

Test Design Automation Tool Development Using VBScript

Kavya Hanisha G U V

Department of Computer Science and Engineering, Amrita University
Amrita School of Engineering, Coimbatore-641112, Tamil Nadu, India

Abstract— Manual test script design and development is a very tedious and time consuming procedure. A lot of time is being spent on this. This in turn increases manual efforts to develop the test cases and documentation issues for the test scripts. To avoid such issues test design can be automated. Therefore, to minimize these complications, instead of writing the test cases manually for each and every requirement, test script generation can be automated, using VBScript by following the standard format of test cases. Input requirements from the Excel sheet. In the user interface of the tool, we mention the Location (path) of the requirements file. Tool parses the requirements sheet and generates test scripts in the output folder as specified by the user. The generated test scripts will be in .VBS format which have to be executed to verify the test results.

Keywords— Test Scripts, Automation, VBScript, VbsEdit.

I. INTRODUCTION

Software testing is an activity to check whether the actual results match the expected results. Test Case is a set of actions executed to verify a particular feature or functionality of your software application. Expected Results should be documented.

Apart from your test case, it may have the field like Pre-condition – which specifies things that me be in place before the test can run. It may also include Post-condition which specifies anything that applies after the test case completes.

During the test case execution, we will document the results observed in the actual results column and may even attach some screen shots based on the results give Pass/Fail status.

Fig.1 exemplifies input data.

	A	B	C	D	E	F	G
1	Test Case ID	Test Scenario	Test Steps	Test Data	Expected Results	Actual Results	Status: Pass/Fail
2							
3							
4							
5							

Fig.1 Format of Input Excel Sheet

II. METHODOLOGY

Input File consists of Test Script Name, Test case Number, Test Case ID, Software Requirement, System Functionality, Test Description, Pass Criteria and Status. Number of rows in the excel sheet denote the number of test cases for the script.

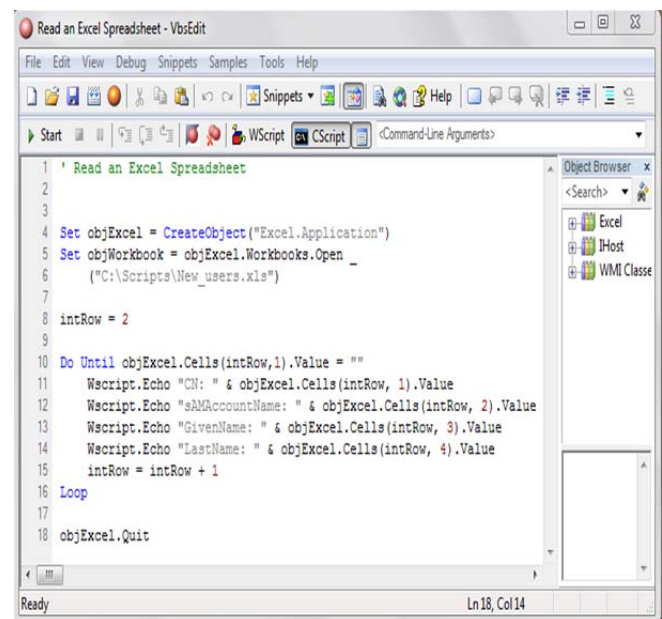


Fig.2 VbsEdit: Editor for VBScript Coding

Code in VBScript for the following user defined functions for the generation of Test Scripts using VbsEdit, a Shareware Windows editor for VBScript development. Fig.2 shows the Editor Integrated Development Environment (IDE) with a sample code to read the excel sheet. Hundreds of samples are available in the editor itself which is very helpful for VBScript beginners. Fig. 3 shows a basic input text box using VBScript.

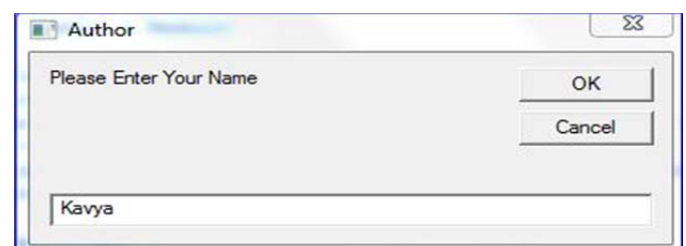


Fig.3 Example GUI using VBScript

(To insert name in the test script to be generated)

The following are a few examples of user defined functions involved in the designing of test scripts.

```

10 .....
11 'Testcase_ID      : @[ATE_Connections_001_001]
12 'Sys_Requirement_ID :
13 'SW_Requirement_ID :
14 'TestCaseDescription:
15 '
16 '
17 'Pass Criteria    :
18 .....
19 TCID= "[ATE_Connections_001_001]"
20 Content =
21 Call write_testResult (Content)
22 Shell.run " "
23 MsgBox
24
25 '1.Run the interface
26 '2.Click on the capture button
27

```

Fig.4 Example ForHeader () function

```

Call ReportSummary ()
Call Reset ()
Call Move_Results (CD&"\" & "ATE_Connections ",Result_Path)
'-----
                        END OF TEST
'-----

```

Fig.5 Example ForFooter () function

Similarly other functions which are necessary for Test Script are TestCase (), GetFolder () and GetExcelPath ()

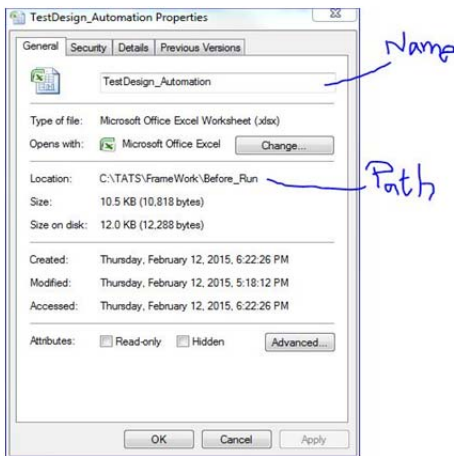


Fig.6 Name and Path of the input file

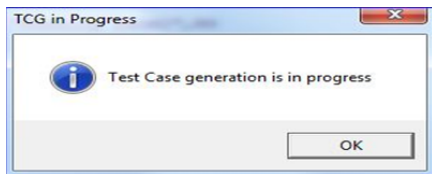


Fig.7 Example GUI

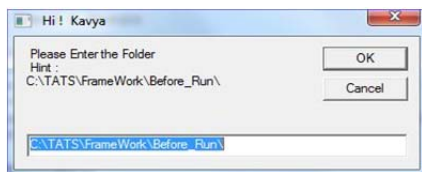


Fig. 8 GUI to get the Output Folder Path

III. ARCHITECTURE OF THE TOOL

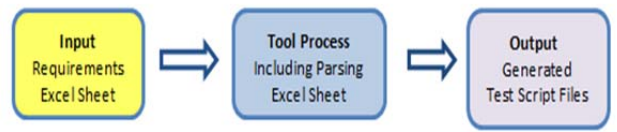


Fig. 9 Automation Tool for getting Test Scripts

The tool gets the input from the user, through the user interface; we can authorize the test script and enter the location and name of the input requirements file. The tool then parses the excel sheet column wise to extract the important rather required data to design the test script. Firstly header function is called as shown in Fig.4; it includes Test Case ID, System Requirement ID, Software Requirement ID, Test Description and Pass Criteria from the requirements worksheet. Secondly, and most importantly the Test Case function is executed. It should be coded as per the requirement. Depending upon the number of test cases in the test script, the function is called iteratively. Finally, the footer function is called as shown in Fig.5; it includes call for Report summary and reset functions. Eventually the test scripts will be generated in the output folder which the user enters as shown in Fig.8 for the number of test cases as specified in the input requirement sheet. Thereby we can save a lot of time by just executing the .VBS files which have been generated automatically by the tool, instead of coding all the test scripts for each and every test case manually.

CONCLUSIONS

It is a big break through that we have saved so much human efforts and Time being spent on coding the test scripts manually. Having built this tool incrementally with requirements clearly defined, there is a lot of scope for further modules or builds.

Automation for self-review of the generated scripts, Automatic creation of Log File to summarize the test script generation, After each test script, Handling of exception cases needs to be done.

ACKNOWLEDGMENT

I would like to express my sincere thanks to Mr Surendra Kumar Reddy, Senior Engineer in Software verification and validation domain and also to Mrs P.Subathra, Faculty in the department of computer science and engineering at Amrita University for their innovative and creative ideas by extending their scope for research facilities. I remain indebted to my family and friends for providing moral support, encouragement and, for their invaluable suggestions and constant encouragement that led to improve the presentation quality of this paper.

REFERENCES

- [1] Hitesh Tahbaldar and Bichitra Kalita , “Automated Software test Data Generation”, International Journal Of Computer Science and engineering survey , Vol 2,No 1 Feb 2011.
- [2] Jayashree Ghorpade, “Automation tool for a layman to test remedy applications”, International Journal Of Advanced Research in Computer Science, Vol 2, Issue 2, Feb 2014.
- [3] Abhijit, Pranit, “Software Testing Techniques and Strategies”, International Journal Of Engineering Research and Application, Vol 2, Issue 3, May-June 2012.
- [4] Software unit test coverage and adequacy, Hong Zhu, Patrick A. V. Hall, John H. R. May, ACM Computing Surveys (CSUR) December 1997, Volume 29 Issue 4
- [5] An empirical study of regression test selection techniques, Todd L. Graves, Mary Jean Harrold, Jung-Min Kim, Adam Porter, Gregg Rothermel , ACM Transactions on Software Engineering and Methodology (TOSEM) April 2001 ,Volume 10 Issue 2
- [6] Using model checking to generate tests from requirements specifications, Angelo Gargantini, Constance Heitmeyer, ACM SIGSOFT Software Engineering Notes, Proceedings of the 7th European Engineering Conference held jointly with the 7th ACM SIGSOFT symposium on Foundations of software engineering October 1999 Volume 24 Issue 6
- [7] Pressman, Roger S. (2005). Software Engineering: A Practitioner's Approach, Sixth. McGraw-Hill. ISBN 0072853182.
- [8] M. Balcer, W. Hasling, and T. Ostrand, Automatic generation of test scripts from formal test specifications. In Proceedings of the ACM SIGSOFT Third Symposium on Software Testing, Analysis, and Verification, pages 210–218. ACM Press, 1989.
- [9] Astels, D. (2003). Test Driven Development: A Practical Guide. Upper Saddle River, New Jersey: Prentice Hall PTR.
- [10] Kaner, Cem; Bach, James; & Pettichord, Bret (2002) Lessons Learned in Software Testing, Wiley.

AUTHOR PROFILE



Ms. Kavya Hanisha G U V pursued her under-graduation in B.Tech [CSE] from Amrita School of Engineering, Amrita Vishwa Vidyapeetham at Coimbatore. Her areas of interest are Software Engineering and Computer Networks. Currently, she is an intern trainee at Honeywell, Bengaluru.