

# Comparative Study of Software Test Automation Frameworks

<sup>1</sup> Khalid Eldrandaly, <sup>2</sup>Mahmoud Abd ElLatif, <sup>3</sup>Nora Zaki

<sup>1</sup>Professor of Information Systems, Faculty of Computers and Informatics, Zagazig University.

<sup>2</sup> Professor of Information Systems, College of Business, Jaddah University, SA & Faculty of Computers and Informatics, Helwan University.

<sup>3</sup> Assistant Teacher, Faculty of Computers and Informatics, Zagazig University.

**Abstract:** *Software testing is one of the most critical phases of software development life cycle. The time and cost consumed by software testing are one of the most critical limitations of software testing. The testing process can be done manually or automatically. Recently, software automation testing is applied in many software organizations to guarantee the software quality and to reduce the cost and time consumed in manual testing. Software test automation framework is an independent application which maximizes the automation efforts by facilitating the execution of the automated test scripts. There are many software test automation frameworks (STAFs) available in the marketplace. The automation testers face a problem in selecting the best STAF that meet their testing requirements. The main objective of this paper is to provide the automation testers with a good understanding of STAFs. This work aims to evaluate each STAF in terms of their scripting approach, features, advantages and disadvantages. Furthermore, it conducts a comparative analysis among STAFs by using the essential parameters of automation projects such Scripting capabilities, time, application size, scripting approach, modularity, scalability, reusability, maintainability, and complexity. This analysis aims to help the testers to select the excellent fit STAF.*

## KEYWORDS

*Software testing, automated software testing, STAF, agile methodology, BDD framework.*

## I. INTRODUCTION

There are different definitions of software testing [1], [2], [3], [4], [5] but, the most used definition states that software testing is a useful process by which some level of

software quality can be verified. Where software testing analyzes the developed software to determine to what extent it contains errors or bugs and try to fix them as early as possible in order to improve the quality of software. Software testing is one of the most critical and costly phases in software development life cycle SDLC.

The Software testing process is divided into four basic tasks: first, generating a set of test cases which contain a set of conditions that are used by the testers to check the software under test SUT against the specified requirements. Second, the test cases should be documented into scripts. Third, the real execution begins by running the generated test cases on SUT and fourth, the test evaluation is performed by comparing the expected and actual output to determine whether the tests pass or fail then reporting the results. These tasks can be done either manually or automatically [6].

Manual testing [7] is performed by a human tester who follow a written test plan but, due to the increasing complexity and size of software product, the cost and time of manual testing increases and some tasks become difficult to perform manually. Therefore, tests should be processed in automated manner in order to improve the efficiency and effectiveness of testing process by decreasing its time and costs [8], [9].

Automated software testing (AST)[10], [11] is an interested area in recent testing research which relies on using the special software tool to automate test activities and control the execution of tests by running the tests quickly and repeatedly. Automated software testing offers the following benefits for software testing process [12], [13]:

1. AST improves testing process by increasing its effectiveness and efficiency,
2. Test coverage, confidence and fault prediction are increased by implementing AST,

3. AST improves the accuracy and quality of software product,
4. Reusability of tests became possible using automation,
5. The consumed time and costs of testing process are saved, and
6. It reduces the human effort.

The previous studies in the field of software testing automation [14], [15], [16], [17], [18] focus mostly on automation of test execution because the execution of test cases consumed a significant amount of the testing process time. Especially, for testing types with a repetitive nature such as stress, reliability and regression testing. Therefore, there is a great benefit by automatically executing tests. A large number of tests can be executed many times in less time by automation [19], [20].

The success of the automation requires a good designed architecture that consists of a set of guidelines, rules and tools which control the execution of test cases and support the automation efforts. This architecture is called Software Test Automation Framework STAF [21]. The STAF offers several advantages to software tester. The tester builds his framework from scratch or chooses one from commercial components available in the market. Nowadays, there are different types of test automation framework available in marketplace. These frameworks are different from each other based on different key factors such as scalability, modularity, scripting capabilities,...etc

The testers always need to choose the suitable SAFT for their testing requirements Because of the great benefits offered by STAF in supporting the automation process. The selection process is not easy because of the different features presented by each framework. Therefore, in order to address this issue, the main objective of this paper is to address the SATF in more details and to conduct a comparative study of the available STAFs by using a set of key parameters that help testers to select the right framework that is suitable for his application. This paper emphasis the difference between the traditional test automation frameworks and agile frameworks.

The structure of this paper is as following: section 2 briefly outlines the concept of Software Test Automation Framework, section 3 shows the difference between traditional and agile development methodology. Also, a section 4 reviews the related works which are done on different testing automation framework; section 5 discusses the different types of Software Test Automation Framework in detailed and conducts a comparison among them. Section 6 shifts to agile automation

framework. Finally, section 7 contains the conclusion.

## **II. SOFTWARE TEST AUTOMATION FRAMEWORK**

A software test automation framework [22] is an independent application which provides an execution environment for automated test scripts. The framework is a set of different guidelines, tools, concepts, libraries and practices [23]. Thus testers and developers can follow it to design, execute and report the automated test scripts efficiently. A good test automation framework can further support the automation effort and improving the efficiency of testing process by reducing the test effort and cost. [24] Concluded that the overall project testing effort can be reduced by applying automated testing framework with automation tool. The Software Test Automation Framework offers the following benefits [25], [26]:

1. Makes the code and test more reusable,
2. Optimizes the utilization and implementation of the automated tools,
3. Reduces the manual testing effort by applying the automation,
4. Improves the product quality.
5. Reduces the maintenance of test scripts and makes it easier,
6. Minimizes the cost and time of testing process by running many tests quickly and repeatedly,
7. Increases the confidence with the application by executing multiple tests frequently,
8. Increases the return on investment by maximizing the reusability and maintainability of tests,
9. Increases the test coverage, and
10. Provides reporting capability by creating a test report automatically.

## **III. WATERFALL AND AGILE DEVELOPMENT METHODOLOGY**

Recently, there are different STAF frameworks in the market. This paper categorizes STAF frameworks based on the used software development methodology into two categories that will be discussed later in section 5. But, in this section, we emphasis the difference between traditional and agile development methodology.

Software development methodology is known as a software process model which represents a set of process in an abstract manner to develop the software through Software Development Life Cycle (SDLC) [27]. Waterfall methodology is one of the traditional development methodologies (waterfall model, V-shaped model, incremental model, spiral model, and red model) which are widely used in

SDLC. In waterfall model, the software is developed in a linear and sequential manner [28] where each phase can't be started until the previous phase is completed. The testing is performed as a separate phase after completing the analysis and design phase which can't begin early in SDLC [29].

The waterfall model is not suitable for the incremental projects which frequently face changing requirements and also it is impossible to clearly define all the requirements at the beginning of the project. The need for a methodology which can support requirement changing during the software development process is raised. Therefore many software companies have started to shift to agile model [30]. On the other side, agile model is considered as the most popular methodology of SDLC. It uses an incremental and iterative approach for software development [31]. Agile methodology offers the following features [29], [32]:

1. It has an ability to adapt with continuous changes in requirements,
2. Supports the collaboration between the developers and the customers in order to validate their requirements, and
3. Provides high flexibility and management of the process.

Many studies conducted a comparative study of traditional models of software development process and agile methodology. [33] was shown that the agile methodology achieves success in many software projects due to its ability to adopt with frequent changes in requirements to provide customer satisfaction and collaboration, and to deliver early working software. [30] made sure that each methodology has its own advantages and disadvantages. The decision to select the appropriate software development methodology for the project must be addressed and taken into account the characteristics of both the project and organizational environment.

#### **IV. RELATED WORK**

This section states the related works that are done on different testing automation frameworks. [34] Proposed a new testing framework for web application to improve the automation process. This framework uses both record/play back scripting techniques (its development is easy) and programmable scripting techniques (its maintenance is easy). This framework tries to extend the automation to all testing tasks. The results show that about 75% of the total time of the automation process can be saved by the proposed framework and 21% can be saved by using Selenium IDE tool.

The selenium webdriver tool is used [35] to support the keyword driven testing framework for testing web application. It ensures that the testing effort is minimized due to the reusability feature of keyword driven framework. Where [36] used the same tool (the selenium webdriver tool) to create a data driven framework for testing web application instead of a keyword driven framework. It also ensures that the data driven framework is the best solution for a big data sets since it can separate data in excel file from the code for reusability purpose and it is time saving.

A comparison among different testing frameworks is conducted [37] based on selenium that is one of the best testing tools and supports different frameworks. It emphasizes that the hybrid framework of selenium tool is characterized as the best due to its functionality, reusability, reliability, performance, and low cost. [38] Presented an adaptive framework which supports the keyword driven test under different test environments. The proposed framework consists of three layers: automation engine, test driver, and test execution layer. It used XML to write the commands of the keyword for the test cases. But, the proposed framework lacks synchronization and concurrence.

The benefits and limitations of the three generations of testing frameworks: record/playback, modular, data driven, and keyword driven frameworks are stated by [39]. He uses the keyword driven framework for testing web application. Moreover, the available tools for the keyword driven accelerate the maintenance of test cases in object repositories by using Html, Xml, and Spreadsheets. [40] Designed a keyword and behavior driven test automation framework using MBT that can be effectively used in agile environment. The proposed framework improves flexibility, maintenance, and coverage through using MBT. Furthermore, it offers effective continuous integration and cost reduction.

#### **V. TYPES OF SOFTWARE TESTING AUTOMATION FRAMEWORKS**

The different types of STAF can be classified into two categories based on the used software development methodology:

- Traditional STAF
- Agile STAF.

Figure 1 shows the different types of STAF available in the market place.

### A. Traditional Software Testing Automation Frameworks

The traditional Software Testing Automation Frameworks that are used by many testers are:

1. Linear test automation framework.
2. Modular test automation framework.
3. Data driven test automation framework.

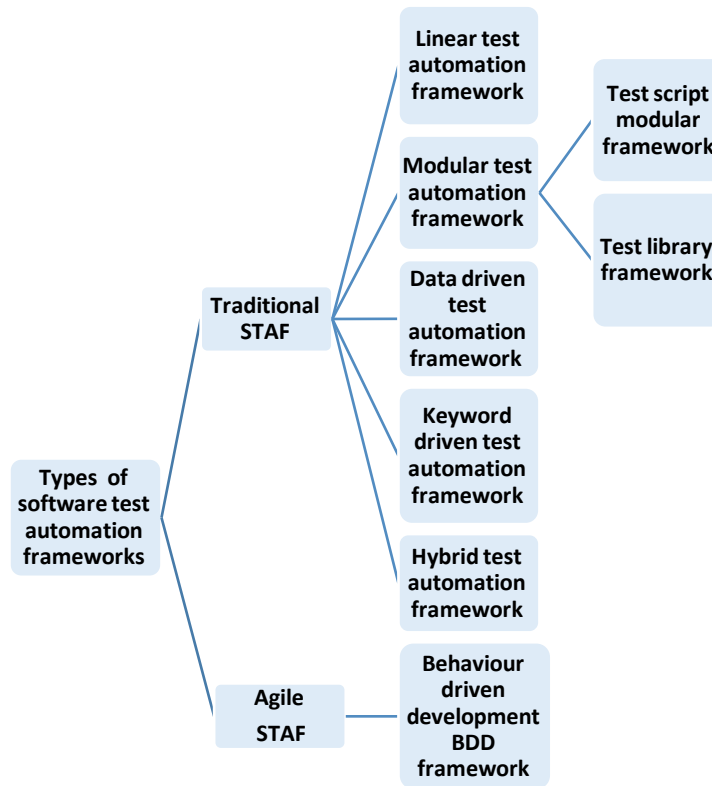


Figure 1. Types of software test automation frameworks.

4. Keyword driven test automation framework.
5. Hybrid test automation framework.

In the following sub sections, we analyze these five traditional testing frameworks in terms of their scripting approach, features, advantages and disadvantages.

#### a) Linear test automation framework (record and playback)

##### • Scripting approach of linear framework:

Linear framework is a first generation of test automation frameworks. It is based on record and playback approach [37] where the tester can record his step by step actions into a test script at the first time and then playback the recorded script without reusing the test script. The test scripts are created in a linear or sequential order with no modulation.

Selenium IDE [41] is a good example of this linear framework because it has a recording option where the testers can record their performed actions in the first round and

just playback the recorded script in the other round. Figure 2 shows the components of linear framework. The basic components are number of test scripts and its local object repository which interact directly with the software under test without any external functions.

##### • Features of linear framework: The linear test automation framework offers the following feature [25]:

1. It is a simple test automation framework,
2. is suitable for testing a web application user interface (UI),
3. Is suitable for smoke test suites where just fundamental tests are needed to perform, and.
4. Doesn't require high level of automation expertise.

##### • Advantages of linear framework:

1. Test scripts can be written by easy and fast way with no or minimal involvement of planning,
2. Test scripts turn out to be understandable since it is written in a linear fashion, and

- The record and playback feature can be learned without programming or design expertise.

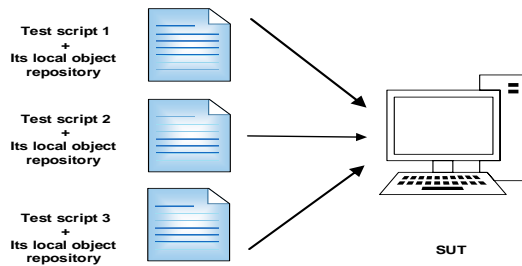


Figure 2. The linear test automation framework (record and playback).

- Disadvantages of linear framework:**

- This framework is inefficient since the tests lack reusability,
- Is difficult to be maintained because any modification made in the application require other scripts to be updated, and
- Test script can't be executed with multiple data sets because input data is embedded into the script.

*b) Modular test automation framework*

- Scripting approach of the modular test automation framework:**

Modular testing framework is the beginning of the second generation of test automation framework. This framework is based on the abstraction concept [43]. There are two versions of this framework. The first one is a test script modular framework [25] where the software under test is divided into a number of independent and logical modules, components or functions and then builds separate test scripts that represent these modules or functions. These modules are hierarchy introduced and then can be combined to create large test cases.

The second extension of the modular framework is a test library modular framework [44]. This framework tries to improve the maintainability through building external and shared test library that combines a number of test functions where they can be called by multiple test scripts as required thus when any changes are made in the system under test, only specific functions can be updated in the test library without impacting the other parts of the application. Figure 3 shows the components of modular framework. The basic components are a number of separate test scripts that contains its test data, and a shared

function library by which test script interact with software under test.

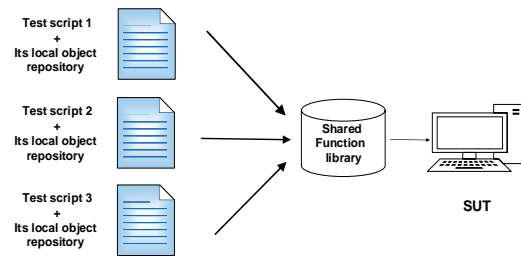


Figure 3. Test library modular framework [42].

- Features of the modular test automation framework:** this framework offers the following features[2], [44]:

- It introduces high level of modularization,
- Offers more scalability since test scripts are written independently,
- Builds test library as an abstraction layer to facilitate the interaction between test scripts and the system under test, and
- Is suitable for large and stable operation.

- Advantages of the modular test automation framework:**

- Test scripts can be written easily and faster for different tests, and
- It is easy to be maintained since an individual change in one part of the application doesn't impact on the other parts of application.

- Disadvantages of the modular test automation framework:**

- The reusability of test scripts is little,
- Test script can't be executed with multiple data sets because input data is still embedded into the script in this framework, and
- The creation of the test library is not easy and requires technical expertise.

*c) Data driven test automation framework*

- Scripting approach of data driven test automation framework:**

This Framework is the beginning of third generation test automation framework. It is based on the data driven approach where test data no longer stuffed within the test script itself but it can be stored in an external files such as XML files, Excel files, Text files, CSV files, ODBC sources,... etc. [43]. This framework overcomes the limitations of previous frameworks by improving the reusability of test script where the same test

script can be executed multiple times with different input data since the test data is stored in external files not in the script [44]. The script became just a mechanism for driving the test data therefore it is called a driver script.

The test data is represented in key-value or tabular format. It includes both input test data and expected output data [42]. First, the driver script uses a reading mechanism to read input data from external file then executes the script code to compute the actual output. Finally, it compares the actual output with expected outputs from external data files.

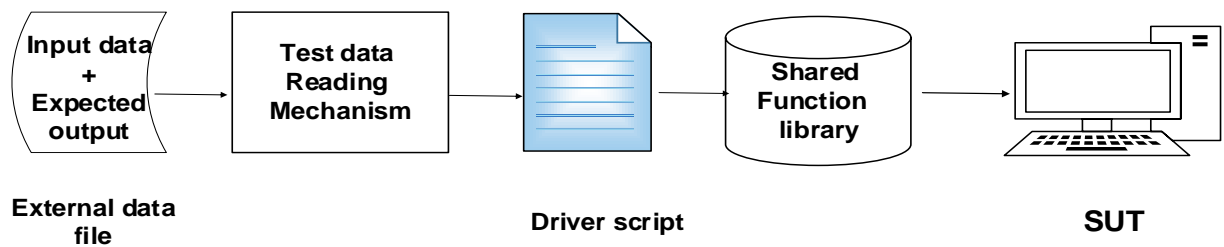


Figure 4. The Data driven test automation framework [25], [42].

4. requires less coding to generate all test sceneries,
5. offers high reusability of test scripts since the same test script can be executed multiple times with different input data,
6. Provides a well-defined architectural design,
7. offers more flexibility and maintainability, and
8. The development of test data is independent of test implementation and can be prepared even before the tested system is ready.

● **Advantages of data driven test automation framework:**

1. It overcomes the problem of hard coded test data into the scripts,
2. doesn't require programming skills to edit test scripts,
3. Is stable and power full in handling errors,
4. Test data and test scripts are separate so any change made in one doesn't affect the other, and
5. The test scripts can be executed in multiple environments, and

● **Disadvantages of data driven test automation framework:**

1. The big limitation of the data driven framework is the need for creation different test script that can understand different sets of data,
2. Requires an initial set-up effort that results in additional time to develop both test scripts and test data, and

Figure 4 shows the components of data driven framework.

● **Features of data driven test automation framework [37], [45]:**

1. It is suitable for large scale test automation,
2. Is suitable for application under development,
3. Creates a little number of test scripts to execute all test sceneries by just changing the test data in the external file.

3. Requires a good programming skill to identify approaches for editing and storing test data in the external data file.

d) **Keyword driven test automation framework**

● **Scripting approach of keyword driven test automation framework:**

This framework is based on keyword driven approach. The keyword driven approach is offered by [46] and [47] as a solution for the biggest limitation of data driven approach which mentioned previously. Keyword driven framework is an extension of data driven framework where not only test data is segregated from test scripts but also directives which telling what actions and steps must be performed on the data are segregated from test scripts and stored with test data into an external files. These directives known as keywords [42]. In order to keep the keyword driven framework modular, the functionality of system under test is determined by creating the basic functions and stores them into an external function library (test library) and then calling these functions as keywords when it is required [48].

This Framework is application independent framework which is based on data tables and keywords. Data tables and keywords should be created independently of test automation tool which used to execute them and the test script which drives them [25]. Since keywords and test data are represented in tabular format, table-driven framework is an alternative name for keyword

driven framework [44]. Figure 5 shows the components of keyword driven framework.

- **Feature of Keyword driven test automation framework:** Keyword driven framework offers the same features of data driven framework in addition to the following features [2], [48]:

1. The keyword and function libraries are independent and can be generally reused with different application,
2. The tester doesn't require high scripting knowledge unlike data driven approach, and

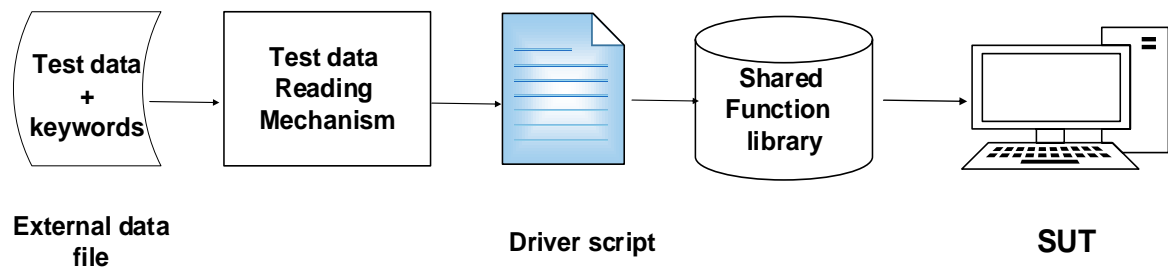


Figure 5. Keyword driven test automation framework [25], [42].

- **Disadvantages of keyword driven test automation framework:**

1. Keyword driven framework is more complex than data-driven framework , and
2. Test cases become more complex and longer because of the greater flexibility.

e) *The hybrid test automation framework*

- **Scripting approach of the hybrid test automation framework:**

The hybrid framework [44], [49] is developed to include more than one approach to integrate their benefits and reduce their weakness. It may be a combination of modular, data driven and keyword driven framework or a combination of at least two of them based on the requirement of the tester. This combination enables the data driven scripts to exploit the libraries that typically go with the keyword driven testing.

- **Features of the hybrid test automation framework:** The hybrid framework integrates the features of all other frameworks while the most important feature of the hybrid approach is its flexibility especially if it is carefully designed.
- **Advantages of the hybrid test automation framework:** The hybrid framework merges the advantage of the included frameworks.

3. Keywords can be reused with different scripts.

- **Advantages of keyword driven test automation framework:** Keyword driven framework has the same advantages of data driven in addition to the following advantages:

1. It is not dependent on a specific tool or programming language, and
2. Can be used by non-programmers who can easily create new kinds of scripts.

- **Disadvantages of the hybrid test automation framework:** Logically the hybrid framework has high complexity and difficulty to setup where high level of programming expertise is needed to build it.

The previous mentioned STAFs use a traditional scripting approach (linear, data driven, and keyword driven). After analyzing these various test automation frameworks, a comparison of key parameters is conducted to help in selecting the right framework for the specific requirements of the project. The comparison is based on the most important parameters for automation projects such as scripting capabilities, time, application size, scripting approach, modularity, scalability, reusability, maintainability, and complexity. Table 1 summarizes the comparison among the five traditional software testing automation frameworks by using key parameters. Testers can assign values to these parameters based on their project requirements to create a scorecard which can be used to measure these parameters. The STAF with the highest score can be selected for further investigation in automation efforts.

Data driven and keyword driven framework can be considered the best frameworks since the data driven overcomes the problem of hard coded test data into the scripts by separating test data from test scripts so any change made in one doesn't affect the other. It

provides high reusability of test scripts since the same test script can be executed multiple times with different input data. Although, it needs to create different test script that can understand different sets of data. While the keyword driven framework increases the reusability of test scripts by separating the keyword from function libraries and can be generally reused with different application. Although, it is more complex than data-driven framework. For best implementation, the testers can benefit from the advantages of both data driven and keyword driven framework by combining them in hybrid framework. Despite of the high complexity and difficulty in designing the hybrid framework but, it provides more flexibility.

## VI. AGILE SOFTWARE TESTING AUTOMATION FRAMEWORKS:

Any traditional testing frameworks can be used in agile environments but they are not a good solution for the agile environment. [2], and [50] ensure that traditional testing frameworks are not fit for the agile environment because agile testing is characterized by short iterations which face rapid changing in requirement and this makes the maintenance of test automation very difficult by the traditional testing framework.

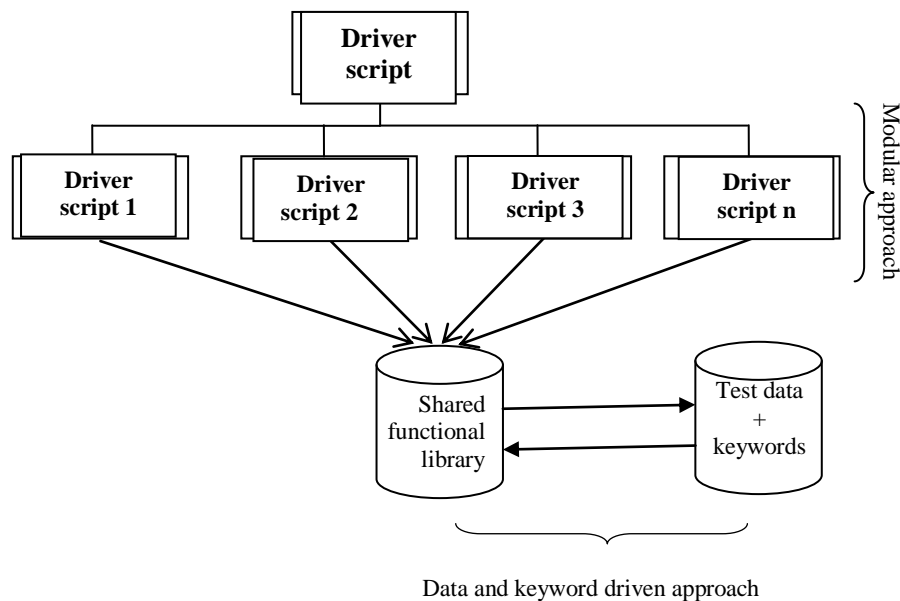


Figure 6. The hybrid test automation framework [49].

Agile teams face different challenge related to continuous integration, unit testing, Collaboration and fast feedback which are difficult to solve by using traditional testing framework. So, Agile teams need testing frameworks that can support early test automation and achieving maximum code and functionality coverage. Agile testing automation frameworks are based on agile software development methodologies which are popular in the software industry. The widely used agile development methodologies are TDD, BDD and ATDD where acceptance testing has a massive affect in their success [51].

**1- Test driven development TDD:** relies on unit tests which guide the design of software. The most featured aspects of TDD are that the code is written after the creation of tests [52].

**2- Acceptance test driven development ATDD:** ATDD is an improvement of TDD where it relies on the acceptance tests. In ATDD, the developers transform the

stakeholders' requirements into acceptance tests which responsible for verifying system's functionality [53]

**3- Behavior driven development BDD:** BDD combines the concepts of TDD and ATDD. The main feature of BDD is it writes the test cases by ubiquitous language that can be understood by everyone. It encourages collaboration between the designers and the stakeholders [54].

### A. Behavior Driven Development (BDD) Framework

This an agile testing framework which is based on behavior driven development approach which is a generally modern agile software development approach that proposed by [55] and focuses on the behavior aspects rather than implementation aspects of the software.

There is little number of the published papers that study BDD; most of them expose it



as specific techniques of the software development. One paper which was introduced by [56] who directly address BDD and its characteristics. They introduced six BDD characteristics through a literature review and seven toolkits (JBehave, NBehave, RSpec, MSpec, StoryQ, Cucumber, SpecFlow) which support BDD. The two basic BDD characteristics are: the first one is that BDD writes test cases in ubiquitous language which would understand easily by everyone. Therefore, the BDD approach could fill the communication gap between developers and testers result in a good collaboration between them. The second characteristic is that BDD is a decomposition and iterative process which starts with identification of user stories which describe the feature delivered by the system, and for each user stories a scenario is identified which describes the context of this user story and show how the system behave according to a specific event.

- **Features of BDD agile framework [40,**

**51].** It offers the following features:

1. It supports the TDD and ATDD development methodology,
2. The early testing through software development life cycle since testing is not a separate phase,
3. Easy and understandable natural language,
4. The communication between developers and testers,
5. A rapid feedback to accommodate with various changes in requirements, and
6. Continues integration, unit testing and acceptance testing.

**Table 1 comparison of the five traditional software testing automation frameworks using different key parameters.**

<b>Parameter</b>	<b>Linear framework</b>	<b>Modular framework</b>	<b>Data driven framework</b>	<b>Keyword driven framework</b>	<b>Hybrid framework</b>
<b>Scripting capabilities</b>	Fast in writing script	Easy and fast in writing script	Easy and fast in writing script	Easy and fast in writing script	Easy and fast in writing script
<b>Time</b>	Minimal time is required for scripting	Less time is required for scripting	Additional time is required for scripting	Time consuming task	Time saving
<b>Application size</b>	suitable for very small application	Suitable for large stable application	Suitable for application under development	Suitable for small application	Suitable for medium to large application
<b>Scripting approach</b>	Record and playback	Abstraction concept Or Modularization	Data driven Approach	Keyword driven Approach	Hybrid approach
<b>Modularity</b>	No modularity	High level of modularity	Modular	Modular	Modular
<b>Scalability</b>	Not scalable	Scalable	Scalable	Scalable	Scalable
<b>Reusability</b>	Lacks of reusability	Little reusability	High reusability	High reusability	High reusability
<b>Maintainability</b>	Difficult to maintain	Easy to maintain	Easy and fast to maintain	Easy and fast to maintain	Additional effort is needed for maintainability
<b>Complexity</b>	Simple	Less complexity	Complex	More complexity	More and more complexity

• **Advantages of BDD agile framework [50]:**

1. It improves the quality and efficiency of software product,
2. Speeds the testing process by reducing the time,
3. Encourages the collaboration between the developers and testers, and
4. The user doesn't necessary to be familiar with the programming language.

It is obvious that the implementation of BDD framework is beneficial when it is used for communication purposes. But, if it is only used for automation support, the using one of the traditional testing frameworks is more suitable than using BDD framework.

**VII. CONCLUSIONS AND FUTURE WORK**

The studied software test automation frameworks STAFs are some of the most popular test automation frameworks that are recently used by the automation testers. The selection of the right automation framework is a real key to success in any test automation projects. A detailed analysis for these frameworks must be conducted before selecting one of them. Therefore, this research does more effort to evaluate each framework in terms of their scripting approach, features, advantages and disadvantages. This analysis gives the testers a good understanding of the concept of each framework so that they can make a right decision about whether a given framework will be a good fit.

Furthermore, after the detailed analysis of these six STAFs, it is concluded that the best framework is a one which provides the testers with easy and maintainable test scripts, high quality test scripts, high reusability of test components, saving on effort and better return on investment ROI.

Last but not least, the selection of the right test automation framework is affected by some key factors that are related to the automation project such as: nature, size, complexity and duration of the project. Other factors such as the requirements of the user, the experience level of both the user and developer, involvement level of customer, team size and skills, the available tools and the application or product type.

For future work, the studied frameworks could be evaluated by additional parameter such as testers' view by a questionnaire for determining which STAF is preferred by the testers.

**REFERENCES**

- [1] Sharma C, Sabharwal S, Sibal R. A survey on software testing techniques using genetic algorithm. *IJCSI International Journal of Computer Science*. Vol. 10, Issue 1, No 1, January 2013.
- [2] Jamil MA, Arif M, Abubakar NS, Ahmad A. Software Testing Techniques: A Literature Review. In 2016 6th International Conference on Information and Communication Technology for the Muslim World (ICT4M). 2016 Nov 22. (pp. 177-182). IEEE.
- [3] Hooda I, Chhillar RS. Software test process, testing types and techniques. *International Journal of Computer Applications*. 2015 Jan 1;111(13).
- [4] Chauhan RK, Singh I. Latest research and development on software testing techniques and tools. *International Journal of Current Engineering and Technology*. 2014 Aug;4(4):2347-5161.
- [5] Nidhra S, Dondeti J. Black box and white box testing techniques-a literature review. *International Journal of Embedded Systems and Applications (IJESA)*. 2(2),2012 Jun; 29-50.
- [6] Amannejad Y, Garousi V, Irving R, Sahaf Z. A search-based approach for cost-effective software test automation decision support and an industrial case study. In2014 IEEE Seventh International Conference on Software Testing, Verification and Validation Workshops 2014 Mar 31 (pp. 302-311). IEEE.
- [7] Thummalapenta S, Devaki P, Sinha S, Chandra S, Gnanasundaram S, Nagaraj DD, Kumar S, Kumar S. Efficient and change-resilient test automation: An industrial case study. In2013 35th International Conference on Software Engineering (ICSE) 2013 May 18 (pp. 1002-1011). IEEE.
- [8] Kaur T, Goel SG. A Framework for Enhancing Performance and Minimizing Automation Testing Cost and Time Using Efficient Methods (Doctoral dissertation). 2015.
- [9] Kumar D, Mishra KK. The Impacts of Test Automation on Software's Cost, Quality and Time to Market. 7th International Conference on Communication, Computing and Virtualization. *Procedia Computer Science*. 2016 Jan 1;79:8-15.
- [10] Gandhi GM, Pillai AS. Challenges in gui test automation. *International Journal of Computer Theory and Engineering*. 2014 Apr 1;6(2):192.
- [11] Rafi M. Automated Software Testing. A Study of State of Practice/Dudekula Mohammad Rafi & Kiran Moses. School of Computing Blekinge Institute of Technology. Sweden. 2010:5.
- [12] Taipale O, Kasurinen J, Karhu K, Smolander K. Trade-off between automated and manual software testing. *International Journal of System Assurance Engineering and Management*. 2011 Jun 1;2(2):114-25.
- [13] Sharma S, Vishawjyoti M. Study and analysis of automation testing techniques. *Journal of global research in computer science*. 2013 Jan 10;3(12):36-43.
- [14] Thummalapenta S, Sinha S, Singhania N, Chandra S. Automating test automation. InProceedings of the 34th International Conference on Software Engineering 2012 Jun 2 (pp. 881-891). IEEE Press.
- [15] Böhme M, Paul S. A probabilistic analysis of the efficiency of automated software testing. *IEEE Transactions on Software Engineering*. 2016 Apr 1;42(4):345-60.
- [16] Vishal S, Vaishali N. A review on Automating Test Automation. *International Journal of Advance Research in Computer Science and Management Studies* Volume 2, Issue 12, December 2014 pg.79-86.
- [17] Clark JA, Dan H, Hierons RM. Semantic mutation testing. *Science of Computer Programming*. 2013 Apr 1;78(4):345-63.
- [18] Puolitaival OP, Kanstrén T, Rytty VM, Saarela A. Utilizing domain-specific modelling for software testing. In3rd Int'l. Conf. on Advances in System Testing and Validation Lifecycle (VALID2011) 2011 Oct 23.
- [19] Rodrigues A, Dias-Neto A. Relevance and impact of critical factors of success in software test automation lifecycle: A

- survey. In Proceedings of the 1st Brazilian Symposium on Systematic and Automated Software Testing 2016 Sep 19 (p. 6). ACM.
- [20] Vishal S, Vaishali N. Generic Test Automation. *International Journal of Science and Research (IJSR)* (2013). 2319-7064
- [21] Cervantes A. Exploring the use of a test automation framework. In 2009 IEEE Aerospace conference 2009 Mar 7 (pp. 1-9). IEEE.
- [22] Artzi S, Dolby J, Jensen SH, Møller A, Tip F. A framework for automated testing of javascript web applications. In Proceedings of the 33rd International Conference on Software Engineering 2011 May 21 (pp. 571-580). ACM.
- [23] Lamba S, Rishiwal V, Rana A. An automated data driven continuous testing framework. *International Journal Of Advanced Technology In Engineering And Science*. 2015 Feb;3(1).
- [24] Yalla M, Shanbhag M. Building automation framework around open source technologies. In Proc. of Software Testing Conference 2009 (pp. 6-9).
- [25] Rajesh P. Test Automation Framework. *IJITKM Volume 7 • Number 2 Jan– June 2014* pp. 214-219
- [26] Bajaj KS. Hybrid Test Automation Framework for managing Test Data. *International Journal of Pure and Applied Mathematics*. 2018;118(9):265-76.
- [27] Raval RR, Rathod HM. Comparative study of various process model in software development. *International Journal of Computer Applications*. 2013 Jan 1;82(18).
- [28] Balaji S, Murugaiyan MS. Waterfall vs. V-Model vs. Agile: A comparative study on SDLC. *International Journal of Information Technology and Business Management*. 2012 Jun 29;2(1):26-30.
- [29] Aitken A, Ilango V. A comparative analysis of traditional software engineering and agile software development. In 2013 46th Hawaii International Conference on System Sciences 2013 Jan 7 (pp. 4751-4760). IEEE.
- [30] Singht W, Denwattana N. An experience in blending the traditional and Agile methodologies to assist in a small software development project. In 2016 13th International Joint Conference on Computer Science and Software Engineering (JCSSE) 2016 Jul 13 (pp. 1-5). IEEE.
- [31] Umbreen M, Abbas J, Shaheed SM. A Comparative Approach for SCRUM and FDD in Agile. *International Journal of Computer Science and Innovation*. 2015;2:79-87.
- [32] Kumar G, Bhatia PK. Comparative analysis of software engineering models from traditional to modern methodologies. In 2014 Fourth International Conference on Advanced Computing & Communication Technologies 2014 Feb 8 (pp. 189-196). IEEE.
- [33] Heba E, Mazen S. Agile and traditional requirements engineering: A survey. *International Journal of Scientific & Engineering Research* 4.9 2013. 473-482.
- [34] Milad H, Amal EA, Mostafa-Sami M. Automated Software Testing Framework for Web Applications. *International Journal of Applied Engineering Research*. Volume 13, Number 11 (2018) pp. 9758-9767.
- [35] Singla S, Kaur H. Selenium Keyword Driven Automation Testing Framework. *IJARCSSE* 2014 Jun;2277.
- [36] Kumar A, Saxena S. Data driven testing framework using selenium WebDriver. *International Journal of Computer Applications*. 2015 Jan 1;118(18).
- [37] Koneru S, Chandra P. A Comparative Study of Testing Framework with Special Emphasis on Selenium for Financial Applications. *International Journal of Soft Computing* 12.3 (2017): 148-155.
- [38] Tang J, Cao X, Ma A. Towards adaptive framework of keyword driven automation testing. In 2008 IEEE International Conference on Automation and Logistics 2008 Sep 1 (pp. 1631-1636). IEEE.
- [39] Bajpai N. A keyword driven framework for testing web applications. *International Journal of Advanced Computer Science and Applications*. 2012;3(3):8-14.
- [40] Sivanandan S. Agile development cycle: Approach to design an effective Model Based Testing with Behaviour driven automation framework. In 20th Annual International Conference on Advanced Computing and Communications (ADCOM) 2014 Sep 19 (pp. 22-25). IEEE.
- [41] Kaur H, Gupta G. Comparative study of automated testing tools: selenium, quick test professional and testcomplete. *Int. Journal of Engineering Research and Applications*. 2013 Sep;3(5):1739-43.
- [42] Laukkanen P. Data-driven and keyword-driven test automation frameworks. Master's thesis. Helsinki University of Technology. 2006 Feb 24.
- [43] Divya A, Mahalakshmi SD. An efficient framework for unified automation testing: a case study on software industry. *International Journal of Advanced Research in Computer Science & Technology*. 2014;2.
- [44] Sadiq M, Firoze F. A method for the selection of software testing automation framework using analytic hierarchy process. *International Journal of Computer Applications*. 2014 Jan 1;102(1).
- [45] Kumar P, Dr. Kavita. Automation Framework for Database Testing. *IJARSE*, Vol.No.5, issue No.6, June 2016.
- [46] Fewster M, Graham D. Software test automation: effective use of test execution tools. ACM Press/Addison-Wesley Publishing Co.; 1999 Jul 1.
- [47] Kaner C. Pitfalls and strategies in automated testing. *Computer*. 1997 Apr;30(4):114-6.
- [48] Gustavsson R, Kostopoulos D. From Requirements Specification to Test Scripting: Towards Automated Support . Master Thesis in Computer Science with Specialization in Software Engineering. 2015.
- [49] Samik R, Uday B. A Survey on Test Automation. *International Journal of Innovative Research in Computer and Communication Engineering*. Vol. 4, Issue 6, June 2016.
- [50] Heet P, Danish C. Agile Testing Automation Framework. *International Journal Of Engineering And Computer Science*. Volume 5. Issue 09. September 2016. Page No.17996-18000.
- [51] Rahman M, Gao J. A reusable automated acceptance testing architecture for microservices in behavior-driven development. In 2015 IEEE Symposium on Service-Oriented System Engineering 2015 Mar 30 (pp. 321-325). IEEE.
- [52] Hammond S, Umphress D. Test driven development: the state of the practice. In Proceedings of the 50th Annual Southeast Regional Conference 2012 Mar 29 (pp. 158-163). ACM.
- [53] Gómez LA. Analysis of the impact of test based development techniques (TDD, BDD, AND ATDD) to the software life cycle (Doctoral dissertation).
- [54] Wanderley F, da Silveria DS. A framework to diminish the gap between the business specialist and the software designer. In 2012 Eighth International Conference on the Quality of Information and Communications Technology 2012 Sep 3 (pp. 199-204). IEEE.
- [55] North D. Introducing behaviour driven development. *Better Software Magazine*. 2006.
- [56] Solis C, Wang X. A study of the characteristics of behaviour driven development. In 2011 37th EUROMICRO Conference on Software Engineering and Advanced Applications 2011 Aug 30 (pp. 383-387). IEEE.