Express Release Validation: Enhancing Quality in tight development timelines

José Ricardo B. Moreira
Sidia Institute of Science and Technology - SIDIA
Manaus, Brazil
jose.ricardo@sidia.com

Maria Deolinda de O. Santana
Sidia Institute of Science and Technology - SIDIA
Manaus, Brazil
deolinda.santana@sidia.com

Gleiça Gabriela J. Silva
Sidia Institute of Science and Technology - SIDIA
Manaus, Brazil
gleica.silva@sidia.com

Raimundo F. Rocha
Sidia Institute of Science and Technology - SIDIA
Manaus, Brazil
raimundo.rocha@sidia.com

Erick Costa Bezerra
Sidia Institute of Science and Technology - SIDIA
Manaus, Brazil
erick.bezerra@sidia.com

ABSTRACT

The industry's strong demand for increasingly faster software releases has led testing professionals to seek agility in their processes. This article proposes the Express Release Validation approach, based on essential quality criteria and the experience of quality assurance analysts in the project. This new strategy seeks to optimize testing time by focusing on the most critical aspects of the system without compromising the product reliability.

KEYWORDS

Express Release Validation, Quality Assurance, Software Testing, Test Optimization, Testing Strategy

1 Introduction

Technological advances in companies require more robust and reliable systems, driving the expansion of software quality. This requires rigorous monitoring of software development without compromising the efficiency and agility of deliveries. In this way, the quality assurance team provides data to management and technical staff, ensuring that actions for product quality are effective [5].

According to [1], software quality is widely recognized and integrated into agile methods, which incorporate quality practices into the development process to ensure fast deliveries aligned with changing customer requirements. In global teams, impediments that require urgent resolution are handled differently. To this end, several tools are used to optimize communication and activity management, promoting greater integration between the teams involved in the project, as applied in [3], wich refers to the BluePrint methodology that reduces distance between teams.

This paper presents the Express Release Validation (ERV) and its challenges, emphasizing essential criteria and the experience of quality assurance (QA) analysts to balance speed and quality in software testing without compromising delivery deadlines.

The paper is structured as follows: Section 2 addresses the context of the work and the formal testing strategy used previously. Section 3 presents the methodology, detailing the solution based on fundamental testing practices. Section 4 presents the results applying the ERV compared to the Usual Quality Delivery (UQD) approach. Finally, Section 5 contains the conclusions.

2 Background

The software development teams in this study operate in a distributed environment, with teams located in Brazil, South Korea and Vietnam. The South Korean team primarily requests new features, Vietnam team executes full test cycles, while the Brazilian team develops and test solutions maintaining a continuous delivery schedule. This global collaboration ensures efficiency but also presents challenges in maintaining consistency and quality across different time zones and work cultures.

The Brazilian development team enhances a system for security release updates for smartphones, which is widely used in the global ecosystem. This system contains modules that frequently undergo requirement changes, especially those handling critical data for the release. Therefore, rigorous validation is crucial to ensure that modifications meet the established criteria before deployment.

Initially, the QA team followed a structured workflow known as UQD, involving requirements specification, test plan creation, execution of complete test cycles, and detailed reporting with quality indicators. However, as the system evolved, the constant increase in complex requirements, demand, shorter deadlines, and a limited number of testers made the full execution of test cycles challenging. Requiring a reformulation of the existing testing strategy to meet demands with agility and quality.

The new approach proposes the adoption of express deliveries, prioritizing essential criteria and leveraging the QA team's expertise to ensure a balance between efficiency and reliability in software release cycles.

3 Express Release Validation

To adapt the software testing process and optimize version deliveries, a new approach called Express Release Validation (ERV) was developed. This strategy emerged from the need to adapt to a critical scenario where the demand for more frequent and agile deliveries became increasingly evident. The proposed approach aims to balance speed and quality, allowing testing in short periods without compromising product reliability.

Unlike the traditional strategy, which relies on complete test cycles and an extensive coverage, the new approach is based on strategic prioritization principles, exploring more dynamic and

efficient methods. To achieve this, the solution includes six fundamental practices, executed according to the sequence described below:

- Risk-Based Testing Strategy – Prioritization of critical functionalities, focusing on areas where the impact on the end user is greatest. By prioritizing risks, resources can be allocated where they will have the most significant effect [5]. Critical functionalities with a history of defects are targeted for more rigorous testing.
- Incremental Automated Testing – Continuous automation to validate essential functionalities before each release; Reduction of manual testing workload, enabling faster cycles; Repetitive execution of critical tests for early defect detection [2].
- Smoke Test – This type of testing provides many benefits when applied to complex software engineering projects with critical deadlines [5]. It is used before each release to quickly validate if recent changes have not compromised essential system functionalities.
- Scripts for Complex Requirement Validation – Automation of complex validations to reduce human errors; automatic comparison of large data volumes; optimization of multiple simultaneous requirement checks. Scripts should be applied whenever feasible, particularly when software modifications are made [6].
- Exploratory Test – Leverages QA analysts' expertise to identify unexpected defects in the project. This type of testing is most effective when conducted by an experienced QA professional with deep domain knowledge and essential skills such as analytical thinking, curiosity, and creativity [2]. Unlike traditional tests that follow predefined scripts, exploratory testing offers a more flexible and adaptive approach, allowing the discovery of defects that may not be covered by formal test cases.

  When the tester becomes familiar with the system under test and refines validation scenarios based on knowledge gained from exploration without relying on predefined instructions, the exploratory approach can demonstrate effectiveness comparable to more structured methods such as script-based testing [4].
- Bug Retest – Continuous verification of fixes in subsequent versions; Prevention of regressions and side effects; Increased system reliability throughout releases. Also known as confirmation testing, its goal is to verify whether a previously identified defect has been successfully fixed. This involves re-executing the same steps that originally caused the failure, ensuring the issue no longer occurs. This process is essential to validate the effectiveness of the implemented fix and guarantee the stability of the tested functionality [2].

By implementing these practices, the ERV strategy reduces the overall testing cycle time without compromising quality. This approach combines traditional and modern techniques to ensure fast and reliable deliveries, allowing for more frequent releases with security and efficiency.

4 Results

To analyze the results, similar deliveries were analyzed in terms of difficulty and scope. Considering the time of testing activities and the number of reported issues as the main factors for this analysis, data were collected over a period of 11 working days (3 deliveries) for ERV and 14 working days (1 delivery) for UQD:

- ERV was 3.82 times faster than UQD.
- In terms of reported issues, an average of 24 issues per delivery was reported for ERV, compared to 18 issues for UQD. Based on the numbers obtained, ERV's delivery rate was significantly higher, proving to be more efficient.

Adopting ERV is advantageous because it improves adaptation to changes and provides a continuous flow. It is associated with a more agile testing cycle, allowing effective feedback from users. The main focus of this model is on the accelerated availability of functional software, where new features and corrections are implemented more frequently. This process allows iterative and incremental adjustments, reducing response time to changes and improving the user experience.

Furthermore, ERV can facilitate greater test coverage even in a shorter execution time. This is because there is less need for detailed documentation during the testing process and in the final delivery. At the end of each test delivery cycle, a summary message is sent to stakeholders, highlighting the main validation points linked to a main task of each delivery. This ensures tracking and control over a timeline. This mechanism reduces bureaucratic overhead and maintains the continuous flow of essential information about the project's progress.

However, ERV has some limitations, mainly due to the reduced time for complete execution of test cycles. The lack of comprehensive tests can result in uncovered scenarios, allowing bugs to escape into the production environment. To avoid this problem, the new approach proposes prioritizing essential criteria and the experience of the QA team to ensure a balance between efficiency and reliability in the release of software versions.

5 Conclusion

ERV is a good alternative when there is reduced time and resources allocated to fully execute all stages of the conventional testing process. It improves the team's responsiveness, reducing the time to release new versions and keeping the software aligned with the customer's needs. Despite the inherent risks addressed in the results, we can highlight that by applying the use of several existing testing techniques it is possible to mitigate the negative impacts and succeed in delivering quality.

ACKNOWLEDGMENTS

This work is the result of the R&D project *Projeto de Engenharia de Software e Ciência de Dados aplicados ao Desenvolvimento de Sistemas*, performed by Sidia Instituto de Ciência e Tecnologia in partnership with Samsung Eletrônica da Amazônia Ltda., using resources from Federal Law No. 8.387/1991, and its disclosure and publicity are under the provisions of Article 39 of Decree No. 10.521/2020.

REFERENCES

[1] Mauricio D. Arcos-Medina, G. 2019. Aspectos da qualidade de software aplicados ao processo de desenvolvimento ágil de software: uma revisão sistemática da literatura.. In *Proceedings of the 3rd. annual workshop on Librarians and Computers*. Int J Syst Assur Eng Manag 10, 867–897. doi:10.1007/s13198-019-00840-7

[2] International Software Testing Qualifications Board. 2023. ISTQB® Certified Tester Foundation Level (CTFL) Syllabus (Version 4.0). 4.0 (april 2023), 39–40. https://bstqb.online/files/syllabus_ctfl_4.0br.pdf

[3] Cristiano P. Godoy, Andre F. Cruz, Elisangela P. Silva, Lanier M. Santos, Rafael S. Zerbini, and Cícero A. L. Pahins. 2019. Blueprint Model: A new Approach to Scrum Agile Methodology. In *2019 ACM/IEEE 14th International Conference on Global Software Engineering (ICGSE)*. 95–99. doi:10.1109/ICGSE.2019.00014

[4] Karina, Ana. 2018. Testes Exploratórios: Características, Problemas e Soluções. *Annalen der Physik* (november 2018), 28–29. https://repositorio.ufrn.br/bitstream/123456789/34172/2/TestesExploratorios_Castro_2018.pdf

[5] Roger Pressman. 2021. *Software Engineering A Practitioner's Approach* (9th ed.). McGraw-Hill. 681 pages.

[6] Ian Sommerville. 2018. *Software Engineering* (10th ed.). Pearson. 228 pages.