Bug Tracking System

Introduction

This guide is useful for:

- Software developers who aren't sure if their team needs a bug tracking tool,
- Team Leads struggling to organize their bug management workflow, and
- QA team members wanting to learn more about refining their process.

The first part of this guide will go over the basics of what bug and issue tracking tools do. The second and third parts will show you how Backlog can improve your workflow, and how to use Backlog for bug tracking. The fourth part of this guide will cover some bug tracking best-practices.

Generally, a bug can be defined as an error in the product's code. Bugs may lead to improper work of the developed software. That is why they must be detected before the final product is provided to the customer. Software development teams usually associate bug tracking with the process of testing.

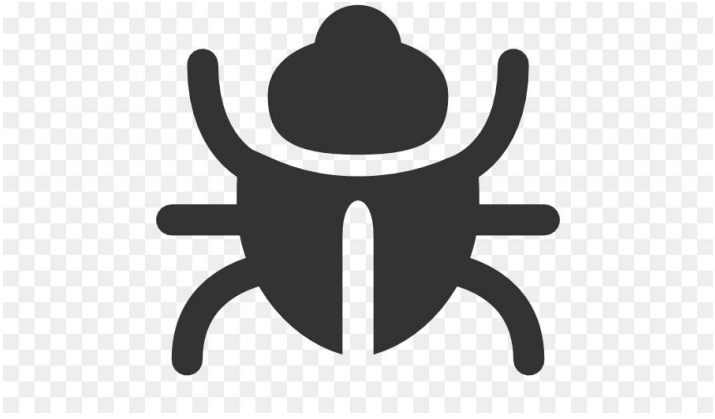

Figure 1

Bug tracking systems are applications that help the teams of software developers detect bugs, report on them, and create bug lists. Bug tracking is an important stage of software testing. Usually, the bugs are tracked with the help of special software. There are lots of such applications in the

modern software industry. All of them were designed to make the work of software developers easier.

Why use a bug and issue tracking tool?

For software development teams, finding, recording, and tracking bugs is an important part of creating a quality product. Bug and issue tracking tools streamline that process, making it easy for teams to prioritize, assign, and fix bugs quickly and efficiently. In lieu of investing in a quality bug tracking tool, some software teams choose to use spreadsheets for bug tracking and reporting.

While this may work as a temporary solution for small teams and projects, this method isn't sustainable.

Consequently, teams end up with a few predictable problems:

- **Lack of communication.** It's difficult to have conversations about each bug on a single shared spreadsheet.
- **Lack of visibility.** Team members end up having conversations via email, and those conversations lack any transparency to other team members.
- **Lack of real-time updates.** While you can check a shared spreadsheet at any time, there is no formal notification feature for alerting members of new bugs, progress, and updates.
- **Lack of fluid priorities.** All of the above lead to problems prioritizing and fixing the highest priority bugs first.
- **Lack of a central repository.** Descriptions, screenshots, feedback, etc. are spread across too many platforms, and rarely easily accessible to everyone.
- **Lack of insights.** Without an organized set of data about your bugs to glean insights from, it can be challenging to discover patterns and trends in your software defects.
- **Lack of integration with other work.** With spreadsheets or even tools that are only meant for bug tracking, you'll always eventually run into the same problem: your bug tracking workflow is happening in a silo, disconnected from the other tasks and projects your team is working on.

Bug tracking tools provide a transparent, centralized location for all of your bug tracking needs. They facilitate communication and provide real-time updates. Plus, tools like Backlog create a living archive of every bug your team encounters, so every team member today and in the future can access information about them.