

## Regression Testing Techniques: A Review

**Maria Ashfaq Ahmad**

Higher Education Archives and Libraries Department

Peshawar, Pakistan

Email: mariaashfaq2018@gmail.pk

### ABSTRACT

The escalation in constant modifications results in veiling faults in software applications that evade detection, compromising the reliability of agile systems. This increasing difficulty of today's software requires a more strategic assessment of technique trade-offs. If changes are not handled carefully, they may lead to errors in a cascading manner, propagating failure, leading to their downfall. Therefore, the maintenance is performed throughout the software's lifecycle. Whenever a change is implemented, the software undergoes some predefined tests to ensure its correct working. One of the many testing techniques that validate the proper functioning of software after modifications have been introduced is Regression Testing. This paper categorizes Regression Testing algorithms into five categories, i.e., retest all, test selection, prioritization, test reduction, and hybrid approaches. It aims to evaluate these categories against features such as cost effectiveness, fault detection rate, etc. At the end of the paper, the findings are summarized and future work is proposed.

**Keywords:** software testing, regression testing, prioritization, selection, retest

### INTRODUCTION

The success of software is not limited to its development time or the delivery of correct results after the software is completed. Rather, its ability to adapt to changes requested by users is also a contributing factor to the software's success. This requires the software to be constantly monitored and maintained. Maintenance focuses on rigorous testing of the software whenever a change is incorporated. Testing is considered an effective process that boosts quality assurance.[1][2] Many testing techniques have been developed over time to address this goal, of which regression testing is one. Software maintenance consumes 70% [3] of the software development lifecycle, while regression testing consumes around 80 % of that maintenance phase [4]. Regression testing is defined as:

“Re-running functional and non-functional tests to ensure that previously developed and tested software still performs after a change is in effect” [5]

Thus, regression testing is intended to validate software functionalities following each modification. Traditional regression testing techniques are required to keep up with the ever-increasing complexity and adaptability demands by balancing effectiveness (error detection) with efficiency (usage of resources). Researchers have developed several regression testing techniques over the years. Some of the techniques are more

Online ISSN

3007-3197

Print ISSN

3007-3189

<http://amresearchreview.com/index.php/Journal/about>

comprehensive, but they consume a lot of resources, while others may be cost-effective, but they might miss critical defects. This tradeoff between resource consumption and error detection necessitates the evaluation of

these techniques under varied conditions. Another reason for the evaluation of regression testing techniques is the advent and increasing development of agile development, continuous integration/development (CI/CD), and distributed environments, which demand a real-time response.

This review article aims to synthesize the current state of regression testing by addressing three research questions:

RQ1: To what extent is the effectiveness and efficiency balanced in different regression testing techniques?

RQ2: What challenges are faced when different regression testing techniques are applied in ever-changing real-time environments?

RQ3: To what extent do hybrid approaches overcome the current regression testing limitations?

To answer these questions, this paper presents a comparative evaluation of regression testing techniques, grouping them into five categories. The rest of the paper is organized in a way that follows from foundational knowledge and leads to actionable insights as follows: Section 2 presents current research work on Regression Testing, highlights important advancements, identifies challenges faced in each category, Section 3 empirically evaluates each category by comparing and analyzing the behavior of each category against critical parameters, and subsequently determining the optimal environmental conditions for each category, providing practitioners and future researchers with evidence-based decision making. Section 4 concludes the work by summarizing key findings and identifying research gaps.

## RELATED WORK

Regression testing plays a critical role in ensuring software quality during maintenance and evolution. As software systems undergo frequent changes, it becomes necessary to retest previously validated components to verify that no new faults have been introduced. This results in a substantial cost in terms of time and resources, particularly in large-scale systems. Aiming to optimize the process of regression testing, researchers have developed new techniques which can be broadly categorized as Retest-All, Test Selection, Prioritization, Test reduction, and Hybrid approaches.

In Retest-All technique the entire test suite is re-run whenever a change is introduced. Although this strategy guarantee fault free solution yet it is not a very practical approach due to high computation cost and time consumption [6]. While Test Selection techniques reduces this exhaustive approach by selection method that checks only the affected areas as a result of recent modifications. Although these are cost effective yet they have a high risk of missing critical areas which disturbs the effectiveness and efficiency concerns. [7]. This directly relates to the first research question (RQ1), which explores how different regression testing techniques manage this tradeoff.

Nowadays, systems being developed are in a race with each other to compete in the practical environment by producing faster release cycles. These systems face continuous adaptation in the face of a high integration rate [8], creating new challenges for

traditional regression testing techniques. It becomes hard to maintain test suites as changes are also affecting them. Scalability, tool integration, and automation compatibility are also persistent concerns [9]. Thus, introducing the second research question (RQ2) in order to research the difficulties faced when applying regression testing techniques on these contemporary agile systems.

Hybrid approaches have been introduced in an effort to face these difficulties by combining various techniques using symbolic methods with embedded intelligent solutions. For example, some hybrid techniques use historical data and fault patterns to better prioritize or select test cases [10]. These new advancements try to answer the third research question (RQ3), which explores how hybrid techniques can deal with the current limitations faced when regression testing is applied to modern systems.

This research work aims to contribute new insights based on the current work done in regression testing. This is achieved through focused literature-based evaluation.

## METHODOLOGY

This review addressed the above-mentioned three research questions through a systematic literature survey, which it used as the primary research methodology. This method enables the in-depth examination of the given techniques across various academic and practical environments. The targets are to access critically and compare the techniques in terms of their effectiveness, efficiency, real-world implementations and innovative capabilities.

Data is collected from these platforms, such as IEEE Xplore, ACM Digital Library, Google Scholar, were utilized by passing keywords such as “regression testing techniques,” “test case selection”, “test prioritization and reduction”, “regression testing in agile environments”, etc. Papers from the past decade were analyzed, prioritizing articles that featured empirical research based on practical cases. Through this methodology, we were able to identify new trends, research gaps, and future advancements in the field of regression testing.

| Online ISSN |                      | Print ISSN       |                                        | <a href="http://amresearchreview.com/index.php/Journal/about">http://amresearchreview.com/index.php/Journal/about</a> |                                                |                                                             |
|-------------|----------------------|------------------|----------------------------------------|-----------------------------------------------------------------------------------------------------------------------|------------------------------------------------|-------------------------------------------------------------|
| S . N O     | FEAT URES            | TEST EST ALL     | TEST SELECT ION                        | PRIORITIZ ATION                                                                                                       | HYB RID                                        | TEST REDUC TION                                             |
| 1           | Test cases           | All are executed | A portion of the test case is executed | Prioritized test cases based on achieving some goal are executed                                                      | Test cases are executed based on historic data | Selected tests are further reduced based on different goals |
| 2           | Cost efficiency      | Very low         | High                                   | High                                                                                                                  | Very High                                      | Very High                                                   |
| 3           | Test Suite           | Whole            | Partly                                 | Ordering suites                                                                                                       | Partly/ Order                                  |                                                             |
| 4           | Fault detection rate | Very High        | Moderate                               | High                                                                                                                  | High                                           | Varies                                                      |
| 5           | Precision            | Low              | High                                   | Medium                                                                                                                | High                                           | Depends                                                     |
| 6           | Execution Speed      | Very Slow        | Fast                                   | Faster                                                                                                                | Fastest                                        | Very Fast                                                   |
| 7           | Runtime Impact       | High             | Medium                                 | Low                                                                                                                   | Optimal                                        | Low                                                         |
| 8           | Overall Balance      | Poor             | Good                                   | Good                                                                                                                  | Best                                           | Risky                                                       |

This methodology ensures that the study is not only grounded in current academic discourse but also capable of delivering practical insights into how regression testing is evolving and where opportunities for improvement remain.

## COMPARATIVE ANALYSIS OF REGRESSION TESTING TECHNIQUES

Regression testing is essential in software maintenance to ensure that future modifications do not introduce errors. To date, extensive research has been conducted to develop a more efficient and effective methodology for optimizing the implementation of regression testing. As the number of techniques is varied and would be difficult to follow individually, for clarity and ease of understanding, this review has categorized them into five categories: retest all, test selection, test prioritization, test reduction, and hybrid. These categories are assessed based on a set of features selected to determine the optimal conditions for their application

and to identify the limitations and challenges associated with each category. This section evaluates widely-used regression testing techniques by comparing how they balance effectiveness (e.g., fault detection, accuracy) with efficiency (e.g., time and resource cost). Based on the literature surveyed, we compare the four primary techniques: Retest-All, Test Selection, Test Prioritization, Hybrid Approach, and Test Reduction.

Retest-All verifies the changes in the program by running all the tests that are included in the existing test suite. It is costly as it performs all the test scenarios to ensure the absence of faults that may arise due to the incorporation of the change. [11]

Test Selection verifies the changes by selecting a portion of test case to run on the modified code. Test Selection techniques have been developed to fulfill different goals. The motivation behind developing this technique was to reduce cost in terms of time and resources. [12]

Test Reduction technique focused on reducing the number of selected test cases for retesting of the software developed or the changes introduced.[13]

The Prioritization technique was introduced to plan the test cases in such a manner as to achieve a goal successfully. [14][5] It targets providing the optimized methodology for locating discrepancies as quickly and as economically as possible.[15] [6]

The Hybrid approach is a mixture of Selection and Prioritization techniques. It combines the best features of both techniques to accomplish improved results. Hybrid Techniques proposed works on minimizing the test part by analyzing the history of the tests that are conducted.[16][7]

The following table evaluates five regression testing techniques—Retest-All, Test Selection, Prioritization, Hybrid Approach, and Test Reduction—based on key metrics like cost efficiency, fault detection, precision, execution speed, run-time impact, and overall balance. As per the values of this table, retest-all is justified only in scenarios where the fault tolerance rate is near zero, while Test selection is best for well-modularized environments. Prioritization techniques are focused on high-risk areas to accelerate defect discovery. Test reduction is risky when the changes being introduced are too many. Hybrid remains a balanced option for software developed nowadays.

TABLE 1. COMPARISON OF REGRESSION TESTING CATEGORIES

## IMPACT OF PRACTICAL CHALLENGES ON REGRESSION TESTING TECHNIQUE

The comparative analysis highlights how various regression testing techniques respond to practical challenges commonly encountered in modern software environments. Retest-All, while thorough, struggles significantly with scalability due to its exhaustive nature and shows limited adaptability to dynamic or evolving codebases. Test Selection and Test Prioritization techniques demonstrate better handling of frequent code changes and make more efficient use of resources,

Online ISSN

3007-3197

Print ISSN

3007-3189

<http://amresearchreview.com/index.php/Journal/about>

though their effectiveness heavily depends on the availability of reliable change data and integration tools. Hybrid Approaches stand out by offering a balanced response to most challenges, particularly excelling in managing test suite maintenance and leveraging historical data for improved accuracy. Test Reduction techniques show moderate efficiency, especially in addressing maintenance overhead, but may fall short in handling non-functional requirements and frequent dynamic changes. Overall, the analysis suggests that while no single technique overcomes all limitations, hybrid methods tend to provide a more adaptable and efficient solution in real-world, high-velocity development contexts

| S<br>·<br>N<br>O | FEATU<br>RES                                             | RETES<br>T ALL | TEST<br>SELECTIO<br>N | PRIORITI<br>ZATION | HYBRID | T<br>E<br>S<br>T<br><br>R<br>E<br>D<br>U<br>C<br>T<br>I<br>O<br>N |
|------------------|----------------------------------------------------------|----------------|-----------------------|--------------------|--------|-------------------------------------------------------------------|
| 1                | Scalabilit<br>y and<br>resource<br>constrain<br>ts       | High           | Low                   | Medium             | Medium | M<br>e<br>d<br>i<br>u<br>m                                        |
| 2                | Tool<br>integrati<br>on and<br>compatib<br>ility         | Medium         | Medium                | High               | High   | M<br>e<br>d<br>i<br>u<br>m                                        |
| 3                | Frequent<br>and<br>Dynamic<br>Code<br>Changes            | Low            | High                  | High               | Medium | L<br>o<br>w                                                       |
| 4                | Lack of<br>Quality<br>Historic<br>Data                   | Low            | High                  | High               | High   | M<br>e<br>d<br>i<br>u<br>m                                        |
| 5                | Mainten<br>ance<br>Overhea<br>d of Test<br>Suite         | Low            | Medium                | Medium             | High   | H<br>i<br>g<br>h                                                  |
| 6                | Limited<br>support<br>for non-<br>function<br>al testing | Low            | Medium                | Medium             | Medium | L<br>o<br>w                                                       |

TABLE 2. IMPACT OF PRACTICAL CHALLENGES ON REGRESSION TESTING CATEGORIES

ROLE OF HYBRID APPROACHES

The introduction of Hybrid regression testing techniques has overcome the challenges faced by traditional testing techniques in the face of constant change and adaptability in varying environments. They aim to combine salient features of

other regression testing techniques to balance trade-offs between them. Thus, integrating these features provides a way that is more change-sensitive, cost-effective, and scalable. As an example, we see that hybrid techniques first narrow down the scope by applying selection and then prioritize the selected areas further based on the systems' faced risks and past defective areas, outperforming traditional techniques in terms of fault detection rate and execution time. Moreover, some hybrid techniques adapt test plans by incorporating historical test performance data to the changing nature of software. Thus, these techniques perform outstandingly when it comes to adaptability, change awareness. And overall test optimization to make Hybrid techniques more suitable for modern development practices.

## CONCLUSION AND FUTURE WORK

This research work has touched three aspects of regression testing, the first being the comparative analysis of regression testing techniques to find which creates a balance between effectiveness and efficiency while applying regression testing techniques, revealing that Retest All and Prioritization fall short in this challenge. Hybrid approaches showed stronger overall performance by combining the strengths of individual methods. The second focus has been to discover the practical challenges faced in modern environments by regression testing techniques characterized by rapid code changes, limited resources, and complex systems, which in turn demand adaptability and automated testing techniques. The third research question has been to identify the role of hybrid techniques when applied to agile environments. Although they behave significantly better yet they still face issues like model complexity and data dependency.

**Future work** will focus on enhancing the reliability of AI-driven hybrid techniques, developing lightweight integration frameworks, and expanding empirical validation across diverse software environments.

## REFERENCES

- Zaidman, A. (2024). An Inconvenient Truth in Software Engineering? The Environmental Impact of Testing Open Source Java Projects. 66, 214–218. <https://doi.org/10.1145/3644032.3644461>
- Raju, S. S., & Leong, W. Y. (2025). AI-Driven Automation in Software Testing: Enabling SME Adoption. INTI Journal, 2025(1). <https://doi.org/10.61453/intij.202511>
- Ogheneovo, E. E. (2013). Software Maintenance and Evolution: The Implication for Software Development. West African Journal of Industrial and Academic Research, 7(1), 81-92.
- Harrold, M. J., & Orso, A. (2008, September). Retesting software during development and maintenance. In 2008 Frontiers of Software Maintenance (pp. 99-108). IEEE. (2015). Software Quality Assurance, Testing and Metrics. PHI Learning Private Limited, Delhi, India.
- Saraswat, B. K., Parashar, B., Takuli, A. K., & Singh, S. (2021, August). Case Study for Regression Testing using Genetic Algorithm. In Journal of Physics:

- Conference Series (Vol. 2007, No. 1, p. 012060). IOP Publishing.
- Rahmani, A., Min, J. L., & Maspupah, A. (2021, April). An empirical study of regression testing techniques. In *Journal of Physics: Conference Series* (Vol. 1869, No. 1, p. 012080). IOP Publishing. [4] Yoo, S., & Harman, M. (2012). Regression testing minimization, selection and prioritization: a survey. *Software testing, verification and reliability*, 22(2), 67-120..
- Pan, R., Bagherzadeh, M., Ghaleb, T. A., & Briand, L. (2022). Test case selection and prioritization using machine learning: a systematic literature review. *Empirical Software Engineering*, 27(2), 29.
- Kolawole, I., & Fakokunde, A. (2024). Improving Software Development with Continuous Integration and Deployment for Agile DevOps in Engineering Practices. *International Journal of Computer Applications Technology and Research*, 14(01), 25-39.
- Shivashankar, K., Hajj, G. S. A., & Martini, A. (2025). Scalability and Maintainability Challenges and Solutions in Machine Learning: Systematic Literature Review. arXiv preprint arXiv:2504.11079.
- Pan, R., Bagherzadeh, M., Ghaleb, T. A., & Briand, L. (2022). Test case selection and prioritization using machine learning: a systematic literature review. *Empirical Software Engineering*, 27(2), 29.
- G.Duggal, B.Suri,"Understanding Regression Testing Techniques", COIT, 2008, India
- Elsner, D., Hauer, F., Pretschner, A., & Reimer, S. (2021, July). Empirically evaluating readily available information for regression test optimization in continuous integration. In *Proceedings of the 30th ACM SIGSOFT international symposium on software testing and analysis* (pp. 491-504).
- Yadav, D. K., & Dutta, S. (2020). Regression test case selection and prioritization for object oriented software. *Microsystem Technologies*, 26(5), 1463-1477.
- Mansour, N., & Bahsoon, R. (2002). Reduction-based methods and metrics for selective regression testing. *Information and Software Technology*, 44(7), 431-443.
- Rothermel, G., Untch, R. H., Chu, C., & Harrold, M. J. (2001). Prioritizing test cases for regression testing. *IEEE Transactions on software engineering*, 27(10), 929-948.
- Yoo, S., & Harman, M. (2012). Regression testing minimization, selection and prioritization: a survey. *Software testing, verification and reliability*, 22(2), 67-120.
- Wong, W. E., Horgan, J. R., London, S., & Agrawal, H. (1997, November). A study of effective regression testing in practice. In *PROCEEDINGS The Eighth International Symposium On Software Reliability Engineering* (pp. 264-274). IEEE.
- Chen, J., Lou, Y., Zhang, L., Zhou, J., Wang, X., Hao, D., & Zhang, L. (2018, October). Optimizing test prioritization via test distribution analysis. In *Proceedings of the 2018 26th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering* (pp. 656-667)