Software Requirements
Specification

For

Automated Bug Triage System

Done by:
M.Thanvi Varma | AP21110010093
Vinuthna Mallela | AP21110010108
P.Durga Mahesh | AP21110010911
Nandini Gogineni | AP21110010912

03-02-2024

Table of Contents

1.Introduction

1.1 Purpose

The purpose of an automated bug triage system is to efficiently categorize and prioritize incoming bug reports in software development. It can enhance accuracy in categorizing bugs, reduce manual effort, and provide insights into the overall health of a software project by tracking and managing reported issues systematically.

1.2 Document Conventions

1.2.1Bug ID Naming Convention

Create a standard naming method for identifying bugs. By using this convention, every defect will be recognized uniquely and traceable with ease throughout the bug triage process.

1.2.2 Bug Triage Status Codes

Create a set of status codes to track the bug triage process's progress. "New," "In Progress," "Verified," "Closed," and other codes may be among them. A common set of status codes improves communication and gives insight into the current condition of the bug.

1.2.3 Bug Severity and Priority

Clearly state the severity and priority levels that each bug has been assigned. This makes it easier to sort bug fixes according to their importance and urgency.

1.3 Intended audience and reading suggestions

1. Software developers
2. Testers
3. Project managers
4. Anyone involved in the bug resolution process

1.4 Product scope

The product scope of an automated bug triage system encompasses the functionalities related to efficiently processing, categorizing, and prioritizing bug reports in software development. This includes features such as automated analysis of incoming bug reports, assigning bugs to appropriate developers or teams, generating insights and reports on bug resolution progress, and ensuring a systematic approach to managing, reported issues.

1.5 References

- Academic databases
- Research papers
- Industry publications for more in-depth information on automated bug triage systems.

2.Overall Description

2.1 Product Perspective

The Automated Bug Triage System performs inside the larger processes of software development and bug fixing. It automatically prioritizes, classifies, and assigns reported defects to the right developers through interaction with social networks, development repositories, and bug-reporting systems. The system is an important component of the pipeline for fixing bugs, improving efficiency, and reducing the requirement for manual intervention.

2.2 Product Functions

The Automated Bug Triage System's primary functions are the automated extraction of important data from bug reports, classification using specified guidelines or machine learning algorithms, impact and severity-based bug prioritization, intelligent bug tasks to developers based on workload and expertise, and notification systems to notify the appropriate stakeholders about assigned bugs.

2.3 User Classes and Classification

Developers: Users who are responsible for fixing issues are known as developers. Interacting with the system allows them to obtain assigned bugs as well as appropriate data for effective bug fixing.

Bug Reporters: Users who submit the bug reports, Their reports are automatically processed, sorted, and assigned to developers, which is a benefit of the system.

System Administrators: Users are responsible for establishing and maintaining the Automated Bug Triage System.

2.4 Operating Environment

Utilizing social networks, version control repositories, and bug-tracking systems, the system functions within the software development environment. For automated triage activities to be carried out, it needs access to relevant developer profiles, source code repositories, and bug reports.

2.5 Design and Implementation Constraints:

Design Quality: The completeness and quality of bug report data decide how effective the system is.

Integration Compatibility: The seamless integration of the system with the current version control and bug-tracking systems is crucial to its working.

2.6 User Documentation:

User documentation will be provided, covering system interaction, administrator configuration, and bug reporting guidelines for developers, administrators, and bug reporters to ensure informative submissions.

2.7 Assumptions and Dependencies:

Assumption: The bug reports submitted are comprehensive and precise enough for automated processing.

Dependency: The system utilizes bug-tracking systems and version control repositories to access pertinent data.

3. External Interface Requirements

3.1 User Interfaces:

Web-based interface: Useful on a variety of devices and accessible through any normal web browser.

Clear navigation, simple controls, and quick access to pertinent information are all examples of user-friendly design.

Dashboards that can be customized: Give users the option to alter workflows and views to suit their preferences and roles.

Search and filtering: Use keywords, categories, priorities, and other criteria to effectively locate particular bugs.

Bulk actions: Take action on several bugs at once (assign, modify severity, etc.).

Analytics and reporting: Create graphs and reports on the performance of the team, activity in the triage process, and trends in bugs.

3.2 Hardware Interfaces:

Mobile app: An optional interface for managing bugs while on the go.

Combining wearable technology: Directly record bug reports from field-based devices (e.g., for hardware testing).

3.3 Software Interfaces:

REST API: Connect to additional reporting platforms, development tools, and bug-tracking systems.

Data import/export: For simple data interchange, support popular data formats including JSON, XML, and CSV.

Webhooks: Notify external systems in real time about bug events and upgrades. Secure access control for various user roles and permissions is achieved through authentication and authorization.

3.4 Communications Interfaces:

Requirements for secure communication: To safeguard data transmission, use HTTPS and other encryption techniques.

Notifications and changes in real-time: Push alerts for important updates and events.

Integration of email: Notify recipients via email when there are changes in status, comments, or bug assignments.

Multilingual support: By providing localized interfaces and notifications, you may appeal to a worldwide audience.

4.System Features

4.1 Use Cases

4.1.1 User Verification

Using a special username and password, users may log in. Secure session management guarantees that only authorized users may access the necessary functionality.

4.1.2 Reporting Bugs

Through an easy-to-use interface, users can enter problem descriptions and define bug categories. The system makes it easier to efficiently enter information about bugs.

4.1.3 Automated Classification of Bugs

A machine learning model that has been trained to categorize bugs according to their descriptions is included in the system. By automating the bug triage process, this feature increases efficiency and precision.

4.1.4 Bug Handling

Users may monitor and control the status of each reported problem on the system, which keeps a categorized list of them. Users can monitor if a problem is in the "New," "In Progress," or "Resolved" states.

4.1.5 Bug States Management

Different statuses can be ascribed to bugs: 'New' for newly reported bugs, 'In Progress' for defects that are being worked on, and 'Resolved' for bugs that have been successfully fixed.

5.Other Nonfunctional Requirements

5.1 Usability Standards

5.1.1 Intuitive User Interface

The Bug Triage System should have an easy-to-use and intuitive user interface. Without requiring in-depth training, users should be able to utilize the system with ease.

5.1.2 Accessibility Description

People with impairments should be able to utilize the system. The system must meet accessibility guidelines so that users of different skill levels may utilize it efficiently.

5.2 Requirements for Compliance

5.2.1 Data Protection

The system needs to abide by privacy laws and data protection rules. Sensitive and personal data must be handled in compliance with current data protection regulations.

5.2.2 Standards of Security

The Bug Triage System must follow security procedures that are accepted in the industry. To guarantee data integrity and user privacy, the system should have encryption, secure communication protocols, and frequent security audits.

5.3 Conditions for Compatibility

5.3.1 Browser Compatibility

The system's front-end ought to work with contemporary web browsers. The solution must offer a uniform user interface for widely used web browsers.

6. Other Requirements

Technical details and hosting infrastructure requirements for the Bug Triage System. Servers using the system need to meet certain hardware specifications, such as RAM and disc space. Additionally, it must properly connect with database management systems like MS SQL Server and Access.