ISSN 2278-3091
Volume 9, No.1, January – February 2020
International Journal of Advanced Trends in Computer Science and Engineering
Available Online at http://www.warse.org/IJATCSE/static/pdf/file/ijatcse105912020.pdf
https://doi.org/10.30534/ijatcse/2020/105912020

Regression Testing: Analysis of its Techniques for Test Effectiveness

Omdev Dahiya[1], Kamna Solanki[2], Sandeep Dalal[3], Amita Dhankhar[4]
[1,2,4]University Institute of Engineering and Technology, Maharshi Dayanand University, Rohtak-124001, India
[3]Department of computer science and applications, Maharshi Dayanand University, Rohtak-124001, India
[1]Omdahiya21792@gmail.com, [2]Kamna.mdurohtak@gmail.com,
[3]Sandeepdalal.80@gmail.com, [4]Amita.infotech@gmail.com

ABSTRACT

Software Testing is one of the phases in the lifecycle of software development that consumes plenty of effort and time. Once the software is delivered to the users, it moves to the maintenance phase. In the maintenance phase, it becomes excessively pricy to run all test cases available within the regression test suite to confirm the correct working of the software application or program in terms of stipulated time and allocated budget. Even if a minor amendment is made to the program, it becomes pricy if we have to run all available test cases to verify that no new errors are introduced because of the amendment made. Amongst the developed techniques for regression testing, Test case prioritization is the most popular, as no test cases are deleted in it like the other two approaches. With time, the test suite size keeps on increasing rapidly, so it is neither practically possible nor economically feasible to run all the test suite. So, prioritization of the test cases becomes a mandatory action. Prioritization strategies facilitate in overcoming these issues by prioritizing the test cases, so many parameters like fault detection rate and statement or path coverage, etc. are maximized. This paper has summarized the research findings of numerous researchers in the regression testing field.

Key words: Software, Software Testing, Regression testing, and its techniques, Systematic study.

.

1. INTRODUCTION

Software refers to a collection of related programs, its operating system procedure along with its documentation. Generally, software code is written by humans, and humans can make mistakes so, it can be said that software has bugs, though it is often unavoidable. Mistakes can be made by developers while writing the code, and when the assumptions which are valid in the past were changed, then the code which was once executing properly will no longer hold good. In Software testing, the developed system is executed to check if it is functioning as desired [1]. In the process of development of software, much time is utilized in its testing so that it may be termed as the costliest phase. It is performed to ensure whether quality standards are met by the developed software or not [2]. If due care is not taken while testing the software, then unexpected things can happen. Depending on the type of software, these unexpected things vary from image of the developer company being tarnished to customers getting angry and even people being killed.

2. REGRESSION TESTING AND ITS TECHNIQUES

"Regression testing is selective retesting of a system or component to verify that modifications have not caused undesirable effects and that the system or component still complies with its specified requirements" [3-4]. With the evolution of software concerning the time, the size of the software test suite also increases, making it very costly to execute. It has been proven by numerous researchers that regression testing is a costly process, and most of the aggregate expenditure of the software is consumed by it [5-6]. Yoo and Harman examined various regression testing approaches and have elaborated regression testing approaches as- minimization, selection, and prioritization [7]. Regression test case selection selects those test cases which focus on the modified part of the software. Instead of removing the test cases, they are generally filtered. Test case minimization reduces the number of test cases by eliminating certain test cases and lays stress on using a subset of the test suite from an economic point of view. This sometimes decreases the rate of fault detection. In the test case prioritization elimination of test cases is not performed; instead, it arranges them according to priority, and they are thus executed with higher priority cases first, followed by the lower priority ones.

3. RELATED STUDIES

This section discusses the research work of different researchers in the regression testing field. The need for this review is to provide meaningful insight into the different aspects of regression testing techniques.

Suri and Singhal have implemented ant colony optimization for test case selection and prioritization, showing how all the possible paths are explored. They have concluded that results obtained by them are close to the optimal solution [9]. Parsa and khalilian have proposed an optimization technique for test suite minimization. They have experimented with showing that the proposed algorithm has significant fault detection capacity [10]. Mayan and Ravi have proposed a hybrid algorithm for test sequence selection in regression testing. They have shown in their results the worthiness of their algorithm in terms of time-bound execution and improving the testing effectiveness [11]. Dahiya *et al.* have performed a comparative analysis of different techniques for test case prioritization. They have shown that two approaches cannot be compared based on limiting factors, and every developed approach has shown results according to software size, requirement, and execution environment [12].

Img.Source
https://www.sketchbubble.com/en/presentation-regression-testing.html [8]

Figure 1: Picture depicting the use of regression testing

Kandil *et al.* have proposed an approach to enhance test case selection and prioritization for agile regression testing. They have proved how much their approach is effective via APFD (average percentage of faults detected) metric [13]. Binkley has proposed a technique to reduce the cost of regression testing by test case selection based on various parameters and have proved the efficiency of their technique [14]. Dahiya and Solanki have conducted a systematic literature study of approaches for regression test case prioritization. They have deduced that although many techniques are there, the current scope lies in developing a requirement-based test case prioritization technique based on a nature-inspired algorithm [15]. Suri and Singhal have proposed a tool-based approach for test case selection and prioritization by ant colony optimization to reduce the effort and time required for prioritizing the test cases. They have demonstrated their applicability in real-life situations [16].

Yoo and Harman proposed a hybrid algorithm for test suite minimization. They have proved how a multi-objective approach can make the testing process more efficient [17]. Dahiya and Solanki have performed a comprehensive review of regression test case prioritization approaches. They have shown that although many approaches for regression testing have been developed by numerous researchers but new and better results can be obtained by combining various approaches [18]. Huang *et al.* have proposed a cost-based method for prioritizing the test cases with the help of historical records. They have shown the effectiveness of their approach in terms of enhanced fault detection via some controlled experiments [19]. Muthusamy and Seetharaman have proposed a new technique for test case prioritization, which is based on the number of factors. They have shown their results via the APFD metric, and experimental findings have shown the effectiveness of their approach in terms of enhanced fault detection rate [20].

Li *et al.* have performed a study to compare the performance of search algorithms and the performance comparison is made by APRCI (Average Percentage of Requirement Coverage Improved) metric [21]. Yoon *et al.* have proposed a risk and requirements-based approach for test case prioritization. They have performed a comparative analysis of their approach with other existing approaches and have shown the effectiveness of their method by values of the APFD metric [22]. Wang *et al.* proposed a method for test case prioritization based on the severity of faults. They have experimented, and from the results obtained, they have shown the efficiency of their method [23].

Khalilian *et al.* have presented an enhanced technique of their previously proposed technique for test case prioritization based on historical information. They have compared their present technique with their previous technique and also with another technique given by Kim and Porter [24]. The results deduced by them have shown the effectiveness of their technique [25]. Nardo *et al.* have presented a case study of

coverage-based techniques for regression testing in a real-world industry system. They have deduced their results using various factors and have shown which technique is more efficient [26].

Srikanth *et al.* have proposed a requirement-based test case prioritization technique using risk factors as a criterion on the industry level system. They have shown that their approach performs better than existing approaches, and it can be used in industry to address budget and schedule constraints at the testing phase [27]. Elbaum *et al.* have conducted a study in which they have presented various factors under which a test case prioritization technique performs well or not. They said that this would help researchers in selecting an appropriate technique [28].

To summarize the different regression testing techniques covered by various authors, a list has been made in Microsoft word in which all the articles covered in this paper for the literature review are included. It is then shown in figure 2. Then to get a high-level picture of the covered/proposed techniques or areas in regression testing, a word cloud of the paper titles has been generated with the help of an online application at - https://www.wordclouds.com/ [29], which is shown in fig. 3. The size of each word as appearing in the cloud shows the frequency in which that particular word is present in the titles of the included papers. From the word cloud, it can be observed that some techniques are studied more as compared to others.

Regression testing is not always possible due to the lack of resources of enough time to execute the testing procedure. In that scenario testing team only inspects those modules of the software system where amendments have been made. They bypass a full regression testing procedure. This is known as Non-regression testing. In this testing technique, it is assumed that the modifications made in the code have not affected the modules which were previously functioning correctly. In this way, time and resources were saved by the testers. Regression testing can also include non-regression testing [30]. Effective testing will eventually result in good quality software [31-34].

While the researchers do regression testing, it is also necessary for the newcomers or research scholars to know the regression testing tools. So, the following section will discuss the popular tools for regression testing.

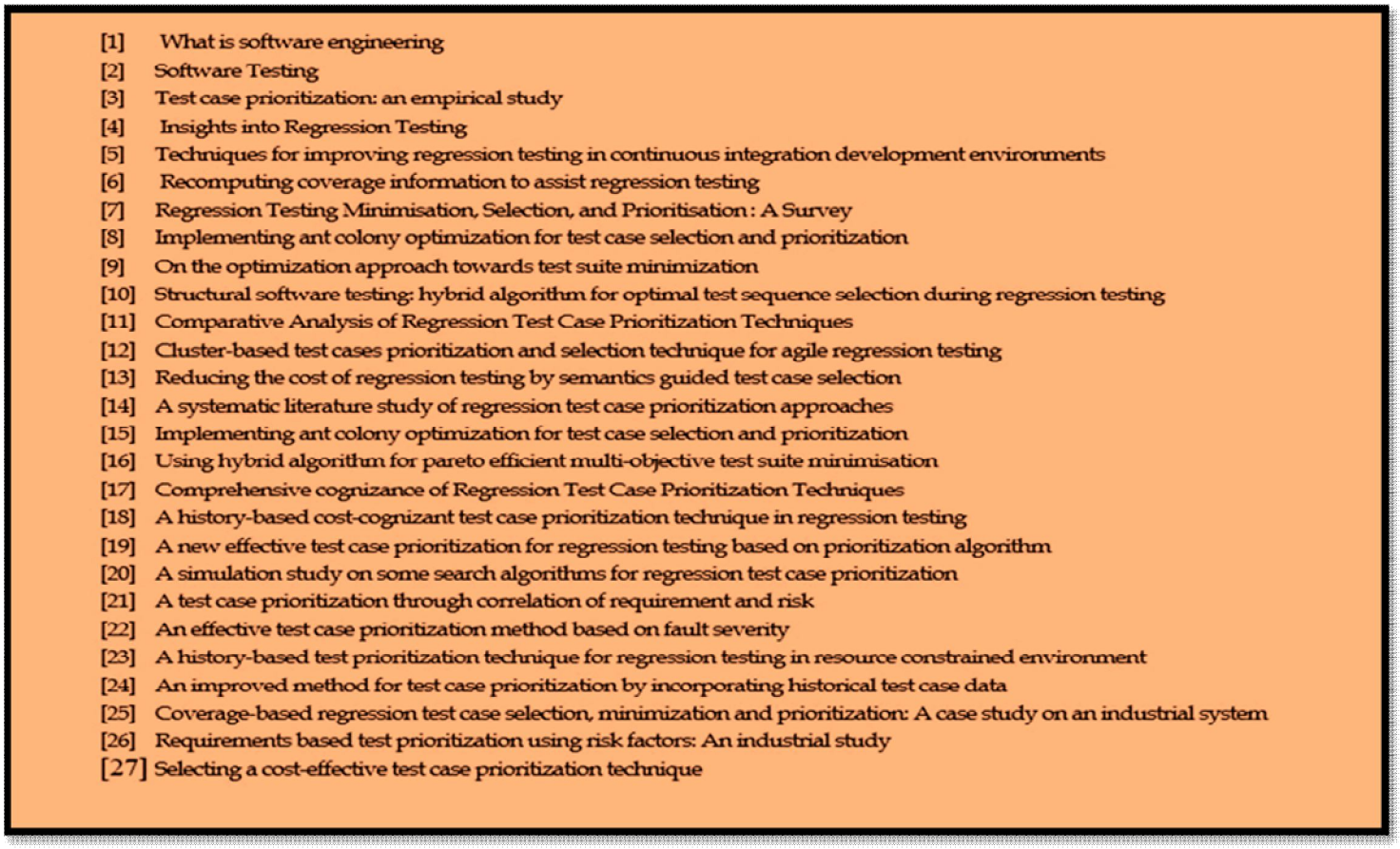


Figure 2: Titles of papers used for making a word cloud

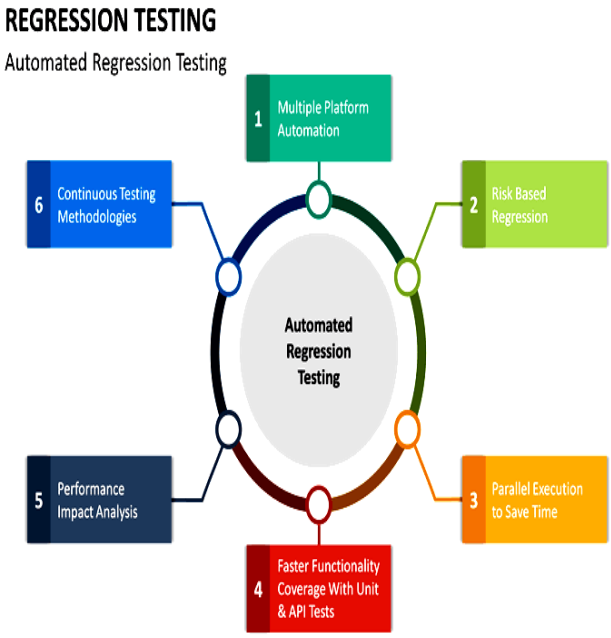
Figure 3 : Word cloud made by using paper titles via online application at- https://www.wordclouds.com/ [29]

4. SIGNIFICANCE OF REGRESSION TESTING AND ITS TESTING TOOLS

Regression testing is a challenging task due to various factors such as the size of the test suite keeps on increasing gradually, time and resource constraints are there, and testers have to work by maintaining the balance between all the factors. Regression testing is performed using various automation tools. The automated regression testing process is illustrated in figure 4.

As changes were made in the software frequently to improve its quality, it will eventually increase the regression testing cost [39-41]. In this scenario, if test cases are executed manually, it will affect the execution time and cost. So, the automation of regression test cases will be an excellent choice to make. The steps incurred in this process are presented in the above figure.

The significance of regression testing lies in its approaches, which helps to ease the tasks of testers due to their unique features. There are various tools available for regression testing [36]. Some of them are documented in figure 5.



Img. Source- https://www.sketchbubble.com/en/presentation-regression-testing.html [35]

Figure 4: Automated Regression Testing

In this figure, popular regression testing tools are listed, which immensely helps in executing test cases quickly, thus saving much time for the testers. Software testing is an activity that keeps ongoing. However, there must be some criteria where the testing activity should stop. The next section will discuss when to stop the testing activity.

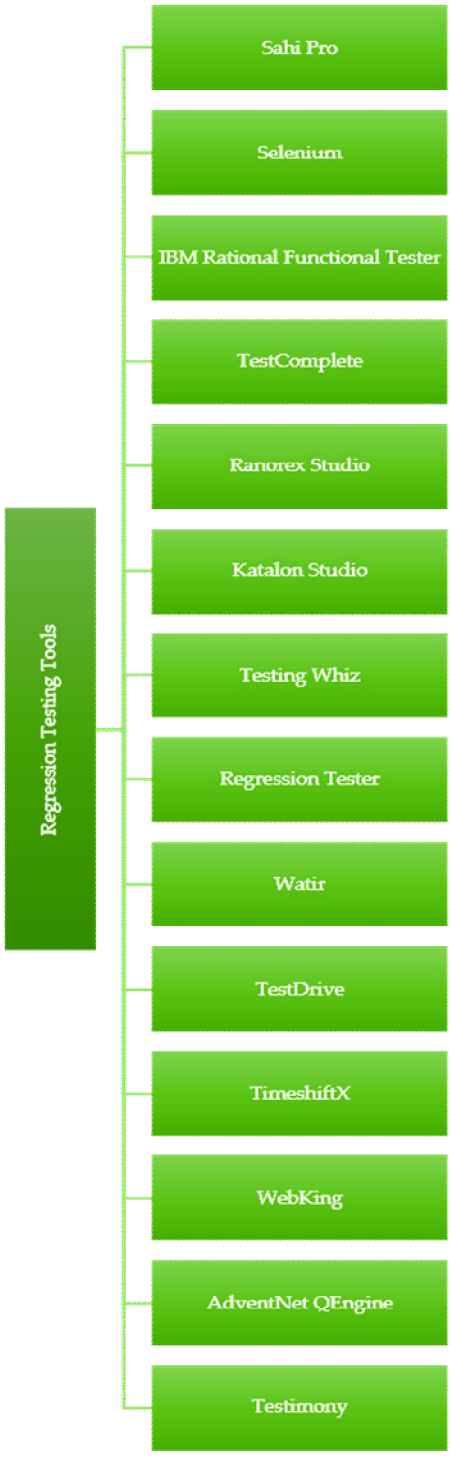
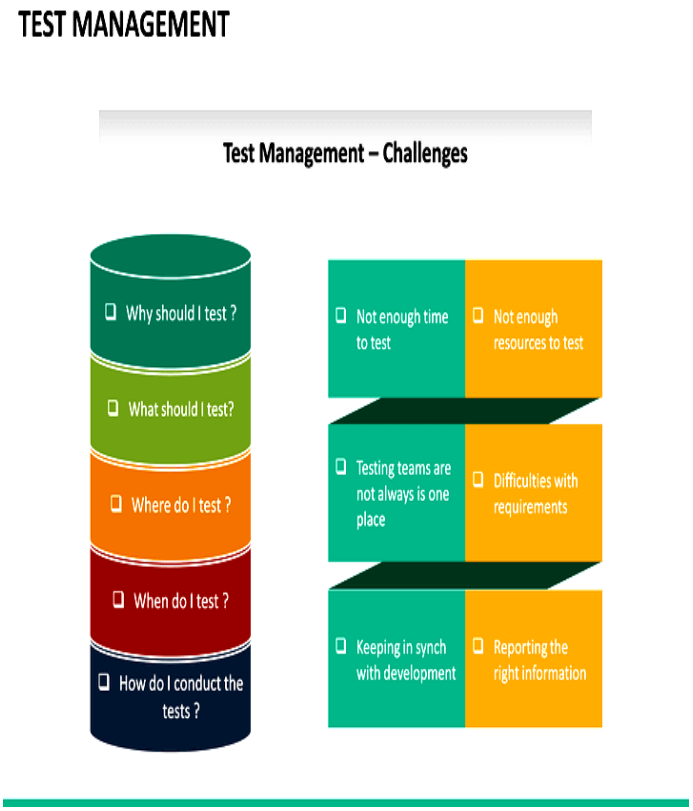


Figure 5: Tools for Regression Testing

5. TEST STOPPAGE CRITERIA

It can be quite challenging to determine when testing activity should be stopped. As most software systems are quite complex, and they are executed in an independent environment, so complete testing is not practically possible owing to various constraints such as limited time and budget. Certain factors help in deciding when to stop testing. Some of them are listed below-

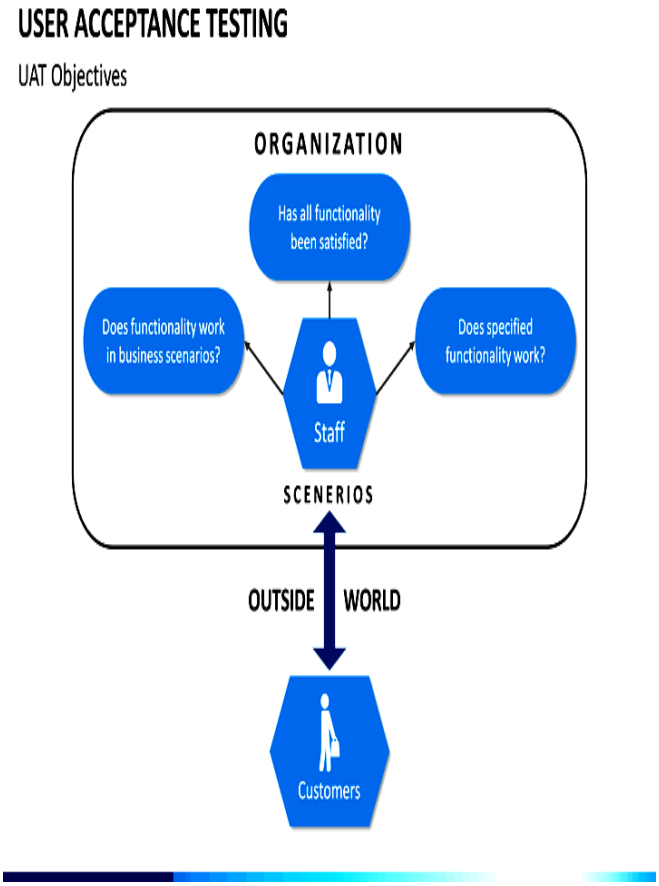
- Deadlines are met, whether it may be testing deadlines, release deadlines, etc.
- The finding of errors falls below a certain level.
- The budget allocated for testing depleted.
- Test cases completed execution with a specific pass percentage.
- Coverage of functionalities/requirements/code has reached a specific point.
- When the developers and testers have reached an acceptable level of risk.
- It is ensured that all the critical test cases are passed.
- High priority defects are identified and fixed.



Img. Source
https://www.sketchbubble.com/en/presentation-test-management.html [37]

Figure 6: Pictorial Representation of Test Management Challenges

It is not practically and economically feasible to do enough testing, but defining how much testing is to be performed so that it can be said; it is enough; it is complicated. Though, before test execution, it is planned what to test and test cases were designed for it but still, testers are not sure about whether how much should be planned. There are numerous challenges in test management, which are explained in figure 6.



Img. Source- https://www.sketchbubble.com/en/presentation-user-acceptance-testing.html [38]

Figure 7: Picture depicting User Acceptance Testing objectives

Defining the specific test stoppage criteria is not possible as it is a vast area with many endless possibilities. A client can also call off the testing if the developer team can assure him how the developed software system is robust by proving the efficiency of the system by performing various testing techniques on it, such as stress testing. For it, the organizational staff will communicate amongst themselves to test almost all aspects of the software system ranging from its performance in favorable circumstances to the unfavorable ones. For its alpha and beta testing will also play a vital role. Some of the objectives of user acceptance testing are shown in figure 7 above.

Whether the performed testing is enough can only be confirmed by analyzing the results of testing. If the test cases were able to discover a considerable number of faults, then testers generally plan a few more execution of test cases to assure that the required level of software quality is achieved. On the contrary, if very few faults are found then (provided that the designed test cases have a good potential of uncovering maximum faults), no more tests will be required.

Why testing is not stopped and is termed as a continuous process even when it is proved that the software system is working as desired because it is not possible to prove without performing exhaustive testing (which is not feasible for real systems).

So, testing can be stopped if it has achieved the points mentioned above and has reached a particular threshold satisfying all the conditions as well as developers, testers, and clients.

6. CONCLUSION

Regression testing is a testing activity conducted to check whether no new faults have crept in after modifications were done in the source code. It is an expensive and time-consuming activity, so various researchers have done their work to make it an effective process. This paper has presented the work of numerous authors on different approaches for regression testing. The researchers have developed various techniques for test suite optimization. They have developed algorithms to make regression testing efficient. From the analysis of the selected studies included in the review part of this paper and after making the word cloud, it has been observed that a considerable amount of work is done on the test case prioritization approach for regression testing. This study has also discussed the significance of regression testing in the software testing field, along with the most widely used regression testing tools. What could be the test stoppage criteria is also discussed with a considerable number of points in this study. So, this study believes there is future scope, which can be achieved by combining various approaches for test case prioritization so that new and better results can be obtained and the testing process be termed as an effective one.

REFERENCES

1. Sommerville, Ian "***What is software engineering***?". Software Engineering (8th ed.). Harlow, England: Pearson Education p.p. 7, 2007.
2. Gordon Fraser and José Miguel Rojas, "***Software Testing***."
3. G. Rothermel, R. H. Untch, C. C. Chu, and M. J. Harrold, **"Test case prioritization: an empirical study,"** *in Software Maintenance. (ICSM '99) Proceedings. IEEE International Conference on software maintenance*, pp.179–188, 1999.

https://doi.org/10.1109/ICSM.1999.792604

4. H. K. N. Leung, **"Insights into Regression Testing,"** *Proceedings of the International Conference on Software Maintenance*, pp. 60–69, 1989.
5. S. Elbaum, G. Rothermel, and J. Penix**, "Techniques for improving regression testing in continuous integration development environments,"** *in Proceedings of the 22nd ACM SIGSOFT International Symposium on Foundations of Software Engineering* - pp. 235–245, 2014.
6. P. K. Chittimalli and M. J. Harrold, **"Recomputing coverage information to assist regression testing,"** *IEEE Transactions on Software Engineering*, vol. 35, no. 4, pp. 452–469, 2009.
7. S. Yoo and M. Harman, **"Regression Testing Minimisation, Selection, and Prioritisation : A Survey,"** Software Testing, Verification and Reliability, 22(2), pp.67-120, 2007. https://doi.org/10.1002/stv.430
8. https://www.sketchbubble.com/en/presentation-regression testing.html
9. B. Suri and S. Singhal. **"Implementing ant colony optimization for test case selection and prioritization."** International journal on computer science and engineering 3, no. 5 (2011): 1924-1932.
10. S. Parsa and A. Khalilian**. "On the optimization approach towards test suite minimization."** International Journal of Software Engineering and its applications 4, no. 1 (2010): 15-28.
11. J. A. Mayan and T. Ravi. **"Structural software testing: hybrid algorithm for optimal test sequence selection during regression testing."** International Journal of Engineering and Technology (IJET) 7, no. 1 (2015).
12. O. Dahiya, K. Solanki and S. dalal, **"Comparative Analysis of Regression Test Case Prioritization Techniques,"** International journal of advanced trends in computer science and engineering, Volume 8 No. 4, pp. 1521-1531, 2019.
13. P. Kandil, S. Moussa, and N. Badr**. "Cluster□based test cases prioritization and selection technique for agile regression testing."** Journal of Software: Evolution and Process 29, no. 6 (2017): e1794.
14. D. Binkley, **"Reducing the cost of regression testing by semantics guided test case selection."** *In Proceedings of International Conference on Software Maintenance*, pp. 251-260. IEEE, 1995.
15. O. Dahiya and K. Solanki**, "A systematic literature study of regression test case prioritization approaches."** International Journal of Engineering & Technology, Vol. 7, No. 4, pp.2184-2191, 2018.
16. **B.** Suri and S. Singhal. **"Implementing ant colony optimization for test case selection and prioritization."** International journal on computer science and engineering 3, no. 5 (2011): 1924-1932.
17. S. Yoo and M. Harman. **"Using hybrid algorithm for pareto efficient multi-objective test suite minimisation."** Journal of Systems and Software 83, no. 4 (2010): 689-701.
18. O. Dahiya and K. Solanki, **"Comprehensive cognizance of Regression Test Case Prioritization Techniques,"** International journal of emerging trends in engineering research, Volume 7 No. 11, pp. 638-646, 2019.
19. Y. C. Huang, K. L. Peng, and C. Y. Huang. **"A history-based cost-cognizant test case prioritization technique in regression testing."** Journal of Systems and Software 85, no. 3 (2012): 626-637. https://doi.org/10.1016/j.jss.2011.09.063
20. T. Muthusamy and K. Seetharaman**. "A new effective test case prioritization for regression testing based on prioritization algorithm."** Int. J. Appl. Inf. Syst. (IJAIS) 6, no. 7 (2014): 21-26.
21. S. Li, N. Bian, Z. Chen, D. You, and Y. He. **"A simulation study on some search algorithms for regression test case prioritization."** *In 2010 10th International Conference on Quality Software*, pp. 72-81. IEEE, 2010.
22. M. Yoon, E. Lee, M. Song, and B. Choi. **"A test case prioritization through correlation of requirement and risk."** Journal of Software Engineering and Applications 5, no. 10 (2012): 823.
23. Y. Wang, X. Zhao, and X. Ding**. "An effective test case prioritization method based on fault severity."** *In 2015 6th IEEE International Conference on Software Engineering and Service Science (ICSESS)*, pp. 737-741. IEEE, 2015.
24. J. M. Kim and A. Porter, **"A history-based test prioritization technique for regression testing in resource constrained environment,"** *in Proc. of the 24th Int'l Conf. Soft. Eng.*, 2002, pp. 119–129.
25. A. Khalilian, M. A. Azgomi, and Y. Fazlalizadeh**. "An improved method for test case prioritization by incorporating historical test case data."** Science of Computer Programming 78, no. 1 (2012): 93-116.
26. D. Di Nardo, N. Alshahwan, L. Briand, and Y. Labiche. **"Coverage□based regression test case selection, minimization, and prioritization: A case study on an industrial system."** Software Testing, Verification, and Reliability 25, no. 4 (2015): 371-396.
27. H. Srikanth, C. Hettiarachchi, and H. Do. **"Requirements based test prioritization using risk factors: An industrial study."** Information and Software Technology 69 (2016): 71-83.
28. S. Elbaum, G. Rothermel, S. Kanduri, and A. G. Malishevsky. **"Selecting a cost-effective test case prioritization technique."** Software Quality Journal 12, no. 3 (2004): 185-210.
29. https://www.wordclouds.com/
30. https://blog.qatestlab.com/2011/03/20/non-regression-testing-what-is-it/
31. K. Solanki, and Y. Singh**, "Importance of Selecting Test Cases for Regression Testing,"** IOSR Journal of

Computer Engineering (IOSRJCE) e-ISSN, pp. 2278-0661, 2014.

32. K. Solanki, and Y. Singh, **"Novel Classification of Test Case Prioritization Techniques,"** International Journal of Computer Applications, Vol. 975, pp. 8887, 2014.
33. K. Solanki, Y. Singh, S. Dalal, and P.R. Srivastava**, "Test case prioritization: an approach based on modified ant colony optimization."** *In Emerging Research in Computing, Information, Communication and Applications*, Springer, Singapore, pp. 213-223, 2016.
34. K. Solanki, Y. Singh and S. Dalal, **"A Comparative Evaluation of "m-ACO" Technique for Test Suite Prioritization,"** Indian Journal of science and technology, Vol. 9, No. 30, pp.1-10, Aug. 2016. https://doi.org/10.17485/ijst/2016/v9i30/86423
35. https://www.sketchbubble.com/en/presentation-regression-testing.html
36. https://www.softwaretestinghelp.com/regression-testing-tools/
37. https://www.sketchbubble.com/en/presentation-test-management.html
38. https://www.sketchbubble.com/en/presentation-user-acceptance-testing.html
39. R. Mukherjee and K. S. Patnaik, **"A survey on different approaches for software test case prioritization,"** Journal of King Saud University-Computer and Information Sciences, 2018.
40. M. Khatibsyarbini, M. A. Isa, D. N. Jawawi, and R. Tumeng, **"Test case prioritization approaches in regression testing: A systematic literature review."** Information and Software Technology, 93, pp.74-93, 2018. https://doi.org/10.1016/j.infsof.2017.08.014
41. O. Dahiya and K. Solanki, S. Dalal, A. Dhankhar, **"An Exploratory Retrospective Assessment on the Usage of Bio-Inspired Computing Algorithms for Optimization,"** International journal of emerging trends in engineering research, Vol. 8 No. 2, 2020. *[Accepted] [In Press]*.