

A Survey of Bug Tracking Tools: Presentation, Analysis, and Trends

Marko Trajkov¹, Aleksandar Smiljković²

In the building phase of every extensive project, the one thing that will happen for sure is a bug. Therefore, it becomes very important to have an appropriate bug tracking system. Poorly designed bug tracking systems are sometimes blamed for bad exchange of information and misunderstanding between developers, product managers and end-users. The purpose of this survey is to present some of the most common used bug and issue tracking systems, to address its problems and to suggest directions for their enhancements.

MSC2010: 68-02

Key Words: Software development, Bug tracking, Project management, Reports

1. Introduction

Term “bug” used to describe an error, mistake, failure, problem or fault in computer hardware or, much more often, software, which produces unexpected behavior. Most bugs arise from developers during the software building process inside the source code, and rarely are made by program compilers. Term “bug” has been introduced when Rear Admiral Grace Murray Hopper [1], USNR, (1906-1992) found a moth trapped between points at Relay # 70, Panel F, of the Mark II Aiken Relay Calculator, while it was being tested at Harvard University, on 9th September 1947. That is the first time when “bug” expression is mentioned as an error and since then, any cause of the problem in the computer software and hardware is called a “bug”.

The software industry has put much effort to find methods for preventing programmers from inadvertently produce bugs while writing software. These include programming style conventions, techniques, methodologies and code analysis. All these methods can help in reducing the number of bugs, but they do

not guaranty bugs-free software, since bugs are a consequence of the nature of human factors in the programming task. So, there is a need for tools and techniques that help to pursue the bugs after they are detected. Those tools are called bug tracking systems.

Nowadays, the use of bug tracking systems for pursuing and resolving the bugs and the other issues is well-spread. Bug tracking systems are used for organizing and monitoring the bugs. Without such systems, monitoring of a large amount of bugs can be almost impossible or very hard. Because of that, there are a lot of bug tracking systems available today. Differences between them can be observed in respect to quality, security, costs, and functionality which they offer to the users.

We can deal with bug tracking system as a part of much wider system for tracking of the project or as a separate, “stand alone” system. Both types have their own advantages and disadvantages but from this point of view only those that work as separate systems will be observed. The reason for this is to reduce the scope of this survey and therefore treat the problem in easier and more precise manner. From the other side, the project tracking systems are more tightly related to the specific problem.

Previous researches have indicated that items such as stack traces, reproducing steps, observed and expected behavior, test cases, and screenshots are often omitted by reporters [2] and that items are preferred information for developers. The effect of delay is that the bugs take longer to be fixed and more and more unresolved bugs accumulate in the project’s bug tracking system.

In this study, we pointed out the reasons for the importance of bug tracking systems and gave a review of the existing solutions and their characteristics. This survey proposes new trends and the optimal solutions for the bug tracking systems. The results presented in this study may be useful for two groups of users. The first group contains the individuals or teams who work on the projects that require bug or issue tracking system. This study may help them to better understand the ground and chose the appropriate solution for their need. In the second group are people who are developing the bug tracking tools. This survey should help them to build the system that will be better than the others by the set of features and functionality, and which will satisfy the wider user group.

2. Problem statement

Bug Tracking Systems allows individuals or teams of developers to keep tracking of outstanding bugs in their product effectively. Having complete information in the initial bug report, helps developers to resolve the bug problem quickly. Therefore, the adequate choice of a good bug tracking system for the product development helps developers to reduce downtime, increase productivity, raise customer satisfaction, and improve communication inside and outside a team.

Nowadays, projects are more complex, robust and harder to control. Obviously, it is much harder to keep in track a huge amount of bugs. On the other side, the projects are passing through many phases such as alpha, beta, release versions, etc, and many iterations are being launched since the first version. Because of this, it is important to track the all malfunctions that are detected in the projects life cycle.

Complexity of the projects is naturally growing over time. Versions of the existing projects increase regularly. The same bugs will continue to appear in every generation of developers. So, it is recommended to keep in records all malfunctions to improve the future software development.

3. Existing solutions of the problem

In this section we provide a short overview of existing solutions according to the problem statement, and their good and bad characteristics, from the point of their usability. In general, the presented solutions give excellent results under the specific problem of interest, but they do not address the general problem of interest for this survey.

Existing approaches for solving the problem will be presented in the following way: (a) The basic information; (b) Specific details about selected solution; (c) Further development trends of the approach; (d) A criticism of the solution; and finally (e) Possible improvements that could overcome the noticed drawbacks.

The section concludes with a classification of each elaborated solution. The classification criteria were chosen to reflect the essence of the basic viewpoint of this survey. At the end, we will summarize all the important parts of elaborated solutions.

3.1. Overview of existing solutions and their drawbacks

There are a plenty of bug tracking solutions available for developers that are used during the software building process. The purpose of this survey is not to cover all of them but only the most popular and representative ones that are widely used. The following subsections provide short descriptions and evaluations of the bug tracking systems that are the author's choice.

3.1.1. Bugzilla

Bugzilla is a very popular, actively maintained, bug tracking system [4], used and developed together with Mozilla, giving it considerable credibility. Bugzilla is based on Perl and once it is set up, it seems to make its users quite satisfied. It is not highly customizable, but in an odd way, that may be one of its features: Bugzilla installations look very similar wherever they are found, which means many developers are already accustomed to its interface and will feel they are in familiar environment. Bugzilla involves a system that can send the results of a particular search on a schedule to another user, or a group that is previously specified. Bugzilla has an advanced reporting system, which enables us to create different types of charts including line graph, bar graph or pie chart.

3.1.2. Mantis

Mantis is a free web-based bug tracking system [5]. It is written in the PHP scripting language and works with MySQL, MS SQL, and PostgreSQL databases and web server. Mantis can be installed on Windows, Linux, Mac OS and OS/2. Almost all web browsers are able to function as clients. Mantis is released under the terms of the GNU General Public License (GPL). The main complaint about this solution is its interface, which does not meet modern standards. On the other hand, it is easy to navigate, even for inexperienced users. Another shortcoming is the lack of advanced features such as charts and reports. Moreover, there are plenty of bugs and very little functionality.

3.1.3. BugTracker.NET

BugTracker.NET is a free, open-source, web-based bug tracker or customer support issue tracker [6], written in ASP.NET, C#, and Microsoft SQL Server Express. BugTracker.NET is easy to install and learn how to use. When first installed, it is very simple to setup and one can start use it right away. Later, users can change its configuration according to their needs. BugTracker.NET

has a very intuitive interface for generating lists of the bugs. It has two very useful features. The first one is a screen capture utility that enables to capture the screen, add annotations and post it as bug in few steps only. The second feature is the fact that it can integrate with subversion repository, so it is possible to associate file revision check-ins with bugs.

3.1.4. Flyspray

Flyspray is a web-based bug tracking system written in PHP [7]. It is free software, released under the General Public License, which means that Flyspray can be downloaded and used free of charge. The source code is available, and everyone is welcome to modify it to suit its needs. It is described as "uncomplicated", and the list of features includes: multiple database support (currently MySQL and PGSQL), multiple projects, 'watching' tasks, with notification of changes (via email or Jabber), comprehensive task history, CSS themes, file attachments, advanced search features, RSS/Atom feeds, wiki and plaintext input, voting, dependency graphs.

3.1.5. Redmine

Redmine is flexible web-based project management web application [8], written in Ruby on Rails framework that is cross-platform and cross-database. Redmine is open source and released under the terms of the GNU General Public License. Redmine is flexible issue tracking system. User can define its own statuses and issue types. Redmine supports multiple projects and subprojects. Each user may have a different role on each project. Interface is very simple, intuitive and easy to navigate. In general, this is very good product and deserves recommendation.

3.1.6. Bug-Track

Bug-Track is a web-based defect and bug tracking software [9], which allows user to document, manage and assign all bugs and tasks and empowers its users to organize bugs, defects or issues into distinct projects. It can run on any web-server like Microsoft, Linux, Unix, etc. Since it is commercial application it is expected that it is better than free products. However, it has nothing new and better than other free bug tracking systems. The only advanced thing is a fact that it has more intuitive interface and that is the only benefit compared to the other solutions.

3.1.7. Bugzero

Bugzero is a web-based bug, defect, issue and incident tracking software [10]. Its single code base supports both Windows and Unix (based on Java) and supports database systems including Access, MySQL, SQL Server, Oracle, etc. Bugzero can be customized for software bug tracking, hardware defect tracking, and helps desk customer support issues and incidents tracking. Bugzero has intuitive interface, but lacks form features. The main drawback is the fact that Bugzero is a commercial product that offers less advantages compared to other solutions that are free of charge.

3.2. Classification criteria and classification tree

In this survey, for the classification criteria we selected the most common features of the observed bug tracking tools. Those features are useful for an average user and they offer a user-comfortable bug tracking system. Some more advanced criteria will be given in Section 4. All classification criteria considered in this survey are given in Table 1. Each classification criteria are given in the caption of Table 1, and further explained.

	Search	Email notifications	Reports	Charts	Time Tracking	RSS/Atom Feed	Configurable	Free of charge
Bugzilla	yes	yes	yes	Yes	yes	no	no	yes
Mantis	yes	yes	no	No	yes	yes	no	yes
BugTracker.NET	yes	yes	yes	Yes	yes	no	yes	yes
Flyspray	yes	yes	yes	No	no	yes	yes	yes
Redmine	yes	yes	yes	yes	yes	yes	no	yes
Bug-Track	yes	yes	yes	yes	no	yes	no	no
Bugzero	yes	yes	yes	no	no	no	no	no

Table 1: Classification criteria.

Table 1 summarizes the criteria used in the decision making process of choosing suitable bug tracking system, such as: (a) Search, (b) Email notifications, (c) Reports, (d) Charts, (e) Time tracking, (f) RSS/Atom Feed, (g) Configurable and (h) Price. The brief description of each criterion follows.

- (a) **Search** is very useful criteria and it is present in all selected products.
- (b) **Email notifications** give users the opportunity to be informed about news on the issues of their interest. The user does not need to check manually bug tracking system to check if there were some changes.

- (c) **Reports** provide users with a brief and concise overview about past happenings in the bug tracking system.
- (d) **Charts** give a clear graphical view of the selected criteria which is very intuitive to the human being.
- (e) **Time tracking** is a feature that keeps tracking the information about changes of some specific bug trough time.
- (f) **RSS/Atom feed** gives users the opportunity to be notified about happenings in the current bug tracking system in the form of e-mail.
- (g) **Configurable system** is capable to be configured to meet certain user needs.
- (h) **Price** is often an important decision making criterion about whether to use some system or not.

The simplified classification tree, derived from the above introduced classification criteria, is presented in the Figure 1. Each leaf of the classification tree is given a name, as described in the caption of Figure 1. For each leaf, the list of existing solutions is given.

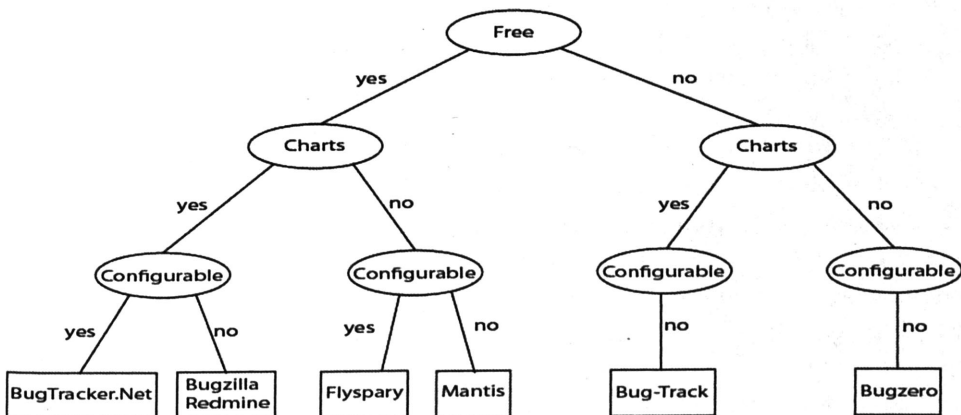


Figure 1: Classification tree.

From all above presented, we conclude that, among the existing solutions, none of them can be treated as the best one, for the general solution of this survey. Every one of them has some advantages and disadvantages. Some of them have more features than the others but in the general, the set of features are the same.

4. The proposed expert system

When a user or developer submits a bug report, he is usually asked many questions [11], such as: What is the name of the product? To which plug-in/component it belongs? What is the Build ID? What is the bug about? What are the steps to reproduce the bug? Is there any additional information? However, the initial information provided in a bug report is often incomplete and developers often have follow-up questions: Do you have all utilities installed? Is there any screenshot? Getting replies by users takes time (often weeks) and slows down the progress of open bugs.

The current bug tracking systems rely on some different selection criteria, which sometimes cannot give satisfying results. The authors of the paper “What makes a good bug report?” [2] believe that including selection criteria like stack traces, steps to reproduce, observed behavior, expected behavior, test cases, screenshots and dependencies will improve current bug tracking systems.

Therefore, we propose to build expert systems that automatically ask relevant questions and gather all required information once an initial failure description has been provided. The selection and order of the questions should not be static (as in current bug tracking systems), but rather depend on previous responses. In Table 2 we present selected criteria of our new expert system: Stack trace, Steps to reproduce, Observed behavior, Expected behavior, test cases, Screenshots and Dependencies. As it can be seen from Table 2, the existing bug tracking tools lack from some important properties (with few exceptions). We believe that this improved set of criteria will provide users with better and more useful information about the current bug. This will lead more efficient locating of the current defect and thus, a faster respond to it.

5. Conclusions

Current bug tracking systems do not effectively collect all of the information needed by developers. Without this information developers cannot resolve bugs efficiently, so we believe that there is a space for improvements on how

	Stack trace	Steps to reproduce	Observed behavior	Expected behavior	Test cases	Screenshots	Dependencies
Bugzilla	no	no	no	no	no	no	No
Mantis	no	yes	no	no	no	no	No
BugTracker.NET	no	no	no	no	no	yes	Yes
Flyspray	no	no	no	no	no	no	no
Redmine	no	no	no	no	no	no	no
Bug-Track	no	no	no	no	no	no	no
Bugzero	no	no	no	no	yes	no	yes

Table 2: The better selection criteria for proposed approach.

the bug tracking systems collect needed information. We summarized criteria that are used in modern bug tracking systems. Such criteria often do not give appropriate results in describing the bug. We further proposed an improved set of criteria that will give a much more satisfying solution for the current system.

This survey can be helpful to the designers of the future bug and issue tracking systems. They may consider the importance of selected criteria for describing of the bug because a well described bug will be easier to find and to be resolved. Designing such a system will give an answer to our assumption, which will confirm our work and apply new ideas to the current bug tracking systems.

At the end, we have noticed that current bug tracking systems are web-oriented, and such trend will be continued. It is very important to design systems that will use selection criteria for better bug description.

References

- [1] Rear Admiral Grace Murray Hopper, USNR, (1906-1992), <http://www.history.navy.mil/photos/pers-us/uspers-h/g-hoppr.htm> (Jan. 10th, 2012).
- [2] N. Bettenburg, S. Just, A. Schroter, C. Weiss, R. Premraj, and T. Zimmermann. "What makes a good bug report?" In FSE'08: Proceedings of the 16-th International Symposium on Foundations of Software Engineering, pages 308–318, November 2008.

- [3] S. Breu, J. Sillito, R. Premraj, and T. Zimmermann. What developers ask about your bug? Technical report, University of Calgary, December 2008.
- [4] Bugzilla, Bug tracking used by the Mozilla projects, www.bugzilla.org
- [5] Mantis, Popular free web-based bug tracking system, www.mantisbt.org
- [6] BugTracker.NET, issue tracking, bugtracker.net
- [7] Flyspray, Bug tracking system written with a focus on ease-of-use, flyspray.org
- [8] Redmine, A flexible project management web application, www.redmine.org
- [9] Bug-Track, Web-based bug tracking software and project management system, www.bugtrack.net
- [10] Bugzero, A bug, defect, and issue tracking tool, www.websina.com
- [11] S. Breu, J. Sillito, R. Premraj, and T. Zimmermann. “What developers ask about your bug?”, Technical report, University of Calgary, December 2008.

*Faculty of Mathematics,
University of Belgrade,
Studentski trg 16/IV,
11 000 Belgrade, SERBIA,*

¹ *markostrajkov@gmail.com,*

² *aleksandarsmiljkovic@gmail.com*