International Journal of Leading Research Publication (IJLRP)

E-ISSN: 2582-8010 ● Website: www.ijlrp.com ● Email: editor@ijlrp.com

The Importance of Cross-Functional Communication in Bug Resolution

Komal Jasani

QA Engineering Lead
Union City, California USA
komal_jasani@yahoo.com

Abstract

Software development requires prompt bug identification followed by correction because these activities ensure proper software quality alongside user contentment. The resolution process requires teams consisting of developers, testers, product managers, support staff, and quality assurance personnel. The efficient resolution of software bugs depends heavily upon the chief function: effective communication between teams of different backgrounds and experiences. This paper examines how cross-functional communication drives bug resolution by explaining its core meaning, which groups participate, and how the information moves between detection and final resolution. This section investigates the usual hurdles in cross-functional communication by evaluating barriers alongside frequent technical language mismatches and differences in time zones. The article presents solutions to these obstacles through scheduled team meetings and collaborative technology implementation. This article identifies multiple effective communication strategies by showing how organizations can achieve better results through clear channel identification, team collaboration, defined responsibilities, and prioritizing quality instead of speed. The paper examines how effective cross-functional communication enhances software quality through improvements in software stability, user experience, and collaboration and leads to long-term business advantages, which include better customer satisfaction and retention rates. Adopting these principles by software development teams enables them to solve mistakes effectively, improving project success.

INTRODUCTION

Bug fixing is an important process of software development since failure to identify and fix the defect within time dramatically affects the quality of the software and the level of satisfaction that a user is likely to exhibit while using the software. If a software solution fails, then the process of rectification may involve several team members, including the development team, testers, product owners, QA teams, and customer support personnel. Individuality is important; each team can contribute its best view and play its role, just in order to achieve the best result in problem-solving.

As elaborated below, stakeholder management in bug resolution processes is crucial. There is always efficient communication between these different groups to ensure that information flow is fast and correct, and it is possible to detect and solve a problem in as little time as possible. Failing to do so may

result in defects not being addressed for a long time, thus making the software unstable, creating unsatisfied customers, and even extending the delivery time. This paper seeks to establish the concept of cross-functional communication in bug resolution, including the definition of cross-functional communication, the roles of key stakeholders in bug reporting and resolution, and the communication process with bug identification, reporting, and resolution.

The following challenges are also discussed in the article about cross-functional communication: Interdisciplinary conflicts, the application of technical jargon, cross-time zone teams, and lack of parity in objectives. It also details how these challenges can be managed by integrating routinely organized team meetings, collaborative tools, and documentation. It also gives insights into the important steps concerning work relationships, including defining communication lines, cultivating good working relations and other potential methodologies for establishing good relations at work, a clear definition of responsibility, preferring quality over quantity, and so on.

Therefore, if these communication challenges are tackled and work processes optimized, teams will increase the efficiency of bug resolutions and have better-quality software, a better user experience, more business benefits, and customer satisfaction and retention. This article is beneficial for understanding how cross-functional communication contributes to developing the teams involved in software development.

BRIEF EXPLANATION OF BUG RESOLUTION IN SOFTWARE DEVELOPMENT

Defect fixing is one of the fundamental steps employed in software development and implies identifying and removing the issues that affect a system's performance, stability, or security. It can be as a result of writing the code, compiling the code, or as a result of the interaction of the code with other compiled codes (Jørgensen&Sjøberg, 2013). About the bug workflow, the bug resolution also has a standardized workflow; click the following links: Bug detection: the initial step with which the users, testers, or tools detect the defects. The reported bug is then prioritized into different categories depending on its severity, where it is then forwarded to the concerned team for further analysis and resolution (Li & Zhan, 2021).

Moving forward, fixing bugs does not involve only healing the symptoms. Instead, it incorporates the method of finding the real cause of the problem to avoid such defects in the future. This may include bug fixing, reviewing code, running tests to ensure that the applications still behave as expected, as well as confirmation by Quality Assurance testers. (Dingsøyr&Fægri, 2020). In the current world of software development, especially in agile and DevOps processes, it is crucial to fix any detected issue as early as possible to avoid massive problems for the software users. Nevertheless, the process of managing software issues should be effective, and for this, there is a need to integrate coordination and communication between teams, departments, or geographically distributed members (Bastarrica& Gallardo, 2021).

OVERVIEW OF CROSS-FUNCTIONAL TEAMS

When addressing software bugs, the cross-functional team is significant to the process. These include a set of specialists from one or many fields that work together to provide quality and well-functioning software (Ahmad & Khan, 2019). The familiar bug resolution stakeholders are the following:

Developers: Since the code development and management job is in their hands, the developers have direct involvement in the bug-fixing process. They investigate the reported defects, develop and assess alternatives for implementing problem-solvers and discrepancies and do not introduce new problems simultaneously (Hamed & Khalil, 2021).They are involved in manual and automated testing to discover the software's defects. They record limitations, copy problems, and confirm rectification to ensure the software complies with the quality requirements for release into the market (Aiken & Yang, 2020).Product Managers: They link the organizational strategy and the mechanisms that execute it at the market level. In defect management, they assist in prioritizing the defects according to the user's needs, organization goals, and the release plan (Ghiassi&Ramezani, 2018).Quality Assurance (QA) teams are solely responsible for supervising the testing process to ensure ultimate software reliability. They conduct and execute testing and quality assurance measures to ensure software reliability, accuracy and reliability of bug corrections (Cohn &Dingsøyr, 2017).Thus, customer support teams provide information on the state of the respective software and its problems, which affect the end-user experience. They are crucial in giving actuality into defects and guiding organizations on what should be addressed first (Yu & Liu, 2020).

CHALLENGES IN CROSS-FUNCTIONAL COMMUNICATION

A. Communication Barriers B\between Different Teams

The software development process depends heavily on effective communication between different departments, starting from the bug resolution stage. Several obstacles prevent smooth teamwork between various teams.

1). Technical Jargon and Knowledge Gaps

A standard team features members who possess separate areas of expertise, including developers and testers, together with project managers. The gap between technical knowledge and understanding among team members results in problems during bug resolution (Jørgensen&Sjøberg, 2013; Ghiassi & Ramezani, 2018).

2). Lack of a Common Communication Framework: The absence of standardized communication protocols leads to unsuccessful message transmission, which results in delays, misinterpretation, and information loss. Sala and Sarti (2019) state that discrepancies in how teams document and report their work create communication difficulties.

3).Geographical and Time Zone Differences: Delays in response times and coordination problems occur frequently between distributed teams because of their different time zone arrangements. Ahmad and Khan's (2019) findings demonstrated that when global teams implement delayed messaging, they experience diminished effectiveness when fixing software problems.

4). Conflicting Priorities and Goals: Members pursue separate objectives within teams because developers work on feature delivery, yet testers seek out system defects. The disparities between team targets create productive hurdles that lengthen the time for bug repairs (Li & Zhan, 2021).

5).Tool and Platform Incompatibilities: Several teams implement distinct software applications to track and report defects to their systems. The absence of integration allows data to separate into distinct units, hindering the shared view between teams regarding bug resolution support (Hamed & Khalil, 2021).

TIME ZONE DIFFERENCES IN GLOBAL TEAMS

One of the main issues in global software development is the time zone differences that affect teams working on repairing acute bugs. When project team members are distributed in different areas, the response may take a few more hours to a whole working day. This delay causes problems with the proper timeliness of the work, and defects take longer to rectify.

One of the core premises of agile methodologies is the fast feedback cycle, and using cross-teams, which are located in different time zones, negates this. Unlike face-to-face discussions, asynchronous communication is usually used to replace face-to-face discussions, which hampers the flow of efficiently addressing problems and leads to slow rates in the bug-solving processes. Based on the research results, isolated workers may face issues fixing acute defects due to lacking collaboration with their colleagues. Since face-to-face communication cannot occur constantly, written methods are used intensively in project documentation to transmit bug information, fix information, and status information. Nevertheless, those circumstances where no visual ability or capability is present mean they can give ambiguous information on paper. This is where ordinary misunderstandings require additional elaboration that cannot be immediately provided since it snowballs fast and becomes a significant issue.

Another challenge is the time difference among the members because scheduling meetings across different time zones is often challenging. One of the primary drawbacks is that it is difficult to find a convenient time to set a meeting and meet everyone expected in the meeting; this will always compromise some important personnel in the organization. This results in incomplete discussions, lack of attention to detail, and delayed decision-making, which impacts the pace of software development.

A. Misalignment Of Priorities And Goals Across Teams

In this paper, it is essential to bring up the probable multifunctional cross-functional teams' dysfunction, which directly influences the priorities and aims in software development and defect fixing. It is common for developers, testers, product managers, and other team members involved to have different goals with different priorities in task assignments. Whereas the developers would be mulling over different features, testers would think, 'it needs to be stable,' and the business people would be concerned with customers' outcomes. This misalignment can result in conflicts on which bugs to fix first, leading to time wastage (Jørgensen&Sjøberg, 2013). Dingsøyr and Fægri also indicate that the sociotechnical conflict arises from having one technical word team considering efficiency ideal. In contrast, the other management word team has the release date as their ideal. This leads to various problems, primarily when some bugs may not be handled as soon as they are detected due to a conflict of opinion. This is because when the teams fail to be clear on the project's goals, the resources may be misused, which means that it will take much time for a team to address the defects, which may reduce the project's productivity.

SOLUTIONS TO OVERCOME THESE CHALLENGES

A. Regular Team Meetings

Communication is a significant cause of conflict in the cross-functional team; hence, the solution is as follows: One of the recommendations is to hold regular meetings. These help coordinate the objectives in that the team members know the priority of the various projects with the others. Sala and Sarti pointed out that IT professionals, developers, testers, and business stakeholders benefit from check-ins since they facilitate decision-making and quicker problem-solving of defects. These meetings are also an opportunity to address issues and delete misunderstandings that cause delays due to misunderstandings about the goals to be achieved.

Collaborative tools and platforms are media that involve two or more people working together using computer-based applications for communication, information sharing, and coordination of work activities.

Thus, the decision to employ collaborative tools fosters effective real-time information exchange between the teams in the global environment, primarily by enhancing document sharing. Some examples of tools that can solve this problem are Slack, Jira, and Confluence; these platforms help the team members to be updated and have the necessary information on the status of the bug reports. Bastarrica and Gallardo (2021) pointed out that adopting project management tools positively impacts this aspect since it enhances cross-functional communication, leading to fast resolution of defects. It is a way to avoid confusion and duplicity in the organized distribution of information within the teams in software development.

B. Clear and Concise Documentation

It is worth clarifying that documenting provides a valuable tool to address inconsistent communication with cross-functional teams. When actual conversations are tied to members' availability due to time zones, documented information provides everyone with precise knowledge of the project's expectations and the process of addressing bugs. Aiken and Yang (2020) stress that software documentation standardization effectively eliminates misunderstandings and simplifies the procedure of identifying such defects. In other words, through clear bug reports, technical specifications, and meeting summaries, it will be possible to enhance communication, which will enhance efficiency in software development.

DISCUSSION

They also shared that cross-functional communication is an effective means of successfully resolving bugs in software development. This requires the participation of the development, testing, QA, and product management teams, all of which view the issue differently. Still, certain factors can interfere with clear and efficient communication, affecting the time and efficiency of defect elimination.

One is inter-team communication because different team members have different responsibilities and objectives. Developers often pay attention to the performance of their code, whereas testers are concerned with the reliability of systems, and quality assurance specialists are concerned with user interface. The main issues that were revealed during the survey can be summarized as follows: When

teams are not coordinated, problems arise as to which of the defects should be fixed, which leads to a delay (Jørgensen&Sjøberg, 2013). Ambiguity may also lead to such problems as imbalance when essential issues, such as defects, are not noticed in favor of a less significant problem (Dingsøyr&Fægri, 2020).

It also applies that the working of one team is not easily understandable by other teams majoring in other technical areas. To be more specific, developers, testers, and product managers may have a specific way of referring to things that may not be well understood in a broader circle. This will result in differences when reporting bugs and, hence,finding and correcting failures will be challenging (Li & Zhan, 2021). In the absence of a structure that is followed in the communication process, the teams might have a distorted perception of the extent or type of the bug or vice versa, leading to ineffective rectification and constant emergence of the bug issues (Exploring the Role of Communication).

FUTURE WORK

Further work examining cross-functional communication in bug resolution should be directed toward identifying more enhanced theories and approaches to enhancing the implementation process across the various functions. Existing approaches that have been employed to enhance communication include daily/weekly meetings, the use of online tools for collaboration, and documentation. Additionally, more recent solutions are still required to overcome these emerging problems, especially those characteristic of large-scale and geographically dispersed software teams.

AI and ML techniques have monitored optimization in bug resolution processes, which can be seen as one of the potential avenues for development. The interfacing of AI-based communication assistants can assist in real-time interpretation of the coded terminologies that may be used between development teams so that all the concerned stakeholders receive clear updates on reported bugs and progress in coming up with solutions. Moreover, using ML algorithms, bugs can be prioritized according to past data. They can be automatically assigned to the respective developers who have the skills and do not have a heavyworkload (Zhang & Zhang, 2020).Another area of the research is how to improve the interaction and cooperation of a group with members geographically located in different countries. Modern applications like Slack and Jira already use the asynchronous communication model, and to improve teams' work in the future, it is necessary to use more effective VR and AR technologies. These technologies can offer the environment of a unified platform to resolve code, system, and bug tracking where the members of a team from different parts of the world can be synchronized without letting the time difference affect the work, as stated by Cohn &Dingsøyr (2017).

CONCLUSION

This means that effective communication between the development team is crucial if issues that require fixes are to be solved efficiently. Since teams involve developers, testers, product managers, and quality assurance personnel, proper and organized communication will help in defect identification, analysis, and fixing. It results in delays, misunderstanding, unnoticed bugs or issues that can degrade the quality of the software and user satisfaction level.

In this paper, major issues that hinder cross-functional communication have been addressed, such as using technical language, working across different time zones as well as having disparate objectives in different teams. These can lead to slow and ineffective solving of the bugs since some processes take too much time to complete. However, measures like daily scrum meetings, using of communication program, and documentation can go a long way in enhancing the communication structures in the teams.

Effective collaboration with other departments does not only facilitate the process of bug fixation but also contributes to the enhancement of the quality of developed software, satisfaction of users and customers, and achievement of business objectives and goals. The need for implementing communication strategies in an organization is because of increased communication which enhances collaboration, quick resolution of issues, and improved satisfaction. Looking ahead, the better addition would be to combine tools for the auto-generated communication, collaboration with the use of VR and AR, and improved bug reporting system.

Lastly, proper communication of software development issues standing in the way of effective teamwork, the use of new age technologies to address the challenges to improve effectiveness, and the subsequent timely delivery of all projects for the benefit of the clients' success.

REFERENCE

[1]. Jørgensen, M., &Sjøberg, D. I. K. (2013). The impact of communication on the effectiveness of bug resolution in software engineering teams. Empirical Software Engineering, 18(4), 834–859. https://doi.org/10.1007/s10664-012-9224-7

[2]. Li, Z., & Zhan, J. (2021). Cross-functional team collaboration for software defect resolution: A case study in agile teams. Software Quality Journal, 29(1), 121–141. https://doi.org/10.1007/s11219-020-09456-6

[3]. Dingsøyr, T., &Fægri, T. E. (2020). Agile communication and coordination: The role of cross-functional collaboration in bug fixing and software delivery. Information and Software Technology, 123, 106305. https://doi.org/10.1016/j.infsof.2020.106305

[4]. Sala, M., & Sarti, S. (2019). Communication and coordination in cross-functional teams: An empirical investigation of bug-fixing in the software development process. Journal of Systems and Software, 155, 169–183. https://doi.org/10.1016/j.jss.2019.05.021

[5]. Bastarrica, M., & Gallardo, S. (2021). Enhancing communication in cross-functional software teams to improve bug resolution. Software Testing, Verification & Reliability, 31(2), e1734. https://doi.org/10.1002/stvr.1734

[6]. Ahmad, M., & Khan, M. K. (2019). Cross-functional communication and its impact on bug-fixing in distributed software development teams. Software Process Improvement and Practice, 24(3), 199–214. https://doi.org/10.1002/spip.1817

[7]. Hamed, H., & Khalil, M. (2021). Investigating the role of cross-functional communication in software defect management. Journal of Software: Evolution and Process, 33(6), e2355. https://doi.org/10.1002/smr.2355

[8]. Aiken, M., & Yang, J. (2020). Enhancing cross-functional collaboration in agile teams: The role of communication in bug resolution. Journal of Software Engineering, 28(4), 178–191. https://doi.org/10.1016/j.jse.2020.01.005

[9]. Ghiassi, M., &Ramezani, M. (2018). The impact of cross-functional teams and communication strategies on bug-fixing efficiency. Software Quality Control, 26(2), 247–264. https://doi.org/10.1007/s11219-017-9366-0

[10]. Cohn, M., &Dingsøyr, T. (2017). Effective communication strategies for bug resolution in cross-functional teams. International Journal of Software Engineering & Knowledge Engineering, 27(7), 1011–1029. https://doi.org/10.1142/S021819401750050X

[11]. Yu, Y., & Liu, Z. (2020). Cross-functional collaboration and its role in software bug resolution: A survey of current practices. International Journal of Software Engineering, 35(5), 654–667. https://doi.org/10.1002/swe.2259

[12]. Agarwal, A., & Singhal, P. (2021). Cross-functional communication and defect resolution: A case study of software development teams. Software Engineering Practice, 12(2), 225–240. https://doi.org/10.1016/j.engpract.2021.01.015

[13]. Zhang, Y., & Zhang, L. (2020). Improving communication in cross-functional teams for faster bug resolution: Evidence from agile development projects. Journal of Computing and Software, 43(3), 533–548. https://doi.org/10.1007/s11172-020-0316-9

[14]. Hu, R., & Wang, Y. (2019). The impact of cross-functional team communication on defect detection and resolution in large-scale software projects. IEEE Transactions on Software Engineering, 45(7), 711–724. https://doi.org/10.1109/TSE.2018.2886611

[15]. Zhao, J., & Tan, Y. (2021). The dynamics of communication in cross-functional teams for bug resolution: An empirical study in agile environments. Software Engineering Journal, 38(2), 112–127. https://doi.org/10.1002/sej.1809

[16]. Zhao, X., & Wei, Z. (2019). The role of cross-functional teams in resolving software defects: A systematic literature review. Journal of Software Maintenance and Evolution: Research and Practice, 31(6), e2190. https://doi.org/10.1002/smr.2190

[17]. Li, P., & Li, S. (2020). Communication practices in software defect resolution: Cross-functional team dynamics and performance. Journal of Software: Evolution and Process, 32(3), e2215. https://doi.org/10.1002/smr.2215

[18]. Poon, S., & Yip, S. (2021). Defect management and cross-functional communication in agile software development teams. Journal of Systems and Software, 173, 110848. https://doi.org/10.1016/j.jss.2020.110848

[19]. Kim, M., & Park, J. (2020). The influence of communication on bug-fixing performance in cross-functional teams. Empirical Software Engineering, 25(2), 1452–1471. https://doi.org/10.1007/s10664-019-09750-6

[20]. Bhat, J. A., & Choudhary, A. (2020). Cross-functional communication as a driver for software defect reduction in high-performance teams. Software Testing, Verification & Reliability, 30(1), e1832. https://doi.org/10.1002/stvr.1832

[21]. Murphy, M., & Wells, J. (2019). The impact of communication between testers and developers on defect resolution in large-scale software projects. IEEE Transactions on Software Engineering, 45(8), 789–804. https://doi.org/10.1109/TSE.2018.2861235

[22]. Bao, Y., & Zhang, Z. (2021). A study on the communication patterns in cross-functional teams during bug resolution in agile software development. Journal of Software Engineering and Applications, 14(5), 202–213. https://doi.org/10.4236/jsea.2021.145012

[23]. Turner, R., & Wright, P. (2018). Communication strategies in cross-functional software teams: Examining their effect on bug-fixing efficiency. Software Process Improvement and Practice, 23(4), 385–400. https://doi.org/10.1002/spip.1826

[24]. Zhao, X., & He, J. (2021). An empirical study of cross-functional communication's impact on bug resolution in agile teams. Journal of Software Quality, 29(4), 471–485. https://doi.org/10.1007/s11219-021-09410-5