

Extreme Programming Methodology For Automated Regression Test Suite Development

Amogh Bendre, Sachindeo Shaha, Gireendra Kasmalkar



VeriSoft InfoServices Pvt. Ltd.
Pune

20th Jan 2004

Abstract

Developing Regression Automation suite for large applications can itself be a gigantic effort in terms of planning, designing, developing and maintaining the entire test suite. For large applications, to be able to develop and maintain equally large automation suites, a formal Software Development Lifecycle has to be followed. In addition, to accommodate the changes made to the applications while automation activity is being carried out, extreme programming paradigm is found extremely suitable for automation suite development. This paper explains in detail how Extreme Programming methodology is adopted by VeriSoft to implement regression automation of large applications. It includes case studies of 2 such projects carried out successfully.

Introduction

VeriSoft is a leading third party testing company that partners with its customers for their software testing requirements. Significant number of projects executed by VeriSoft are in the regression automation area. VeriSoft's typical customers are large services companies or product companies with large complex applications.

For the development of automated regression suite, VeriSoft has evolved a unique collaboration model, where the development phase of the regression automation is executed by the VeriSoft team following VeriSoft methodology and processes with support from resources from the customer side, and the maintenance phase is carried out by the customer.

During the development phase, a large number of test cases are to be automated in a short time, while the application is undergoing changes. For this phase, VeriSoft has successfully applied the extreme programming paradigm to automation suite development.

This paper describes in detail this methodology. Specific case studies of actual projects have been presented. Finally, the suitability of this approach has been demonstrated and specific benefits listed.

Background

Applications to be regression tested

The automation suite is expected to be reused over a number of builds. Thus products or large applications where number of releases are expected are ideal candidates for regression automation. The applications must also be stable (not changing drastically from build to build) so that the automation suite can be reused over a number of builds.

The following two regression automation projects have been studied in this paper.

1. A large web-based e-Sales System for global sales offices. The application uses Microsoft technology. The development work has been going on for 2 years after which the application has now gone in production at a few sites.
2. An Airlines proration product from Kale Consultants, which is already in the market.

Both are large applications that have reached reasonable levels of stability to be taken up for automated regression testing. Also the manual test cases are already documented and can be used as inputs for the development of automation suite.

Attributes of an automation suite

Automated regression test suite has to be Robust, Maintainable and Data-driven.

- a. Robust, because the suite should be able to handle unexpected conditions while execution as a result of defects in the application being tested.
- b. Maintainable, because in case of any functional / requirement changes, the scripts developed can be modified or new scripts can be added to the suite with minimum efforts.
- c. Data Driven, because the automation suite should be able to execute from different machines for a number of builds, so data should be a part of the suite but separate from the scripts.

To meet these objectives, formal software development life cycle has to be followed.

Phases in automation suite development

The automation suite development cannot start too early in the SDLC, while the application to be tested is still not stable. However, once the application starts becoming stable, there is a need to quickly automate the large number of manual test cases. This is the “development” phase.

Once majority of test cases have been automated, the further updation of the automation suite is based on the build cycle of the application. For each build, incremental addition to the automation suite can be made based on what parts of the application have stabilized in that build. This is the “maintenance” phase.

While conventional SDLC can be applied in the maintenance phase, for the development phase, where large number of test cases have to be automated relatively quickly, while the application is still undergoing changes, extreme programming paradigm [1] has been used.

Methodology

Following is the Comparison of Normal Software Development Life Cycle and Development Life Cycle of Regression Automation Test Suite:

Sr. No.	Normal SDLC	SDLC of Regression Automation
1.	Requirement Analysis	Evaluation of Tool for Automation, Conducting Feasibility Study of Application to select test cases suitable for automation
2.	Designing	Designing the Regression Automation Test Suite. This Includes: Defining folder structure, Identification of Utility functions and Test Data
3.	Coding	Recording of scenarios Programming for robustness, maintainability and data-drivenness Development of Utility functions
4.	Testing	Code review Unit testing of individual test cases Batch testing of entire suite
5.	Deployment	Deploying the automation suite (scripts, data and appropriate configuration / setting files) in the execution environment

Activities during the Development Phase of Regression Automation Test Suite:

Following Activities were carried out for each deliverables during development phase:

1. Designing the test suite directory structure.
2. Identifying modules to be covered for each deliverable.
3. Identifying the test cases and scenarios to be automated for each delivery.
4. Identifying utilities required for the Test Suite.
5. Assigning all the identified test cases among the Testing team.
6. Designing the suite for each delivery.
7. Development of automated test scripts for identified test cases.
8. Identifying test data required for executing each test script.
9. Conducting code review of each test script.
10. Unit testing of each script to verify the logic applied for individual test cases.
11. Execution of each script individually as well as in batch mode.
12. Verifying results of each scripts execution and verifying results of batch script before delivering the scripts to the customer.
13. At the time of each delivery test scripts from earlier deliveries were also executed along with the new scripts to be delivered.
14. Preparing the delivery note for each delivery made to the customer.

Activities during Maintenance phase of Regression Automation Test Suite:

As there are already developed test scripts the scope of activities in maintenance phase is applicable when there is change in any functionality within the application.

During maintenance phase activities, which are to be carried out are as follows:

1. Identification of new test cases to be automated for change in functionality.
2. Measuring the impact of changed functionality on the existing test suite.
3. Identifying utilities required for automating the new functionality.
4. Modification of existing test scripts.
5. Modification of Test Data required for executing the modified scripts.
6. In case any new functionality has been added to the application, new scripts are added to the existing suite. While developing new scripts some of the activities specified in development phase will be carried out.
7. Testing and Execution of Newly developed / Modified test scripts.

Modifications to the SDLC activities in the development phase based on Extreme Programming paradigm

1. Entire Automation Project has been divided into mini-projects of 2 weeks each.
2. While a broad plan is made for the entire development phase, detailed plan with specific deliverables and resource is made only for 2 immediate bi-weekly deliverables.
3. The prioritization of modules and corresponding test cases for automation is based on their criticality according to the customer as well as their stability.
4. The entire cycle of design, development and testing of the scripts for the particular deliverable are as per standard practice described earlier.
5. Each Delivery is a part of entire Regression Automation Test suite. The automated test scripts of any delivery are added to the suite already developed till the previous delivery, tested and deployed as one complete suite.

Observations and Conclusions:

1. The development phase is best carried out by automation experts, with skills on the automation tool and knowledge of automation process. They can quickly automate a large number of test cases, which are already documented for manual testing. The members of the application testing team can participate in this effort, understand the process while still primarily continuing with manual testing. During the maintenance phase, the application testing team can carry out the automation work with much less effort than required in the development phase.
2. By following extreme programming – the schedule of bi – weekly deliverable, one can start using the test suite once automation for the first delivery has been completed. This is a great advantage as one can start using the regression automation suite as early as 3rd or 4th week of the project,

depending upon the duration required to conduct feasibility study. With this approach the automation suite grows at the end of every 2 weeks.

3. As the entire project is divided into small modules and each module is treated as a separate project it is easier to plan in detail for each delivery. This also helps the team management to plan for further deliveries depending upon output of each resource in previous deliverables.
4. In the first deliverables, less number of test cases are automated as more time is devoted to developing utilities. See Appendix.
5. During the execution of each delivery, the scripts delivered in previous deliverables are also executed as one entire suite, thus ensuring the integrity of the entire suite.

Acknowledgment

We acknowledge our customers ISRC Ltd. and Kale Consultants Ltd. for allowing us to publish this paper based on the regression automation projects carried out by VeriSoft for them.

References

[1] <http://www.extremeprogramming.org/>