



International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Self-Healing Test Automation Framework Approach in Testing

Rashmi Popli

Department of Computer Engineering, J.C. Bose University of Science and Technology, YMCA
Faridabad, Haryana, India, rashmimukhija@jcboseust.ac.in

ABSTRACT :

Test Automation has become the need of the hour for the software development industry and the reasons behind, are the benefits it provides - confidence on build quality, lower turnaround time, quick releases, remove repetitive job from tester's scope are few of them. In this research work an adaptive autonomous Self-Healing Test Automation Framework Approach (SHTAF) is proposed. It reduces manual maintenance effort of automation framework. This approach provides ability for the automated test scripts to find the root cause of failure and automatically heal themselves as soon as it fails and thus reduces the maintenance effort. It also helps in reducing turnaround time for a test cycle to execute and report issues which ultimately supports quicker delivery of software applications.

KEYWORDS: Test Automation Framework, Self-healing, Application Under Test (AUT), Test Scripts.

Introduction

Test automation framework, if not implemented appropriately itself can be a big challenge for automation. The test automation solution must not only focus on today's application context but instead should also see application context 2-3 years down the line in terms of tools, technology etc[1]. All software applications projects also include test automation scripts (either unit testing scripts or UI testing scripts) which are written to test different feature and business processes. If these test scripts fail to execute properly then it could trigger test automation to fail, making it a big challenge in automation testing [2].

2. Literature Review

Test automation will be very difficult to successfully implement if the testing is not planned in a structured way or is considered optional, or if while doing testing there is a general unconstructiveness and lack of enthusiasm to implement processes. Rigid deadlines and a high schedule pressure also increase the risk for deviations from processes and automation failure that may cause problems later. If the automated test is perceived as an impediment by its users, there is a risk that the testing is skipped or that the automation is abandoned in favor of manual work. Table 1 presents the critical analysis of literature work. Automated Testing is carried out to scale back testing time drastically - A typical automated test suite will run in much less than 24 hours. For a sophisticated product, manual testing may require dozens of workforce months to perform the similar testing.

Table 1: Critical Analysis of Literature Work

S.No	Year	Author	Title	Premises	Critical Analysis
1.	2017	Kim Kunka et.al	Developing a test automation framework for Agile Development and Testing[3]	The author developed a test automation framework for Agile Development and Testing. Also implemented test automation framework by combining the features of Software testing automation framework & FitNesse.	The new framework referred as NTAF (NHN test automation framework) is an open-sources powerful as it eliminates the drawbacks of Testing Automation Framework & FitNesse.

2.	2018	Milad Hanna, Amal Elsayed Aboutabl, Mostafa-Sami M. Mostafa	Automated Software Testing Framework for Web Applications [4]	The proposed work aims to provide automated software testing Framework which tentatively resolves the limitations of traditional software testing frameworks especially for web applications.	The proposed framework can ease the process of creating test scripts automatically specially when the software is updated again and again and needs regression testing to confirm stability of software.
3.	2019	Ayo Adekanmi	Research on software testing and effectiveness of automation testing [5]	The paper describes how automation testing can be effectively applied to the software projects using different testing tools.	Although automation has brought a large storm in the world of testing but still around 60% work needs to be done manually. The automation can't be considered as a substitution of manual testing.
4	2021	P. C. Shekhar	Next-Gen Test Automation in Life Insurance: Self-Healing Approaches[6]	Presents an architecture for self-healing test automation targeted at complex insurance systems with shifting APIs and UI; demonstrates reduced regression time in case study.	Useful domain case study showing practical gains; however, evaluation limited to one enterprise setting and lacks open datasets for reproducibility
5	2023	Venugopal Tamraparani & Aryendra Dalal	Self generating & self healing test automation scripts using AI for automating regulatory & compliance functions in financial institutions[7]	Proposes AI routines to auto-generate tests and perform locator repair for compliance workflows in finance; emphasizes regulatory traceability.	Strong on application (regulatory traceability) but partly grey literature / industry-oriented; methods and metrics for healing decisions are not described in full reproducible detail
6	2024	M. S. Bari et al	AI-Augmented Self-Healing Automation Frameworks [8]	Describes architectures that combine ML models (feature/locator matching, similarity scores) and adaptive strategies to automatically fix broken test locators and revalidate tests.	Shows promising reductions in maintenance, but experiments often use synthetic UI changes; real-world robustness (flaky tests, false positives) needs broader benchmarking.

2. Proposed Self-Healing Test Automation Framework (SHTAF) Approach

Self-healing is not a new term to the industry now. This is one of the buzzword which is catching the eye of everyone who is into this IT profession. People have already made a lot of effort to make their system equipped with this feature and indeed, test automation systems are also not untouched with this. Test automation tools development companies have not only started thinking about it but few of them have already implemented it (or at least claiming) and it has become one of the major selling point of their tools. It looks like it is the best time when a test automation engineer should also start thinking to make the framework ready with self-healing capability.

“How Self-healing can be applied to test automation” is the main topic of research of this paper and in next few sections, self-healing approach and its implementation is described. The proposed Self-Healing Test Automation Framework (SHTAF) approach can be applied in any type of automation framework be it a data-driven framework, keyword driven framework or hybrid type framework.

With respect to test automation, self-healing is an ability for the test scripts to automatically heal themselves upon failure. All test automation frameworks/tools interact with the application (AUT) and perform certain action on it as per the business scenarios. They also gather some information from application as a pre-requisite and use it to identify the elements/objects on the UI.

2.1 Workflow of Test Automation Framework with Traditional Manual Maintenance Approach

A system can be designed robust enough to self-healed itself for all type of failures but in this research paper, Self-healing technique has been applied to the failure reason when the test automation framework is ineffective to identify the object on application UI based on the locator's type and value given in the OR. If self-healing is not applied to this failure reason, then automation expert has to do this task manually which includes following steps:

Step 1: Identify the failure reason. If “object not identified” is the primary reason of failure then OR need to be updated with correct locator and its value.

Step 2: Automation experts go the application in non-testing mode and locate the object manually. He then tries to find its new locator type and its value with the help of some Inspector tool by going through the DOM structure.

Step 3: Once identified, he opens the object repository and update the old object properties with these new properties (locator type and value).

Step 4: Finally, he re-executes the test method to validate that new properties are working fine, and test methods is successfully passed.

Step 5: These activities take approximately 10 minutes time of an automation expert which is huge if more than 25-30 failures are in one execution cycle.

2.2 Workflow of Test Automation Framework with Proposed SHTAF Approach

If all these activities are being done by any system which automatically identifies the failure reason and fix it, it will save huge about of time. Following steps will be carried out by the system to heal it robotically:

- Step 1: As soon as a test script fails, self-healing engine get to know about this failure, and it tries to find the failure reason. If the failure reason is “object not identified”, it detects the former properties of the missing object.
- Step 2: It pulls all the properties (all locator type and their values) of that missing object from the database which can be used to identify it on UI.
- Step 3: It applies some data analytics to get the best match of the locator based on certain condition and try to execute the test method (at the same point where it halted) with this new locator.
- Step 4: If this new locator works best with the test method, this value is being updated in the object repository so that future test runs used this new locator for test execution.
- Step 5: If this new locator doesn’t work, self-healing engine again find the next best match and use it for test execution. This system tries it again and again till a new locator works successfully for test execution.
- Step 6: It has been observed that this self-healing engine takes only 0.5 seconds to perform all these operations, which is negligible if it is compared with the manual process and thus saves complete 15 minutes for one failure (Figure 3).

2.3 Working of Self-Healing engine in Proposed SHTAF Approach

Self-healing engine is another system which works in parallel along with automation framework and come into action as soon as automation framework starts doing its job (testing the application) and constantly monitor the state of framework. Once it finds the automated script failure, it immediately tries finding out the root cause of the problem and provide the solution. The Figure 4 shows the implementation of proposed self-healing engine as a part of test-automation solution.

Test automation framework constantly interacts with the Application Under Test (AUT). In a normal case, when it is not combined with self-healing engine, it only performs action on AUT and it doesn’t take any information or data from the application. But when it is combined with the self-healing engine, this engine constantly monitors the state of application and takes information from DOM which is necessary to work for the engine. The information necessary for self-healing engine is locator’s type and its value for all the objects on application. It not only gets this information but also checks that application objects are recognizable from this data, and once verified, this information is stored in the database.

Self-healing engine has two major components, one component is to examine the current state of framework, to check if any test script is getting failed or not. Once it gets to know that a script is failed, it starts investigating the reason for it. And if it is the same reason, for which this self-healing engine is designed, the second component comes into the action and starts resolving the issue by executing the necessary steps.

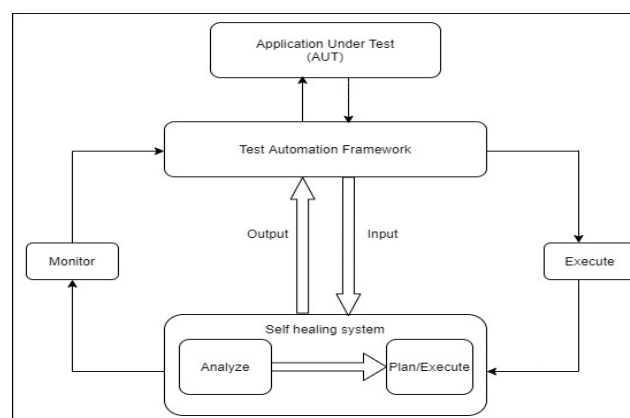


Figure 4: Implementation of Proposed SHTAF Approach in Test Automation Framework

2.4 Role of Self-healing Engine in SHTAF Approach

Updating locator’s data base: Self-healing engine starts working as soon as automation framework starts testing the application. Test scripts jumps from one page to another page performing certain actions on these pages and in parallel, self-healing engine analyzes the objects, its locators and their values on that particular page and updating the database accordingly.

Healing an object: Self-healing engine can easily verify that the failure object is present in the historical data. It will start comparing historical data with the current set of data. Assuming that multiple properties of the same object are present in the historical data. This engine will fetch the next highest weighted property of the same object and will pass it to the automation framework. Framework will restart the execution as soon as it gets the new property without failing and stopping the current test method execution.

3. Test Automation Reports

Once this self-healing engine starts working along with test automation solution, it has to be presented somewhere that it is working as expected. Reports are the best place to see the overall status of execution cycle which shows the test scripts status as passed, failed or skipped, total number of test suits with no. of test methods in each test suits, test environment, test release name, start and end time of a test method. Few reporting utilities also show no. of steps executed in each test methods with a screenshot of application at that step. Once whole test execution is completed, test automation reports are also the best place to present self-healing approach with all the activities performed in it.

Test reports should clearly present the step where exactly test automation framework is failed and what is the actual failure reason. Here, actual error is that framework doesn't have right locator in the OR to identify the object. Test report should also present the failed object name, its locator type, locator value and if possible screenshot of the object. And once self-healing engine works, it finds the best locator from database, restarts the test execution and updates the OR with this locator, all these events must be recorded in the automation report.

3.1 SHTAF Approach represented in Logs and Reports

Test automation results are always presented in the test reports to the project stakeholders and every test automation framework/tool used to have its own default report for status reporting. A selenium-based automation solution normally uses JUnit or TestNG as basic framework which also provides their inbuilt reports. Although these built-in reports provide enough information about the executed steps as part of the test methods, but they still need more customization to be more comprehensive with full of features.

In this case study, another external utility extent report is being implemented which is a customizable HTML report developed. It can be well integrated into Selenium based automation solutions using JUnit and TestNG frameworks and provides PASS, FAIL, SKIP, INFO status of a test method. To implement the SHTAF Approach, this extent report is further customised to present the status of self-healing events.

With the help of Test automation reports of the proposed SHTAF Approach, Automation Analysts can see the overall status of execution cycle which shows the test scripts status as passed, failed or skipped, total number of test suits along with test methods count in each test suits, test environment and test release information, start and end time of a test method. Once whole test execution is completed, test automation reports present SHTAF approach with all the activities performed in it.

Once self-healing engine is implemented and integrated with test automation framework, it should notify the end users about the activities done by this integrated component with complete set of evidence. Evidence in this case is which object is failed at which steps of the test method, when it is failed, which locators are found by self-healing engine in the database, out of those multiple locators which is the best locators according to the their weights, once the best locator is used for re-execution this has to be updated in OR. All this information has to be presented in the test report and ultimately informed to the end user.

A typical example of a test automation report used in the implementation is presented in the snapshot in Figure 8.

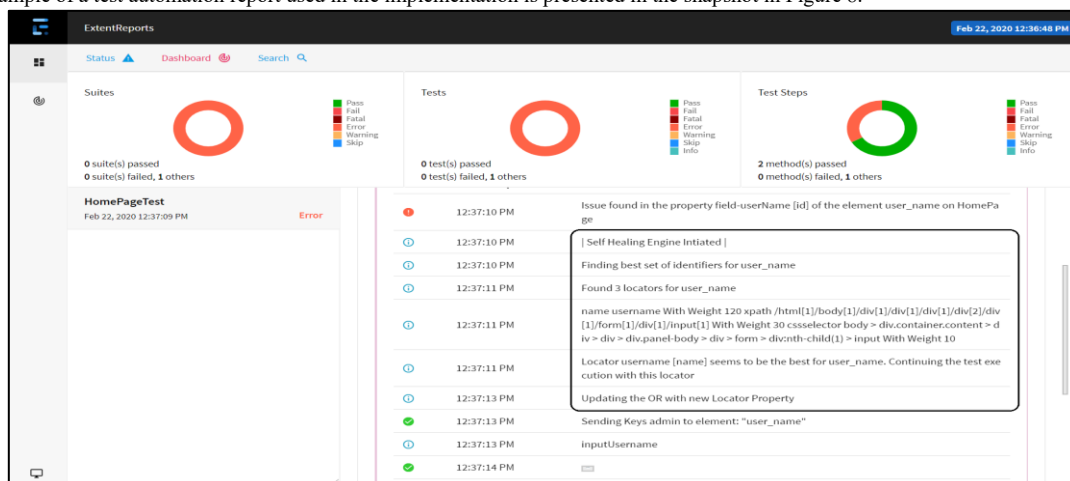


Figure 8: Report Presenting SHTAF Approach

4. Conclusion and Future Scope

This paper proposed SHTAF approach for test automation systems. There is enormous manual effort saved and huge ROI achieved once this approach is

implemented in test automation frameworks. And once this time is saved, test engineers can be involved in more productive tasks like identifying more complex scenarios, automating them and doing exploratory testing on the product instead of just correcting the failure reasons of the scripts. This work can be extended in future by making self-healing engine more sophisticated and sustainable, capable of healing any kind of script failure.

REFERENCES

- [1] S. Eldh, K. Andersson, A. Ermedahl, and K. Wiklund, Towards a Test Automation Improvement Model(taim). In 2014 IEEE Seventh International Conference on Software Testing, Verification and Validation Workshops, pages 337–342, March 2014.
- [2] E. Alegroth, R. Feldt, and H. H. Olsson, Transitioning Manual System Test Suites to Automated Testing: An Industrial Case Study, IEEE Sixth International Conference on Software Testing, Verification and Validation IEEE;Luxembourg,Luxembourg,2013,pp.56–65.
- [3] Kim Kunka et.al, Developing A Test Automation Framework for Agile Development and Testing, International Conference on Agile Processes and Extreme Programming in Software Engineering,2017.
- [4] Milad Hanna, Amal Elsayed Aboutabl, Mostafa-Sami M. Mostafa, Automated Software Testing Framework for Web Applications, International Journal of Applied Engineering Research, Volume 13, Number 11, pp. 9758-9767, 2018
- [5] Adekanmi, Ayo, Research on Software testing and Effectiveness of Automation Testing, 2019
- [6] P. C. Shekhar, Next-Gen Test Automation in Life Insurance: Self-Healing Approaches, *International Journal of Advanced Research in Computer Science and Software Engineering*, Vol. 11, Issue 5, pp. 45-52, 2021.
- [7] Venugopal Tamraparani and Aryendra Dalal, Self Generating & Self Healing Test Automation Scripts using AI for Automating Regulatory & Compliance Functions in Financial Institutions, *International Journal of Advanced Computer Science and Applications (IJACSA)*, Vol. 14, No. 2, pp. 210-216, 2023.
- [8] M. S. Bari, A. Rahman and T. Ahmed, AI-Augmented Self-Healing Automation Frameworks, *International Journal of Software Engineering and Applications (IJSEA)*, Vol. 15, No. 1, pp. 33-42, 2024.