International Journal of Core Engineering & Management
Volume-4, Issue-11, February-2018, ISSN No: 2348-9510

QA SUCCESS WITH UFT: UNLOCKING THE FULL POTENTIAL OF TEST AUTOMATION

Asha Rani Rajendran Nair Chandrika

Abstract

In the dynamic landscape of software development, the demand for effective testing methodologies is paramount. Unified Functional Testing (UFT) stands out as a comprehensive solution that empowers QA testers to enhance software quality while optimizing resources. This article examines the evolution and capabilities of UFT, highlighting its integration of both GUI and API testing functionalities. UFT facilitates keyword-driven testing, business process testing, and seamless management of test processes through Application Lifecycle Management (ALM) integration. With its intuitive interface and robust features, UFT maximizes testing efficiency and supports both technical and non-technical users, enabling streamlined workflows and the delivery of high-quality applications. Furthermore, the article delves into practical automation techniques and best practices, showcasing how UFT can transform testing processes and contribute to successful project outcomes.

Keywords: Unified Functional Testing, UFT, software quality, test automation, GUI testing, API testing, keyword-driven testing, Application Lifecycle Management, business process testing.

I. INTRODUCTION

Unified Functional Testing (UFT), previously known as Quick Test Professional (QTP), is a pivotal tool for quality assurance professionals seeking to enhance software quality and efficiency in testing processes. Originally developed by Mercury Interactive Corporation, the first iteration of this tool, Quick Test Professional, paved the way for a comprehensive testing solution. Following HP's acquisition of Mercury Interactive in 2006, the tool was rebranded as HP Quick Test Professional, which later evolved into HP UFT in December 2012 after merging with HP Service Test, a product designed for testing web services. [1]

UFT stands out as an advanced solution for automating both functional and regression testing across various platforms, enabling QA teams to uphold high standards of software quality. This powerful tool integrates GUI testing with API (service) testing, allowing users to evaluate functionality at multiple application layers, from front-end GUI interfaces to back-end service layers. Furthermore, UFT's Business Process Testing (BPT) features cater to a diverse user base, encompassing both technical and non-technical testers, thereby maximizing opportunities to create comprehensive automated tests while significantly reducing manual efforts.

International Journal of Core Engineering & Management
Volume-4, Issue-11, February-2018, ISSN No: 2348-9510

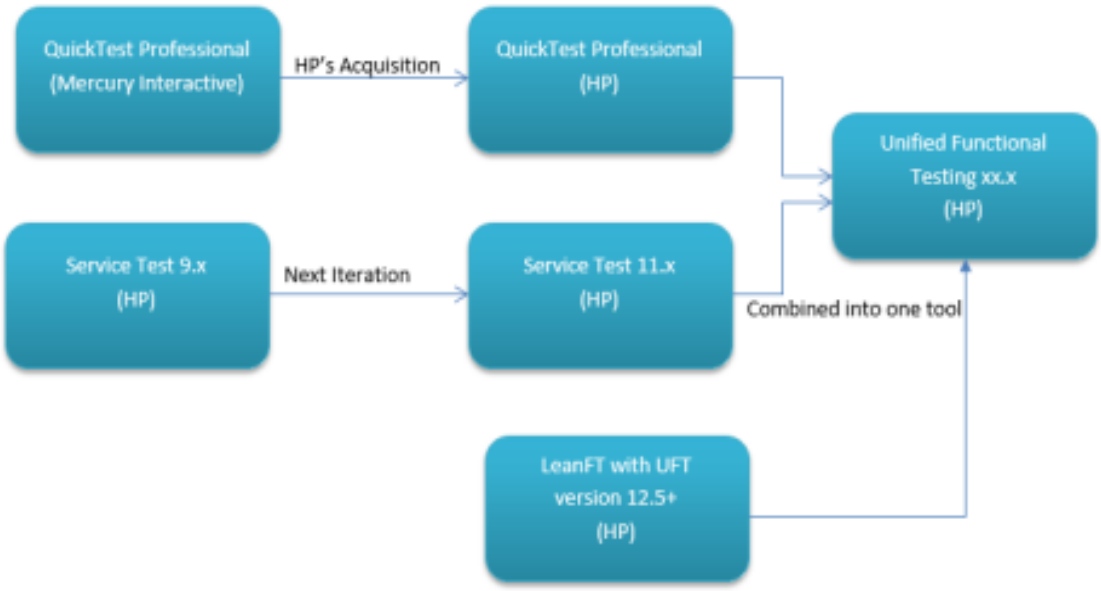

Figure 1: Evolution of HP UFT from Quick Test Professional [1]

Mastering VBScript, the foundational scripting language behind UFT automation, is essential for crafting robust and efficient test scripts. This article aims to provide a thorough overview of UFT's capabilities, explore effective automation techniques, and share best practices tailored for QA testers.

II. INTEGRATED SERVICES OF UFT

UFT (Unified Functional Testing) offers a comprehensive testing environment with the following core services:

1. UFT GUI Testing

Enables automated testing of graphical user interfaces, allowing you to validate and verify the visual and functional aspects of an application's interface across different platforms.

2. UFT API Testing

Provides tools for testing APIs and other service layers, enabling users to verify backend functionality, data processing, and integration points without relying on a GUI.

3. Business Process Testing

Supports the automation of complex business workflows by allowing users to define, manage, and test critical business processes across different layers of an application.

In addition, UFT allows seamless integration of GUI and API tests within a single automated test, which means GUI tests can call API tests and vice versa. This integration empowers testers to build versatile test suites that ensure application reliability from both the user interface and backend perspectives.

International Journal of Core Engineering & Management
Volume-4, Issue-11, February-2018, ISSN No: 2348-9510

III. HOW TO START UFT

To open UFT after it has been installed, navigate to the Start menu, go to All Programs > HP Software > HP Unified Functional Testing, or simply double-click the UFT shortcut on your desktop if one is available. On startup, UFT will display the Add-in Manager dialog by default, listing all currently installed add-ins. You can choose specific add-ins to load or adjust your settings to skip this step on future launches. In the Add-in Manager dialog box, select the add-ins you want to load and click OK. The UFT window opens, displaying the "Start Page." [2]

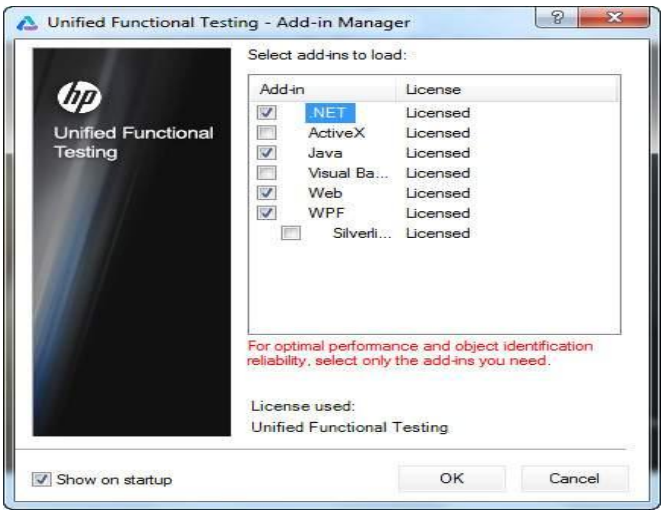

Figure 2: Add-in Manager Dialog Box [2]

A. UFT Welcome page

When UFT opens, the Welcome page introduces you to the latest features and enhancements in the current release, with links to relevant documentation that can help you explore and understand these updates.

1. **Recent Tests/Components:** The "Recent Tests" section enables you to quickly access previously created tests, components, or solutions. By clicking the "New" or "Open" buttons, you can initiate the creation of a new test or open an existing one. This feature streamlines your workflow by allowing easy navigation through your recent work.

2. **Startup Options:** Users can customize their experience with options available for viewing the Start Page. By selecting Display the Start Page on Startup, UFT will automatically show the Start Page each time it launches. Alternatively, choosing Close Start Page after Test Loads will close the Start Page automatically after a test has been loaded, helping to minimize distractions during testing.

3. **What's New Area:** To stay informed about the latest advancements, users can click the What's New button. This feature provides a comprehensive overview of the most recent enhancements, new features, and supported environments included in the latest version of UFT.

International Journal of Core Engineering & Management
Volume-4, Issue-11, February-2018, ISSN No: 2348-9510

4. **Community Area:** Engagement with the UFT community is facilitated through the Community button, which directs you to important updates about UFT. Here, you'll find links to forums dedicated to GUI Testing, API Testing, and Business Process Testing (BPT), as well as access to the UFT blog for additional insights and discussions.

5. **Links Area:** For quick reference, the Links button opens up essential resources, including the UFT Help documentation, the HP UFT Object Model Reference specifically for GUI Testing, and tutorials that cover both GUI and API Testing. These resources are designed to enhance your understanding and usage of UFT.

6. **Support Area:** Lastly, the Support button offers access to vital support resources, including links to UFT support services, the Knowledge Base, and troubleshooting sections on the HP Software Support website. This area is crucial for resolving any issues you may encounter during your testing processes.

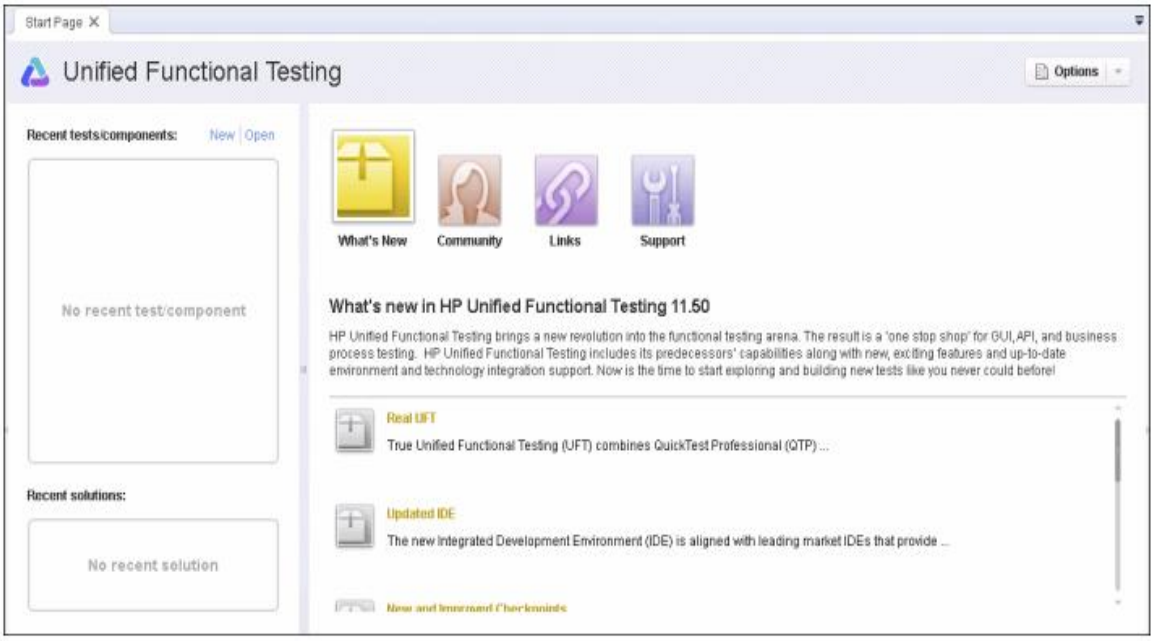

Figure 3: UFT Welcome Page [2]

B. Creating a New Standalone Document

To create a new standalone document:

1. Select File > New and select the type of document to create, or
2. Click the New button-down arrow and select the type of document to create from the drop-down list.
3. A dialog box opens. If you are creating a test or component, you will see the "New <Document> Dialog Box." This dialog box enables you to specify the type of test or component you want to create, the location in which the new item is stored, and, optionally, the solution in which it is included.

International Journal of Core Engineering & Management
Volume-4, Issue-11, February-2018, ISSN No: 2348-9510

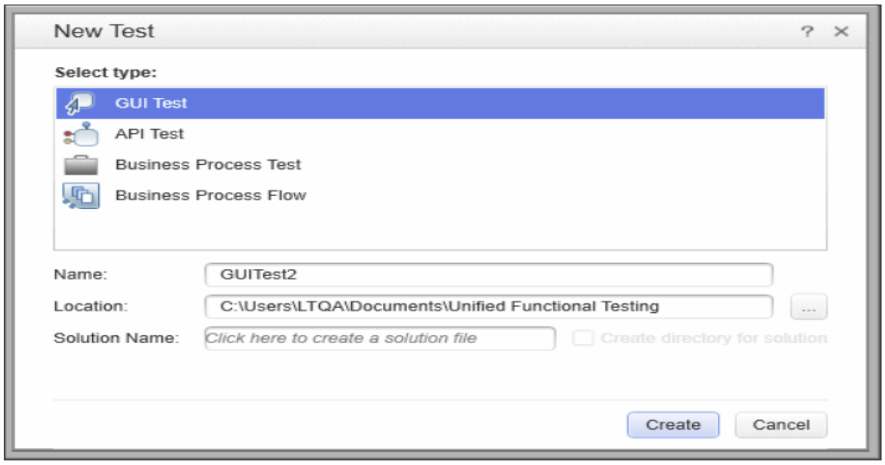

Figure 4: Create a new test (File > New > Test) [2]

C. UFT GUI Testing

UFT's GUI testing solution deploys the concept of keyword-driven testing to enhance test creation and maintenance.

1. Basic GUI Test Structure

- A test is comprised of calls to actions. Tests can be organized into logical segments, known as actions, which represent distinct parts of a website or specific tasks within an application. By creating tests that call multiple actions, you can design tests that are more modular and efficient.
- Each action is comprised of steps. As you add steps to an action, they are displayed in the UFT window, either as part of the Keyword View or the Editor. In the Keyword View, every step includes automatically generated documentation that provides a plain language textual description of what the step does.
- When you perform a run session, UFT performs each action in your test. After the run session ends, you can view a report detailing which steps were performed and which ones succeeded or failed.
- Keyword-driven testing is a technique that separates much of the programming work from the actual test steps. This enables you to create test steps earlier and maintain them with only minor updates, even when there are significant changes in your application or your testing needs.
- The keyword-driven testing methodology provides automation specialists comprehensive access to both test and object attributes through a unified scripting and debugging interface that maintains synchronization with the Keyword View.
- Experienced testers knowledgeable in VBScript can utilize the Editor feature to incorporate and modify code statements, thereby augmenting the functionality of their tests.

International Journal of Core Engineering & Management
Volume-4, Issue-11, February-2018, ISSN No: 2348-9510

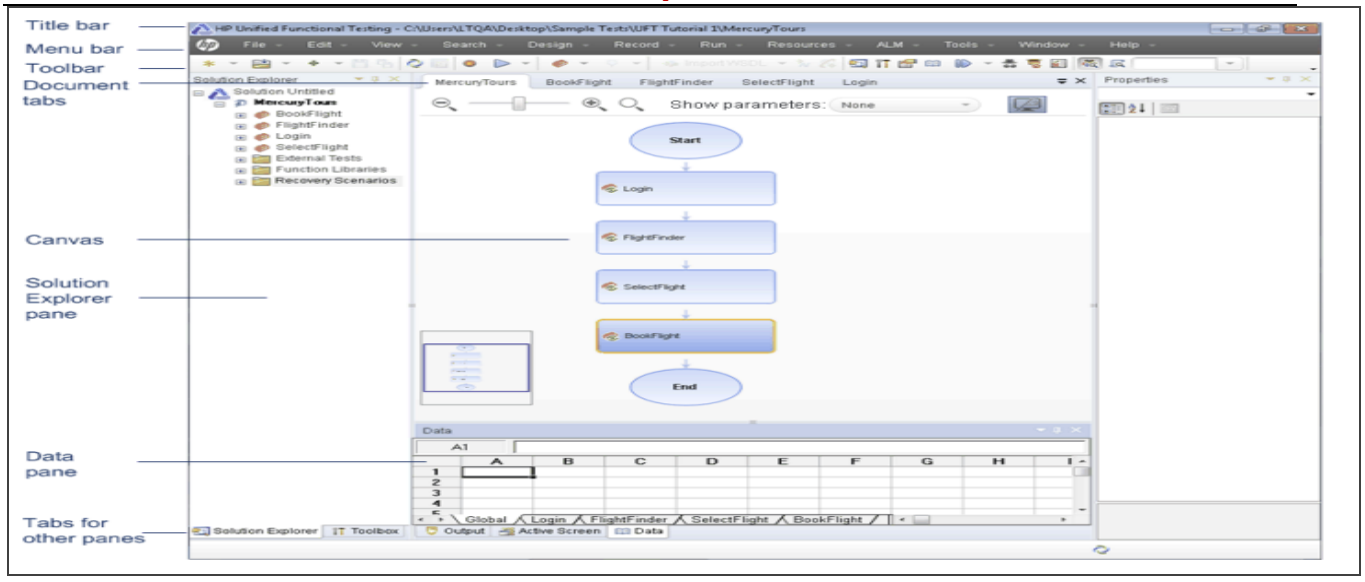

Figure 5: UFT Main Window [4]

2. GUI Testing Workflow

2.1 Stage 1: Application Analysis

Begin by identifying the development environments used to create your application's controls. Establish the core functionalities you want to verify in the testing process. Strategically organize these functionalities into manageable units that can be tested through specific actions. [3]

2.2 Stage 2: Testing Infrastructure Preparation

Assemble the necessary resources required for the testing. Adjust UFT settings to suit the specific testing requirements. Create one or more repository tests that serve as the foundation for action storage.

2.3 Stage 3: Building Action Steps

Actions can be structured using the intuitive, graphical Keyword View, or alternatively, you can use the Editor for direct coding with VBScript. Additional methods to add steps include dragging objects or recording actions directly within the application.

2.4 Stage 4: Test Enhancement

Refine the test process by using advanced testing features or adding programming commands. Examples include:

- Inserting checkpoints and using output values to capture and verify data.
- Substituting static values with parameters to expand the test's versatility.
- Creating function libraries for user-defined functions, which can be called within the test.

2.5 Stage 5: Test Execution and Debugging

Once your test is created, conduct various types of runs to meet different objectives, such as:

- Debugging the test for accuracy.
- Executing the test to validate application functionality.
- Updating the test as necessary.

International Journal of Core Engineering & Management
Volume-4, Issue-11, February-2018, ISSN No: 2348-9510

2.6 Stage 6: Reviewing Results and Reporting Issues

After the test run, results are available in the Run Results Viewer, where both summaries and detailed reports are provided. Defects identified during the session can be reported. The system supports:

- Viewing visual captures (images or videos) of the test.
- Local system performance monitoring.
- Logging issues in ALM, either automatically or manually.

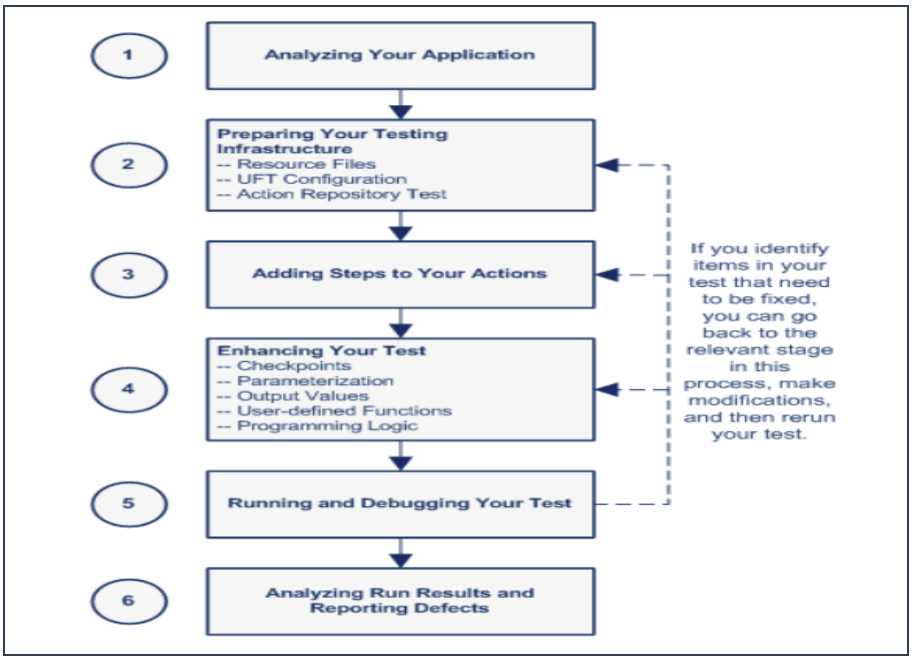

Figure 6: GUI Testing workflow [3]

D. UFT API Testing

UFT's API (service) testing solution provides tools for the construction and execution of functional tests for headless (GUI-less) systems. For example, you can use UFT to test standard Web Services, non-SOAP Web Services, such as REST, and so on.

You create an API test by dragging and dropping activities from the UFT Toolbox pane into the test, displayed in the canvas. The toolbox provides a collection of activities for functional testing in areas such as REST, Web Services, JMS, and HTTP. You can add more activities to the toolbox by importing WSDLs or providing other contract definitions.[4]

International Journal of Core Engineering & Management
Volume-4, Issue-11, February-2018, ISSN No: 2348-9510

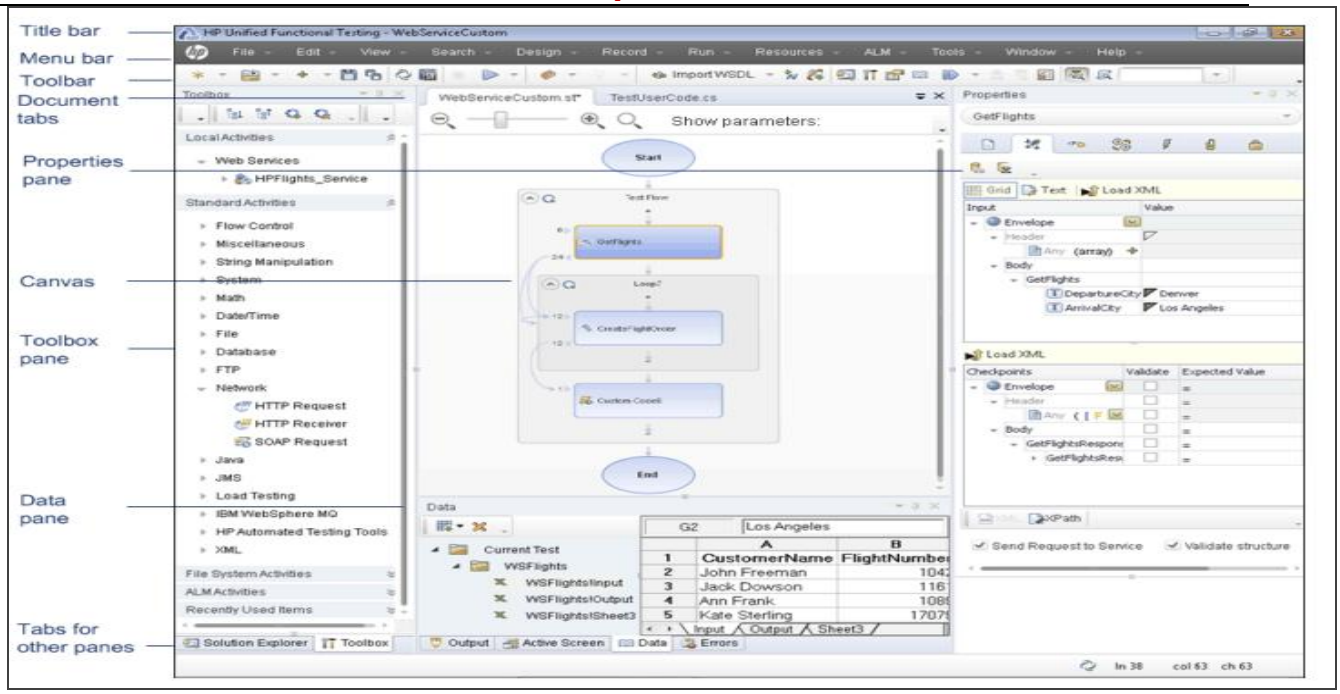

Figure 7: UFT main window when an API test is open. [4]

E. UFT Integrated Testing

You can integrate your GUI and service testing processes in a single test by including calls from your GUI test to API tests, or from your API tests to GUI tests. When you insert a call to another test, the call is displayed as nested under the relevant action in the canvas.

- Insert and modify calls to API tests from GUI tests using the "Call to API Test/Action Dialog Box."
- You can incorporate and adjust calls to GUI tests within API tests by selecting and dragging the Call GUI Action or Test from the HP Automated Testing Tools section in the Toolbox pane onto the test canvas.

F. Integration with ALM

UFT can be seamlessly integrated with Application Lifecycle Management (ALM) to facilitate comprehensive management of the testing workflow. This integration allows for:

- **Create a project:** Establish a central project within ALM to store both manual and automated tests, ensuring easy access and organization.
- **Build test cycles**: Create and manage test cycles to efficiently plan and execute testing phases.
- **Run tests**: Initiate and monitor test runs directly from ALM, streamlining the testing process.
- **Report and track defects**: Log and track defects found during testing, maintaining a clear overview of issues that need resolution.

Additionally, ALM enables the generation of reports and visual graphs, aiding in the assessment of test planning, execution progress, and defect tracking prior to software releases. Tests and components developed in UFT can be saved into the ALM project, allowing users to

International Journal of Core Engineering & Management
Volume-4, Issue-11, February-2018, ISSN No: 2348-9510

execute UFT tests and effectively manage and analyse results within the ALM environment.[5]

G. Business Process Testing in UFT

Business Process Testing (BPT) works within UFT or ALM as a component-based testing framework. Working with a testing framework provides many advantages to enterprises, including streamlining the creation and maintenance of both manual and automated tests and maximizing efficiency for testing complete business processes.

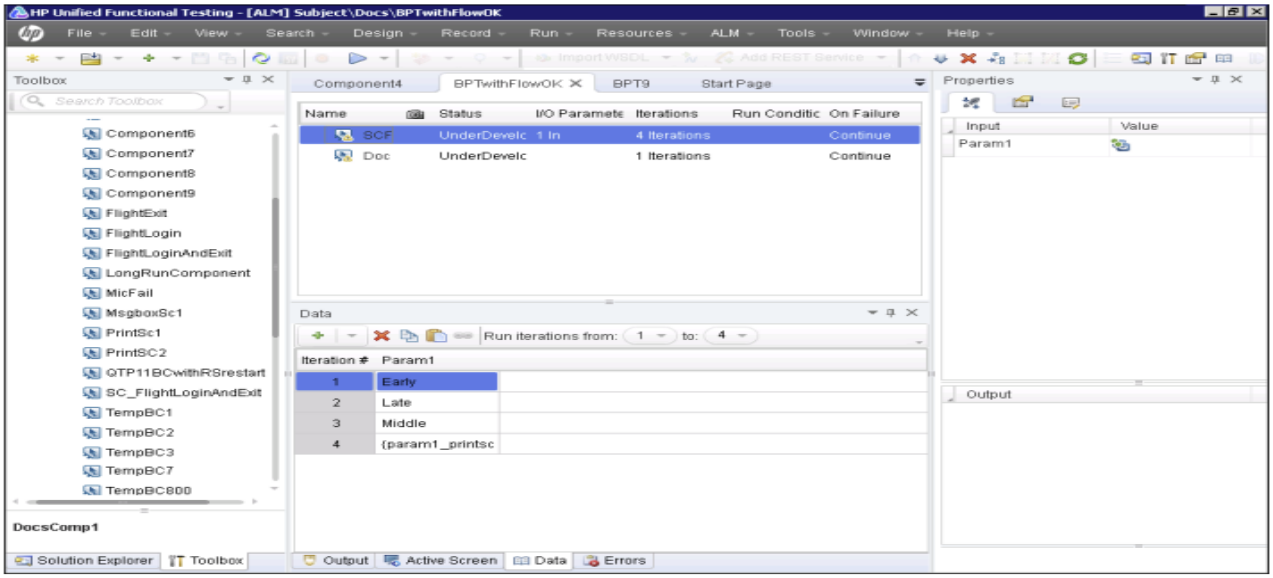

Figure 8: Business Process Test [4]

H. Automatic Export of Run Results

This dialog box enables to define settings that instruct UFT to automatically export run results to a specific folder in the file system after a run session. [6]

1. Accessing Export Settings

Go to Tools > Options > General > Run Sessions to locate settings that control run result exports.

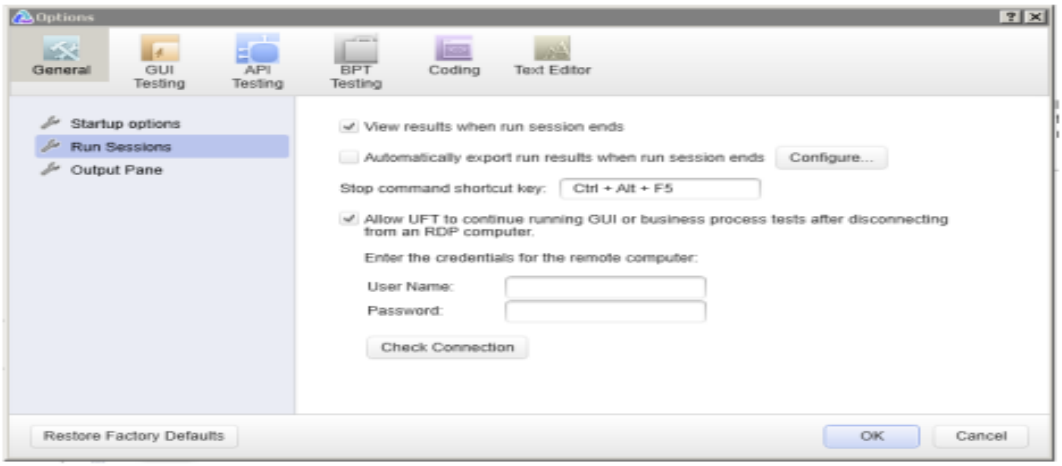

Figure 9: Run result configuration window [6]

International Journal of Core Engineering & Management
Volume-4, Issue-11, February-2018, ISSN No: 2348-9510

2. Setting Export Options

Within the Export Format Settings window, you can specify various settings for your exported results:

2.1 Step Details: Includes each test step in the report.

2.2 Export Type: Choose between HTML or PDF; HTML is the default setting.

2.3 Export Format: Options include:

2.3.1 Short: Provides only a summary for each test step.

2.3.2 Detailed: Exports comprehensive information, including captured images, for each step.

2.3.3 User-Defined XSL: Uses an XSL file for customized report formatting.

2.4 Screen Recorder: Adds recorded video footage from the test execution.

2.5 Data Table: Exports the runtime data table, including any test output values.

2.6 Log Tracking: Logs from the application are included in the exported results.

2.7 System Monitor: Records system metrics captured during the test.

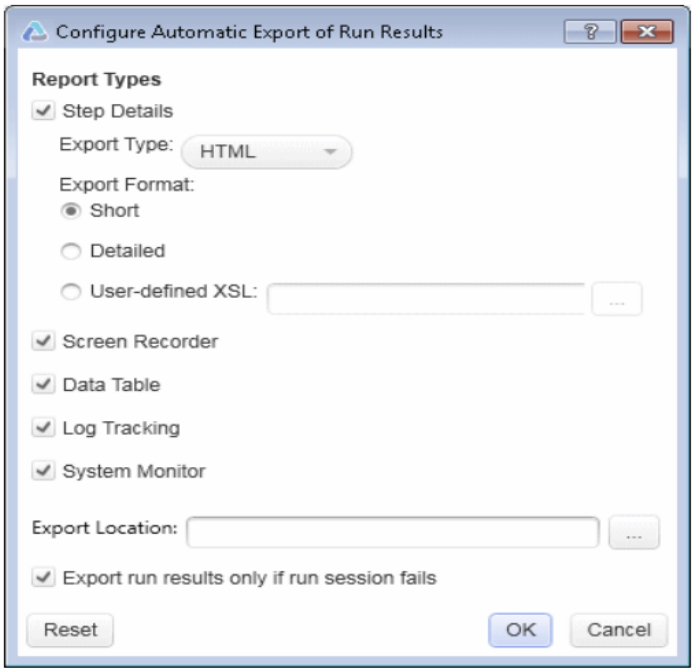

Figure 10: Configure Automatic Export of Run Results Dialog Box [6]

Define the Export Location where files should be saved. By default, this is the Test Folder > Result Folder > Report directory. Additionally, the Export on Failure Only option can be used to export results only when tests fail.

IV. LIMITATIONS/CHALLENGES

1. Resource Intensive Memory and Processing Power:

UFT is a resource-intensive application. Running complex test scripts with multiple actions, especially for GUI testing, often requires high memory and CPU usage. Users working on resource-constrained systems may experience slower response times, leading to longer test

execution periods.

2. Compatibility Constraints

Limited Cross-Browser Testing: UFT's cross-browser support, while improving, may not be as extensive as other dedicated testing tools. Ensuring compatibility across all major browsers and devices can still be challenging, especially with frequent browser updates.

3. Cost Constraints

Licensing Costs: UFT is a commercial tool with licensing fees, which can be prohibitively high for startups and smaller organizations. While it provides extensive features, the cost factor may limit accessibility for teams with budget constraints.

V. CONCLUSION

1. Comprehensive Automation: UFT provides robust capabilities for automating both GUI and API testing, ensuring a thorough evaluation of applications.
2. Unified Framework: The integration of GUI and API testing within a single framework enhances testing efficiency and simplifies the testing process.
3. Business Process Testing: BPT enables testers to create modular, component-based tests that streamline the testing of complex business processes.
4. User-Friendly Environment: UFT's features accommodate both technical and non-technical users, promoting collaboration and enhancing productivity.
5. Efficient Test Management: The ability to manage tests, track results, and generate reports within UFT facilitates better oversight of the testing process.
6. Encouragement to Utilize Features: Testers are encouraged to leverage UFT's capabilities to improve test design, execution, and reporting, ultimately leading to higher software quality and reliability.

REFERENCES

1. https://sumeetkushwah.com/2016/03/07/hp-uft-brief-introduction-and-history/
2. https://www.tutorialspoint.com/qtp/qtp_environment_setup.htm
3. https://ozkanuslu.weebly.com/hp-qtpuft/functional-testing-page-under-construction
4. https://qualitat.solucions.gencat.cat/manuals/hp_man_UFT12.00_UserGd_pdf.pdf
5. https://blogs.perficient.com/2017/08/15/automated-web-testing-with-hp-alm-uft/
6. https://diptichaudhary.wordpress.com/2014/05/28/uftqtp-exporting-run-results/