

# Regression Testing in Agile: Concepts, Strategies and Challenges

Sarika Chaudhary

Department of computer science and engineering, Amity University

Email: sarikacse23@gmail.com

**Abstract-** In agile development environment regression testing support development team focus on added functionality, while keeping up stability with each new release. Developers use regression testing to ensure that deliverable is in alignment with new functionality to avoid reworks and also to maintain the customer's requirement. There can be dependencies between recently included and existing features, it becomes mandatory to verify and validate that the updated version conforms to the older one and that the unchanged part is not influenced in either way. In agile development, since there are many sprints, regression testing turns out to be progressively significant as there are persistent changes that are added to the application.

**Index Terms-** Automated testing, regression testing, agile methodology, test case prioritization, sprint.

## 1. INTRODUCTION

In this era of technology software's are developing every day. Software once developed undergoes changes during maintenance phase as customer requirements are changing. So they have to be refreshed with each modification. It is necessary for the team to validate all the previous functionality and new features, are working in congruity with the fixes in the updated version. This reason provides a base for the importance of regression testing for software's that has to be updated and maintained in due course of time [1].

In agile, new build are released at every cycle that result in addition of various changes in application. So testers should generate new test cases with respect to each sprint and test them accordingly to ensure that new errors have not been introduced and if so they did not affect the functionality of the system. So regression testing becomes essential in agile development environment [4]

The main focus of testing is to test each and every part of the software, to ensure the high quality of the product. In this way, for any group, it will be imperative to execute test cases and guarantee that there are no reactions in the wake of including new upgrades or highlights. Executing regression testing in an agile domain helps in the accompanying ways:

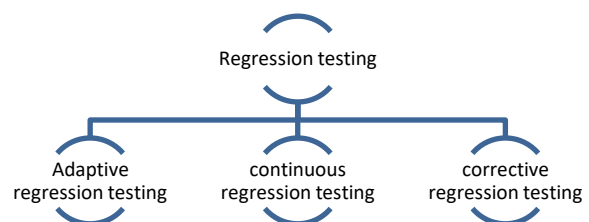
- a) Improved Stability of the software.
- b) Early identification and removal of defects.
- c) Improved quality of the software by ensuring the continuity of various functions.
- d) Reduced rework in programming.



Fig. 1. Regression testing in agile

## 2. BASIC CONCEPTS

Regression testing is an activity of re-executing test cases seamlessly to find out the bugs introduced after the software experience change. Regression test suites is defined as collection of previous test cases and newly generated test cases. In agile after successive releases of sprint the size of test suite keeps on increasing, thus making it inefficient for re-execution of all test cases. This result in increased cost, effort and time for the deliverable. Due to these reasons different techniques are advised by the researchers to handle the test suite growth. They are test suite reduction, test case selection and test case prioritization. Regression testing can carried out in three ways [2]:



a) Adaptive regression testing: When there are changes in execution environment, it might possible that new errors introduced. Re execution of the test cases generated after this is known as adaptive regression testing.

2) Continuous regression testing: Process of re execution of test cases which are more likely to figure out faults that arise after every modification in requirements.

3) Corrective regression testing: In this type of testing error introduced after few modifications in code are uncovered [2].

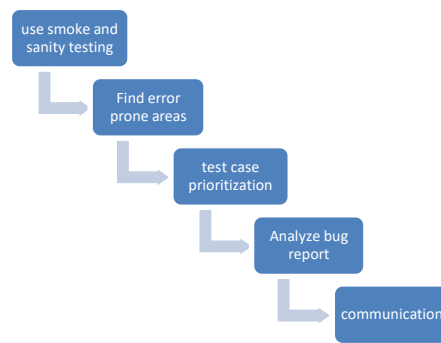


Fig.2. Regression testing strategy

### 3. RESEARCH METHODOLOGY

In agile, developers work in short and quick sprints where each sprint focus on short requirements. So we need to pay extensive attention on documentation as each sprint is time constrained [3]. Also we don't have to generate long test plan but a high level of testing strategy as guidelines should be structured for the testers. Following are the questions designed for this paper:

Q1: What are the conditions that must be satisfied before structuring a RTS (regression testing strategy)?

Q2: What are the phases in building a RTS?

Q3: What considerations should be kept in mind while implementing RTS?

### 4. RESULTS

In this section findings from various selected papers are presented. Results are organized according to the following question discussed in above section.

4.1 Conditions that must be satisfied before structuring a regression testing strategy:

- a) Enhancements should be identified.
- b) Test cases expected to be executed should be gathered.
- c) Execution time should be estimated for test cases.
- d) Figure out what and how can be automated.

4.2 Various phases in building RTS.

To effectively carry out regression testing in agile, proper testing strategy should be considered. Following figure depicts the phases that should be followed.

a) *Smoke and sanity testing*: It is carried out before regression testing. Prior to the testing of updated version, it thoroughly test the basic functionality of the older version, to validate that features works as planned. This saves the time [7].

b) *Figure out error-prone areas*: Test cases that fail frequently are considered in this phase. In this phase focus is on finding out the areas in software that are more prone to errors e.g. Test cases which fail even after a little bit of modification. So the idea is to include those test cases in regression testing.

c) *Test case prioritization*: In this phase priority of test cases are assigned by testing teams. Team selects test cases that are most affected after modification in the software. Test case with higher priority are executed first. Prioritizing test cases reduce the test suite and maintenance time and speed up the regression testing process [2][8].

d) *Error report analysis*: It is always a good idea to integrate error tracking tools with testing tools, so that we can analyze during regression testing what kind of data is affected and what is the impact of this on rest of the code. With error tracking tools identification of bugs is easy and it also helps in effective debugging.

e) *Communication*: In this phase testing team monitors and rate the changes required by the customer. Team communicate continuously to figure out modification and do the needful and customer communicate with team so that they can understand the change in every iteration. .

4.3 Major considerations

a) *Exhaustive code coverage is not feasible*.

Always keep in mind that 100% of code coverage is not possible because only 70-90% of test case are efficient, 10-30% of test cases finds bugs only once and they are not effective further. So it is a total waste of resources to go for 100% testing [5].

b) *Run manual test cases before automated test cases*

It is always advisable to write test cases manually at first because if we start with an automated test case for a functionality without having a prototype of that volatile functionality, it may result in waste of effort and time. Also we can compare test result if both manual and automated run.

c) *Automation only for medium and large scale projects.*

It is advisable to automate the testing process for medium and large scale project and not for small scale projects because it may not worth to indulge so much effort, time and cost in small scale projects. In large scale system multiple sub systems are there which require efficient testing to be carry out in order to conform the requirement and also if something went wrong they results may be catastrophic.

## 5. CHALLENGES IN AUTOMATING REGRESSION

With regression testing test suite size keeps on increasing with time. So test team should limit the test cases such that necessary test case should be executed and obsolete test cases should be removed [6]. Testers should aware of obstacles that can increase automation efforts:

- a) *Validity and Maintenance:* Regression test case are not valid for long time as updates occur frequently. So maintenance is major challenge as keeping obsolete and non-obsolete test cases result in increased time and effort.
- b) *Positive zero:* it means even when the software has no faults, testing results a failure report. The reason for false positive (positive zero) can be anything like timing constraints, obsolete test cases, inefficiently designed test cases that provide inconsistent results [9].

## 6. CHALLENGES ENCOUNTER BY AGILE TEAMS

Various challenges listed below are faced by regression testing team in an agile development environment:

- a) *Lack of expertise:* with the advancement in project, more experts will be required for detailed levels of testing like performance and integration testing. Team should contact specialists within the organization to analyze and gather requirements in order to effectively plan test cases.

- b) *Frequent change in requirements:* Sometimes customers will go for continuous change in requirements. This results in volatile test cases which ultimately roll out the complete iteration. This could be a severe threat to any testing method [9][10].

- c) *Faster test suite growth:* with the release of each sprint, number of test suite also grows. In large scale project it becomes unmanageable to maintain such large test suite. To ensure the good maintenance obsolete test cases should be removed at frequent times [5].

- d) *Automation of test cases:* Automation of test cases should not be followed as mandatory approach, because of more maintenance required. Whenever and wherever necessary test cases can be automated.

- e) *Insufficient communication among stakeholders:* In order to have good understanding of the change occurring in system communication between stakeholders should be seamless. Respective stakeholder should aware of what is going on in other departments.

## 7. CONCLUSION AND FUTURE SCOPE

With the expanding requirements and needs and reduced time of product release it becomes very critical to assure the quality of the product and guarantee that the items you deliver conforms the expectation for the clients and in the meantime is steady and viable. Executing regression testing in a coordinated manner (agile) will permit you convey quality items to the market on schedule. In agile regression testing can be automated by keeping in mind various challenges and considerations discussed.

## REFERENCES

- [1] Williams, L. (2010): Agile Software Development Methodologies and Practices, Advances in Computers, Vol. 80, pp.1-44.
- [2] Chaudhary, S. , (2018): Findings and implications of test case prioritization techniques for regression testing, International Journal of Technical Innovation in Modern, vol. 4, no. 5, pp. 1259-1266.
- [3] Williams, L. (2010): Agile Software Development Methodologies and Practices, Advances in Computers, Vol. 80, pp.1-44.
- [4] R. a. G. O. S. a. R. G. Rosero Miranda, (2016):15 Years of Software Regression Testing Techniques --A Survey," International Journal of Software Engineering and Knowledge Engineering, pp. 675-689.

- [5] Pravin, A., Srinivasan, S.,(2013): Effective test case selection and prioritization in regression testing,J. Comput. Sci., Vol. 9 Issue:5, pp. 654–659.
- [6] Chauhan, R., Batra, P., & Chaudhary, S., (2014): An efficient approach for test suite reduction using density based clustering technique. International Journal of Computer Applications, Vol.97 Issue:11.
- [7] Verma, A., Khatana, A., Chaudhary, .,(2017) “A Comparative Study of Black Box Testing and White Box Testing,” International Journal of Computer Sciences and Engineering, vol. 5, no. 12, pp.301-304, 2017.
- [8] Inayat, I., Salim, S.S., Marczak, S., Daneva, M. and Shamshirband,S., (2015): A Systematic Literature Review on Agile Requirements Engineering Practices and Challenges. Computers in Human Behavior, Vol. 51, pp.915-929.
- [9] Baruah, N., (2015): Requirement Management in Agile Software Environment. Procedia Computer Science, Vol. 62, pp.81-83.
- [10] Ramesh, B., Cao, L. and Baskerville, R.,2010 :Agile Requirements Engineering Practices and Challenges:An Empirical Study. Formation Systems Journal, Vol. 20, pp.449-480.