AUTOMAN

-A COLLABORATIVE TEST AUTOMATION FRAMEWORK

Balamurali L (Group Manger SQA)
Diana Ambrose (Senior Lead Engineer SQA)
Anildas H (Lead Engineer- Testing)
Geethu K Nair (Engineer SQA)

Network Systems and Technologies (P) Ltd.
A3, Periyar, Technopark,
Thiruvananthapuram, Kerala

Submitted to:

13th Annual International Software Testing Conference in India 2013
04 - 05 December 2013, Bangalore, India
1. Abstract

Test automation is advocated to reduce overheads, improve productivity and efficiency; the bane of which is heavy investment in terms of purchase of the tools, license, training and maintenance. Add to it the dependence on domain/language and availability of plug-ins; and teams prefer to avoid automation.

AutoMan is an easy to deploy and use framework that contains test script generation tools, script execution, reports retrieval, auto notification and most importantly populates the results of tests to complete the test execution process without manual intervention. Recurring manual effort in sequencing tests, documenting results in tools like DevTest etc is completely eliminated…advantage?- leveraging the best of various test automation tools to seamlessly integrate testing activities right till the end- status reporting.

2. Introduction

The market is flooded with tools for all sorts of testing activities, be it test management, scripting, execution etc. How does AutoMan differ from them? Is this just another tool to increase overhead in the name of automation?

The problem with test automation in most organizations is that there is rarely an end-to-end test management tool in place. The cost for buying such tools with plugs-ins for scripting, various type of tests, test data generation etc, is huge; especially with licensing, training and support involved. Organizations end up investing in just parts or individual components rather than an entire suite, resulting in documentation overheads to tie the gaps together for management to get the big picture of the test process.

3. Test Management – Practical Scenario

Consider a scenario, where you are using a scripting tool like Test complete to generate tests and execute them, but your customer insists on having custom template for reports. And you end up moving all this data to Test management tools like DevTest which will give the customer dashboard view, viewable over the net. A good amount of time is spent in analyzing the On an average, a test engineer would require atleast 5 minutes to key in the results in DevTest per TC report. On an average if your tool has executed 100 TCs in 14 hours, it would require almost the same time for the test engineer to move the results to the test management tool. Is it just a small price to pay for automation? Or a waste of valuable resource time that can be avoided, considering this is not a onetime effort, but effective for each iteration

4. The Solution – Automan

The solution is a simple and light Test management framework named “AutoMan”. The four layers of this framework are described below:
The Test Management Layer:

This layer contains traditional test management tools like DevTest, JIRA etc. The framework layer shall load the final results to the Test management layer; where the existing features of the tool can be leveraged for advanced reporting, analysis and decision making.

The Result Management Layer:

This layer contains the results of test execution tools. Tools like QTP, Test complete, or any other tool which is used in the project for scripting and test execution; generates results in its propriety format. Interpreting the test results very often is a tedious task. The logs generated are parsed by the framework layer and converted into various formats that provide test manager/senior management with potential data for quicker decision making.

The Test tool layer:

The Test tool layer contains actual tools used in the project for test script generation like QTP, Test complete etc. The tools shall be used to generate scripts for the system under test. The framework layer can be configured to identify the test tools in use, and sequence tests for execution.
The Framework Layer:

This layer acts as an interface between user and other layers of the Automan framework. Automan framework can be configured for different test tools (refer Figure 2: Settings Screen)

![Figure 2: Settings Screen](image)

The various components of the Framework layer include: Test Execution Configuration, Result file manager, Test Scenario Collection and Report - Mail Server manager.

Test Scenario Collection module will collect all the test scenarios to run; while the Test Execution configuration shall set the sequence and the repeatability of tests (refer Figure 3: AutoMan- Main Screen)

![Figure 3: AutoMan- Main Screen](image)

Test Scenario Collection and Report generates the test results in user specified formats. The framework supports configuration to support multiple customers. User has the option to select the custom template designed to suit customer requirements or use the default template provided by the framework to store
and display test results. This template (e.g shown in Figure 4: Summary Report Email) will be used by Result management Layer for generation of result file.

Mail Server manager shall be responsible for mailing the reports generated to concerned stakeholders.

Figure 4: Summary Report Email
5. Workflow

The test sequencing and controlling function lies between the ‘application under test’ and the test management layer. One time settings involve identification of automation and management tools for the project, set the path for the automation tool to save reports and allows configuration of auto mails generated to notify stakeholders of crashes/ issues or test successes.

Once the settings are done, Test engineers work on generating scripts for the identified Test cases. A unique link is established between the TCs and the script files to enable Automan Server to sequence and trigger the test execution. Test Engineer can sequence the tests that require to be executed. AutoMan triggers the execution of the scripts generated using the tool identified in the settings; the resulting reports are stored on the server and mails sent to stakeholders.

The idea is to keep the report simple and provide relevant stakeholders with a snapshot view of the test iteration. Links within the report can be used to access the test management tool to view the details of the tests. The individual TC reports are by default stored in Excel based templates; this allows Managers with the flexibility of doing custom analysis and reporting even without the use of test management tools like DevTest, if required. Standard templates or customized templates as per customer preferences can be used to store TC execution results.

In parallel, the result are ported to a test reporting tool (like DevTest) to provide managers with access to test status, not just for status reporting but also in assisting with decision making. Stakeholders get notified on TC failures or show stoppers, next day all they need is to login to the system to get a host of data ready to be analyzed and ported in whichever way they could best utilize it.
5. Case Study

A Project executed in the Power Domain required the automation of test scripts done using QTP. Since it was a multi-location project, the customer preferred the use of DevTest to have a centralized view of the test results. The team was struggling to meet the customer requirement since additional overheads for:

- Manually verifying QTP logs and porting data to custom TC report format
- Log each test result to DevTest along with filled in custom TC report

Approximately 8-10mins was required for executing one TC and another 10mins for parsing the QTP logs, working on custom template and porting result to DevTest. Considering the duration of the project and the multiple iterations expected to be performed, the team began looking at ways to optimize the documentation overhead while reducing effort but saving cost by not burdening the company with purchasing new tools. ‘AutoMan’ provided the team with a simple to use interface to schedule tests, port reports, send notifications etc.

The manual intervention required from start to finish was minimized and the customer/ senior management was able to get summary status reports in their mail box as well have all the details on the DevTest tool for further analysis.
Data for one of the major modules was analyzed to look into the savings by using the tool. At the rate of 8-10 min/TC execution, 200 hours would be required to execute the tests; while almost same amount of time would be required to port data manually to DevTest. 45% savings was brought in by using Automan for each iteration!

6. Advantages

The framework bridges the gap between various tools providing an end-to-end solution to test automation. The framework is lightweight, and not just tests are executed overnight, but even results are ported without any manual intervention saving considerable time each iteration to populate results in the final test management tool. The considerable savings in effort of test engineers is noteworthy.

It is an end to end solution for test automation which requires minimal human intervention. Support to pick tools of choice; different reporting formats, automated mails and alerts; and streamlined data population are some of the salient features of the framework.

At the same time, it provides an efficient and effective means to integrate various tools without having to bother about the technicalities; giving the same benefits that a complete end-to-end Test suite would have offered, at a much lesser price.

7. Scope for Future Work

The framework has been tested with tools like QTP, TestComplete and DevTest with standard as well as customer specified TC report. While in theory it should work for any other tool, more research and trials will need to go into ensuring that AutoMan supports a wider set of test scripting and management tools to make it more robust. Integration to CM tool to pull in latest builds and mechanism to generate a QA test certificate would add further value to the tool.
8. References


9. Acknowledgement

The Authors are thankful to Mr. N. Jehangir, Managing Director, NeST Group and Mr. S. Narayanan, President of NeST, Trivandrum for their support in carrying out this work. We are thankful to Mr. Sebastian Ukken, Associate Vice President & Head of Quality and Testing Group for their useful inputs and suggestions at different stages of this work. All the members of the organization who have helped us in various stages of this work are acknowledged for their support.

10. Author’s Biography

**Anildas H, Lead Engineer Testing- NeST.** With an MCA from Mahatma Gandhi University, Kerala; he has more than 7 years of experience in the industry. He has played a lead role in implementing automation frameworks Industrial and Power Automation domain. He is a Certified Engineer for Rational Functional Tester from IBM. He is also an ISTQB Certified Tester

**Diana Ambrose, Senior Lead SQA-NeST.** She has a Masters degree in Computer Applications from Sikkim Manipal University and has more than 12+ years of experience in Quality and Testing activities. She has played a key role in implementing models like ISO 9001:2008 and CMMI v1.1. She has worked as an assessment team member for CMM v1.1 and CMMIv1.3 and good knowledge on the implementation of standards like ASPICE, ISO 27001, TL9000 and CMMI v1.3. An ISTQB certified Test Engineer and certified Test Manager; she also holds a certificate for Test Manager by Department of IT, Govt. of India and as Internal Quality Auditor from Zandig Solutions. She is a TMMI foundation member

**Geethu K Nair, QA- Engineer –NeST.** MCA from IHRD College of Applied Science, Calicut; she has 4+ years of experience in the industry and plays a key role in defining and streamlining the metrics activities in the organization. She is an expert in Metrics and statistical process control and was a key member in organization’s journey to CMMI DEV V1.3 Maturity Level 5. She is a Certified Tester (foundation level) by ISTQB. She is a TMMI foundation member

**Balamurali L, Group Manager SQA- Network Systems and Technologies (NeST).** He has an M.Tech from Indian Institute of Technology (IIT) Kanpur and has 14+ years of experience in the industry. He has played a lead role in implementing models like CMMIV1.1, CMMIV1.2, CMMIV1.3 and ISO900, ASPICE, ISO 27001,TL900, IEC61508 and ISO13485. An Assessment Team Member for CMMI v1.2 and v1.3; he is certified to CSQA & CQSPE from QAI, ISTQB Certified Tester and Certified Test Manager. He is actively involved with Indian Testing Board and plays a pivotal role in the Trivandrum chapter of Software Process Improvement Network (SPIN). He is a certified Test Manager, Information Security Professional and Internal Auditor by Department of IT, Govt. of India.