

## Development of a Framework for Automated Software Tester (AST)

Mohammad Rafique<sup>[1]</sup>, Mr.Gaurav Goel<sup>[2]</sup>, Dr.Devesh Katiyar<sup>[3]</sup>

<sup>[1]</sup>Students of MCA, <sup>[2]</sup> Asst. Professor, <sup>[3]</sup> Asst. Professor

Department of Computer Science and Information Technology  
DSMNRU Lucknow, Uttar Pradesh, India

### Abstract

My research paper “Framework for Automated Software Tester” aim is to provide a framework to test the software automatic without any human interference. This method will provide an efficient way for time consuming. It is based on different test methods.

### I. INTRODUCTION

Here is model shift in the s/w industry. It's being driven through business demands to create longer & more complex s/w product in no time. How is the model shift changing the style s/w is produced?

The model shift may be characterized by two words: infra arrangement & automation.

The goal of a s/w test automation framework is to provide the infra arrangement that testers need in order the help for the development of auto mated test solution.

The framework evaluation team explored three options; (I) Creating one in house (II) Use a consulting coy. to recommend the commercial product (III) Performing a search on the internet for commercial product.

After checking all available options, the evaluation team chose to use open source test framework called s/w Automation Testing Framework.

The s/w test automation framework provides a collection of general test services that testers can use to create automated test cases. These services unlock s/w functions that testers can call in to simplify the common tasks that testers encounter when testing s/w. The STAF (S/w Testing Automated Framework) also provides the capability to extend beyond its core set of services by allowing tester to create custom services specific to their testing environment. To

Evaluate the benefit of STAF advertised, a system level testing involves two JPL s/w system were chosen, Mission Data's Processing Control Subsystem (MPCS) & Multi-missions Automating Tasks Invocations Subsystem (MATIS). These two s/w products work together as system within JPL's GDS.

There are many frameworks that provided supported for AST such as Framework for Integrated Test (FIT), Fitness, S/w Tasting Automation Framework (STAF), Selenium, Story Test IQ (STIQ), Ranorex, Test automation FX, & concordion, ranging from decent to horrible & free to very expensive.

Fitness is the source for automated customer tests. Fitness is wiki that created on top the fit framework, which is used for automated accepted test cases. The simple ideas of FitNesse is that test could be written tables, much like a spreadsheet. FitNesse give result best in a pair environment though, where one programmer & one business rules & write tests for them. Yet it doesn't allow remote implementation.

STAF is open resource; multi- stage, multi-language framework designed around the idea of reusable components called services. STAF doesn't allow distributed implementation meaning that results could be execute from a STAF maintain machine which are sent to & run locally on either client or server machines. The result returned back to control machine for reporting. Selenium & STIQ is s/w testing framework for web applications, selenium tests could be written as HTML tables or coded in a number of popular programming languages & could be run directly in most modern web browsers. Renorex & Test automation FX are windows GUI test automation Frameworks.

## II. SOFTWARE TESTING

Testing are systematic way to check the correctness of a system through help of experimenting with it. Tests are functional to a system in controlled environment, & a verdict about the correctness of a system is given, rooted in observations during the implementation of the test. When testing s/w, there is often amount of possible test-cases becomes an important part of testing process.

- (I) The code Coverage study of is the procedure of...
  - Find area of code not used by group of test cases.
  - Where Does Code Coverage Fit in The Testing Procedure

- (II) **Black Box Testing :-**  
Black box testing considers the object being tested to be a black box, which is no knowledge of the internals of the object being tested, should be known to the Test. The test simply uses the interface given by the object & ensures that the output is expected.

- (III) **White box Testing-;**  
White box testing is a dynamic testing strategy. White-box techniques are based on having full knowledge of the system with these techniques it's possible to check each branch & decision in the program. When the internal arrangements are known it's interesting to look at different coverage criteria. One of crucial one is decision coverage. The test is successful if tester could check if the programs diverge from its given goal.

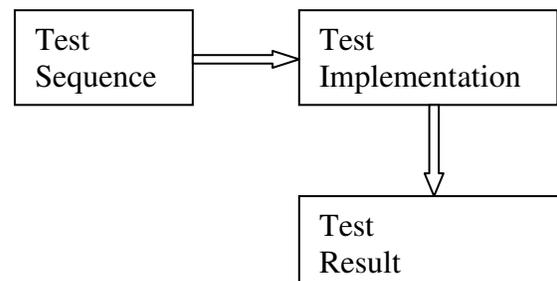
- (IV) **Code Coverage ;**  
This is the white box test methodology, i.e. it needs knowledge of & access to the code itself instead of simply using the interface provided Code Coverage is probably most useful during the module testing face, tough it also has benefit during integration testing & probably at other times, depending on what & how we are testing. Some Basic Matrices:  
The Boolean expression contains the conditions & 0 or more Boolean operators. The decision without any Boolean operator is the condition.

- (V) **Unit Testing;**  
Unit testing's are performed at the most functional level where the smallest testable part of system (unit) is tested to ensure that it's working according to detailed

design/requirements of the system. For this the test environment should provide capabilities to simulate input values at unit level & verify the output at unit.

- (VI) **System Testing;**  
System testing's are performed to ensure that system is capable of delivering as per functional specification or systems requirement. These tests are performed at higher level of system as a whole & usually require input simulation at product interface level.  
The expected outputs are also captured at product interface level.

- (VII) **Test Engine;-**  
The test environment shall provided capabilities to Define test sequences (test procedures), perform test implementation & evaluation of Pass/Fail criteria for generation of tests results. The part of tests environment that provides these capabilities to define tests sequences, test implementation & test evaluation is called Test Engine responsible for driving the procedures associated with testing.



## III. AST

A test interface was provided in s/w, which could receive test commands & send response back to test tool. The automated testing was architected focusing on reusability. We first decided to separate test sequences from the s/w that would actually send the test result to the s/w. It would enable the communicating s/w be reused.

The second factor that decided the architecture of the system was the need to be the event driven. The automated testing was required to be capable of catching an unexpected response from the s/w. To achieve this the automated testing was check if the message was expected or not accordingly writes an error log.

The third factor that influenced the architecture was the requirement for the automated testing tool they

represent & uses the core model also allowed re-usability. Test sequence could now build rooted in core functionality. A test sequence could be composed of group of core functions called in a specific manner.

(I) Group of core functionality specified in readable format similar to specification of the test cases.

(II) Test sequences also specified & read before sending Automated Testing S/w.

(III) A logging module that would record the test action sent & the response received, along with the any noted errors.

Auto mated test primarily starts with defining test sequences i.e. defining the inputs & expiating the outputs as per requirement. It shall provide capabilities to define the test sequences in form of test scripts shall provide capability to be put under configuration control once test sequences are defined.

#### IV. COMPARATIVE STUDY FOR AST

A S/w testing requires automated test implementation & results verification called automated Testing. Test automation is one of main approach has been functional to decrease the testing costs.

##### (I) N- Description Diverse Systems & m-model Program Testing: -

Manolache & Kourie suggested an approach based on N-description Diverse System based on various implementations of a program. To put it differently, testers use various descriptions with independent implementation of application over test. This Idea could be resulted in an expensive process

& this method is unable to assure the efficiency of testing process.

##### (II) Decision Table-;

A decision table is the s/w requirements representation model. This has been functional wherever there are many condition sections, which are s/w response where special conation satisfied. Each row in decision table present a variant as a unique combination of combination conditions.

##### (III) IFN Regression Tester:-

There have several attempts to apply Artificial Intelligence (AI) methods for simulating the Automated Testing behavior & use it as test s/w.

Info fuzzy network is then approach created for knowledge discovery & data mining.

An IFN present the functional needs by an oblivious tree like arrangement.

##### (IV) Input/output Study Based Automatic Expected Output generator—

Previously there were few research on semi automated expected output generation. The combinatorial testing are to test all possible input values combinations, since the number of these combinations could be very large in practice, effective test data reduction is necessary. Note that it's important to maintain the test quality. Structural study is either static or dynamic & could be functional if testers have access to source codes. Static study checks the run time information gathered from code implementation. Static study may overestimate program dependencies & dynamic study not able to assure full detection of Input Output relationships

#### V. CONCLUSION

Everything is work on scale or everyone has a blueprint on which she works best. Similarly, in Computer Science, the framework is used to get the work done quickly and in a short time as we have as we have developed for software testing. This is a profitable process for testing. Framework is set of rules and corresponding tools that are used for building test cases.

#### VI. REFERENCES

- [1] <https://www.testingxperts.com/blog/test-automation-frameworks>.
- [2] <https://smartbear.com/learn/automated-testing/test-automation-frameworks/>
- [3] <https://www.softwaretestinghelp.com/automation-testing-tutorial>
- [4] [https://en.wikipedia.org/wiki/Test\\_automation](https://en.wikipedia.org/wiki/Test_automation)
- [5] <https://www.guru99.com/test-automation-framework.html>
- [6]

<https://www.smartsheet.com/test-automation-frameworks-software>

[7]

Eun Ha Kim, Jong Chae Na, Seok Moon Ryoo “Test Automation Framework for Implementing Continuous integration”

[8]

Xin Chen, R. Scott Harvey, Kyle Robeson  
“Software Testing automation Framework”