



ISSN: 0976-3031

Available Online at <http://www.recentscientific.com>

CODEN: IJRSFP (USA)

*International Journal of Recent Scientific Research*  
Vol. 10, Issue, 06(F), pp. 33098-33101, June, 2019

**International Journal of  
Recent Scientific  
Research**

DOI: 10.24327/IJRSR

## Research Article

### OPTIMIZING THE REGRESSION TESTING

Dr. Shiv Kumar Goel<sup>1</sup> and Nikita Rajendra Patil<sup>2</sup>

<sup>1</sup>Deputy Head of Master of Computer Application (MCA) Department, Vivekanand Education Society's Institute of Technology (VESIT), Chembur Mumbai, Maharashtra

<sup>2</sup>Master of Computer Application Vivekanand Education Society's Institute Of Technology (VESIT), Chembur Mumbai, Maharashtra

DOI: <http://dx.doi.org/10.24327/ijrsr.2019.1006.3605>

#### ARTICLE INFO

##### Article History:

Received 4<sup>th</sup> March, 2019  
Received in revised form 25<sup>th</sup>  
April, 2019  
Accepted 18<sup>th</sup> May, 2019  
Published online 28<sup>th</sup> June, 2019

#### ABSTRACT

Regression testing is a crucial process in testing of an application which is a time and cost consuming activity and hence it is necessary for an organization to discover the challenges of regression testing and establish the solutions to optimize the regression testing.

##### Key Words:

Regression testing, automation, framework, ROI, Business value, test suite, test cases, test plan, test strategy

Copyright © Dr. Shiv Kumar Goel and Nikita Rajendra Patil, 2019, this is an open-access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution and reproduction in any medium, provided the original work is properly cited.

#### INTRODUCTION

Regression testing is the retesting of the parts of the application which are not revised to ensure that they are not negatively affected by the changes done to other parts of the application.

Test suites for regression testing are often maintained for the whole lifetime of an application, even after the original development process.

Few problems are: regression testing is easy to define but difficult to understand when performing it since it's dynamic in nature, as the functionalities of the product expand regression testing becomes more complex, regression testing is time consuming and costly at times.

Automating the regression testing, maintenance, prioritizing test cases, periodic cleanup help reducing the problems and optimizing the regression testing process.

##### Challenges with regression testing and ways to optimize

The challenges with regression testing can be broadly classified as:

- Time consuming
- Complex

- Business value
- Identifying and maintaining the regression test suite
- Expensive
- Workforce

##### Time Consuming

Regression testing involves testing of the test cases that already exist. When regression testing is to be performed the testers can be tempted to perform exhaustive testing, that is, utilizing the whole time to test all modules. The best approach is to consider only those parts or modules of the application which can be affected by the changes to the application or the testing team should devise a proper plan so that the regression is completed in a stipulated time.

The time required for regression testing can be reduced by simplifying the tests in the test suite. This can be done by examining the changes done and finding out the impact done on modular and functional level and based on this impact the related test cases can be assembled and executed.

The time can also be reduced by executing the test suites that test only the changed part of the application and removing obsolete test cases. Automating the test cases and executing the

\*Corresponding author: Dr. Shiv Kumar Goel

Deputy Head of Master of Computer Application (MCA) Department, Vivekanand Education Society's Institute of Technology (VESIT), Chembur Mumbai, Maharashtra

smoke and sanity testing can save a lot of time and efforts. Smoke testing involves early testing of the application to understand if the application is working fine without deeper testing whereas sanity testing involves testing certain areas of application.

**Complex**

As the functionality is added to the application the testing becomes more complex.

The emphasis should be given on the most used parts of the application which includes the common functionalities. Risk-based testing is an approach that can be used to reduce the complexity of an application. This testing is testing of an application with an emphasis on risks. It prioritizes the tests based on the risk of failures and its impact.

It is important to examine the testing requirements in advance and a requirement traceability matrix shows whether there is enough of test coverage for important functionalities.

**Business value**

Communicating and explaining the business value of regression testing and its ROI to the non-technical people within the business can be difficult. Executives focus on the ROI but since regression testing involves focusing on long term profits it becomes difficult to prove the ROI during each cycle.

The communication needs to be smooth between the developers, testers and the top-level management. The activities such as review meetings help to maintain a smooth communication and synchronization.

The active participation of testers is necessary in the code review meetings which keeps the testers updated with the changes in the application and focus on those changes during testing and thus the testers can explain the business value and the time and budget needed for the regression testing to the top-level management.

**Identifying the test cases and maintaining the regression test suite**

**Identifying the test cases**

**Test cases that are frequently error prone**

Some parts in the application are more error-prone and the test cases for that parts usually fail due to a small change in the code. These test cases need to be covered in the regression test suite.

**Test cases that cover the core features**

The core features of the application should be identified and the test cases written for those should be covered in the regression test suite.

**Prioritizing the test cases**

The test cases can be prioritized according to the frequently used functionalities of the application or focusing on critical parts of the application. The test cases can also be prioritized based on the basic or essential functionalities.

**Test cases that are complex**

The complex test cases which do not relate to the recent changes in the application should also be included in the regression test suite.

**Maintaining the regression test suite**

Maintaining the test suites is a crucial part which helps save time and cost of regression testing.

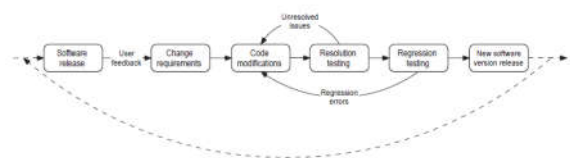
**Selecting the test cases for Regression**

The regression testing is a time and resource consuming process with a large test suites therefore the test cases in the test suite to be executed for regression testing must be limited. This process is called as Regression Test Selection.

Regression Test Selection is selecting a subset of test cases from original test suite that could test the affected but unmodified parts of the application. This helps reduce the testing costs.

**Regression Test Selection has the following activities**

1. Identification of the unmodified parts of the application
2. Selecting a subset that can effectively test the unmodified parts



Activities that take place during software maintenance and regression testing.

**Test suite optimization techniques such as minimization and prioritization**

Minimization can be achieved by finding a subset of test cases so that required coverage can be still achieved.

Prioritization is prioritizing test cases by running those test cases first which cover the affected or modified area.

**Regression test Tracking**

It is necessary to have a tracking mechanism in place in order to maintain the effectiveness of the regression test suite between the features that are developed and the test cases that are added for usability. This helps the testing team to validate the test coverage.

**Need to Monitor the Changes**

A sketched out procedure will guarantee that the test cases that are helpful to the testing strategy get added to the test suite, which guarantees the productivity and ease of use of the testing structure at a more elevated amount.

**Regular Checking**

Regular checking ensures accuracy and efficiency of test suite. This can also help in identifying problems in existing test suite.

**Assessing the metrics**

Assessing the metrics helps to identify the quality and usability of the test suites. Percentage of defects is one such metric used to determine the defects found.

### ***Cleaning up periodically***

Cleaning up maintains the efficiency of test suite. As the application undergoes changes overtime the test suites change accordingly. New test cases are added to the test suite to test a new feature added to the application and some are also removed from the test suite which are no more useful to test the application after the changes are done. The test suite is improved to test the changed application by adding test cases. The test cases which are obsolete or redundant after the addition of new test cases need to be removed.

As the application is retested at a frequent rate the redundant test cases can add to the cost of testing and size of the test suite. Hence, the redundant test cases need to be kept track of and some of them should be timely removed every time new test case is added.

All the existing test cases need to be analysed and can be removed if it does not support the different features in different scenarios.

Clean up ensures robustness of the application for a long time.

### ***Expensive***

Regression testing is retesting of the application over and over again. The company invests a lot of time and money for retesting which is also done previously without an ROI attached to regression testing. The following factors help reduce the costs to a certain extent

### ***Starting the testing in the early stages of software development***

At the initial stages of development it is easy to find the warnings and ambiguities. Thus, the defects found at this stage are comparatively less expensive to resolve.

### ***Documented testing process***

The testing should be kept documented, creating documents like test strategy, test plan and updating these documents with changes and their versions should be preserved.

Automating the test cases, removing redundant or obsolete test cases and keeping the regression test pack limited in size will also help making the regression testing process less expensive.

### ***Using appropriate tools***

Using free, open source testing tools should be encouraged over the expensive tools.

### ***Well-organized allocation and usage of resources***

If the employee skill set does not match with the application requirements, the best practice is to outsource the project which helps save the costs of testing. The resources should not be over pressurized and good relations should be maintained.

### ***Workforce***

The quality manager deals with the group of individuals, so he has to deal with the inspiration level and enthusiasm of his people. The task of regression assigned to a single resource can be tedious and boring and the quality of testing might drop. Hence, it is a good practice to keep rotating the personnel with different tasks.

If the tester continues to perform regression tests constantly, he may get disappointed. To stay aware of his inspiration level, enable him to give his hands a shot at automation testing tools to perform regression testing.

### ***Miscellaneous***

#### ***Planning for the Impact of Changes***

Adding a new functionality, integrating different modules can break the bug-free application. Therefore, regression testing must ensure that the application is not impacted by the changes. To build up a effective regression testing procedure, we should be solid and steady for the outcomes like unsure and vague bugs that the new changes can bring.

#### ***Classifying the Selected test cases***

Regression testing turns out to be troublesome when the application scope is enormous and there are continuous additions or patches to the framework. In such situations it is better to execute selected test cases to save time, cost and effort. This can be done effectively by classifying the test cases like reusable, redundant and obsolete test cases.

#### ***Creating an entry and exit Criteria for Regression Testing***

For entry criteria, a fixed arrangement of conditions like examining bugs and regression tests dependent on those bugs, and so on should be met so as to start the testing.

For exit criteria, a fixed arrangement of conditions like checking all tests are executed, no bugs stayed unfixed, and so on should be met to finish up the testing procedure.

#### ***Code Analysis***

Code analysis dives deep into the test case to find the cause of incorrect output.

The analysis can be static which is done when the code is written or dynamic which is done when the code is being run. Regular code analysis help testers to keep updated with the application capabilities and to update test scripts.

#### ***Optimizing the automated regression testing with Jenkins***

Jenkins is an open source automation server programmed in Java. It is used for continuous integration to monitor and run repeated tasks automatically. Whenever a test case fails, it can be notified via an email or RSS. Each execution in Jenkins is referred to as a build. The build can have successful, unstable and failed as it's possible outcomes.

#### ***The Jenkins has the following process***

- The developers check or change their source code.
- Jenkins will pick the changed source code and trigger a build and run any tests if required.
- The output is available in the Jenkins dashboards.

Jenkins is a powerful way to define when, where and how to run automated tests.

#### ***The key features of Jenkins***

- Jenkins can schedule tests to run at a specific time.
- Jenkins can display test results overtime.
- Jenkins display a summary of the test results.
- Jenkins display the details of a test failure.

- Jenkins can tell how long the tests have been taking to run overtime.

### **Optimizing the automated regression testing framework**

The automation frameworks are Linear frame work, Modular framework, Business Process Testing framework, Data Driven framework, Keyword driven frame work and Hybrid framework. Few of the widely used frame work is the Hybrid framework in automated testing which is a combination of Data Driven framework and Keyword Driven Framework.

For example, in basic terms, for a keyword driven framework a file (eg. an excel file) can be maintained which has a row of functions corresponding to each test case. In data driven framework a file is maintained wherein all the test data required is stored.

Optimizing this can be explained with an example: Two different processes use the Hybrid framework but with different approaches. For Process 1 the test data can be stored in a single file for every test case and for the Process 2 the test data can be in different files for different test cases. The Process 1 seems to be an optimized solution of storing test data but for smooth and time saving automation the Process 2 has a better approach of storing test data.

### **CONCLUSION**

The regression testing is the most crucial part of testing an application since that confirms that the changes have not adversely affected the application. Thus, optimizing the regression testing becomes important to ensure that it is a smooth process as well as time and cost efficient. More optimization techniques are developed to make the regression testing easier and effective. The topics discussed in this article are few of the broader level challenges and solutions for optimizing the manual and automated regression testing.

### **References**

1. ISTQB Glossary 3.2 GA Release Notes final
2. Graham D., van Veenendaal E., Evans I. & Black R. (2008) Foundations of Software Testing: ISTQB Certification. Cengage Learning EMEA, London, UK.
3. Based on personal experience during the Internship.
4. Based on research by S. Narasimha Reddy *et al./ Indian Journal of Computer Science and Engineering (IJCSE)*
5. Ranorex Regression Testing Guide

#### **How to cite this article:**

Dr. Shiv Kumar Goel and Nikita Rajendra Patil., 2019, Optimizing the Regression Testing. *Int J Recent Sci Res.* 10(06), pp. 33098-33101. DOI: <http://dx.doi.org/10.24327/ijrsr.2019.1006.3605>

\*\*\*\*\*