



Automated Software Test Data Generation

Harish Shinde

Students,

Department of MCA,

Tulsiramji Gaikwad patil College Of engineering ,Mohgaon.

Abstract-A challenge for IT industry is to develop software system that meets business needs. The fact is we are to deliver software that is free of bugs. Software testing is important parameter developing software that is free from bugs and defects. Software testing is performed to support quality assurance. Test data can be designed either manually or automatically. Software engineering research puts large emphasis on automating the software development process that produce large more complex quantities of code with less effort. A number of test data generation techniques such as random test data generator, path oriented test data generator, goal oriented test data generator and intelligent test data generator have been automated. Nowadays testing on networking environment i.e. to improve the scalability of software testing is emphasized Most of the industry is using object oriented techniques for software development due to high productivity. UML has got a great importance for software testing of object oriented programs. Different UML diagrams are used for different types of

testing. Mutation testing is used to improve reliability of object oriented software

Keywords: Time, scale and performing methods we can classify testing as unit testing, integration testing, system testing, alpha testing, beta testing, acceptance testing, regression testing, mutation testing, performance testing, stress testing etc..

1. INTRODUCTION

1.1 What is Automated Software Test?

Automated software testing is a process in which software tools execute pre-scripted tests on a software application before it is released into production.

The objective of automated testing is to simplify as much of the testing effort as possible with a minimum set of scripts. If unit testing consumes a large percentage of a quality assurance (QA) team's resources, for example, then this process might be a good candidate for automation. Automated testing tools are capable of executing tests, reporting outcomes and comparing results with earlier



test runs. Tests carried out with these tools can be run repeatedly, at any time of day.

The method or process being used to implement automation is called a test automation framework. Several frameworks have been implemented over

the years by commercial vendors and testing organizations. Automating tests with commercial off-the-shelf (COTS) or open source software can be complicated, however, because they almost always require customization. In many organizations, automation is only implemented when it has been determined that the manual testing program is not meeting expectations and it is not possible to bring in more human testers.

I. Automated Software testing in Future

Software projects, especially large and complex ones, require a significant investment in testing to ensure that they are successful. This specific investment is in time, staff resources, and additional costs, which can quickly add up and make

the cost of testing increase the overall scope of the development project. Luckily, automation testing can often be implemented to speed up the process and reduce these costly testing investments. Automation testing is testing without using manual labor or testing that creates automated scripts for previous repetitive, but necessary, testing in a formalized testing process already in place. It also

adds additional testing that would be difficult to perform by hand.

UML based Object oriented software Testing

The UML is a set of techniques for specification, visualization and documentation. UML is used by testers for

getting greater flexibility. The basic idea is to test software whose design is modeled using UML. The UML based testing is useful for model based system testing of distributed, component based systems. UML sequence diagram, state chart diagrams, UML communication diagram, class diagram, and activity diagram are used for test data generation where we require constraint solving. The main challenge here is to collect information from one or combined UML diagrams and store in an efficient data structure. Test specifications and test data are collected from the data structure.

What Automation Testing Can Add to Your Project

An automation specialist is a more highly skilled person than a regular tester, since he/she would already be an experienced manual tester with additional automation skills. Bringing in a person with these kinds of skills would enhance the testing process even more by reducing manual testing hours and decreasing the number of test cases and test data. Furthermore, the tools and frameworks you use to achieve automation can become a dependable resource for larger projects if there is a budget to cover that aspect of testing. Manual testing should still occur for a while even though automation helps speed up the testing process as a whole. However, automation in general should not be the real goal of your testing efforts for a big project, because if you focus on the



whole picture, the main goal should be to support new development efforts by providing quick feedback to the whole team.

Things to Consider with Automation

1. Cost and Resources: What is your budget and how many resources can you afford?

2. Stakeholders: Who is involved in the project? (i.e. Project Management, Quality Control, Development, Requirements)

3. Manageability: What is the scope of the project? Can Automation be included and be still be cost effective?

4. Execution Time: Manual vs. Automation. How much time can you save with automated tests vs. manual?

it is also important to keep your test cases/scripts and test data independent of the selected test automation tool due to the changes in format, layout, design, technical changes and updates to the code. For example, creating tests with hard-coded test data, system configuration, and properties makes them difficult to maintain, because in the long run, configuring the test data makes it difficult to change your test updates midway through a project if you were to run into any unanticipated issues and problems. Keep in mind that the most important part of your automation is the tests.

The Value of Automation Testing

A lot of project teams spend most of their time and effort creating a nice framework with a lot of features but forget about the tests. Don't let the main code become more important than the test code since what you use to test should be the main priority. The real value of any automated testing effort is derived from the test results it produces, instead of the quantity of automated test scripts. Additionally, you should not automate just for the sake of automation. Consider the costs involved, the resources and stakeholders, and the main concerns like maintainability and execution time before adding new tests. Automated tests reduce the testing time as a whole, but you need to comprehend that they become part of the production code base and therefore must be maintained just like the rest of the code, for the entire life of the application. Adding tests that are overly complex or difficult to maintain can slow down the feedback cycle to the team and should be avoided, which would lessen the importance of automation.

I : Functional requirements

Functional requirements are associated with specific functions, tasks the software must support. what are the work software supposed to do. It deals with product features and product functionality

II : Nonfunctional requirements

Non-functional requirements are constraints on various attributes of functions or tasks. It deals with quality of product. Testing for maximum ability, speed, efficiency, reliability, safety, and scalability etc. For example how many users can simultaneously try to vote in online voting software where millions of voters may cast vote on the same day.



Testing this type software manually is not possible. Non functional testing's are key area for automation where manual testing is impossible. Both functional and non functional requirements are included in software requirement specification document

1. International Society of Automation (ISA): -www.isa.org
2. International Organization of Standardization (ISO): -www.ios.org
5. International Telecommunication Union (ITU): -www.itu.int
6. Internet Engineering Task Force (IETF):
www.ietf.org

- **Future Challenges**

- 1. Improvement of code Coverage**

Empirical study for code coverage in different existing test data generation algorithm is a key area of research. The computational complexity of test data generation algorithms are very high. Therefore optimal solution or heuristic can be derived for complexity and coverage to facilitate the test automation process in minimum cost.

- 2. Efficient predicate constraint Modification**

Major problems with path oriented testing is many paths are non executable or infeasible. To decide the feasibility of paths we may generate a set of constraints

and then decide their satisfiability. Improvement of constraint solver is a challenging research area till today specially if it contains non linear constraints. Most of the test data generation algorithms complexity is high. It is basically designed for unit testing. But unit testing has high cost for huge quantity of extra coding necessary for simulating the environment where the unit will be run. Therefore the test data generation algorithms can be improved for making them scalable, i.e. for performing system testing.

- 3. Loops handling in path oriented testing**

Infinite looping is a common error in programs. In fact it is impossible to detect all kinds of infinite looping fully automatically. But many infinite loops can be detected automatically. Therefore research can be done on early detection of infinite loops automatically and it is more challenging if loops are nested.

- 2. CONCLUSIONS**

Automated test data generation is an important area of research for reducing cost of software development. Test data generation is done to satisfy functional, non functional, and business requirements. Some non functional requirement testing can be done only by automation; where manually it is not possible.

Depending on test information requirements, test data generation methods, different program analyzers are used. The mapping between program analyzers and test data generators can help



in this regard. The paper emphasizes the basic concepts of automated test data generation. The paper does not focus much on the test data generation using UML and object oriented methodology. In future, we may hunt the area of object oriented program test data generation works, but without having the concepts of this paper, we can not go directly to the concepts of object oriented programming.

3. REFERENCES:

UML based Object oriented software Testing

[1] Xiao Qu, Myra B. Cohen, Katherine M. woolf, “Combinatorial Interaction Regression

Testing: A study of Test Case Generation and Prioritization”, 2007.

[2] Shahid Mahmood, “A Systematic Review of Automated Test Data Generation Techniques”,

Master Thesis Software Engineering Thesis no: MSE-2007:26 October 2007.

[3] Gautam Kumar Saha, “Understanding Software Testing Concepts”, ACM Ubiquity Vol 9,

Issue 6, February 12-18, 2008.

[4] M. Prasanna, S.N. Sivanandam, R. Venkatesan, R. Sundarrajan, “A survey on automatic test

case generation”, Academic Open Internet Journal, www.acadjournal.com , Vol 15, 2005