A Study of various Defect Tracking systems in Software Tools

Adarsh[1] Pawan Kumar[2]
[1]Research Scholar, NIILM University, Kaithal
[2]Assoc. Professor, NIILM University, Kaithal

Abstract - **This paper introduced lots of Defect tracking systems in the market deciding on a bug tracking tool to use for the development of your app can be confusing. The sheer amount of tools that have surfaced in the recent years and the similar functionalities they offer can make it difficult to determine which one will be best for you and your team. For this work some defect tracking tools have been studied and discussed. Later out of those tool we have compared only some top tracking tools with some parameters (Price, issue/Defect handling, Suitability with Team size, Integration, Workflow Customization, Manufacturer name and etc.)**

Keywords: Software issues, Bug tracking.

I.   INTRODUCTION

A bug tracking system, also known as a defect tracking system**,** is considered to be a software application that helps to keep track of the reported software bugs in all the software development projects. Bug tracking tools are regarded as a type of issue tracking system. It is kind of a computer program, used by the team of application support professionals, to keep track of the various issues that the software developers face [8]. If a tester is testing a software, s/he might come across several terms like failure, defect, bugs, mistakes, fault, error, etc., which are used by them interchangeably. However, have you ever wondered, whether these terms are synonyms of one another or do they specify different aspects of software testing process**?** Well, as an answer to this question, here is a detailed differentiation of all these terms, which will help us differentiate them from one another.

II.   DIFFERENT ASPECTS OF THE SOFTWARE ISSUES

 All the terms- issue, error, mistakes, faults, bugs, failures, and defects are used interchangeably by the majority of persons all around the world, but they signify and represent different aspects of the software. These terms are an integral part of the software testing process and without detecting and identifying them the team of testers cannot validate the quality, effectiveness, functionality, and more of the software.

Error & Mistake:
During the process of software testing, errors are the most basic discrepancies found by the team of testers. These are the mistakes made by the software developer or programmer, while preparing the code or design of the software. Errors are mainly a deviation from the results expected by the team, which further changes the functionality of the software.

Fault:
Introduced in the software because of an error, fault is another discrepancy found by the team of testers during the process of software testing. Unlike error, the reason for a fault to occur while implementing the software is not because of a miscalculation of some value or discrepancy between actual and expected result, but is merely a manifestation of an error in a software. Moreover, a fault in the software system inhibits it from performing its intended function and forces the system to act in an unanticipated manner.

Bugs

Bugs are the most integral part of a software system and can be termed as the errors, flaws, and faults present in the computer program that impact the performance as well as the functionality of the software can cause it to deliver incorrect and unexpected results. These not only impact the performance of the software, but also cause it to behave in an unanticipated way.

Failure

When a software is incapable of performing the required functions and is offering results that are not adequate and far from the expected results, then it is termed as a failure. These failures are incorrect external behaviour that leads a software to deliver services that are not in compliance with the specifications.

Defect

If the actual result of the software deviates from the one expected and anticipated by the team of testers, while testing the software, then it results into a defect. Defects, are therefore, defined as any deviation or irregularity from the specifications mentioned in the product functional specification document by the client or any other stakeholders of the project. These are the discrepancies and issues in the software, found during the process of testing, that impact its functionality. Moreover, defects are errors found after the application goes into product[1,2].

III.   SOFTWARE DEFECT PREDICTION (SDP)

The objectives of SDP are as follows:

- The efficiency of the testing phase is improved through defect prediction and also helps the developers evaluate their software product's quality and identify its proneness (Umar SN, 2013).
- The managers are helped through the allocation of resources, rescheduling, training plans, and allocation of budgets.
- Based on the trends of forecasting, there is an efficient ramping up of resources up or down, and gaps in skills and training could be overcome.
- Defect leakage into production is predicted.

Software artifacts such as subsystems of files are classified appropriately through most of the SDP instigation as to whether Fault-Prone (FP) or not. The numbers of defects are investigated through other SDP, which may appear in software artifacts so that ranking can take place (Xia et al. 2014). An outline of the SDP scheme is depicted in Figure 1.1.
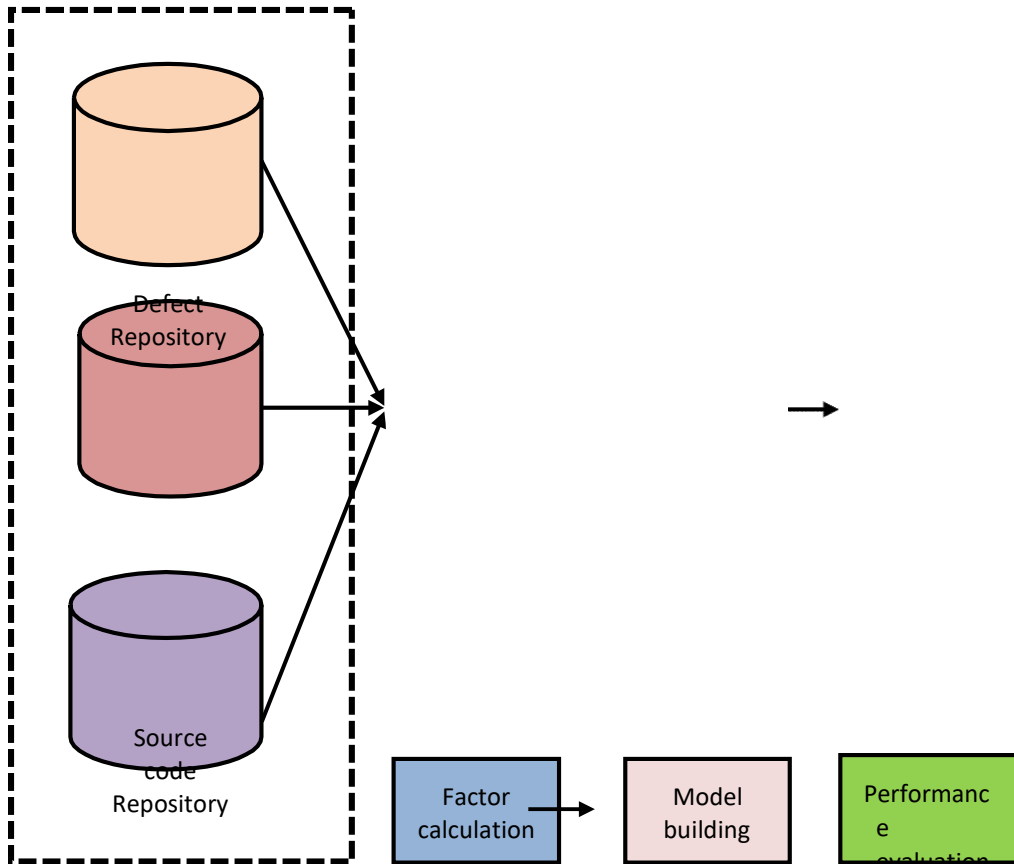
Figure 1.1 Overview of Software Defect Prediction (SDP)

At first, a collection of project data takes place from the software repositories (e.g., defect and source control repositories). From the data, factors are calculated. The locations are predicted through statistical and machine learning models, which have a better potential for the defects contained. Ultimately, using prediction models, various measures are evaluated, such as precision, recall, and explanative power (Shihab E, 2012).

SDP is focused on an extensive body of work. Their unique data is made use of by each of these works, with dependent and independent variables, techniques in modelling and evaluation of their models differently.

IV. VARIOUS DEFECT/BUG TRACKING TOOLS

Lets discuss some Defect Tracking tool and their works on issues Table 1 is showing the Defect Tracking and management tools [3].

Table 1.1 Defect Tracking and management tools.

Sr. No.	Name of DTS/BTS	Graphic Symbol	Description
1	Airbrake		Airbrake is the leading bug tracking software that provides error tracking for +50,000 developers.
2	Backlog		Backlog is an online bug tracking and project management software built for development teams. It's easy for anyone to report bugs with a full history of issue updates, comments, and status changes. Reported issues are easy to

			find with search and filters.
3	ReQtest		ReQtest is a powerful bug tracking software that allows Developers & Testers to collaborate on fixing bugs using the "Agile board". There is a dedicated bug module to report bugs
4	Bugzilla		Bugzilla[8] has been a leading bug tracking tools widely used by many organizations for quite some time now. It is very simple to use, web-based interface. It has all the features of the essence, convenience, and assurance. It is completely open sourced and is free to use.
5.	Asana		Asana is a web and mobile application designed to help teams organize, track, and manage their work. Forrester, Inc. reports that "Asana simplifies team-based work management." It is produced by the company of the same name[7].
6	Pivotal Tracker		Pivotal Tracker was built by Pivotal Labs in 2006 to simplify and facilitate the development process for teams using agile methodologies for software development. It is an opinionated project management tool that supports bug tracking as part of the project management process[10].

V. CONCLUSION

There are lots of Defect tracking systems in the market deciding on a bug tracking tool to use for the development of your app can be confusing. The sheer amount of tools that have surfaced in the recent years and the similar functionalities they offer can make it difficult to determine which one will be best for you and your team. Here we are discussing only some top tracking system and will compare them using some parameters (Price, issue/Defect handling, Suitability with Team size, Integration, Workflow Customization, Manufacturer name and etc.). Perhaps the biggest deciding factor affecting your choice of a tracker will be the size and distribution of your team. Larger and more distributed teams require more powerful solutions that can handle a high volume of issues and make team communication and collaboration easier

REFERENCES

[1]A.Gantes and j. stucky, "A platform on a Mobile Ad hoc Network challenging collaborative gaming," international symposium on collaborative technologies andsystems, 2001.
[2][Lyu 2007] Lyu, M.,(2007) "Software reliability engineering: A roadmap". In FOSE '07: 2007 Future of Software Engineering, pages 153–170, Washington, DC, USA, 2007. IEEE Computer Society. ISBN 0-7695-2829-5. doi: http://dx.doi.org/10.1109/ FOSE.2007.24.
[3][Jones 2008 b]Jones,C., "Applied Software Measurement: Global Analysis of Productivity and Quality". McGraw-Hill, New York, NY, USA, 3rd edition, 2008b. ISBN 978-0-07-150244-3.
[4] https://www.softwaretestinghelp.com/popular-bug-tracking-software/
[5]"Why is Trello Called Trello?". rewindandcapture.com. Retrieved July 10, 2020.
[6]Rao, Leena (September 13, 2011). "Joel Spolsky's Trello Is A Simple Workflow And List Manager For Groups". TechCrunch. Retrieved February 10, 2012.
[7]"The Forrester Wave™: Enterprise Collaborative Work Management, Q4 2016". www.forrester.com. Retrieved May 18, 2020.
[8]"Facebook co-founder Dustin Moskovitz unveils new company, Asana". LA Times Blogs - Technology. November 2, 2011. Retrieved May 18, 2018.
[9]https://www.bugzilla.org/about/

[10] Dash, Anil (November 14, 2017). "Introducing Manuscript". Retrieved March 20, 2018.
[11] https://en.wikipedia.org/wiki/Pivotal_Software
[12] https://www.atlassian.com/software/jira
[13] "What does JIRA mean?". *Atlassian.com official website*. Retrieved 5 November 2019.