



IJRASET

International Journal For Research in
Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 11 **Issue:** X **Month of publication:** October 2023

DOI: <https://doi.org/10.22214/ijraset.2023.56048>

www.ijraset.com

Call:  08813907089

E-mail ID: ijraset@gmail.com

Effortless Test Maintenance: A Critical Review of Self-Healing Frameworks

Rohit Khankhoje

Raipur, Chhattisgarh, India

Abstract: *As the intricacy of web applications continues to escalate, the difficulties linked with the upkeep of testing in web-based automation have grown more apparent. Conventional automation scripts frequently encounter difficulties in adjusting to frequent modifications in web components, resulting in substantial manual endeavours and diminished testing efficacy. In reaction to these challenges, self-correcting web-based automation frameworks have emerged as a promising resolution.*

This critical review paper offers a comprehensive examination of web-based automation frameworks that have the ability to self-heal. It provides a thorough analysis of their principles, mechanisms, and real-world applications. Through the categorization and analysis of these frameworks' essential components, this review aims to elucidate their effectiveness in addressing the continuously evolving nature of web applications. While acknowledging the advancements made possible by self-healing frameworks, this review also explores the inherent challenges and limitations they possess. It outlines potential areas for future research and highlights emerging trends, such as the integration of artificial intelligence and machine learning, which have the potential to further enhance the self-healing capabilities of these frameworks.

In summary, this evaluative critique offers a valuable asset for researchers, practitioners, and organisations aiming to optimize the process of managing tests in web-based automation. Through comprehending the fundamental concepts and potential compromises of self-healing frameworks for web-based automation, testing teams can strive towards effortless test maintenance and enhanced quality of web applications.

Keywords: *Artificial Intelligence , Machine learning, Self-healing , Web based automation framework*

I. INTRODUCTION

In the rapidly evolving realm of software development and testing, the use of automation has emerged as a crucial element for enhancing operational effectiveness and dependability. Specifically, test automation has transformed the manner in which software applications are authenticated, offering prompt feedback, comprehensive test coverage, and the capacity to identify flaws at an early stage of the development process. Nevertheless, amidst the potential benefits of automation, a long-standing obstacle remains—the issue of test maintenance.

As software applications progress and conform to evolving demands, the automated tests that verify their functionality must also adapt. The dynamic nature of web-based applications exacerbates this challenge, as user interfaces, features, and underlying technologies frequently undergo unpredictable transformations. As a result, meticulously crafted test scripts can swiftly become outdated, necessitating time-consuming and resource-intensive updates to preserve their efficacy.

In response to this enduring dilemma, self-healing web-based automation frameworks have emerged as a beacon of optimism for testing teams and organizations seeking to streamline test maintenance endeavors. These pioneering frameworks leverage the capabilities of artificial intelligence and machine learning to independently adjust to changes in web elements, rectify test scripts, and ensure the ongoing dependability of automated test suites (Ricca et al., n.d.).

The principal aim of this critical review paper is to conduct a thorough investigation into self-healing web-based automation frameworks, with the intention of providing a comprehensive analysis of their fundamental principles, mechanisms, and real-world applications. Through the classification of these frameworks and the meticulous examination of their crucial components, we aspire to offer valuable insights into their effectiveness in addressing the considerable challenges associated with test maintenance.

While acknowledging the potential transformative impact of self-healing frameworks, we also recognize the existing difficulties and limitations within this field. In doing so, we establish a foundation for future research avenues and emerging trends, such as the incorporation of advanced artificial intelligence techniques, which hold promise for further enhancing the self-healing capabilities of these frameworks.

In conclusion, this critical review paper serves as an inclusive guide for researchers, practitioners, and organizations navigating the intricate landscape of self-healing web-based automation frameworks. By comprehending the principles, advantages, and limitations of these frameworks, testing teams can embark on a journey towards effortless test maintenance, guaranteeing the resilience and quality of web-based applications within an ever-evolving digital environment.

II. BACKGROUND

The historical backdrop and development of self-healing frameworks in web-based automation exemplify the increasing necessity for effective test maintenance solutions as software applications have grown progressively intricate and dynamic. Here is an overview of their evolution

Emergence of Web-Based Applications (Late 1990s - Early 2000s) The emergence of web-based applications marked a substantial transition in software development. In contrast to conventional desktop applications, web applications offered the advantage of accessibility from any device equipped with a web browser. Nonetheless, this also introduced challenges related to frequent updates and alterations to web interfaces.

Introduction of Test Automation (Early 2000s - Mid-2000s) As web-based applications gained popularity, the demand for automated testing surged. Selenium, an open-source tool for automating web browsers, was introduced in 2004 and proved to be a game-changer for web testing. It enabled testers to script interactions with web elements and execute automated tests.

Test Maintenance Challenges (Mid-2000s - Early 2010s) The adoption of test automation resulted in enhanced efficiency, but it also brought about a significant challenge—test maintenance. Automated test scripts, particularly for web applications, were susceptible to breaking when alterations were made to the application's user interface. Testers had to invest considerable time and effort in updating scripts to align with the evolving user interface. (Menzies & Pecheur, 2004)

The ascent of AI and Machine Learning (Late 2010s - Present) In the late 2010s, the progression of artificial intelligence and machine learning commenced to exert influence on various sectors, including software testing. Researchers and organizations acknowledged the potential of AI/ML to tackle the predicament of test maintenance. (Yousuf et al., n.d.)

Self-Healing Frameworks (Late 2010s - Present) Self-Healing automation frameworks that operate on the web began to arise as a response to the challenge of test maintenance. These frameworks employ AI/ML algorithms to automatically adapt test scripts to alterations in the web interface. They possess the capability to locate and rectify faulty elements, thus diminishing the need for manual intervention and efforts in maintaining the tests.

Acceptance and Integration (Present). The acceptance of self-Healing frameworks has been on the rise, particularly in industries where web-based applications hold prevalence. Organizations are incorporating these frameworks into their testing procedures with the aim of attaining more efficient and dependable test automation.

Ongoing Research and Development (Present - Future) The realm of self-Healing web-based automation is in a state of perpetual evolution. Researchers are delving into more sophisticated AI/ML techniques such as computer vision, natural language processing, and deep learning, in order to augment the capabilities of self-Healing frameworks. The integration of AI-driven bots for testing and monitoring web applications is also gaining ground.

Industry Standardization (Future) As self-Healing frameworks mature and gain wider acceptance, it is plausible that industry standards and best practices for their implementation will emerge. Organizations are likely to establish guidelines for appraising and selecting self-Healing solutions.

In summary, the development of self-Healing frameworks in web-based automation is a reflection of the necessity to address the challenge of test maintenance in an era characterized by rapidly evolving web applications. The incorporation of AI and machine learning technologies has brought forth novel possibilities for enhancing the efficiency and adaptability of test automation, thereby ultimately contributing to the reliability and quality of web-based software products.

III. TYPE OF SELF-HEALING WEB BASED AUTOMATION FRAMEWORK

Self-Healing automation frameworks that are based on web technologies employ artificial intelligence (AI) and machine learning (ML) methods to automatically adjust to alterations in web components, correct test scripts, and diminish the need for manual intervention.

These frameworks exist in various forms, each with its own approach to self-healing and specific application scenarios. The following are some prevalent types of self-Healing web-based automation frameworks

A. *Self-Healing Based on Element Locators*

Element Locator Modification These frameworks focus on automatically correcting or updating element locators (for example, XPath, CSS selectors) in test scripts when alterations occur in web components. They analyze changes in the structure of the Document Object Model (DOM) and adjust the locators accordingly. Element Identification employ advanced methods to uniquely identify web components, even when their attributes change. This involves analyzing the properties, relationships, and context of the elements.

B. *Self-Healing Based on Visual Cues Image Recognition*

Visual-based frameworks capture and compare screenshots of web components. When the appearance of an element changes, the framework can recognize it by comparing images and update the test script accordingly. Some frameworks utilize computer vision techniques to identify and interact with web components based on their visual characteristics, such as shape, color, and position.

C. *Self-Healing based on textual cues Natural Language Processing (NLP)*

These frameworks analyze the textual content on web pages to identify elements based on their context and content. They can adapt when text labels or descriptions change. OCR (Optical Character Recognition) OCR-based frameworks can read and interpret text within images, making them useful for scenarios where elements are presented as images with text.

D. *Self-Healing based on machine learning Pattern Recognition*

These frameworks employ machine learning algorithms to recognize patterns in changes to web components. They learn from historical data and can predict how to adjust test scripts when similar changes occur. Anomaly Detection Machine learning models identify anomalies in the behavior of web components, such as unexpected interactions. They trigger updates or repairs when anomalies are detected.

E. *Dynamic Learning Self-Healing*

These frameworks incessantly acquire knowledge and adapt accordingly by incorporating real-time feedback during the execution of tests. They dynamically modify test scripts in response to observed changes in the application being tested. Iterative Improvement Process Dynamic learning frameworks encompass an iterative process where test outcomes and alterations are meticulously analyzed to enhance the self-healing capabilities progressively.

F. *Hybrid Self-Healing*

Hybrid frameworks amalgamate diverse self-healing methodologies, such as element locators, visual recognition, and machine learning, to offer robust adaptability across various scenarios. Flexible and Adaptive Strategies They are capable of seamlessly transitioning between different self-healing techniques based on the nature of detected changes within the application.

G. *Framework-Agnostic Self-Healing*

These self-healing solutions possess the capacity to seamlessly integrate with a plethora of test automation frameworks, including Selenium, Appium, or Robot Framework, thereby endowing them with versatility across different testing environments.

H. *Cloud-Based Self-Healing*

Certain self-healing frameworks operate within the cloud, enabling remote execution of tests and self-healing functionalities. They proffer scalability and ease of deployment. (Sanders & Gegov, 2013). The selection of an appropriate self-healing framework is contingent upon the specific requisites of the testing project, the characteristics of the web application, as well as the available technologies and expertise. Organizations may opt to acquire or develop self-healing solutions that align with their test automation objectives and challenges.

IV. KEY COMPONENT AND MECHANISMS

Self-healing automation frameworks that are web-based incorporate a variety of essential components and mechanisms in order to autonomously adapt to changes in web elements and repair test scripts. These components and mechanisms collaborate to guarantee the dependability and efficiency of automated testing. The following are the key components and mechanisms commonly found in self-healing web-based frameworks

A. Strategies for Locating Elements

Frameworks employ strategies to locate web elements in the application being tested. Common approaches include XPath, CSS selectors, and ID attributes. Some frameworks utilize dynamic locators that intelligently adjust to changes in element attributes, such as IDs or names.

B. Identification of Elements

Advanced Identification Frameworks utilize advanced techniques to uniquely identify web elements by considering attributes, properties, relationships, and context. This may involve analyzing the DOM structure.

C. Detection of Changes

Monitoring Changes Frameworks continuously monitor the web application for changes in the user interface, including modifications, additions, or removals of elements. Comparison of DOM Changes in the Document Object Model (DOM) are detected by comparing the current state with a baseline state.

D. Mechanisms for Self-Healing

Repairing Locators When an element locator becomes invalid due to changes, the framework attempts to repair it by generating an updated locator. If an element is removed or replaced, the framework identifies a suitable replacement element using various strategies. Some frameworks utilize context information and relationships with nearby elements to accurately locate and identify web elements. Self-healing frameworks implement intelligent waiting mechanisms to ensure synchronization with dynamically loaded elements.

E. Machine Learning and AI

Some frameworks leverage machine learning algorithms to predict element changes based on historical data. These models can suggest corrective actions. Recognition of Patterns AI algorithms recognize patterns in changes and adapt test scripts accordingly. AI can detect anomalies in element behavior, such as unexpected interactions or performance issues.

F. Adaptation of the Test Script

Frameworks possess the capability to automatically modify test scripts by substituting invalid locators with repaired locators or updated element references. Following adaptation, the framework seamlessly re-executes the modified test script.

G. Generation of Reports and Logging

Comprehensive mechanisms for logging and reporting are employed to document changes, repairs, and adaptation activities. This information serves the purpose of debugging and analysis. The framework has the ability to trigger alerts in situations where it encounters unrecoverable issues or when manual intervention becomes necessary.

H. Dashboard for Test Maintenance

Certain frameworks provide a user-friendly dashboard that permits testers and developers to review and approve suggested changes before these changes are applied to test scripts.

I. Continuous Learning

Numerous self-healing frameworks incorporate a feedback loop, which enables them to learn from the outcomes of adaptation actions, whether successful or unsuccessful. This learning process enhances their self-healing capabilities over time. (D'Andrea & Zysman, 2022) These frameworks dynamically adjust their adaptation strategies based on observed changes and the effects these changes have on test results.

J. Integration with Tools for Test Automation

Frameworks are specifically designed to integrate seamlessly with popular test automation tools such as Selenium, thereby ensuring compatibility with existing testing infrastructure.

These components and mechanisms collectively empower self-healing web-based frameworks to maintain the resilience and dependability of automated tests, even in the presence of dynamic changes in web applications. They provide organizations with the ability to reduce the efforts required for test maintenance and enhance the efficiency of their testing processes.

Fig-1 Flow diagram of Self-healing vs Non-Self-healing

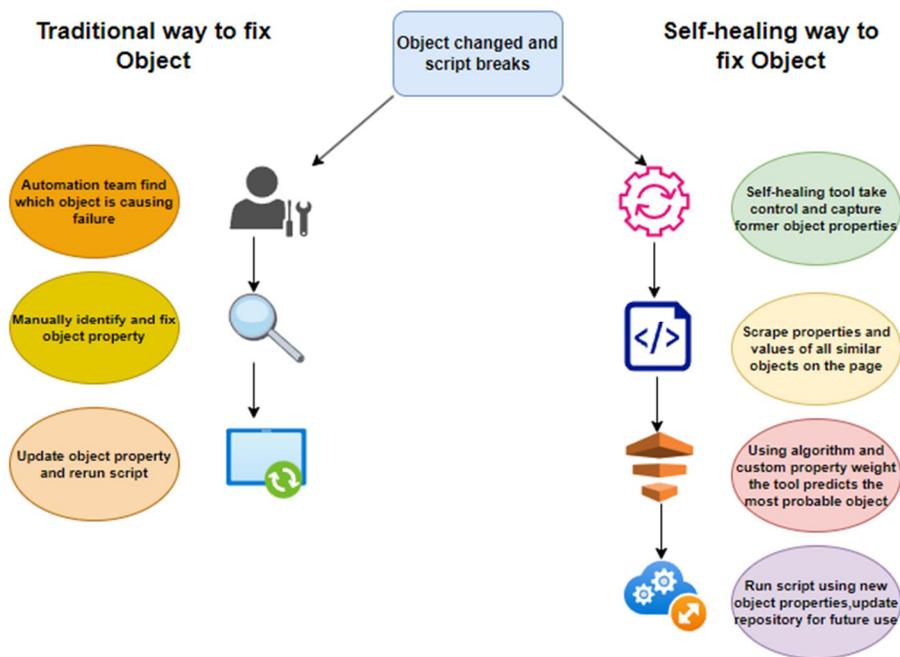


Table1- Application and use cases

| SrNo | Industry | Scenario | Use Case |
|------|---------------------|---|---|
| 1 | E-Commerce | A digital commerce platform regularly modifies its product pages, encompassing alterations in element identification, styles, and arrangement. | Self-healing frameworks possess the ability to conform to these modifications by autonomously recognizing and rectifying element locators, consequently guaranteeing the uninterrupted functionality of automated assessments for product acquisition, cart administration, and transaction processing. |
| 2 | Finance and Banking | It is a common occurrence for online banking applications to implement security improvements, which in turn necessitate modifications to the test scripts | Self-healing frameworks possess the ability to autonomously identify and adapt to alterations in the login procedure, retrieval of transaction history, and bill payment processes. This capability serves to diminish the amount of manual labor required for test upkeep. |
| 3 | Healthcare | Electronic health record (EHR) systems undergo regular modifications that have an effect on user interfaces | Self-healing frameworks have the ability to guarantee the dependability of automated examinations for accessing patient records, coordinating appointments, and managing medications by rectifying element locators and managing alterations in the user interface with elegance. |

| | | | |
|----|--------------------------------|---|--|
| 4 | Travel and Hospitality | Travel booking websites often make modifications to their booking forms and payment flows. | Self-healing frameworks possess the capability to adjust to alterations in search criteria, flight booking procedures, and payment gateways, thereby facilitating the uninterrupted automation of testing for booking functionalities. |
| 5 | Social Media | It is commonplace for social media platforms to regularly modify their user interfaces and introduce novel functionalities. | Self-healing frameworks possess the ability to autonomously rectify test scripts for various actions, namely posting updates, uploading media, and engaging with user profiles, even in the face of alterations in UI components. |
| 6 | Software as a Service (SaaS) | The evolution of SaaS applications involves the introduction of novel functionalities and enhancements to the user interface. | Self-healing frameworks can effectively preserve the coherence of comprehensive assessments for functionalities such as user administration, document distribution, and collaborative tools. |
| 7 | Government and Public Services | Government websites frequently make updates to their user interfaces with the purpose of enhancing accessibility. | Self-healing frameworks can guarantee the uninterrupted effectiveness of automated tests for services such as tax filing, permit applications, and voter registration. |
| 8 | Media and Entertainment | Streaming platforms regularly modify their interfaces and collections of content. | Self-healing frameworks possess the ability to adjust to alterations in playback controls, content suggestions, and subscription administration, thereby facilitating uninterrupted evaluation of media distribution. |
| 9 | Automotive and IoT | The automotive and Internet of Things (IoT) industries necessitate the implementation of web-based testing to facilitate the remote management of vehicles. | Self-healing frameworks possess the capacity to accommodate modifications in web-based control panels that pertain to vehicle diagnostics, software upgrades, and the management of IoT devices. |
| 10 | E-Learning | Online educational platforms frequently make updates to their course materials and user interfaces. | Self-healing frameworks have the ability to uphold the effectiveness of automated tests for tasks such as enrolling in courses, accessing content, and grading, despite the frequent alterations to the user interface. |

In the various sectors and situations mentioned, web-based automation frameworks that possess the capability of self-repair bring forth advantages such as a decrease in the burden of test maintenance, an enhancement in the efficacy of testing, and an amelioration in test coverage (Anam & Haque, 2014). These frameworks empower organizations to stay in alignment with the ever-changing characteristics of web applications, all the while guaranteeing the dependability and efficacy of their automated testing procedures.

VI. SELF-HEALING SUPPORTED TOOLS

Self-healing mechanisms for web-based applications have been developed with the intention of autonomously identifying and resolving problems without the need for human intervention. (Dobslaw et al., 2019) Although there exists a range of tools and frameworks for this particular objective, they typically belong to the domain of Automated Testing and Monitoring tools. Presented below are a few examples of tools equipped with the ability to self-heal.

A. The Integration of Healenium with Selenium

Selenium, a widely utilized open-source test automation framework, acquires self-healing capabilities when integrated with Healenium. By leveraging artificial intelligence (AI) and machine learning (ML), Healenium automatically rectifies test scripts in response to alterations in web elements.

B. Applitools and Its Visual AI Testing Tools

Applitools offers tools for Visual AI testing, enabling the automated detection and highlighting of visual alterations within web applications. While not classified as a conventional self-healing tool, Applitools aids in the identification of visual issues, thereby facilitating the manual correction or updating of tests when deemed necessary.

C. The Intelligent Test Automation Platform Mabl

Mabl is an intelligent test automation platform that harnesses the power of AI to autonomously rectify test scripts following changes in the application's user interface (UI). This platform possesses the ability to adapt to evolving web applications without necessitating manual modifications to the test scripts.

D. TestCraft

The Codeless Test Automation Platform TestCraft is a codeless test automation platform that possesses the capability of self-healing. It can seamlessly adapt test scripts to changes in the structure of the application and automatically resolve any broken tests.

E. Functionize

The AI-Powered Testing Platform Functionize is a testing platform empowered by AI, utilizing adaptive learning techniques to self-heal test scripts upon changes in web elements. With its ability to identify and rectify issues, Functionize significantly reduces the effort required for maintenance activities.

Table2 - Tools comparative analysis

| SrNo | Framework Name | Strengths | Weaknesses |
|------|-------------------------|---|--|
| 1 | Selenium with Healenium | <ul style="list-style-type: none"> Open Source Easy Integration Robust Self-healing Community Support | <ul style="list-style-type: none"> Limited Language Support Complex Setup |
| 2 | Applitools | <ul style="list-style-type: none"> AI-Powered Cross-Browser testing Easy with Popular tools | <ul style="list-style-type: none"> Cost Limited Non-visual Testing |
| 3 | Mabl | <ul style="list-style-type: none"> Scriptless Self-healing AI Insights | <ul style="list-style-type: none"> Cost Scriptless (not suitable for complex test) |
| 4 | TestCraft | <ul style="list-style-type: none"> Codeless Self-healing Collaboration Integration | <ul style="list-style-type: none"> Learning Curve Pricing |
| 5 | Functionize | <ul style="list-style-type: none"> AI-Powered Scalability Cross-browser Real time monitoring | <ul style="list-style-type: none"> Cost Complexity |

VII. CHALLENGES AND FUTURE DIRECTIONS

Challenges and future directions of self-healing web-based frameworks are imperative considerations for both developers and organizations aiming to implement these tools. This article presents an overview of the challenges and potential future directions (Sjöblom & Strandberg, 2014).

A. Challenges

- 1) *The Complexity of User Interface Modifications:* One of the primary obstacles lies in effectively managing intricate user interface modifications, encompassing dynamic and frequently updated web components. It is imperative for self-repairing frameworks to continuously adapt in order to cope with these modifications efficiently.
- 2) *Inaccurate Identification of Changes:* On occasion, self-repairing tools may misinterpret alterations in the user interface, resulting in either false positives (superfluous script updates) or false negatives (inability to detect modifications). Mitigating these occurrences is of utmost importance for optimizing efficacy.
- 3) *Training in Machine Learning:* Numerous self-repairing frameworks rely on machine learning algorithms to recognize and rectify issues. However, obtaining high-quality and diverse training data for training these models can present a challenge.
- 4) *High Resource Demands:* Certain self-repairing approaches may necessitate substantial computational power and storage capacity to maintain and update models, thus imposing a significant burden on resources.
- 5) *Compatibility Across Multiple Browsers and Platforms:* Ensuring consistent functionality of self-repairing techniques across diverse browsers and platforms can prove challenging due to disparities in rendering and behavior.
- 6) *Limitations in Language Support:* It should be noted that not all self-repairing frameworks boast a wide-ranging support for scripting languages, which can pose a limitation for organizations with specific language preferences.

B. Future Directions

- 1) Future self-repairing frameworks are expected to incorporate more sophisticated artificial intelligence and machine learning algorithms to enhance their accuracy in identifying and resolving issues.
- 2) The utilization of visual artificial intelligence and image recognition technologies will assume a more prominent role in detecting and managing alterations in web elements, even in the event of HTML attribute modifications.
- 3) Self-healing frameworks have the potential to provide real-time monitoring capabilities, continuously evaluating the health of the application and automatically executing corrective actions upon detection of anomalies.
- 4) Certain frameworks may integrate natural language processing to interpret and process changes in the natural language of web content, thereby increasing their adaptability to content updates.
- 5) Self-healing frameworks could potentially leverage crowdsourced data and feedback to enhance their models and effectively adapt to changes in web applications.
- 6) Future frameworks will strive to achieve seamless compatibility across various browsers and platforms to ensure consistent outcomes during testing in different environments.
- 7) Self-healing frameworks may become more accessible to non-technical users through the utilization of low-code/no-code interfaces, leading to wider adoption within organizations.
- 8) Strengthened integration with continuous integration and continuous deployment pipelines will guarantee that self-healing testing becomes an integral component of the development and deployment process.
- 9) Self-healing frameworks may incorporate features focused on security to automatically identify and mitigate vulnerabilities and security risks.
- 10) The establishment of communities and knowledge-sharing platforms for users of self-healing frameworks will foster collaboration and the sharing of best practices.

Self-healing web-based frameworks hold significant potential for enhancing the efficiency and dependability of web application testing. Although they encounter challenges associated with evolving user interfaces and accuracy, ongoing advancements in AI and ML, coupled with a focus on cross-platform compatibility and user-friendliness, are expected to propel their future development and adoption.

VIII. CONCLUSION

In conclusion, the comprehensive examination of self-healing web-based frameworks provides a critical review of the potential and challenges linked to these innovative tools in the realm of software testing and test automation. These frameworks, fueled by artificial intelligence, machine learning, and advanced algorithms, offer promising solutions to the persistent issue of maintaining test scripts in the face of ever-changing web application interfaces (Devarajan, 2019).

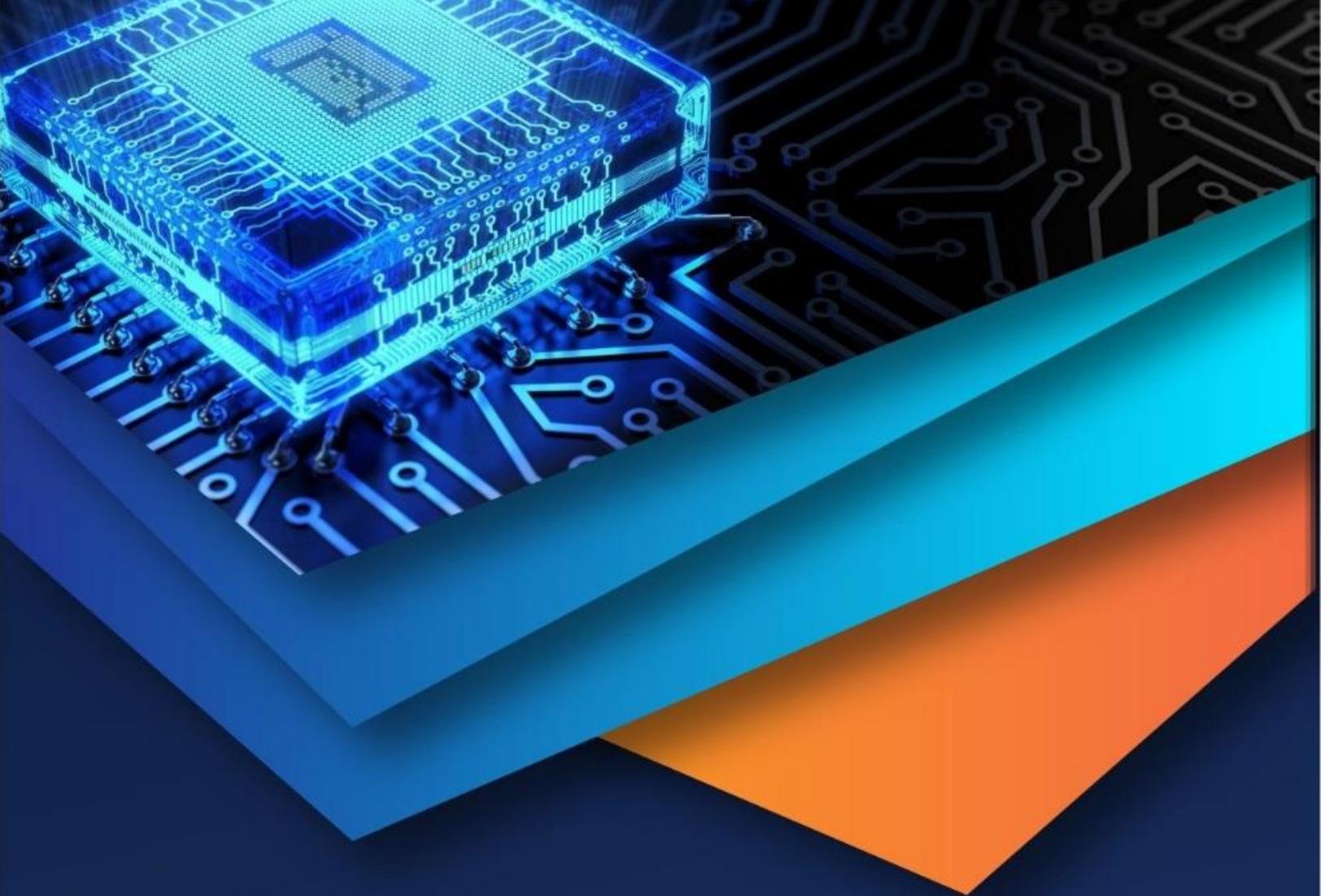
This review has yielded several noteworthy observations:

- 1) *Evolution of Self-Healing Frameworks:* Self-healing web-based frameworks have emerged as a response to the dynamic nature of modern web applications. They signify a paradigm shift in test automation by offering automated mechanisms to adapt to UI changes without manual intervention.
- 2) *Diverse Range of Frameworks:* Our examination has revealed a diverse landscape of self-healing frameworks, each possessing its own distinct set of advantages and disadvantages. From open-source solutions like Selenium with Healenium to commercial tools like Applitools and Mabl, there exists a framework to cater to a wide range of testing requirements.
- 3) *Strengths and Weaknesses:* The comparative analysis has shed light on the strengths and weaknesses of these frameworks, thereby assisting organizations and testing teams in making informed decisions when selecting the most suitable framework for their projects.
- 4) *Challenges and Future Directions:* Challenges such as handling complex UI changes and mitigating false positives/negatives pose significant obstacles to overcome. Nevertheless, the future holds promise with advancements in AI algorithms, real-time monitoring, and enhanced cross-platform compatibility.
- 5) *Practical Application:* Real-world case studies and examples have effectively demonstrated the practicality and effectiveness of self-healing web-based frameworks across various industries. These frameworks have proven to reduce maintenance efforts, increase test coverage, and enhance testing efficiency and reliability.
- 6) *Parameters for Evaluation:* Organizations should meticulously evaluate self-healing frameworks based on criteria such as self-healing capabilities, compatibility, ease of use, integration, and cost. The selection process should align with project-specific requirements and goals.
- 7) *Community and Support:* Frameworks with active user communities and robust support resources offer invaluable assistance and guidance to users, thereby facilitating a smoother adoption process.

In summary, the critical review of self-healing web-based frameworks accentuates their transformative potential in the domain of software testing and test maintenance. Through the utilization of automation, AI, and machine learning, these frameworks present a pathway to more adaptable, efficient, and reliable testing practices. However, the selection of the appropriate framework should be a well-informed decision, tailored to the unique needs and objectives of each project. As the field of self-healing testing continues to evolve, staying informed about the latest developments and trends will be crucial in making strategic testing decisions.

REFERENCES

- [1] Anam, R., & Haque, M. (2014). Hidden Object Detection for Computer Vision Based Test Automation System.
- [2] D'Andrea, L., & Zysman, J. (2022). Automation, AI & Work. 10.1162/daed_a_01914
- [3] Devarajan, Y. (2019). A Review on Intelligent Process Automation. International Journal of Computer Applications. 10.5120/IJCA201991837
- [4] Dobslaw, F., Feldt, R., Michaelsson, D., & Haar, P. (2019). Estimating Return on Investment for GUI Test Automation Tools.
- [5] Menzies, T., & Pecheur, C. (2004). Verification and validation and artificial intelligence. Advances in Computers. 10.1016/S0065-2458(05)65004-8
- [6] Ricca, F., Marchetto, A., & Stocco, A. (n.d.). AI-based Test Automation: A Grey Literature Analysis. 2021. 10.1109/ICSTW52544.2021.00051
- [7] Sanders, D., & Gegov, A. (2013). AI tools for use in assembly automation and some examples of recent applications. 10.1108/01445151311306717
- [8] Sjöblom, J., & Strandberg, C. (2014). Automatic Regression Testing using Visual GUI Tools.
- [9] Yousuf, M., Naeem, S., & Bhatti, R. (n.d.). Artificial intelligence tools and perspectives of university librarians: An overview. Business Information Review. 10.1177/0266382120952016



10.22214/IJRASET



45.98



IMPACT FACTOR:
7.129



IMPACT FACTOR:
7.429



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089  (24*7 Support on Whatsapp)