Enhancing Automated Tools: Reporting Bugs with Bug Builder

Lennon Chaves
Sidia Institute of Technology
Manaus, Brazil
lennon.chaves@sidia.com

Flavia Oliveira
Sidia Institute of Technology
Manaus, Brazil
flavia.oliveira@sidia.com

Leonardo Tiago
Sidia Institute of Technology
Manaus, Brazil
leonardo.albuquerque@sidia.com

ABSTRACT

The detection of issues in software projects is crucial to the quality of the developed product. In particular, a test team within the Institute of Research and Development needs to automate their tests and processes through the creation of tools (software systems) that perform these tasks in an automated manner. Thus, the test team needs to build systems and guarantee that they work adequately. In case of issues, such systems require corrections. In this way, the goal of this work is to share the experience that the test team obtained when creating tools to perform their tasks and report the existing problems in these systems through a tool called *Bug Builder*. To evaluate *Bug Builder*, a qualitative review was performed to obtain the test team's perceptions of the implementation of *Bug Builder* in the team. Thus, this work aims to contribute to the industry and academia by presenting how the test team acts to ensure the quality of their own developed tools.

CCS CONCEPTS

• **Software and its engineering** → **Software verification and validation**; **Software development process management**; **Process validation**; **Software defect analysis**; **Software development methods**.

KEYWORDS

Bug Report, Automated Test, Software Testing

1 INTRODUCTION

A variety of projects have been developed and validated by a software test team at the Institute of Research and Development. Thus, sanity, regression, and functional tests [1] were executed throughout the development of the projects. Such tests are performed in the Android operational system[1], and each test suite [1] aims to validate some of the components of a device embedded with Android, such as: telephony, applications, connectivity, among others.

Furthermore, many processes are performed by the test team, such as reporting issues [4], consulting requirements, collecting test evidence, and test execution. In this way, to optimize each of the processes that are performed manually, the test team decided to develop tools to automate these processes and help the tester with their tasks. Consequently, the manual process is now automated owing to the developed systems. However, these systems are susceptible to failure, given that they are developed by humans and are prone to failure. During the implementation of these systems, some problems were detected by the users, which in this case were the members of the test team that were using them initially [5].

Despite these issues, no standard process or tool exists to report the detected problems. To address this, Jira[2] was utilized to log issues found in the test team's automation systems. Within Jira there is a type of task called "Problem" that includes fields such as title, labels, components, description, steps, found results, and expected results, aiding in problem description. However, this approach posed challenges for the test team, as each system's unique requirements led to varying types and quantities of information being needed. The team struggled to adhere to this process, resulting in significant rework to correct reports before addressing the automation system issues.

As a proposal to mitigate the problems in registering issues for the automation systems, a tool called *Bug Builder* was implemented, which servers as an interface between the user and Jira, that is, instead of the user reporting a system issue in Jira by filling all the information manually, *Bug Builder* contributes with the automated process of registering issues in Jira. Thus, some information such as labels, standard titles, and other fields are filled automatically by the system, and a jira ticket is created at the end of the process.

Contributions: This paper presents a summary of the lessons learned from the implementation of *Bug Builder* to report issues in the automation systems which are developed by the team themselves. In this way, interviews were conducted to collect the perceptions of the team regarding the manual and automated processes, aiming to summarize the benefits and improvement points with the use of *Bug Builder*.

The remainder of this paper is organized as follows: Section 2 details the *Bug Builder* and the learned lessons. Section 3 presents the conclusions of this study.

2 BUG BUILDER

2.1 Overview

Bug Builder is an automation tool that has as main goal the registering of issues and suggestions of improvements for the systems that the test team used to execute tests and processes. Particularly, *Bug Builder* is a web application developed with VueJS[3], which is a JavaScript[4] framework to develop web applications[5].

Furthermore, these systems are called "automation tools" because they represent tools that perform the test team's tasks automatically, that is, the tester uses an automation to perform a task instead of doing it manually. In summary, the automation tools maintained by the team can be organized as follows:

[1] https://www.android.com/

[2] https://developer.atlassian.com/
[3] https://vuejs.org/
[4] https://developer.mozilla.org/en-US/docs/Web/JavaScript
[5] Due to confidentiality policies of the client that owns the project in which the team that participated in this study works on, the source code of the tool could not be disclosed

- **Standalones**: They are tools that perform automated tests, that is, they act as a desktop interface so the tester can set up the test cases that will be executed in an Android device;
- **Farm**: They are modules that are part of a Physical Device Farm,that execute and automatically collect evidence of validation of requirements in Android devices. This way, a tester connects a test sample in a farm, and selects the types of tests that will be executed. This farm works as a server with all the devices connected and in stand-by waiting for test executions.

It is important to highlight that the test team is organized into two groups: (1) the manual tests group, which uses the automation systems that are produced by the (2) automation group, responsible for developing and maintaining the systems.

In this way, to register the issues to the testing team's systems, the tester needs: (1) to use the standalone tools or farm to perform their tasks; (2) in the case where they find issues in the systems, use *Bug Builder* to report the found problems; (3) finally, *Bug Builder* reports the issue in the Jira system. After these steps, the group responsible for the maintenance of the tools corrected the reported issues. The process of using *Bug Builder* is illustrated in Figure 1.Figure 1.

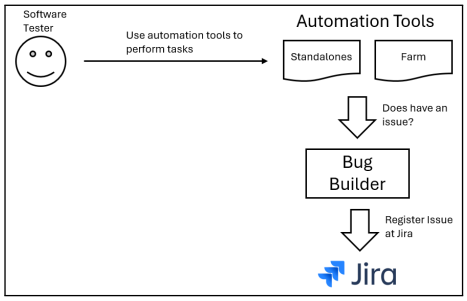

Figure 1: Bug Builder Tool Process

Bug Builder possesses some important requirements that make it relevant for use in the test team:

- **Availability**: It is an online tool, available 24 hours per day, and all the members of the team can access it;
- **Integration**: Possesses integration with the Jira system, which is a platform for managing the team's issues;
- **Auto-Filling**: Automates the filling of some information in Jira, such as: adding labels for issues, including tags in the report's title, organizing and formatting the data automatically;
- **Traceability**: Since the issues are registered in Jira, *Bug Builder* associates each issue to their corresponding tool, allowing tracking to be made between the problem and the tool.

Figure 2 illustrates the interface of the *Bug Builder* tool, in which it is necessary for the tester to fill in some information about the problem, such as the type of problem (bug or improvement), tool/system, title problem, tool version, frequency, priority, problem description, reproduction route, and expected results.

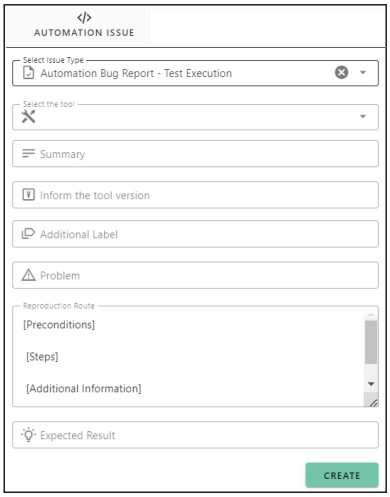

Figure 2: Bug Builder Tool Interface

2.2 Lessons Learned

To perform a qualitative study, 8 testers that register issues for automation systems using *Bug Builder* were selected. All of them possess more than one year of experience with software testing in the institute, are users of *Bug Builder*, and have already performed the process of registering issues manually. This group of testers comprised a representative sample of the team's total population. Furthermore, all participants signed the Informed Consent Form, agreeing to participate anonymously so that the information they provided during the interviews could be used in this study.

The research was structured in three steps: (1) In the first step the participant would register an issue manually, and right after the participant is asked to select two emotions they felt while performing the task from the Geneva Emotion Wheel [2]; (2) The second step was registering an issue using *Bug Builder*, and selecting two emotions again; (3) Lastly, a semi-structured interview [3] composed of 19 questions[6] was performed with the goal of comprehending the perceptions of the participants regarding *Bug Builder*. Based on the participants' answers, the lessons learned were categorized as follows:

Perceptions regarding the Manual Process: The participants reported that manual process for registering an issue is slower because they often had doubts about which information needs to be inputted in which field. They also mentioned that they worry about the information being inputted as expected, given that it is difficult to remember which information needs to be added. Lastly, the manual process is seen as laborious, cumbersome, and slow.

Perceptions regarding *Bug Builder*: When questioned about *Bug Builder*, participants mentioned that the tool is intuitive, practical, easy to use, attractive, effective and objective. Furthermore, they commented that all the necessary fields have already been mapped, which makes the process of registering issues more optimized. Finally, participants considered that *Bug Builder* met expectations and automatically filled in information such as labels and tags.

[6]https://zenodo.org/records/11644335

Performance and Efficiency: Regarding these aspects, the participants stated that *Bug Builder* speeds up the process, given the practicality of not needing to manually inputting information. In addition, the participants mentioned that there was an overall improvement in the speed of the resolutions of the issues reported after the implementation of the *Bug Builder*.

Satisfaction and Recommendation: Another important factor that was observed is that the participants feel satisfied with *Bug Builder*, given how the tool is easy to use, agile and stable. Finally, all of them believe that other test teams that register issues for their systems should use a similar approach and recommend the use of *Bug Builder*.

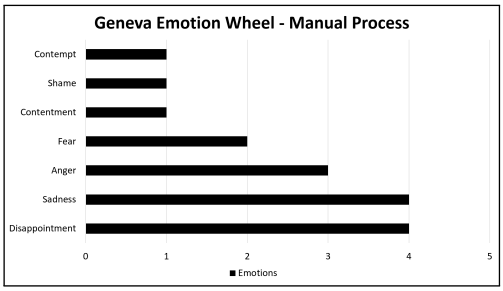

Figure 3: Geneva Emotion Wheel - Manual Process

In regards to the emotions perceived during the manual process for registering issues, the ones that stood out the most were *"Disappointment"* and *"Sadness"*, as illustrated in Figure 3. The reason for these emotions was questioned, and the participants mentioned that the process was exhaustive, unpractical, and required more time to be performed.

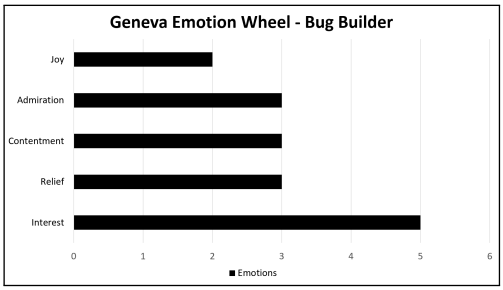

Figure 4: Geneva Emotion Wheel - Bug Builder

Regarding the perceptions of the use of *Bug Builder*, Figure 4 shows that the emotions that stood out the most were *"Interest"*, *"Relief"*, *"Contentment"* and *"Admiration"*. Among the highlighted reasons that motivated these emotions, participants mentioned that they felt that doing the task using the tool was quicker and they did not need to worry about many details, making the task itself more interesting and easy.

3 CONCLUSIONS

This paper presents a *Bug Builder* as a solution to automate the manual process of reporting issues for automation systems in a software test team. To achieve this goal, qualitative research was performed using a semi-structured interview and the Geneva Emotion Wheel to understand the perceptions of the testers with regard to the implementation of *Bug Builder* in the team. The results show that the manual process causes negative feelings such as *"Disappointment"* and *"Sadness"*, while the automatic process with Bug Builder reveals more positive feelings such as *"Interest"* and *"Relief"*. Among the main benefits noticed by the testers are intuitiveness, efficiency, practicality, and ease of use of the *Bug Builder*. As an improvement point, some testers mentioned that it would be interesting if the tool could search for previously reported issues. It is important to mention a few aspects that threaten the validity of this study: (1) it was not possible to generalize the results of this research, given that it was performed in the context of a specific institute that develops its own automation tools; (2) the time for registering issues manually and automatically was not mentioned, so the gain of the time in regard to how quickly an issue can be registered could not be determined. In future work, research in the Large Language Models (LLM) field could be performed to understand how prompts can be created to automatically generate issue reports for automation systems.

ACKNOWLEDGMENTS

This paper is a result of the Research, Development & Innovation Project (ASTRO) performed at Sidia Institute of Science and Technology sponsored by Samsung Eletrônica da Amazônia Ltda., using resources under terms of Federal Law No. 8.387/1991, by having its disclosure and publicity in accordance with art. 39 of Decree No. 10.521/2020.

REFERENCES

[1] Marcio Delamaro, Mario Jino, and Jose Maldonado. 2013. *Introdução ao teste de software.* Elsevier Brasil.

[2] Klaus R. Scherer. 2005. What are emotions? And how can they be measured? *Social Science Information* 44, 4 (2005), 695–729. https://doi.org/10.1177/0539018405058216 arXiv:https://doi.org/10.1177/0539018405058216

[3] David Silverman. 2020. Qualitative research. (2020).

[4] Yang Song, Junayed Mahmud, Ying Zhou, Oscar Chaparro, Kevin Moran, Andrian Marcus, and Denys Poshyvanyk. 2022. Toward Interactive Bug Reporting for (Android App) End-Users. In *Proceedings of the 30th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering* (<conf-loc>, <city>Singapore</city>, <country>Singapore</country>, </conf-loc>) *(ESEC/FSE 2022)*. Association for Computing Machinery, New York, NY, USA, 344–356. https://doi.org/10.1145/3540250.3549131

[5] Kevin J. Valle-Gómez, Pedro Delgado-Pérez, Inmaculada Medina-Bulo, and José Magallanes Fernández. 2019. Software testing: cost reduction in Industry 4.0. In *Proceedings of the 14th International Workshop on Automation of Software Test* (Montreal, Quebec, Canada) *(AST '19)*. IEEE Press, 69–70. https://doi.org/10.1109/AST.2019.00018