

Self healing to overcome the instability of automation tests



Table of contents

Abbreviations	3
Abstract	3
Market challenges	3
Solution	4
Benefits	5
Limitations	6
References	6
Author information	7

Abbreviations

SDLC	Software development life cycle
ROI	Return on investment
GUI	Graphical user interface
DOM	Document object model
AI/ML	Artificial intelligence / Machine mearnings
AUT	Application under test

Abstract

Testing plays a crucial role in delivering a bug-free product, especially in this digital world, where companies strive for rapid time-to-market and frequent upgrades. Relying solely on manual testing to address such rapid releases may not align with product release timelines. Automation of the test scenarios can significantly reduce testing times and increase test coverage.

However, automation also introduces its own limitations. Any change in the product features necessitates relevant changes in automation scripts, adding to maintenance efforts. Additionally, false failures during automation test execution require additional time to analyze the test results. Even minor intentional Graphical User Interface (GUI) changes demand substantial maintenance effort. Implementing a self-healing mechanism can mitigate such unnecessary maintenance efforts.

Market challenges

Automation is a cost-effective solution to address the product's frequent releases and features coverage in Agile and DevOps development models. However, as described above, it requires manual efforts for analysis and test maintenance. Market trends suggest that 15%–20% of the overall test automation budget will be needed for maintenance alone.

Some of the automation challenges are:

- Broken Element Locator, which leads to test failures
- NoSuchElementException test failures
- Known Element location change, which leads to test failures
- Known Element property change, which leads to test failures

- Additional effort to analyze the false failures
- High maintenance effort to make changes in existing automation scripts

Solution

Existing automation tools or frameworks in the market addressing the above challenges are limited and often expensive, with restrictive coverage. HCLTech's self-healing solution stands out by tackling most of these challenges. Integrated with our Falcon Test Automation Framework (a patented AI/ML-based UI automation framework), this solution offers a no-code or scriptless approach.

Our self-healing solution automatically detects element changes, locator changes, control changes and HTML DOM structure changes and fixes the test scripts accordingly at run time. This eliminates the need for SDETs or engineers to spend additional effort analyzing failure logs to fix the test scripts and re-execute them.

The process is explained in the diagram below.

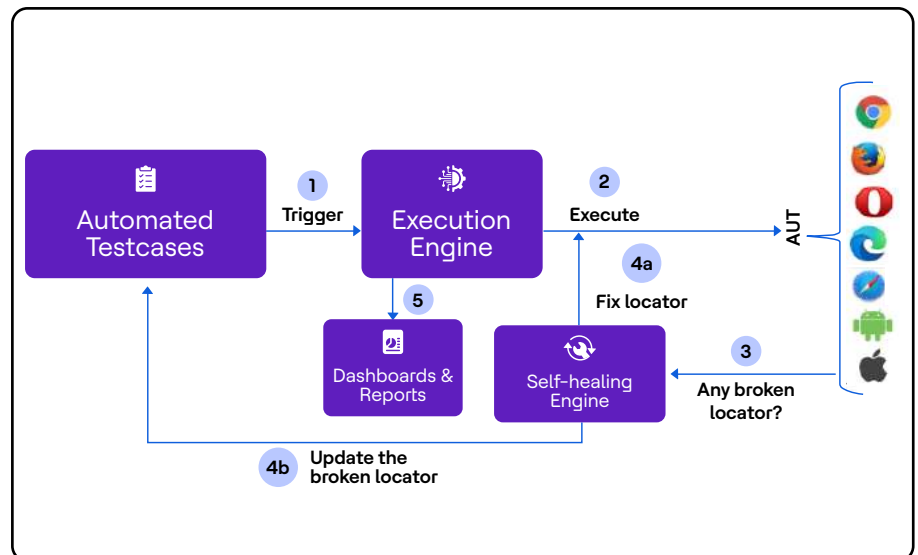


Figure 1: Solution process flow diagram

As demonstrated in Figure 1, the Automated Testcases triggers (1) the Execution Engine to run (2) the test scripts against AUT on a targeted browser. If any broken locators are encountered in the Automated Testcases the Engine will invoke the Self-healing Engine (3) to rectify these broken locators. The Self-healing Engine will fix the broken locators (4a) by getting the new locator details from AUT and executing the test case without stopping the execution or failing the test cases. The self-healing engine will also update the Automated Testcases with new details (4b) once the execution is complete, and reports will be generated for analysis.

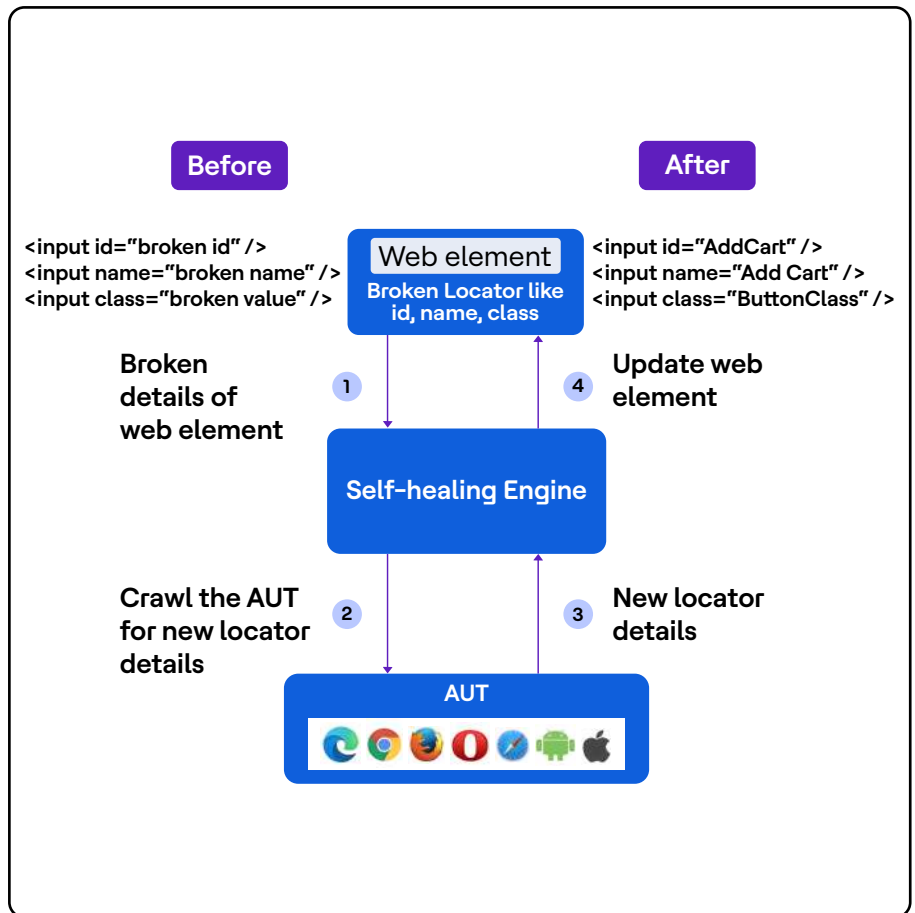


Figure 2: Self-healing process

As shown in Figure 2, if any web element details are broken or changed in the new build (1) those web element details will be shared with the Self-healing Engine, which will crawl the AUT (2) with available details and get the new element details (3); the Self-healing Engine will then update the test scripts (4) with these new element details.

Benefits

- Allows users to overcome the instability of automated test scripts
- Saves 40%-50% of maintenance efforts
- Improves test accuracy and reliability
- With fewer false failures, the failure analysis time is automatically reduced
- No need to manually fix the broken locators, which leads to low maintenance efforts
- Auto handling of NoSuchElementException exceptions at runtime
- Auto handling of element location changes and design layout changes

- Auto handling of element property changes
- Fewer (or even no) code changes to handle element property changes
- Improves test coverage
- Addresses both web and mobile automation tests
- Improves test coverage and release timelines

Limitations

Self-healing of a test case will not be possible in all cases. The limitations are listed below:

- Self-healing is not applicable if the functionality itself is changed
- Self-healing is not possible if all the properties of an element are changed

References

<https://testsigma.com/blog/self-healing-tests-maintenance-testsigma/>

<https://blog.qasource.com/a-complete-guide-to-self-healing-test-automation>

<https://www.accelq.com/blog/self-healing-test-automation/>

<https://www.nextgenerationautomation.com/post/how-self-healing-automation-works>

Author information



Kiran Ayyagari

Kiran Ayyagari is a Solutions Architect with over 19 years of experience in IT, including test automation, framework design, tool development and application testing. He spent more than 8 years at multiple geo locations in Europe and the US, working with top IT customers. He has extensive knowledge in UI, API and Mobile test automation using both open-source and commercial automation tools/frameworks



Narender S

Narender S is a Senior Solutions Architect with over 20 years of experience in product engineering and sustenance engineering and testing in various domains, including product, platform and enterprise customers. Strong experience in creating AI/ML-based automation solutions to address the current market challenges and improve the efficiency of digital customers.

HCLTech | Supercharging Progress™

HCLTech is a global technology company, home to 222,000+ people across 60 countries, delivering industry-leading capabilities centered around Digital, Engineering and Cloud powered by a broad portfolio of technology services and software. The company generated consolidated revenues of \$12.3 billion over the 12 months ended December 2022. To learn how we can supercharge progress for you, visit hcltech.com.

hcltech.com

