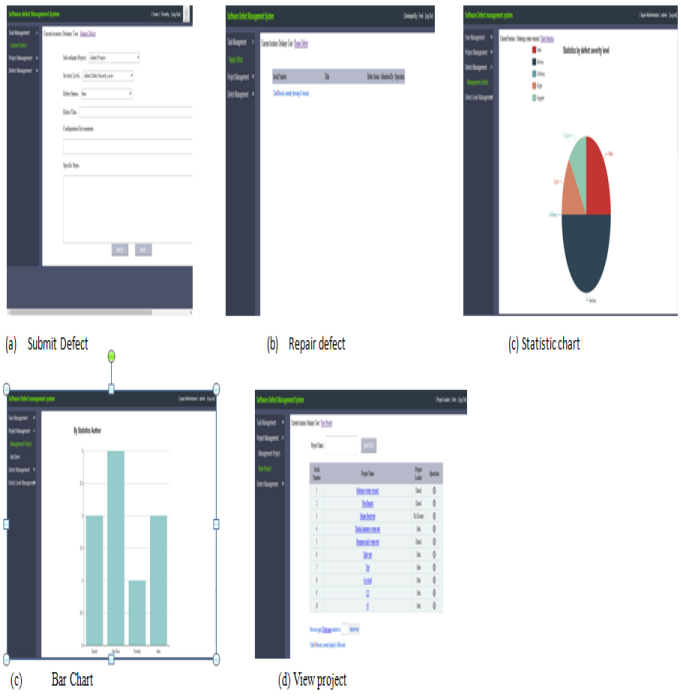

Figure 5 function

Most web pages are written in English, but only a few of the world's population uses English as their first language, this (ODMP) software was developed in two versions order to localize the online platform(Chinese and English).
Designing localization involves adapting visual aspects of a user interface to the other countries, such as the layout and overall aesthetics, beyond simply changing the language.

The core function module user authentication, submission defects and project management were selected for demonstration in the lab, on Operating system compatibility with win7, win8, win10, Mac10.10+OS with a Browser Compatible with four mainstream browsers: Chrome 77.0.3865.90 and Firefox 69.0.1, Safari10.0 above, at a Throughput rate of 10000kb/s, with Average response time Within 5s with Maximum concurrent users of 50.

However, the performance test was tested on the functions of logging in, submitting defects and searching for defects. A comprehensive security vulnerability scan was also conducted on the system with a tool AppScan.

To enable objective comparisons between website designs, we quantified each website's attractive by computing a set of 32 image metrics, such as the colorfulness, visual complexity, the number of image and text areas, and the saturation of colors.

For instance, the same message takes different amount of space in these two languages; the Standard formats of Dates, Numbers, and font-size also differs in Display.

IV. CONCLUSION

This paper first introduces the research background of the subject, points out the importance of software defect management in the software development process, and points out that it is of great practical significance to design and implement a lightweight online software defect management system for small software development teams.

This system uses the SSM framework and MySQL database to implement a software defect management system and has carried out system tests. It provides the team developers and testers with the functions of submitting defects, tracking defects, managing defects and so on, can achieve the expected goal.

Furthermore, the system has been tested for function, performance, and safety. From the test results, the system basically meets the system requirements and can run stably. In the database design, the conceptual model of the database is given in the form of Research and development diagram. Finally, several tables in the database are introduced.

However, this system is mainly used for software defect management; overall software defect management is only a part of the software product development process, which also includes requirements management, test case management, etc. Therefore, this system still needs to be improved and expanded. Therefore, the limitation of the system is relatively large.

V. REFERENCES

[1] Zou Hongting. Research and Application of Web System Based on SSM Framework [J]. Journal of Hunan Institute of Science and Technology (Natural Science Edition), 2017,(0 1):39-43.

[2] Yang Li. Design and Implementation of SSM Framework in Web Application Development [J]. Computer Technology and Development, 2016,(12):190-194.

[3] Shi yuemei. Data paging display processing based on SpringMVC and MyBatis [J]. Information and computer (theoretical edition), 2015,(1 9):138-140.

[4] Rong yandong. Applied research on Mybatis persistence layer framework [J]. Information security and technology, 2015,(12):86-88.

[5] Zhang Miao. Data Export Technology in excel Form Based on springMVC Framework [J]. Information Communication, 2014,(09):27-28.

[6] Chen Yan. Research and Implementation of Software Defect Management Technology [D]. University of Electronic Science and Technology, 2014.

[7] Rida Noor, Muhammad Fahad Khan. Defect Management in Agile Software Development [J]. International Journal of Modern Education and Computer Science (IJMECS), 2014, 6 (3). http://www.mecs-press.net/ijmecs/ijmecs-v6-n3/IJMECS-V6-N3-7.pdf.