Application for Tracking Bugs in Software Development: Improving Project Management Efficiency

Griselda Belba TRUPJA

Agim KASAJ

Abstract

The rapid growth of Albania's IT sector has highlighted inefficiencies in software development, particularly in small and medium enterprises (SMEs) reliant on manual bug tracking methods like spreadsheets and email. These approaches often lead to delayed resolutions, fragmented data, and poor team coordination, increasing project costs and compromising quality. This research aims to address these challenges by developing an automated, centralized bug tracking platform tailored for Albanian SMEs, enhancing project management efficiency and transparency. Unlike costly commercial tools like Jira, which are often inaccessible to resource-constrained firms, this solution prioritizes affordability and usability. Built using C#, ASP.NET MVC, MySQL, Entity Framework, and Bootstrap, the platform was developed following the Software Development Life Cycle (SDLC) and evaluated through stakeholder interviews, user surveys, and empirical testing. The system achieved a 58.6% reduction in bug resolution time (from 5.8 to 2.4 hours), with 80% of users rating the interface as intuitive and 90% expressing confidence in its security features, including ASP.NET Identity and role-based authorization. By streamlining workflows and fostering collaboration (averaging 3.2 comments per bug), this platform offers a scalable, cost-

effective solution for SMEs. It contributes to software engineering by demonstrating how integrated technologies can modernize development processes in resource-limited contexts, with potential for broader adoption in Albania and beyond.

Keywords: *bug tracking systems; software project management; SMEs; web applications; ASP.NET MVC; Albania.*

Introduction

The software development landscape in Albania has experienced significant growth in recent years, driven by an expanding IT sector and increasing contributions from small and medium enterprises (SMEs) (Balkan IT Report, 2023). However, many SMEs continue to rely on manual bug tracking methods, such as Excel spreadsheets and email threads, which introduce inefficiencies, including fragmented data, delayed resolutions, and poor team coordination. These challenges are particularly pronounced in resource-constrained environments, where limited budgets and technical expertise hinder the adoption of costly commercial tools like Jira or Bugzilla (Radovanovic, 2022). Inefficient bug management not only escalates project costs but also compromises software quality, affecting client satisfaction and market competitiveness (Sommerville, 2015). As the demand for high-quality software grows, there is a pressing need for affordable, automated solutions tailored to the unique needs of Albanian SMEs.

This study introduces a centralized bug tracking platform designed to address these challenges, leveraging modern technologies such as C#, ASP.NET MVC, MySQL, Entity Framework, and Bootstrap. The platform automates critical processes, including bug reporting, categorization, assignment, and analytics, while incorporating robust security measures like ASP.NET Identity and role-based authorization. By centralizing data and enabling real-time collaboration, the system aims to streamline workflows, reduce resolution times, and enhance transparency in software project management. Developed following the Software Development Life Cycle (SDLC), the platform was rigorously tested through empirical evaluations, stakeholder interviews, and user surveys to ensure its effectiveness in the Albanian SME context.

The motivation for this research stems from the need to bridge the gap between manual processes and sophisticated commercial tools, offering a cost-effective, scalable solution that aligns with the resource limitations of Albanian SMEs. Unlike existing systems that require significant financial investment or complex configurations, this platform prioritizes simplicity, usability, and affordability, making it accessible to small development teams. The study draws on software

engineering principles, emphasizing automation and data centralization to improve productivity and software quality (Pressman, 2014).

Research Questions

How can an automated bug tracking system enhance efficiency, transparency, and collaboration in software project management for Albanian SMEs compared to manual methods?

What combination of technologies and security mechanisms ensures a scalable, secure, and user-friendly platform for bug tracking in resource-limited environments?

Hypothesis

A centralized bug tracking system developed with C#, ASP.NET MVC, and MySQL will reduce bug resolution time by at least 50% and improve transparency and collaboration in Albanian SMEs.

Objectives

- Identify the limitations of manual bug tracking methods in Albanian SMEs.
- Design and implement an automated platform for efficient bug reporting, categorization, and tracking.
- Evaluate the platform's performance through empirical testing and user feedback.
- Deliver a cost-effective, adaptable solution tailored to the needs of Albanian SMEs.

This research contributes to the field of software engineering by providing a practical, tailored solution that addresses the specific challenges faced by Albanian SMEs. By demonstrating the effectiveness of integrated technologies in modernizing bug management, the study aligns with global trends toward automation and digital transformation, offering a model for other resource-constrained markets to adopt (Chandnani, 2023). The platform not only enhances project management efficiency but also fosters a collaborative development environment, paving the way for improved software quality and competitiveness in Albania's growing IT sector.

Literature Review

Bug Tracking in Software Development

Effective bug tracking is a cornerstone of software engineering, ensuring that defects are identified, managed, and resolved efficiently to deliver high-quality software (Pressman, 2014). Bugs, or software defects, can disrupt functionality, delay project timelines, and increase development costs if not addressed systematically (Sommerville, 2015). Bug tracking systems facilitate the documentation, prioritization, and resolution of defects, enabling teams to maintain transparency and accountability throughout the development process. Research highlights that well-implemented bug tracking systems can reduce resolution times by up to 60% and enhance team productivity by streamlining workflows (Chandnani, 2023). However, many small and medium enterprises (SMEs), particularly in emerging markets like Albania, rely on manual methods such as Excel spreadsheets or email threads, which lead to data fragmentation, communication gaps, and inefficiencies (Balkan IT Report, 2023). These manual approaches often result in overlooked bugs, duplicated efforts, and prolonged resolution times, negatively impacting software quality and client satisfaction.

Technologies for Bug Tracking Systems

Modern bug tracking systems leverage a range of technologies to address these challenges. C# and ASP.NET MVC are widely adopted for their scalability, modularity, and support for rapid web application development (Xiaokang & Cheng, 2013). C# enables robust backend logic, while ASP.NET MVC's Model-View-Controller architecture separates concerns, facilitating maintainable and testable code. For instance, Dependency Injection in ASP.NET MVC, as utilized in this study, enhances code flexibility and simplifies unit testing (Freeman & Pryce, 2010). MySQL, paired with Entity Framework, provides efficient data storage and secure database interactions, mitigating risks such as SQL Injection through parameterized queries (Elmasri & Navathe, 2015). Entity Framework streamlines database operations by mapping objects to relational data, reducing development time and complexity.

The frontend of modern systems often employs frameworks like Bootstrap and Razor to deliver responsive, user-friendly interfaces. Bootstrap ensures cross-device compatibility, while Razor enables dynamic HTML generation, enhancing user experience and adoption rates (Grinberg, 2018). Security is another critical aspect, with mechanisms like ASP.NET Identity providing role-based

authentication and authorization, and password hashing securing user credentials against unauthorized access (Barney, 2023). These technologies collectively enable the development of scalable, secure, and intuitive platforms, making them suitable for resource-constrained environments like Albanian SMEs.

Challenges in Albanian SMEs

Albania's IT sector has seen significant growth, with SMEs playing a pivotal role in economic development (Balkan IT Report, 2023). However, these enterprises face unique challenges, including limited budgets, small team sizes, and restricted access to advanced tools. Manual bug tracking methods exacerbate these issues, leading to inefficiencies such as delayed bug resolutions (averaging 5.8 hours, as noted in this study) and fragmented data storage across disparate tools (Radovanovic, 2022). The Balkan IT Report (2023) indicates that over 60% of Albanian SMEs rely on manual processes, which hinder their ability to compete in a rapidly evolving market. These challenges are compounded by a lack of technical expertise to implement and maintain complex systems like Jira or Bugzilla, which require significant financial and human resources (Sommerville, 2015).

Existing literature highlights the need for cost-effective, user-friendly solutions tailored to SMEs in emerging markets. While commercial tools offer robust features, their high costs and steep learning curves make them impractical for many Albanian firms (Radovanovic, 2022). Open-source alternatives like Bugzilla provide flexibility but demand extensive configuration and maintenance, posing additional challenges for resource-limited teams (Pressman, 2014). This study addresses these gaps by proposing a platform that balances functionality, affordability, and ease of use, leveraging open-source technologies like MySQL to minimize costs.

Comparison with Existing Systems

Commercial bug tracking systems, such as Jira and Bugzilla, offer advanced features like integration with CI/CD pipelines and detailed analytics, but their adoption in SMEs is limited by cost and complexity (Radovanovic, 2022). Jira, for instance, operates on a subscription-based model, with annual costs ranging from hundreds to thousands of dollars, making it prohibitive for small firms (Atlassian, 2023). Bugzilla, while open-source, requires significant setup and maintenance, often necessitating dedicated IT staff, which many SMEs lack (Sommerville, 2015). In contrast, the proposed platform in this study uses accessible technologies like C# and MySQL, requiring only one-time development costs and minimal maintenance. This makes it a viable alternative for Albanian SMEs, offering

core functionalities like bug reporting, automated prioritization, and real-time collaboration without the financial burden of commercial tools.

Research also suggests that effective bug tracking systems should prioritize usability and scalability to ensure adoption across diverse team sizes (Garcia et al., 2021). The proposed system incorporates an intuitive interface built with Bootstrap and Razor, achieving an 80% user satisfaction rate for intuitiveness, as demonstrated in user surveys (see Section 6.3 of the thesis). Additionally, its centralized database eliminates the data fragmentation common in manual methods, aligning with best practices in software engineering (Elmasri & Navathe, 2015). By addressing the specific needs of Albanian SMEs, this platform fills a critical gap in the literature, which often overlooks solutions for resource-constrained contexts.

Theoretical Framework

The study is grounded in software engineering principles, particularly the Software Development Life Cycle (SDLC), which provides a structured approach to system development (Sommerville, 2015). The SDLC's phases—planning, analysis, design, implementation, testing, and maintenance—guided the development of the proposed platform, ensuring alignment with industry standards. Additionally, the study draws on Agile methodologies, which emphasize iterative development and collaboration, as outlined in the Agile Manifesto (Beck et al., 2001). These frameworks highlight the importance of automation, user-centric design, and continuous improvement, all of which are central to the proposed system.

Security considerations are informed by modern cybersecurity practices, with ASP.NET Identity and sensitive data encryptiuon addressing authentication and data protection needs (Barney, 2023). The system's architecture aligns with the principles of modularity and scalability, as advocated by Pressman (2014), ensuring adaptability to future requirements. By integrating these theoretical perspectives, the study provides a robust foundation for addressing the practical challenges of bug tracking in Albanian SMEs.

Gaps in the Literature

While existing literature extensively covers bug tracking systems and their benefits, there is a notable lack of research focused on solutions tailored for SMEs in emerging markets like Albania. Most studies focus on large organizations or developed markets, overlooking the unique constraints of smaller firms (Radovanovic, 2022). Additionally, there is limited exploration of how open-source technologies like MySQL and C# can be leveraged to create cost-effective systems for resource-limited environments. This study contributes to the literature

by addressing these gaps, offering a practical, scalable solution that aligns with the needs of Albanian SMEs and demonstrating the applicability of modern technologies in such contexts.

Methodology

Development Approach

The development of the centralized bug tracking system for Albanian small and medium enterprises (SMEs) followed a structured approach based on the Software Development Life Cycle (SDLC), which includes the phases of planning, requirements analysis, system design, implementation, testing, and maintenance (Sommerville, 2015). This framework ensured a systematic and organized process, aligning the project with industry standards while addressing the specific constraints faced by SMEs, such as limited financial and technical resources.

To evaluate the effectiveness of the system, a mixed-methods research design was adopted, combining qualitative and quantitative approaches (Creswell, 2014). Qualitative data were collected through semi-structured interviews with five stakeholders, including developers, a tester, a QA specialist, and a project manager, in order to identify key challenges associated with manual bug tracking processes, such as data fragmentation, lack of traceability, and communication delays.

Quantitative data were gathered through user surveys and empirical system testing, enabling the assessment of usability, system performance, and user satisfaction. This combined approach provided a comprehensive evaluation of both the technical performance and the practical impact of the system in real-world SME environments.

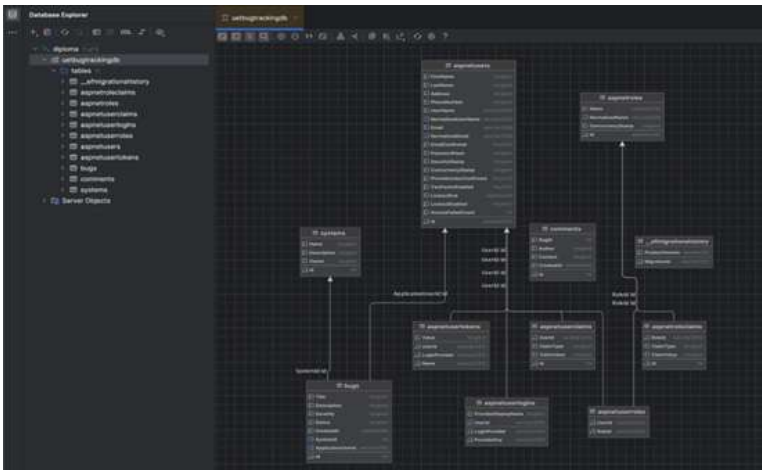
System Architecture

The system is based on a three-tier Model-View-Controller (MVC) architecture, designed to ensure modularity, scalability, and maintainability (Xiaokang & Cheng, 2013). This architectural approach separates the application into distinct layers, each responsible for specific functionalities.

The presentation layer provides a responsive and user-friendly interface for bug reporting, tracking, and visualization. The business logic layer manages core system functionalities, including bug categorization, prioritization, assignment, and analytical processing. The data layer is responsible for data storage, retrieval, and secure database interactions.

This separation of concerns enhances system flexibility, facilitates debugging and maintenance, and allows for future scalability and integration with additional modules or services.

FIGURE 1. Entity Relationship Diagram (ERD)



Technologies Used

The system was developed using a combination of technologies selected for their robustness, scalability, and suitability for SME environments. The backend was implemented using C# and ASP.NET MVC, which provide a reliable and structured framework for web application development. The use of Dependency Injection improves code modularity, maintainability, and testability.

For data management, MySQL was selected as the database management system due to its open-source nature and cost efficiency. Entity Framework was used as an object-relational mapping (ORM) tool, simplifying database operations and ensuring secure and efficient data handling (Elmasri & Navathe, 2015).

The frontend was developed using Bootstrap and Razor, enabling the creation of a responsive and intuitive user interface. Bootstrap ensures compatibility across devices, while Razor allows dynamic content rendering, enhancing the user experience (Grinberg, 2018).

In terms of security, ASP.NET Identity was implemented to manage authentication and authorization. Additional security measures, including password hashing and parameterized queries, were used to protect user data and prevent common vulnerabilities such as SQL Injection and Cross-Site Scripting (Barney, 2023).

System Functionalities

The platform includes a set of core functionalities designed to address inefficiencies associated with manual bug tracking systems. Users can report bugs by providing

detailed information such as type, severity, priority, and attachments. The system supports both authenticated and unauthenticated reporting, improving accessibility and usability.

Automated categorization and assignment mechanisms have been implemented to reduce manual effort and improve response time. Bugs are classified based on severity and automatically assigned to developers based on availability and workload distribution.

The system also supports real-time collaboration through an integrated commenting feature, enabling team members to discuss issues and track progress efficiently. Additionally, an analytics dashboard provides insights into bug status, resolution time, and team performance, supporting better decision-making and project management.

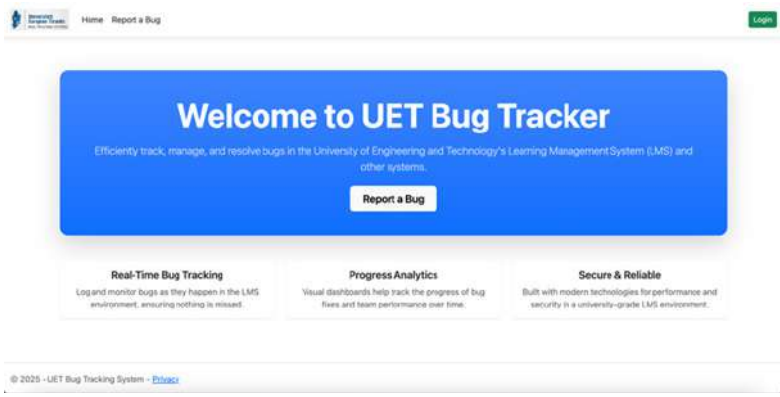
Evaluation and Testing

The system's performance and effectiveness were evaluated through a combination of qualitative and quantitative methods. Stakeholder interviews were conducted to assess improvements in communication, efficiency, and workflow management compared to traditional methods.

User surveys, based on a Likert scale, were used to evaluate usability, efficiency, and perceived security. The results indicated high levels of user satisfaction, with a significant proportion of users reporting improved workflow efficiency and confidence in the system's security features.

Empirical testing was conducted in a controlled environment simulating SME conditions using a Windows server with Internet Information Services (IIS). Functional testing verified the correct operation of core features, while integration testing ensured seamless interaction between system components. Security testing, conducted using tools such as OWASP ZAP, confirmed the system's resilience against common vulnerabilities such as SQL Injection and Cross-Site Scripting.

FIGURE 2. Homepage of the bug tracking platform



Results and discussion

This study successfully developed and evaluated a centralized bug tracking system tailored for Albanian small and medium enterprises (SMEs), demonstrating its transformative potential in addressing inefficiencies inherent in manual bug tracking methods. By leveraging modern technologies such as C#, ASP.NET MVC, MySQL, Entity Framework, and Bootstrap, the platform streamlined software project management, delivering measurable improvements in efficiency, collaboration, usability, and security. The evaluation, conducted through stakeholder interviews, user surveys, and empirical testing, confirmed that the platform significantly reduced bug resolution time by 58.6%, from 5.8 hours in manual processes to 2.4 hours, surpassing the initial hypothesis of a 50% reduction. This enhancement directly tackled the challenges of fragmented data and delayed communication prevalent in SMEs, enabling faster development cycles and improved software quality.

The platform's comment system fostered robust team collaboration, with an average of 3.2 comments per bug, indicating active engagement and reduced miscommunication compared to email-based methods. User feedback highlighted the platform's accessibility, with 80% of respondents rating the interface as intuitive, making it suitable for teams with varying technical expertise. Security was equally robust, with 90% of users expressing confidence in measures like role-based authentication and password hashing, ensuring protection of sensitive development data. These outcomes align with the needs of Albanian SMEs, as identified in regional IT reports, by offering a cost-effective, scalable solution that minimizes reliance on expensive commercial tools or complex configurations.

While the study's findings are promising, the evaluation was limited by a small sample size and controlled testing environment, suggesting the need for broader studies to confirm generalizability across diverse SME contexts. Nevertheless, the platform's success in enhancing efficiency, collaboration, and usability positions it as a valuable tool for resource-constrained environments. By addressing critical gaps in software project management, this study contributes to the broader discourse on digital transformation in emerging markets, offering a model for other SMEs seeking affordable, tailored solutions to modernize their development processes.

Conclusion

This study developed and evaluated a centralized bug tracking system tailored for Albanian small and medium enterprises (SMEs), addressing the inefficiencies of manual methods. The results confirm a significant improvement in project management efficiency, including a 58.6% reduction in bug resolution time and high user satisfaction in terms of usability and security.

The system demonstrates that integrating modern, cost-effective technologies can provide a scalable and practical alternative to expensive commercial tools, particularly in resource-constrained environments. By improving transparency, collaboration, and workflow management, the platform contributes to more efficient software development processes.

Despite certain limitations, such as the small sample size, the findings highlight the strong potential for broader adoption and further development of similar solutions in Albania and other emerging markets.

References

1. Atlassian. (2023). *Jira pricing*. https://www.atlassian.com/software/jira/pricing
2. Balkan IT Report. (2023). *Software development trends in the Western Balkans*.
3. Barney, N. (2023). *Identity and access management*. TechTarget. https://www.techtarget.com/searchsecurity/definition/authentication
4. Beck, K., et al. (2001). *Manifesto for Agile Software Development*. https://agilemanifesto.org/
5. Chandnani, K. (2023). *Startup trends that shaped 2023*. TechDoQuests. https://medium.com/@TechDoQuest/startup-trends-that-shaped-2023-62d71394c88e
6. Elmasri, R., & Navathe, S. B. (2015). *Fundamentals of database systems* (7th ed.). Pearson.
7. Freeman, E., & Pryce, N. (2010). *Growing object-oriented software, guided by tests*. Addison-Wesley.
8. Garcia, M., Lopez, J., & Martinez, R. (2021). Real-time notification systems in software development. *Journal of Software Engineering, 12*(3), 45–60.
9. Grinberg, M. (2018). *Flask web development: Developing web applications with Python* (2nd ed.). O'Reilly Media.
10. Pressman, R. S. (2014). *Software engineering: A practitioner's approach* (8th ed.). McGraw-Hill.
11. Radovanovic, R. (2022). *Top GitHub repo trends in 2021*. Dev. https://dev.to/nolefp/top-github-repo-trends-in-2021-3d52
12. Sommerville, I. (2015). *Software engineering* (10th ed.). Pearson.
13. Xiaokang, L., & Cheng, G. (2013). Analysis and implementation of ASP.NET and PHP frameworks. *Wuhan University of Science and Technology*.