

Streamlining Software Quality: The Comprehensive Bug Tracking Methodology

Prof. A G Vishvanath S^{1*}, Shrinidhi H R²

Assistant Professor, Department of Master of Computer Application, BIT Bengaluru¹

PG Student, Department of Master of Computer Application, BIT Bengaluru²

Abstract: The Default Detect System (DDS) is a comprehensive software program created to make it easier to find, report, and fix software bugs or problems. Throughout the bug lifetime, this approach makes it easier for developers, testers, and stakeholders to collaborate and communicate effectively. DDS has an intuitive user interface that enables users to submit thorough bug reports that include descriptions and prioritization, assuring accurate and transparent statuses of found issues.

Delivering high-quality goods is crucial in the quick-paced industry of software development, thus it's important to ensure a seamless and effective procedure. In contemporary development workflows, a default detect system (DDS) is a crucial tool that allows teams to quickly locate, document, and fix problems. This abstract gives a brief description of a groundbreaking DDS that was created to improve user experiences, accelerate development cycles, and foster collaboration.

Keywords: developers, bugs, quick-passed, stakeholders

I. INTRODUCTION

In the dynamic world of software development, ensuring the smooth functioning and high performance of applications is crucial. However, even the most meticulously crafted software is bound to have imperfections—bugs that can hinder its efficiency and impact user experience. Enter the Default detect system, a powerful and indispensable tool designed to identify, monitor, and rectify these bugs throughout the software development life cycle.

At the heart of every successful software development team lies the Default detect system—a centralized platform that acts as the vigilant guardian of software quality. From the initial stages of code writing to the final deployment, this system serves as a watchful eye, capturing and documenting every anomaly and aberration that arises.

The Default detect system is more than just a repository for bug reports. It serves as a virtual command center, where developers, testers, and project managers collaborate seamlessly. With its intuitive interface and easy-to-use features, it fosters effective communication, allowing team members to share vital information and track bug progress in real-time. One of the most significant advantages of the Default detect system is its ability to maintain a comprehensive historical record of detected bugs. This historical data becomes a treasure trove of insights, aiding developers in identifying patterns, recurring issues, and areas of improvement. As a result, the system contributes to the evolution of software development practices, fostering a culture of continuous enhancement and innovation.

II. LITERATURE SURVEY

During the software validation process, testers will find defects and then submit them to the Project Manager and developer using simple channels of communication like shared email lists and shared documents. The great majority of organizations will include this information in a file that is referred to as a "defect report." Due to the lack of a specialized tracking system, this procedure carries a high risk of error.

Further complicating matters, there is a real chance that some of these problems may go unaddressed, increasing the possibility that some vulnerabilities won't be fixed. A number of variables, including the growing complexity of projects undertaken by teams, have contributed to the development of numerous tools and project management approaches. The cohesion of the elements of open-source project management is of highest importance because using issue tracking software is required for managing open-source projects. The total number of defect reports delivered to various software development projects throughout the course of the last couple of decades has unavoidably increased.



III. METHODOLOGY

In today's dynamic software development landscape, a robust Default detect System (DDS) is essential for ensuring high-quality products. This methodology outlines a streamlined approach to implement an effective DDS that optimizes bug identification, tracking, and resolution processes. By leveraging modern tools and practices, teams can elevate their software quality assurance efforts and deliver exceptional user experiences.

Methodology:

Requirement Analysis

Begin by understanding project requirements, user expectations, and system specifications. Collaborate with stakeholders to define clear bug tracking objectives and establish relevant metrics for assessment.

DDS Customization

Set the DDS up to record pertinent bug data including severity, priority, status, and reproducibility instructions. Make integrations with version control systems, specialized workflows, and automated notifications.

Bug Reporting

Encourage the participation of the entire team in bug reporting. Create an intuitive interface that enables thorough bug explanations, screenshots, and attachments.

Bug Triage

A specialized bug triage team should be established to examine incoming bug reports. Prioritize bugs according to their importance and user effect.

Bug Tracking

Assign each bug to the appropriate developer and provide regular updates on bug resolution progress. Utilize tags or labels for easy categorization and filtering.

Regression Testing

Conduct comprehensive regression testing to ensure bug fixes do not introduce new issues into the system.

Bug Closure and Documentation

Close resolved bugs after validation and document the entire bug tracking process, including lessons learned and improvements for future projects.

Continuous Improvement

Regularly review the DDS process and gather feedback from team members to identify areas for improvement. Adapt the DDS methodology to accommodate changing project requirements and technologies.

IV. MODULES IN SYSTEM ARCHITECTRE

User module

Here user may enter all the required details to Default Detect system. Empower users to report bugs seamlessly and track their status effortlessly with our user-friendly Default Detect System module.

System module

The core engine of our Default Detect system. It manages bug data, user authentication, and database interactions, ensuring a seamless and secure bug tracking experience for all team members.

Database module

Efficiently manage bug data with our Default Detect System's intuitive database module. Organize, track, and analyze bugs seamlessly for streamlined issue resolution and improved software quality.

V. CONCLUSION

A default detect system helps to facilitate the quick localization and correction of any software program issues that may have been suggested for inclusion in a release. The Default Detect System (DDS) can be used to track problems that have been introduced into the different project modules during the testing and development phases of this project and to provide



guidance on how these issues can be corrected. Both of these stages are included in the project's scope. The scope of this plan will enable the elimination of any and all potential bottlenecks in the problem reporting process at all project levels and inside each and every project module within the context of software development.

REFERENCES

- [1] Torky Sultan, Mostafa Khedr, and Ayman E. Khedr ACTA Advise MED. 2013 June; 21(2): 103- 108 Sayed "A Proposed Imperfection Following Demonstrate for Classifying the Embedded Imperfection Reports to Improve Programme Quality Control"
 - [2] Yajie Wang, Ming Jiang, and Yueming Wei. The topic of "A Software Quality System for Mobile Application Testing" was discussed during the annual Global Conference on Propels in Framework Testing and Approval Lifecycle that took place in 2012 for the fourth time.
 - [3] Chetna Gupta, Priyanka Chandani. The Journal of Applied Psychology, Volume 101, Number 1 in 2014, Pages 32–41 In February of 2014, MECS Online published "A Study on Successful Imperfection Avoidance - 3T Approach."
 - [4] An Coordinates Quality Confirmation System for Indicating Trade Data Systems was published in 2009 by Gregor Engels, Stefan Sauer, and Frank Salger under the title Procedures of CAiSE Gathering.
 - [5] "An Coordinates Master Framework System for Computer programme Quality Assurance*",0730-3157/90/0000/0161\$01.OO 1990 IEEE, Stephen S. Yau, Yeou-Wei Wang, Jules G. Huang, and Jinshuan E. Lee. [5] "An Coordinates Master Framework System for Computer programme Quality Assurance*"
 - [6] V.B. Singh, Krishna Kuma, and Chaturvedi International Journal of Software Engineering and Its Applications Volume 5 Issue 4 October 2011 Citation:
 - [7] Dane Bertram, Amy Volda, Saul Greenberg, and Robert Walker The "Bug Following and Unwavering quality Evaluation System (BTRAS)" "Communication, Collaboration, and Bugs: The Social"Nature of Issue Following in Small, Collocated Groups" is one of the topics that will be discussed. The conference for this year will take place in Savannah, Georgia, from the 6th through the 10th of February. 2010 ACM, 978-1-60558-795-0/10/02. Copyright.
- An article titled "Plan of Bug Following Framework" was written by Gauri M. Puranik and published in the July 2014 issue of the Universal Journal of Creative Research in Science, Construction, and Innovation. This publication is an ISO 3297: 2007

BOOKS:

1. "Bug Tracking for Testers" by Paul Gerrard and Neil Thompson
2. "Bug Tracking Basics: Defect Tracking Log Book" by Journals For All
3. "The Definitive Guide to Bug Tracking" by Linus Upson and Andrew Whittaker
4. "Bug Tracking using Jira and Confluence" by Pramod Singh
5. "Bug Tracking, Testing, and Maintenance with TFS 2017" by Nanda L. Yadav