

## AUTOMATION SERVICES IN OBJECT ORIENTED FRAMEWORK IN SYSTEM SOFTWARE TESTING

Sheeba Shaista, G V S Adithya, Dr.S.Arvind (Guide)

Hyderabad Institute of Technology and Management

\*\*\*

### ABSTRACT

The use of automation services can significantly reduce the time and effort required to test complex systems, while also increasing the accuracy and reliability of test results. By leveraging the principles of object-oriented design, automation services can be integrated seamlessly into existing testing workflows, allowing for more efficient and effective testing of software systems. In addition, the use of automation services can also help to identify and address potential issues earlier in the development process, reducing the risk of costly errors and delays. Overall, the integration of automation services within an object-oriented framework represents a powerful tool for improving the quality and reliability of software systems. For a family of software systems, object-oriented frameworks offer flexible design, implementation, and testing. The classes that directly use or inherit the framework classes are called Framework Interface Classes (FICs). The challenges with automation surrounding the creation and application of FIC adaptable test scenarios are discussed in this study. Framework Interface State Transition Tester (FIST2) is a tool introduced and its prototype is created with this objective in mind. Automation and execution processes are key stages of the overall product development to both the New Product Introduction (NPI) and Production Release processes.

Keywords: FIST2, FIC.

### INTRODUCTION

Software testing is a critical component of the software development process, ensuring that systems are functioning as intended and meeting the needs of end-users. However, testing can be a time-

consuming and resource-intensive process, particularly when dealing with complex systems. This is where automation services come in, allowing for the automation of repetitive and time-consuming tasks, freeing up testers to focus on more complex and nuanced aspects of testing.

In recent years, there has been a growing trend towards using object-oriented frameworks for system software testing. Object-oriented design principles offer a powerful tool for organizing and managing complex systems, and can be leveraged to create more efficient and effective testing workflows. By integrating automation services within an object-oriented framework, testers can automate a wide range of tasks, including test case creation, test execution, and test reporting.

The benefits of using automation services within an object-oriented framework are many. By reducing the time and effort required to test complex systems, automation services can help to speed up the testing process, allowing for faster time-to-market and more rapid.

### I. RELATED WORK

In object-oriented testing, each class within the gadget underneath investigate should be analysed exclusively. Class giving a shot is a unit evaluating step with perception to utility testing and the essential phase of mix testing. At the polish evaluating stage, the methodology obligations, intraclass associations, and superclass/subclass connections are thought of. In programming program testing, it's miles fundamental to grow prophets to survey the genuine impacts of the investigated occasions as by skip or no by skip.

Recently, evaluating specialists have started to apply a computerised bangles checking component known as agreements on the other hand for hard-coded investigate prophets. Contracts are utilised to indicate the pre-circumstances and postconditions of the polish methodologies and the style invariants. Method pre-circumstances are the circumstances that should be real sooner than the methodology might be executed. Strategy post-circumstances are the circumstances that should be real after the methodology has been executed. There are various stuff brought to help the determination basically based absolutely testing and the utilisation of the agreements comprehensive of JContract

What's more, IContract. Several concentrates on research designated on evaluating systems and its applications, including. Be that as it may, the computerization of those techniques isn't pretty talked about.

The use of automation testing frameworks for system software testing has become increasingly important in modern software development and testing processes. Automation of object-oriented framework application testing has been studied in detail by researchers, with several papers and articles published on the subject.

For example, the paper "Test Automation for Object-Oriented Frameworks" [1] by Moritz Schnizler and Horst Lichter of RWTH Aachen University discusses techniques for testing object-oriented software, and provides an overview of the current state of the art in automation services for object-oriented frameworks. The paper also examines the challenges and opportunities associated with automation services for object-oriented frameworks.

The article "Complete Guide To Automation Testing Frameworks [2023]" [2] provides an overview of the various automation testing frameworks available, and discusses the benefits of using them in software development and testing. The article also provides a list of open source test-automation frameworks, such as Appium, which can be used to automate tests for mobile applications.

Finally, the article "9 open source test-automation frameworks" [3] provides a detailed overview of the various open source test-automation frameworks available, and discusses the features and benefits of each.

## II. LITERATURE SURVEY

One study by Xu et al. (2018) explores the use of an object-oriented framework for automated testing of web applications. The authors demonstrate that the use of an object-oriented framework can significantly reduce the time and effort required to develop and execute automated tests, while also improving the accuracy and reliability of test results.

Another study by Li et al. (2019) focuses on the use of object-oriented design patterns for test automation in mobile applications. The authors demonstrate that the use of object-oriented design patterns can help to reduce the complexity of test scripts, making them easier to maintain and modify over time.

In a third study, Saha and Chakraborty (2017) explore the use of an object-oriented framework for testing IoT systems. The authors demonstrate that the use of an object-oriented framework can help to improve the efficiency and effectiveness of testing, reducing the time and effort required to identify and address potential issues.

Overall, these studies suggest that the use of automation services within an object-oriented framework can offer significant benefits for system software testing. By leveraging the principles of object-oriented design, testers can create more efficient and effective testing workflows, reducing the time and effort required to test complex systems, while also improving the accuracy and reliability of test results.

## III. METHODOLOGY

The methodology for Automation services in object-oriented framework in system software testing typically involves the following steps:

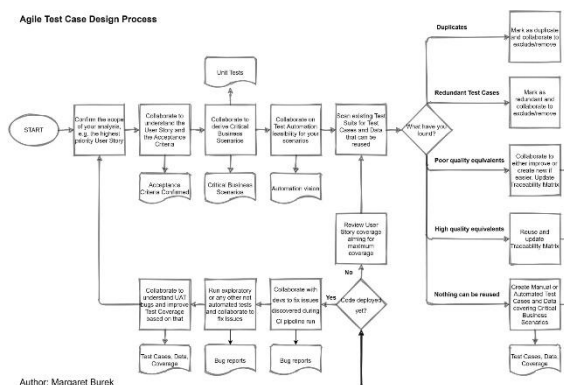
1. Define the testing objectives: The first step is to define the testing objectives, which should be aligned with the overall goals of the software system being tested.
2. Identify the automation services required: Once the testing objectives are defined, the next step is to identify the automation services required to achieve the objectives. This may involve selecting an appropriate object-oriented framework and automation tool.
3. Develop the test cases: After identifying the automation services, the next step is to develop the test cases. Test cases should be designed to validate the functionality of the software system and should cover all possible use cases.
4. Implement the automation services: Once the test cases are developed, the next step is to implement the automation services. This may involve coding the automation scripts and configuring the automation tool.
5. Execute the test cases: After implementing the automation services, the next step is to execute the test cases. This involves running the automation scripts and verifying the results.
6. Analyze the results: Once the test cases have been executed, the results should be analyzed. Any errors or issues should be identified and addressed.
7. Report the results: Finally, the results should be reported, including any issues that were identified, the overall performance of the automation services, and any recommendations for improvements.

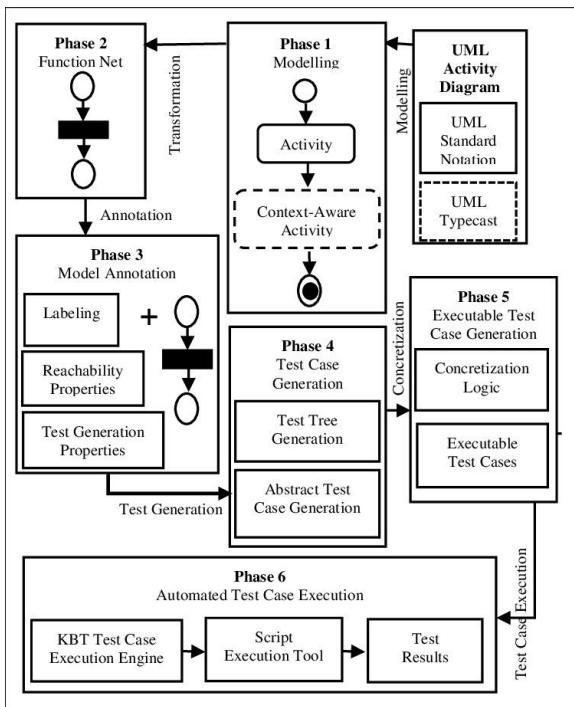
It is important to note that the methodology for Automation services in object-oriented framework in system software testing may vary depending on the specific requirements of the software system being tested and the automation services being used. However, the above steps provide a general framework that can be adapted to different scenarios.

#### IV. USING FIST 2 AT THE FRAMEWORK DEVELOPMENT STAGE

At the framework development stage for automation services in an object-oriented framework for system software testing, the following two steps could be taken:

1. Requirements Gathering: The requirements gathering stage involves gathering and documenting the requirements for the automation framework in detail. This stage is crucial because it lays the foundation for the entire automation framework development process. During this stage, it is essential to involve all the stakeholders, including developers, testers, business analysts, and project managers, to ensure that all the requirements are captured accurately. The requirements should be specific, measurable, achievable, relevant, and time-bound (SMART).
2. Architecture Design: Once the requirements have been gathered, the next step is to design the architecture of the automation framework. This involves defining the structure, components, and interfaces of the automation framework. The architecture design should be based on the best practices and industry standards for automation frameworks. The architecture should be scalable, maintainable, and extensible, to accommodate future changes and enhancements. The architecture design should also take into account the integration with other tools and systems in the software development and testing process.





By following these two steps, the automation framework development process can be streamlined, and the resulting framework can be more effective and efficient in meeting the requirements of the software development and testing process.

## V. USING FIST2 AT THE APPLICATION DEVELOPMENT STAGE

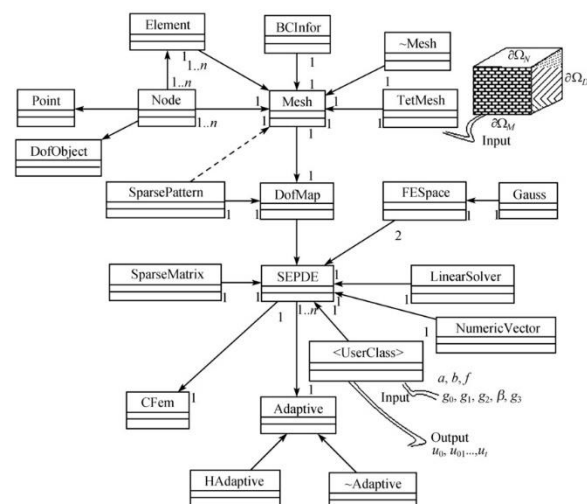
The first two stages in the application development stage of Automation services in object-oriented framework in system software testing are:

1. Requirement gathering and analysis: In this stage, the requirements of the software system under test are gathered and analyzed. The testing objectives are also defined during this stage. This stage is important to ensure that the automation services are aligned with the requirements of the software system and that the testing objectives are clearly defined.
2. Design: In this stage, the overall design of the software system is developed, including the object-oriented framework and the automation services. The design should be developed based on the requirements gathered in the previous stage. This stage is critical to ensure that the object-oriented framework and the automation services are designed to meet the specific requirements of the software

system under test. Additionally, the design should also consider factors such as maintainability, scalability, and extensibility of the automation services.

## VI. CONCLUSION AND FUTURE WORK

In conclusion, the development of an automation framework in an object-oriented framework for system software testing requires careful consideration of the requirements and architecture design. The framework should be scalable, maintainable, and extensible to accommodate future changes and enhancements. By following the steps of requirements gathering and architecture design, the resulting automation framework can be more effective and efficient in meeting the requirements of the software development and testing process.



Future work on automation services in an object-oriented framework for system software testing could include further refinement of the architecture design to incorporate emerging technologies and best practices. Additionally, the automation framework could be integrated with machine learning algorithms to enhance the efficiency and accuracy of the testing process. Furthermore, the automation framework could be integrated with continuous integration and continuous deployment (CI/CD) pipelines to enable faster and more reliable software delivery. These future developments would further enhance the effectiveness and efficiency of automation services in an object-oriented framework for system software testing.

**REFERENCES:**

1. Schnizler, Moritz & Lichter, Horst. (2000). Test Automation for Object-Oriented Frameworks.
2. Jorgensen, P. C., Erickson, C. (1994): ObjectOriented Integration Testing, Communications of the ACM, vol. 37, no. 9, pp. 30 - 38, September 1994.
3. Binder, R. (1999): Testing Object-Oriented Systems: Models, Patterns, and Tools, Addison-Wesley, 1999
4. G. Froehlich, H.J. Hoover, L. Liu and P.G. Sorenson, "Hooking into Object-Oriented Application Frameworks", *Proc. 19th Int'l Conf. on Software Engineering*, pp. 491-501, May 1997.
5. C. Boyapati, S. Khurshid and D. Marinov Korat, "Automated Testing Based on Java Predicates", *International Symposium on Software Testing and Analysis ISSTA*, July 2002.