

The Case Against HP UFT for SAP GUI Testing

Automated testing script languages require specialized programming skills, and the more code that is created, the greater the time and effort to introduce the modifications needed to keep pace with functional changes. Adopting a non-script-based approach to testing can dramatically reduce maintenance costs while improving end-to-end business process coverage and automation rates.



Contents

Introduction	3
HP UFT Shortfalls for SAP GUI Testing.....	4
Case 1 – SAP Customizations	4
Case 2 – Configuration Changes to Standard Business Processes	4
Case 3 – SAP Feature Updates	5
Case 4 – Non-SAP Application Integrations	5
The Hidden Costs of HP UFT.....	6
The Case for Object Orientated Test Automation.....	7
Gains of Using an Scriptless Object Orientated Approach to Testing.....	8
How Worksoft Overcomes the Complexity of SAP Test Automation	8
Go Beyond Testing and Accelerate Time to Value.....	9
Summary	11

Introduction

Companies that invest in test automation tools are already a step ahead of most of their competitors. They understand the value of quality assurance, and the value that automation can bring. Companies that made the initial investment several years ago adopted first generation tools that were script-based — tools that can only take them so far. The reason being that the productivity gains from automated execution are more than offset by the time and cost of developing and maintaining custom script code. Automated testing script languages require specialized programming skills, and the more code that is created the greater the time and effort to introduce the modifications needed to keep pace with functional changes.



The Trouble with Script Based Approaches? Coverage.

On average clients report a 20% automated business process coverage rate using HP UFT.

After migrating to Worksoft clients have been able to achieve a coverage rate of 80% while significantly reducing the number of tests they have to maintain.

HP UFT Shortfalls for SAP GUI Testing

It can of course be argued that being code-based doesn't matter if you never see the code. The issue is that no two SAP implementations are exactly the same, no two SAP versions are precisely identical, and any time changes are introduced either by a user or SAP, the script code itself must also be changed. And, of course, not all enterprise applications are limited to SAP.

Case 1 – SAP Customizations

In the most obvious case, an SAP customer may implement customizations in the form of Z- transactions to accommodate special requirements. In this case, no pre-developed business components exist and so they must be developed from scratch. And while it may seem that the keyword interface to UFT simplifies this task, the code that is generated is strictly limited to linear, simplistic actions. Any sophisticated decision-making or error-handling functionality must be created as code by a programmer in the expert view. And, naturally, this code must be tested and debugged just as any other code must be before it can be made available as a reusable business component.

Case 2 – Configuration Changes to Standard Business Processes

Case 2 arises when the SAP customer introduces any configuration changes that modify the standard business process procedures in any way. In fact, many enterprises deliver competitive differentiation through specialized business processes; it is even possible to patent unique processes, such as the Amazon One-Click ordering technique. No doubt a large degree of SAP's success can be traced to their ability to enable customization of business processes without the development of custom code.

Since this very flexibility is a strength of SAP's design, it is not only common but expected. With HP UFT, the pre-developed script code must be modified by a programmer to reflect any configuration changes, and these changes also must be tested and de-bugged.

Case 3 – SAP Feature Updates

The third case, which is not only inevitable but often the driver for most test efforts, is that SAP will introduce new capabilities in the form of new releases or modules. When this occurs it may require not only new components to be developed from scratch, but also any existing components to be modified, tested and debugged.

Case 4 – Non-SAP Application Integrations

Case 4 and yet perhaps most severe case occurs when any non-SAP application is integrated into an end to end process flow. Because the pre-developed content is specific to SAP, any additional applications must be implemented completely from scratch. Further, these applications may span other platforms that are not supported by the UFT scripting language, thus requiring the introduction and integration of yet another tool and skill set.

In other words, from the moment a script based solution is implemented, the underlying script code shifts from being an asset that accelerates testing to a liability that slows it down by requiring programming resources to develop, debug, test and maintain potentially tens of thousands of lines of code.



The Hidden Costs of HP UFT

Aside from the self-evident costs of time, money and resources to devote programming resources to the care and feeding of a substantial code base, there are also hidden costs that don't become evident until the implementation is well under way. These costs arise because the very structure that is employed to hide the code also serves to limit functionality and flexibility.

For example, it is a common requirement during test execution to retrieve data values that are generated in one transaction and supply them to a later transaction. The total price of an order, let's say, should be verified against the value transferred to an invoice or statement.

But because of the way the pre-developed business components are designed to be reusable, this type of integration across components is rarely available. As a result, this otherwise simple and straightforward concept cannot be implemented within the high level drag and drop view unless supporting code is first developed to enable it.

A more pervasive problem arises whenever the business analyst designing the test needs to make decisions based on the results of the test during execution. This type of flow control is a normal and even essential capability to handle known and unknown risks that may affect the execution of a transaction. For example, something as simple as a pop-up window may need special handling in order for the test to proceed.

Some might argue that predictability is a hallmark of a well-designed test and that is certainly true, but that doesn't obviate the need to address the unavoidable situations and circumstances that may impede test execution. Without the power to make decisions during execution, tests become more fragile and results less predictable,

HP UFT Sample Script with Logic

```
Function MyClose(oWindow)
  Dim iTimer 'Will help us
  count time
  iTimer = Timer
  oWindow.Close 'Initiate
  the VBWindow original
  close command
  Do
    If
      Dialog("micclass:=Dialog").
      Exist(0)
    Then _
      Dialog("micclass:=Dialog"
      ).Type MicEsc
      'Deal with any pop-ups
      Loop Until (Not oWindow.
      Exist) Or (Timer-
      iTimer>15)
      'Wait for the window to
      close or the command to
      time out
  If oWindow.Exist Then
  MyClose = False
  Else
    MyClose = True
  End If
End Function
```

creating additional overhead for diagnosing and re-running failed tests – often manually. However, the HP UFT high level view that supposedly simplifies test design provides no such sophistication. All decision-making and error-handling logic must be implemented as script code.

And because test results are the ultimate outcome – and value – of any automated execution, it is essential that they are easily understood and diagnosed. Unfortunately, this is where a code-based approach reveals one of its greatest weaknesses. When HP UFT is running a test, what is actually being executed is VBScript code. That means any diagnosis of errors or failures must occur at the code level.

But because HP UFT is implemented through multiple layers and products, tracing issues through the various components is itself a challenge. Each product requires training, support and expertise and the integration among them introduces obstacles to understanding, debugging and removing issues.

As you can see, the layers that have been added to obscure the code from test designers ultimately serve to limit the user's ability to develop flexible, reliable, and understandable tests, all adding to the overall costs of maintaining HP UFT test scripts.

The Case for Object Oriented Test Automation

No matter how good the technology is, it is still only a tool. It won't jump out of the box and start testing your software by itself. For automation to work, clients have to invest the effort to design and build a reusable library of business process tests to reap the benefits. Automation has to be integrated throughout the complete life cycle, and continually assess your risks and improve your coverage every time there is a change. You not only have to keep your existing tests current, you have to keep adding new ones, because the longer your application is in use the more functionality it will have and the greater the risk of unexpected impact. So you can't just plan a quick project to automate and then stop: those tests will eventually become obsolete and your application will continue to expand.

All the time you save by liberating your manual resources is returned to you in the form of improved business processes, more enhancements and streamlined operations. And the greater your test coverage the fewer emergencies and outages you will experience, which will improve user productivity and free IT resources to reduce the backlog and improve response time to new business needs.

No matter how good the technology is, it is still only a tool. It won't jump out of the box and start testing your software by itself.

Gains of Using a Scriptless Object Orientated Approach to Testing

Worksoft automation references enterprise applications and dynamic web pages at the object level – from capture to visualization, maintenance and test execution. This means that when application objects move or change, Worksoft tests continue to work as designed and the maintenance burden in maintaining your automation is minimized. This is in stark contrast to legacy approaches that HP UFT uses which requires scripting and programming. With Worksoft, there's no scripting and no need for programming skills. Worksoft Certify was first to market an extremely easy to use solution designed specifically for both business analysts and automation specialists that was powerful enough to validate end-to-end business processes without programming or scripting.

How Worksoft Overcomes the Complexity of SAP Test Automation

Worksoft Certify is an automated business process validation solution that was designed to enable the rapid creation of automation that is resilient to application change and easily maintainable. The single user interface for test creation, execution and maintenance is intuitive and easy to use. Our patented Object Action Framework enables the creation of test steps that are stored in the Certify database. These test steps can then be manipulated much like rows in a spreadsheet to accelerate test creation. There is no code generated and there is no programming required to create and maintain tests. Test case creation is simplified, the tests are more resilient and the effort to maintain tests is dramatically reduced vs legacy code based testing tools.

With Certify, all test assets are stored in a shared relational repository, including the application maps (information about your application objects), tests and test steps, the data used to drive the tests (recordsets) and results. Traceability is automatically generated between every application screen, field, test and data record set, so it's easy to understand how to maintain the test when changes occur in your application.

Certify test cases are built on the foundation of test steps, simple narratives that explain the testing activities being performed. Certify Capture with Analyze allows end users/subject matter experts to capture the test steps by observing as they navigate through the application, and stores all user actions via a visualized business process workflow.

Worksoft Certify was first to market an extremely easy to use solution designed specifically for both business analysts and automation specialists that was powerful enough to validate end-to-end business processes without programming or scripting.

A Worksoft test never needs to be re-recorded or regenerated as the application or process changes. You can easily go into an existing Worksoft process and add/modify/delete new steps where required. If an application object changes, a change would be performed once, any and all related tests will automatically use that new object information.

Complex scenarios can easily be handled through our “out of box” powerful actions, which provide support for more complex items, such as grids/tables/trees, without the need for programming activities. Traditional numeric and text functions (like math, substring, etc) are provided as actions. Looping and decision making (On True / On False) provide for advanced functionality, and flow control. Our many customers have demonstrated that all complex scenarios can be easily handled through our non-programming approach.

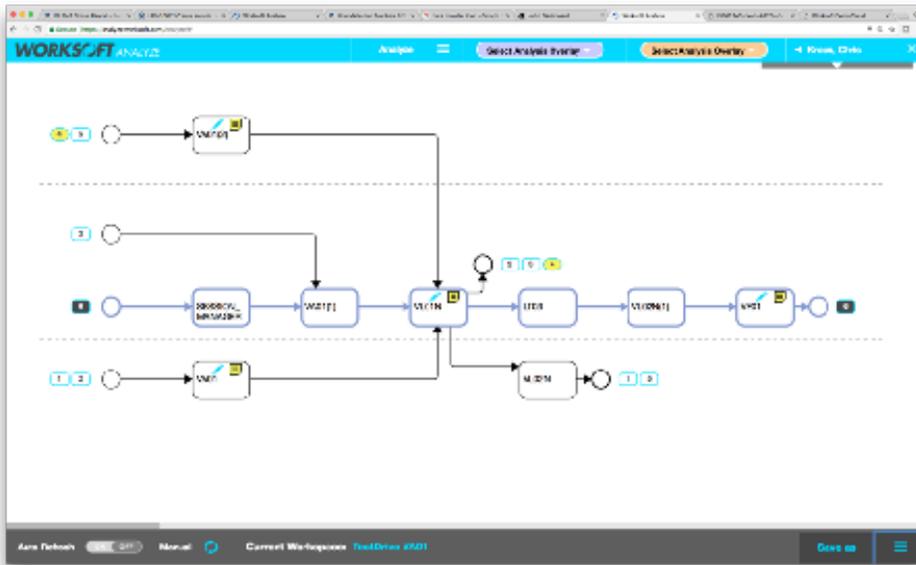
SAP GUI automation can be further simplified with Certify Impact. Certify Impact is an automated risk based testing solution that can identify, recommend, and execute end-to-end business process tests based on a set of proposed SAP application changes directly from the transport(s). With each transport, the Certify Impact engine determines the SAP Programs and T-Codes that are being impacted, and subsequently to be tested, as a result of the transport. It also identifies all SAP screen changes, and uses this information to reconcile against the Worksoft Certify repository for determining the appropriate tests to be adjusted and executed as a result of the proposed transport change. ST03 usage metrics are also used to determine the importance of the changes to the business process.

“Because there is no syntax involved with Worksoft Certify, process and actions are more like executable documentation, making maintenance much easier.”

IT Central Station
Real User Review

Go Beyond Testing and Accelerate Time to Value

Understanding the business process is key and it begins by using Certify Capture and Worksoft Analyze. Analyze will capture the steps they have taken and uses artificial intelligence to connect the tasks of multiple users into an end-to-end process. It is simple, quick, and can handle thousands of process variations. By helping the Business Analyst discover, visualize, document and optimize business processes, Analyze accelerates the communication between the business and IT, giving each team a common language and the technical assets required for successful automation project implementation.



Consider this: Over time, many companies have created sizable repositories of UFT test assets. These assets are often built in large, distributed environments where design and best practices can become lost, and communication between testers and developers is limited. Developers will often decide it is faster to create a new test instead of navigating the UFT functional libraries to find an existing test that meets their requirements.

This duplication of effort is expensive – the larger the number of tests, the more costly the maintenance. It can also result in lower overall test coverage, because you eat up test windows running the same tests. Worksoft Analyze can be used to automatically discover duplicates and similar use cases. Using Analyze for this purpose, we’ve seen an average reduction of 60% to 80% in the total number of UFT cases needed to obtain the same coverage in Worksoft Certify.

Once discovered within Analyze, all captures are easily imported into Certify. Certify was designed so that anyone in the organization can build tests with point and click functionality. Test scripts are easy to read and understood as they are written in plain English.

Fewer Tests. Better Coverage.
 After migrating to Worksoft, clients have been able to achieve a coverage rate of 80% while reducing the number of tests they have to maintain by 60% to 80%.

Summary

Ensuring quality execution of business processes across the enterprise is a challenge most organizations must tackle for success. Doing so is increasingly difficult because of the speed of technology changes, growing demands on systems, and the complexity typical of enterprise SAP landscapes. Your business depends on business process quality and the integrity of its critical transactions. The failure of enterprise systems costs time and money, and brings with it the risk of business disruption. Test automation has come a long way.

First generation script-based automation tools have proven fragile and unable to keep up with high change environments. Modern solutions like Worksoft Certify are specifically designed to address these problems and support the unique requirements of SAP systems and SAP customers. Worksoft Certify is able to do this for four specific reasons:

- **Simplicity** – It's designed to be simple enough to be used by business SMEs yet offers the sophistication for even the most advanced IT users. No scripting and no programming required.
- **Speed** – It delivers results in a matter of days and weeks, thus delivering a rapid return and motivating teams to continue to do more. This also shortens project timelines.
- **Resilience** – Worksoft tests continue to work, even as enterprise applications and business processes change. This reduces maintenance costs and improves quality.
- **Completeness** – It comprehensively covers all SAP and non-SAP applications and interfaces that are typically present in any modern SAP landscape. This means that true end-to-end business process test automation can be achieved with Worksoft Certify.

Worksoft Certify and Analyze makes it easy for companies with prior investments in HP UFT to leverage that investment as they move to the industry's leading solution for end-to-end business process validation.

Ready to Escape the Scripts of HP UFT?

The HP UFT replacement program is designed to help clients quickly and easily convert an existing library of SAP-based UFT (formerly QTP) functional tests into end-to-end Worksoft business process tests – with minimal effort.

For more information on the Worksoft HP UFT Replacement Program, download the [ebook](#) or email sales@worksoft.com.

WORKSOFT®

Worksoft® is a leading global provider of automation software for high-velocity business process testing and discovery. Enterprises worldwide use Worksoft intelligent automation to innovate faster, lower technology risk, reduce costs, improve quality, and deeply understand their real end-to-end business processes. Global 5000 companies across all industries choose Worksoft for high-speed process discovery and testing of digital, web, cloud, mobile, big data, and dozens of enterprise applications, including SAP®, Oracle, and Salesforce.com.

For more information, contact Worksoft at sales@worksoft.com or visit www.worksoft.com.