Software UI Functional Automation Using GUIRobo

G. Mohan Doss Gandhi

Abstract—In Agile world, complexity of the software is keep increasing as applications are getting developed in multiple platforms and environments. In fast-moving software industries, changes and challenges are inevitable to meet customer satisfaction. Especially delivering High quality product on time is essential to beat the competition. Though test engineers have multiple challenges, need to test the entire functionality and deliver the products on time. Test automation is one of the robust and fastest solutions for achieving quality in complex environment. To achieve High quality on time, test engineers should have robust and highly sophisticated automation framework. At the same time, framework should be more generic, economical and reusable. This paper looks at the problems persisting on Graphical User Interface (GUI) Functional test automation tools and describes the proposed new simple and affordable solutions which will be implemented on GUIRobo.

Index Terms—GUI functional test automation, test modes, test types.

I. INTRODUCTION

Software testing is not just for uncovering any defects in the software. Software testing must be performed to ensure that a software program, application or product meets all the implicit and explicit business and functional requirements [1]. Hence robust testing methodologies are essential to discover the issues and achieve best quality. Testing can be accomplished thru reviews (product requirements, software functional requirements, software designs, code, test plans, etc.), unit testing, system testing (also known as functional testing), expert user testing (like beta testing but in-house), smoke tests, etc. Each ‘testing’ activities are playing vital role in determining the High quality [2]. The biggest challenge in delivering high quality product is increased pressure in time to market. Due to time pressure, we cannot shortcut our test strategy and test methods. Still test engineers are committed to perform through functional testing and deliver the high quality product as they promised to the end user. There are number of GUI functional test automation tools available in the market but each tool has its own unique problem such as, lack of reusability with multiple platforms, lack of Integration between front and back end, difficult to maintain. Some of testing tools are not affordable to all startup companies as it is too expensive.

Hence, to overcome all the issues and deliver the better quality on time, test engineer must adapt a Proactive and innovative functional test automation solution. We believe that the proposed and enhanced new tool GUIRobo will sort out all the issues and challenges faced on commercially available tools and provide the robust solution in functional test automation.

II. MANUAL VS AUTOMATION

A. Manual Testing

Though practice of manual testing is still being used, it has significant weaknesses. The primary weakness is once a problem is found; it is difficult or impossible to reproduce the defect because the tester was not following a pre-defined sequence of events. Other weaknesses of manual testing are [2]:

1) Manual techniques cannot provide maximum user interaction over time. Because we cannot manually keep stressing the application for longer period.
2) Manual testing will not provide the extensive test coverage of the product. There may a be chances of missing some scenarios.
3) In general, Manual testing cannot help much in repeating the tests and reproducing the failures on time.
4) Manual testing does not log the discrete values with each test execution sequence. It is very critical for detecting memory leaks while validating particular functionality.

B. Automated Testing

Automated ‘Testing” is automating the manual testing process currently in use. This requires that a formalized “manual testing process”, currently exists in the company or organization. Automation is the use of strategies, tools and artifacts that augment or reduce the need of manual or human involvement or interaction in unskilled, repetitive or redundant tasks [3]. Automated testing can provide several benefits when it is implemented correctly and it follows a rigorous process. The significant automated test benefits are [4]:

- **Reliable**: Tests perform precisely the same operations each time they are run, thereby eliminating human error.
- **Repeatable**: You can test how the software reacts under repeated execution of the same operations.
- **Programmable**: You can program sophisticated tests that bring out hidden information from the application.
- **Comprehensive**: You can build a suite of tests that covers every feature in your application.
- **Reusable**: You can reuse tests on different versions of an application, even if the user interface changes.
- **Better Quality Software**: Because you can run more tests in less time with fewer resources.

Manuscript received September 15, 2015; revised February 6, 2016. This is a follow up paper to the “Software test automation using GUIROBO” which was published on ICCTD 2011 conference.

G. Mohan Doss Gandhi is with the Microsoft India R&D India Pvt Ltd, Microsoft Campus, Hyderabad, Telangana, 500032 India (e-mail: gmdg75@gmail.com).

doi: 10.18178/Lnse.2016.4.3.248 189
• **Fast:** Automated Tools run tests significantly faster than human users.
• **Cost Reduction:** As the number of resources for regression test are reduced.

### III. PROBLEMS IN EXISTING AUTOMATION TOOL

A number of Functional Test Automation Tools are available in the market to verify the functionality of the GUI-based applications. However, many of them, will not conquer an entire solution for Functional Test Automation. Sometimes due to the complexity of the tool, several people have failed to use them effectively. Most of the commercial functional automation tools contain the automation solution for multiple programming languages in single package. Though the tester has to automate only the particular application developed in particular platform, still they are forced to pay more and buy an entire automation solution. In some scenarios, we may have to install the automation solution and run the test on device under test. Incase If device under test does not have enough memory, then we cannot install the automation solution and execute the tests on device under test. Hence still we have to rely on manual execution for such scenarios.

In fact, the present automation tools in the market forces the tester to do lot of customization and develop test suites to run their functional automation. It makes test engineers to spend more time in analyzing and debugging features of the Functional automation tool. If they spend most of their time in writing test scripts and creating test harness, then it will delay in findings early defects in Software life cycle. Finding a bug in later stage of software life cycle costs more. The greatest ROI is to find bugs early [5].

Software managers and developers are being asked to turn around their products within ever-shrinking schedules and with minimal resources. In order to start testing in early stage, we require a routine (test harness or executable) and robust automation tool for running test in repeatable fashion. Another major challenges in GUI test automation are maintainability, reliability and meeting all testability requirements.

However, if automation tool is not robust enough to handle the testability challenges, then finally user has to end up with using coordinates for identifying and automating the GUI controls.

Hence the repetitive nature of testing, leads people to use record and playback techniques.

There are many tools that allow scripts to be recorded and then played back, using screen captures for verification. The problem that always crops up is that play back will fail in case if layouts or resolution of the applications are getting changed. Due to that the scripts must be re-recorded from the scratch. Record and playback tools provide an easy way to create throwaway test suites and not for long term solution. Test creation should be a cumulative process, with parts of existing tests being recycled to make new tests. A good tester will always try to reduce the repro steps to the minimal steps to reproduce; this is extremely helpful for the programmer who has to fix the bug. It can be possible only if tool has efficient and detailed logging /report facility. The proposed solution contains the solution for above described issues.

### IV. PROPOSED SOLUTION USING GUIROBO

Test automation can be done for Functional testing, Performance testing, Load testing and Stress Testing. To carry-out GUI functional testing, we have numerous tools available in the market. Since we do not find any robust, simple and cheapest automation solution/ tool for functional test automation, enhanced GUIRobo stress testing tool to support the functional testing. Using GUIRobo, the test engineer can quickly develop their automated test cases and execute them all without any user intervention. It allows user to come up with their custom solution by just invoking dynamic link libraries (dlls). It can be installed on device under test which has less memory space as user does not have to install any huge application software to use this tool. Also User does not have to spend more time on learning/developing and executing test cases using GUIRobo.

Initial version of GUIRobo tool will provide an easy to use, consistent and cost-effective way of testing functionality of GUI applications developed in C/C++ /VC++, C# and VB.Net. WPF, WCF, ASP.Net and REST based Web APIs. It provides robust solution to validate the front end and back end testing as part of system Integration testing. It supports all the windows UI controls and custom controls such as Source grid, Data grids etc. It does not force user to come up with custom solution and user can develop their test cases in any managed programming languages. GUIRobo uses various techniques such as UI Automation element [6], hook process, win32 APIs, Text and Name property to identify and automate GUI controls. Hence it can provide support for wider ranges of GUI controls including owner draw and customized UI controls.

### V. WHAT IS GUI FUNCTIONAL TEST?

GUI Functional testing is form of testing to verify the UI functionality of the software application under test. Functional testing is done against the business requirements of application. Basically, it ensures the basic functionality mentioned in the requirement specification and functional specification document. It also cross check whether software application is meeting the customer expectations. As part of functional testing the following verification will be done:

- Is UI functionality is working as expected?
- Is UI functionality is not working as it should not do?
- Is software is not doing as it not supposed to do?

Examples of UI Functional testing include Unit Testing, Integrated testing, System testing and regression testing. Functional testing includes following steps:

- Preparation of test data based on the specifications of functions.
- Business requirements are the inputs to functional testing.
- Based on functional specifications find out output of the functions.
- The execution of test cases.
- Observe the actual and expected outputs.
VI. FUNCTIONAL TEST AUTOMATION USING GUIROBO

A. How Does It Work?

GUIRobo tool consists of simple VB.NET template and supportive dynamic link libraries(dlls). The template contains sample User Interface and supportive methods required for functional test automation. The supportive dlls have APIs (Application Programmable Interface) required for automating various UI functionalities like click, verify etc. Some of the built-in methods are developed as part of the template which provides extensive logging mechanism to store the Test results/memory usage of device under test. Using this template test engineers can quickly automate/develop their functional test suits. It allows user to control their test execution based on impact on each module and schedule.

Once all the test cases are developed, then user can create a standalone executable file and can be used on multiple device under test. This way the users don’t have to compile the test suite for every time it runs. GUIRobo front panel has options for selecting the desired path for log file, various Test types and Test modes.

Test engineer can select the appropriate options and start the functional test by selecting Start button. At any point of time, test execution can be aborted by selecting Scroll Lock key. GUIRobo also supports command line option to execute the test remotely on scheduled time.

B. Different Types of Testing

The sample testing types have already been defined in GUIRobo front Panel (Fig. 1). Tester has flexibility to design their test cases for each testing types or they can customize their own UI controls according to their functional test requirements. The test execution can be varied based on the test types.

C. Test Modes

GUIRobo provides the support for executing tests in different test modes based on the regression, impact and schedule. It helps tester to reduce the test execution time and wrap up and qualify the build quickly with High quality. For example, it provides the following execution options:

- **Short**
- **Standard**
- **Long**

As specified in the front panel, by default it supports the Standard mode. By selecting the optional radio button, pops up multiple option to select the different execution modes as specified in Fig. 2. If user wants to run in Short or Long mode, user has to select Optional radio button and then select Test type which they want to run.

Fig. 2. Test modes.

**Short:**
In this mode, user can run only selected master regression test cases. Test execution will be completed within 4hrs. For example, UI verification for text box, only one value will be validated. The value will be picked up randomly.

**Standard:**
In this mode, test cases can be designed to complete within 2 days (0.5 days to 2days). For example, range of values can be validated randomly for text box.

**Long:**
In this mode, user can run the full test and it covers entire test cases. All the values can be tested for text box.

Each Testing types can be interlinked with corresponding Test Modes. For example, Weekly test can be interlinked with Long modes as it executes all the test cases.

VII. LOGGING

GUIRobo creates two log files Report Summary and Detailed Log files. Both the Log files will be created in csv format to help the user for further analysis. Report Summary Log file provides a test summary like number of passes and failures, warnings, execution details, Test Start and End Time, Elapsed time and memory details. The detailed log file will have entire failure and Pass information for each test cases and machine configuration of device under test. So that in case of failures, test engineer can easily debug and identify the root cause.

VIII. SALIENT FEATURES OF GUIROBO

GUIRobo makes UI functional testing easy and robust. Unlike other automated tools, GUIRobo does not require an expensive license and yearly maintenance fees.

The following salient features are included in GUIRobo:

- Tests wide range of environment, platforms (x86, x64 etc.) and multiple programming languages.
- Allows Functional tests to run in different modes such as Standard, Long and Full mode.
- Built-in methods to store Memory usage, Physical memory, Virtual memory USER, GDI for each test execution.
- Extensive logging helps tester to analyze and reproduce the problem quickly.
- Simplified way of creating and verifying test cases using...
robust verification APIs.

- Ability to run only certain branches and narrow down the defect.
- Support randomness of field inputs wherever applicable to increase more test coverage.
- Allows tester to run in command Line mode.
- Generates detailed log files with memory and GDI counters.
- Using Hook APIs to automate customized UI controls [7]
- Easy to validate and correlate front end and back end results- Verify UI and GPIB interface output.
- Support for customized, Owner Drawn and third party controls E.g.: Source Grid [8].

IX. CONCLUSION

GUIRobo helps test engineers to move away from recorded or programmed tests. It makes the GUI test process more stable, more efficient and, ultimately, reduces the cost while increasing the quality of the delivered product. Compare to other automation tools, GUIRobo provides quick and easiest automation solution to deliver our products on time with High quality. High investments are not necessary, not only for GUIRobo, but also for test case implementation, test case maintenance and more test coverage. User just need only few dlls and it does not require huge memory to install and run the GUIRobo. Only in this way, each new UI functional requirements can be tested thoroughly and efficiently on time.

REFERENCES


Mohan Doss Gandhi holds a master degree in applied electronics from PSG CAS, Coimbatore, Tamilnadu, India and is currently pursuing a doctoral program (Ph.D.) in software quality engineering at Hindustan University, Chennai.

He is presently employed in Microsoft India (R&D) Pvt. Ltd, Hyderabad, as a software engineer. Overall, he has 15+ years of experience in the IT industry.

He has vast experience in GUI test automation tool development and has done a lot of research in GUI test automation. So far Mr. Gandhi has published one paper each on ICCTD2011 and JICTE journal and three poster papers to PNSQC.