Reducing the Allocation of Software Testing Demand: A Study on the Pros and Cons using STELA Tool

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

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

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

ABSTRACT

The speed at which new technology is developed and implemented in commercial products directly impacts the flow and rhythm of the software test field. In the context of global software development, different companies follow their unique approaches to assign their test demands to their employed testers; however, the dependency on a fully manual process that is prone to human error leads to a growing trend of possible issues, given that demand is expected to grow even more, making it even harder to keep up with them. To address this problem, we performed a study to determine the significance of the time gain and effort reduction when comparing the assignment task performed manually and automatically with support from the STELA tool developed within the research and development institute in which this study was conducted. Analysis of the data obtained through the collection of execution time and one-on-one interviews with the main assigners of a test team within the institute shows that using automation to simplify the assignment task not only makes it more efficient, with a reduction of 54% in the time invested when compared to manual execution, but also reduces the possibility of human error, even though the tool itself is prone to occasional errors.

CCS CONCEPTS

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

KEYWORDS

Software Testing, Automated Process, Test Management

1 INTRODUCTION

Software tests help ensure that software is stable and works with quality when it reaches the end users [5]. However, given the current state of globalization, there are teams working from different locales in different time zones to develop products, a work setting known as Global Software Development (GSD) [4]. This scenario impacts the product's development directly, with new technologies being developed and incorporated into products in parts of the world, while needing to be tested by people on the other side of the planet.

Therefore, in order to avoid any conflicts during development that may compromise the release of the product, the demand for test requests - which are documents received by the test teams that specify what needs to be validated - needs to be well managed, so that all tests are performed with quality and, consequently, the product has a healthy development cycle [1].

Within an institute of research and development, test requests are assigned by project leaders and received by the test team, who perform functional and exploratory tests [2]. Furthermore, in the software test team, the process of validating the information present in the request and determining whether they are all correct before starting the test is known as acceptance, and the process of delegating the request execution to a tester according to their availability and knowledge is known as assignment. In this context, each project represents a smartphone or tablet device with the Android operating system[1].

However, the management of acceptance and assignment can be costly, especially when there are many test requests, because of the necessity of manual verification of all information contained within the request, including, but not limited to, checking the environment, type of device, and scenario in which the test needs to be performed, and determining who from the test team is available and has the theoretical and practical knowledge needed to perform the test.

All of these validations need to be made in a task management system known as Jira[2], and the information is spread between many different pages, so the amount of clicking, screen transitions, and searching for specific data can reach an exorbitant amount when the demand is very high. To minimize the manual effort required to validate the test requests, the automation team, which is composed of members of the test team, developed a module for managing the test requests within the STELA tool.

Thus, this work presents an experience report on the use of the STELA tool to manage the demand for test requests by a test team within a research and development institute, with the goal of highlighting the time gain and effort reduction achieved through the use of automation to perform complex and repetitive tasks, and consequently resulting in an overall better test cycle.

The remainder of the paper is organized as follows: Section 2 presents the assignment module of the STELA tool; Section 3 presents the obtained results and discussions, and lastly, Section 4 presents the conclusions we reached.

2 STELA TOOL

In the institute in which this research was performed, the projects under development have a leader, who is responsible for delivering the software releases which must be tested by the team responsible for testing. The test requests are made through the Jira platform, where it is possible to follow the project and view information about the type of test being performed, its due date, the software's ID, the version of the operational system, among others.

[1] https://www.android.com/
[2] Jira, available at: developer.atlassian.com

Within the test team, a small group is responsible for checking the test request, confirming if all the necessary information is filled in. The members of this group are called "assigners". The members of the test team who receive the test request and perform the tests are referred to as "testers" in this paper.

The assigners were also responsible for managing the tester's test demands. Therefore, if the information on the request is deemed correct, the assigner checks if testers are available to execute the test and assigns the request to the selected tester.

The manual process of accepting a request is shown in Figure 1, which starts when a test request is created by the project leader.

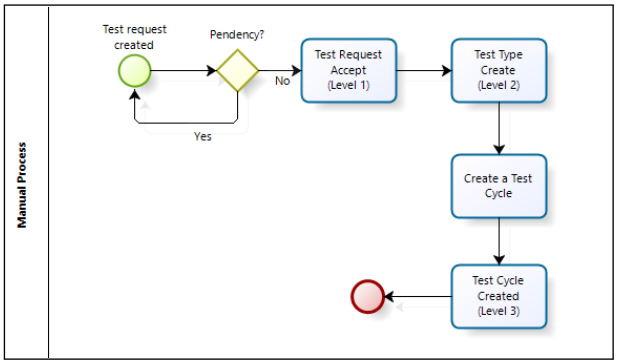

Figure 1: Manual Management Process

In step 1, the assigner checks if there are any pendencies in the request's main page, such as missing information from the test environment or the application that needs to be tested, unavailability of the files needed to perform the test, and incoherence with the history of prior tests.

In step 2, the assigner performs acceptance, the request's test scopes are separated by category, and for each category, the corresponding test cycle must be created, which contains all test cases to be performed.

Finally, in step 3, the categories are divided into tasks to be performed by the tester, containing summarized information from the request and features so that the tester can interact with the page and indicate their progress, make comments and attach evidence of the test execution.

To minimize the time spent assigning test requests, a module that manages and supports the acceptance of requests was developed using the STELA tool and made available to the team by the end of January 2024. STELA is a system developed by the automation team to manage and control test executions.

The tool's management module was developed using VueJS[3] and TypeScript[4] for the front-end, and Python[5] for the back-end, however further details of the implementation (such as the source code itself) cannot be disclosed because the institute in which the study was conducted possesses strict confidentiality policies for code developed within the institute.

The process of automatic acceptance with support from STELA is illustrated in Figure 2.

[3]https://vuejs.org/
[4]https://www.typescriptlang.org/
[5]https://www.python.org/

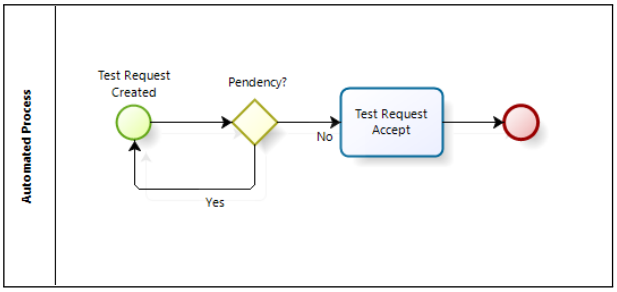

Figure 2: Automated Management Process

As illustrated in Figure 2, after the creation of the request, STELA automatically checks the test request's information fields and the software release made available. In this way, there are two possible statuses that STELA provides to test requests:

- **PASS**: With the PASS status, it is possible to accept a test request within STELA, and the creation of the test task is performed automatically;
- **FAIL**: A test request with this status possesses discrepancies, and the tool indicates which information fields need to be fixed.

It is important to highlight that STELA's request management module is viewed by both the assigners and the project leaders, so they can also see the request's status and act on it. In the case of FAIL status, the project leader can find the discrepancy and act on a correction. It is also considered good practice for the assigners to monitor this module to check if there are new test requests that the team needs to work on, and when finding the FAIL status, contact the project leader to inform them of the discrepancies, so that they can be fixed and the acceptance can be performed.

3 RESULTS

To perform the experiments, 4 participants who were responsible for this task for over a year and had experience with both the manual process and support from STELA were selected. It is important to emphasize that the number of assigners is small given that only this group is responsible for distributing the demand for the team. Thus, the sample selected for this study was representative of participants within the context of this study. During the study period, the assigners measured the time for accepting a request both manually and using the automatic acceptance in STELA. In all, 65 assigned test requests were considered and accepted with and without the use of the tool.

It is noteworthy that the requests covered different types of devices and test scopes. The process of assigning requests may sometimes require negotiations and adjustments, which leads to an even greater time cost for resolving the discrepancies before proceeding with acceptance. Therefore, to measure the time spent, the median was used as the measurement of the central tendency because it is not affected by outliers [3]. Considering the time spent by the selected assigners to accept the requests, the results are compiled in Table 1. Manually, it is possible to observe a total time of $74min20s$ to assign requests, while with support from STELA, the total time was $33min49s$.

Table 1: Time for accepting requests

Form of Acceptance	Total time	Median
Manual	$74min20s$	$65s$
Automated	$33min49s$	$30s$

Thus, with the data displayed in Table 1 it is possible to see a reduction of approximately 54% with the use of the tool when compared to the manual process. Besides the experiments made to determine the time gain, the participants of this study were interviewed[6] to assess the perceptions of the assigners regarding the tool. All participants signed an Informed Consent Form, ensuring that they consented to use the information obtained in the interviews to develop this study while maintaining their anonymity. The following is a discussion of the interpretations.

Perceptions regarding the Manual Process: Based on the interviews, it is valid to affirm that the time they had been on the team and the experience acquired during this time did not lead to higher efficiency in the process of manually assigning requests, indicating how high is the cost of time and effort for this task. It is valid to mention that his cost is not only associated with the task itself, but also with Jira, the system in which it is performed. Participants emphasized that manual flow requires many clicks and screen transitions, extending the time and increasing complexity, which could cause them to make mistakes, given that it is a task performed every day.

Perceptions regarding the Automated Process: Based on the perceptions of the participants regarding the manual process, it is possible to understand why all of them agreed that the automation of the assignment task within the STELA tool made their job much easier, as the tool performs the analysis of the request's necessary information in Jira, assertively points out discrepancies, moves the important information to the task that will be received by the tester, and is able to accept multiple requests at once.

Opportunities for Improvement: That all being said, there are still cases in which the STELA tool can fail and make the assigner have to do more work, such as when - as mentioned in the interviews - it was mentioned that the system can slow down and crash, and when this happens it is necessary to check all of the requests it was working on individually, as it does not say which of them have discrepancies and what is said discrepancy, an opinion shared between 75% of the interviewed team members, which show how the tool violates some aspects of user experience and interface, and also how it is still prone to issues due to the Jira system.

4 CONCLUSIONS

This study examines a software test team's use of an automation module to manage the demand within the STELA tool. The research institute's test team receives numerous daily test requests, requiring significant time for the assigners to verify the information before acceptance and assignment. Team assigners were interviewed to assess the tool's impact. They suggested improvements such as an execution report to identify errors, but all agreed that the tool accelerates the process and reduces manual clicks. Additionally, the assigners compared the time spent accepting requests manually with STELA. The median values exhibited a reduction of approximately 54% with the automated tool.

It is important to highlight the aspects that threaten the validity of this study: 1) The research was developed in an institute with confidentiality policies and a specific context of development; 2) The study had a small sample of participants, as the group of assigners was small (only four members); 3) This study did not consider how the type of project or test scope would affect the time spent; 4) The participants mentioned that more clicks were necessary when working manually, however the exact amount was not measured, so this was affirmed only based on their perception.

For future work, it is suggested to perform experiments focused on understanding the impact of using an automation tool to accept requests in specific test scopes, as the differences between the information that needs to be checked for each test scope may impact the acceptance process and it is also suggested to perform a study with novice members to understand the impact of using the tool without an experience bias.

[6]https://zenodo.org/records/11658222

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] Tulasi Anand, Chittoor Reddy, and VS Mani. 2016. System testing optimization in a globally distributed software engineering team. In *2016 IEEE 11th International Conference on Global Software Engineering (ICGSE)*. IEEE, 99–103.

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

[3] Luiz Paulo Fávero and Patrícia Belfiore. 2017. *Manual de análise de dados: estatística e modelagem multivariada com Excel®, SPSS® e Stata®*. Elsevier Brasil.

[4] Babur Hayat Malik, Saeed Faroom, Muhammad Nauman Ali, Nasir Shehzad, Sheraz Yousaf, and Hammad Saleem. 2018. Geographical distance and communication challenges in global software development: A review. *International Journal of Advanced Computer Science and Applications* 9, 5 (2018).

[5] Yi Zhao, Yun Hu, and Jiayu Gong. 2021. Research on international standardization of software quality and software testing. In *2021 IEEE/ACIS 20th International Fall Conference on Computer and Information Science (ICIS Fall)*. IEEE, 56–62.