accenture

StormTest

WHITEPAPER

10 Top Tips for Getting Started with Multiscreen Test Automation

Contents

This paper was written by Robert Colbert, Software Architect, Accenture Digital Video.

Robert has been working in technical leadership roles in the Broadcast and Communications domain for 24 years, with specialist knowledge in the areas of video platforms, consumer electronics and platform integration.

Introduction

When transitioning from manual to automated testing, there are a number of challenges that must be addressed to create a successful automation program.

In this whitepaper, Accenture Digital Video (ADV) provides advice to those considering automating their multiscreen testing process – whether that be a Set-Top Box (STB), gateway, game console, connected TV or mobile application.

These insights are informed by our work with Accenture StormTest, an advanced automated video test platform. This platform is replied upon globally by more than 50 customers to reduce test cycle time, and improve platform stability.

Creating a successful automation program is not achieved solely by selecting an automated test solution. When getting started with test automation, in addition to choosing a fit-for-purpose test automation product, to facilitate a smooth transition to automated test and ensure the long-term sustainability ADV recommends that particular attention be paid to three areas:

- Architecture and maintainability
- Script development
- Test result management

"Test Automation teams typically have a different profile to manual teams"

Architecture and Maintainability

#1 Establish a maintainable architecture

A key objective should be to put in place an easily maintainable scripting architecture that can handle change.

Platforms, colors and layouts will change, along with frequent channel swapping in the EPG lineup. Scripts must be written to handle change quickly and easily. Note, it is not necessarily the changes themselves that take time, but the effort to verify that all the scripts still work once changes have been implemented.

The more scripts you have, the more time that is needed to ensure that these scripts will work correctly, after changes have been made. The target is to avoid having to update manually thousands of scripts for a minor change.

Therefore, the architecture should enable the reuse of code so that when changes need to be made they need only be carried out once, in a unique location. This allows new test cases to be developed smoothly, as well as providing for example the ability to easily handle the same application running on STBs from different manufacturers.

#2 Create a plan to integrate automated & manual tests results

Test automation must work in collaboration with the current manual test process, including integrating thousands of test results. The architecture for processing these results needs to be efficient so that automated testing contributes to reducing the overall time and effort, while improving quality.

The main challenge with manual and automated testing running in parallel is with the results analysis. Correlating manual and automated results is an important outcome. From a high-level reporting view, the ability to see the overall status of your tests is crucial, regardless of whether these were run in an automated or manual fashion.

#3 Establish a robust, reliable & sustainable libraries

The ability to reuse code and to easily develop new test cases is of great importance. When writing scripts, test developers should think long-term, and work in such a way that any other developer or tester may be able to reuse it as it is, adapt it, or re-use elements for new test cases.

Creating reliable, sustainable and multi-platform script libraries provides an opportunity to have different levels of scripting expertise and knowledge.

A key objective of automation is to save man hours. It is recommended that test modules be organized in a way that allows for quick and straightforward identification of required code, to avoid spending time locating the desired code elements.

The stability of code impacts the testing process, and test outcomes. A great part of successful automation lies in the stability of code. At the risk of stating the obvious, in a manual testing world when an STB exhibits a slight glitch, perhaps unrelated to the feature under test, the manual tester may take a look, decide it is unimportant and carry on. This is not possible with automation. This is because the script flow may not be able to continue reliably at that stage if the 'glitch' is unexpected. In order to generate reliable results, the code must handle unexpected events and know what events affect the test outcome and what events can be worked around.

"The target is to avoid having to update manually thousands of scripts for a minor change"

Script Development

#4 Robust software development from the outset

Scripting tools should accommodate both power users and those preferring a more GUI driven approach, allowing developers and testers to get started with test automation immediately. However, rather than launching straight into script creation, a robust software development approach should be adopted when creating the initial scripts and libraries.

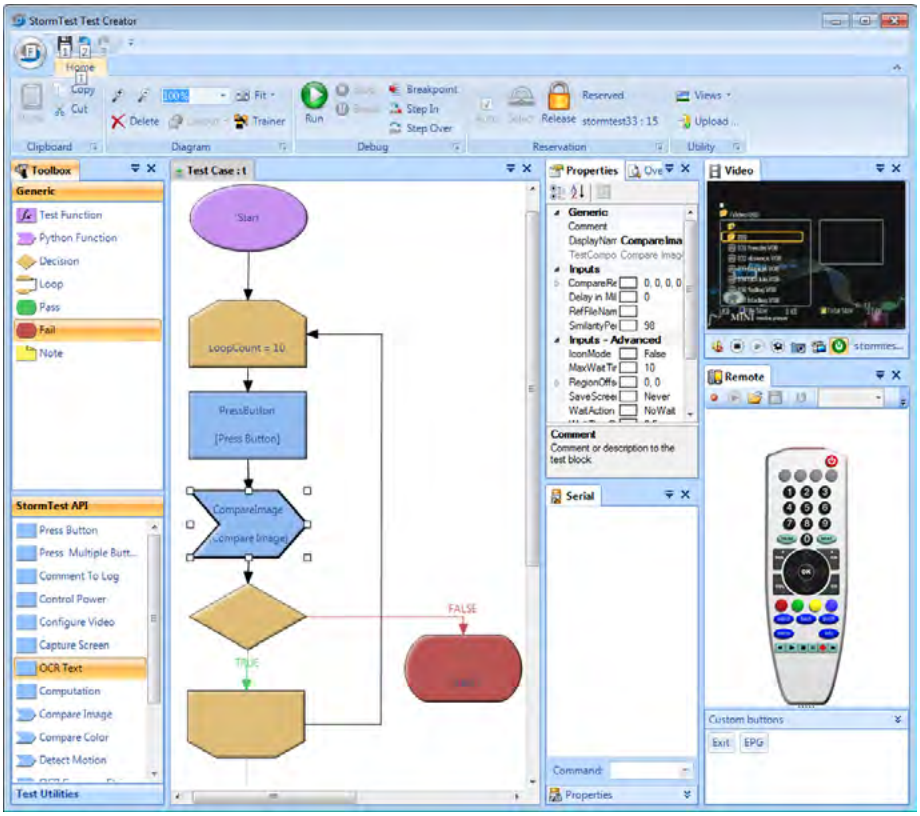
Testers may then use these scripts without requiring an in-depth knowledge of the scripting language, while developers maintain the library architecture and to overcome tougher issues.

As long as those preparing test scripts are familiar with the test and the application being tested, they should be capable of automating them. ADV has seen scenarios where test automation teams include personnel with profiles that have a deeper knowledge of the device under test compared to being expert developers in the scripting language.

While manual scripts are easy to read and understand, they may not be easily automated. When developing test cases for automation, test developers have to use a standard method that will be easy to automate and adapt later on. Having this in mind, it is considered good practice to encourage the manual test team to write test scripts in a way that would match the requirements of an automation process.

"When writing scripts, test developers should think long-term and work in such a way that any other developer or tester may be able to reuse"

Figure 1. Creating test scripts with StormTest



#5 Develop tests to the lowest common denominator

When developers have to update an automated script, it can take considerable effort to understand it, in order to identify the elements that require updating and enhancement. It is strongly recommended that effort be put in from the start to develop scripts to the lowest common denominator of receiver device. By 'lowest' this could be the device with the slowest response time to IR key presses, or to the lowest common software release version.

It is our opinion that the cost of updating existing scripts is actually much higher than putting this effort in from the start and will allow for the proportion of automated tests to grow at a more regular pace.

#6 Give priority to easily automatable tests

Once the decision to automate a test script has been made, another key element is to take the time to clearly define the purpose of the test, i.e. to define what is being automated and understand how clear the purpose of the test is.

The recommended test implementation strategy is to prioritize those scripts that are the easiest to automate. A few factors should be taken into account in defining which tests are to be automated first:

- How many man hours will be needed to automate vs. man hours saved once automation is implemented?
- How frequently is a test executed? Tests run very frequently (i.e. nightly) should be automated before those that are run infrequently.
- How many similar tests may be grouped together? Once a test is automated, similar tests may be automated very quickly based on this.

As such, one of the first tasks for the developers is to get the manual run times for all tests. Tests should be categorized according to how easy they will be to automate.

#7 Who should write test scripts?

Test Automation teams typically have a different profile to manual teams. When planning your test automation program, considerations must be given to the composition of the automation team. A strong team will typically contain a balance of the following profiles:

- Technical people whose role will be to overcome technical issues
- Python developers whose role will be to develop libraries and utilities that will ensure tests are scripted in the best possible way
- Testers with strong domain knowledge who have progressed to become developers, whose role will be to provide the team with test scripts that will benefit from their test and validation experience

#8 Identify unique on-screen elements for reliable navigation

Successfully automating tests relies on being able to reliably navigate through the menu structure of the device under test. There are a number of different methods that can be used to establish which screen or menu a device is on, including image comparison, color analysis and optical character recognition (OCR).

The careful choice of a unique and constant element on each screen helps to ensure that the navigation process can occur in a reliable way.

"While manual scripts are easy to read and understand, they may not be easily automated."

Test Result Management

#9 Divide tests down into a number of blocks,

For those unfamiliar with test automation, there is a surprising amount of tester effort required, mainly due to the significant increase in test results and investigation. While manual testing requires a small number of people to look into the results generated, automated testing requires more effort to be put into the investigation of results because there are so many more results being generated.

ADV has seen that up to 20% of test cycle effort is spent investigating failed tests and then checking that the defects can be reproduced.

In order to understanding of the purpose of a test script and facilitate the identification of failures in the test results files, it is recommended that testers divide tests down into a number of blocks, i.e. do not test too much functionality in a single step.

#10 30-second rule for test analysis

Since manual investigation of results is one of the elements that impacts on test cycles the most, having easy to read tests results will lead to time, effort and cost savings. The goal should be to take no more than 30 seconds to understand the scope of any test script. This golden rule may obviously not apply to some particularly sensitive tests.

The more tests that are run, the more results you get. More results means more failures as well as more defects being identified and fixed, thus leading to an overall improvement of build reliability.

Logging features that enable the rapid identification of the point of failure in the test, is a critical requirement.

"When dealing with automated testing, there is a surprising amount of manual effort that needs to be spent in doing the analysis and investigation of the results generated"

Conclusion

This white paper has outlined a set of recommendations which we believe should be applied when transitioning from manual to automated testing. These recommendations are based on direct feedback from our customers who use Accenture StormTest test automation platform.

Recommendations apply to the architecture, development and also to the way test results may be handled and interpreted.

Through careful application of automation, very valid results may be reached, leading to a major reduction of the effort spent on DUT testing and validation.

About Accenture Digital Video

Accenture Digital Video is a business unit within Accenture. We deliver business results for companies where video is of strategic importance, helping them pivot to capture new growth opportunities in an ever changing market. Partnering with clients, we use our agile methodologies, deep skills, and open technology platforms and apply them in every phase of a change journey—from thinking to doing. The end result: more predictability in the face of a complex and volatile landscape. Accenture Digital Video has a 20 year track record in driving video innovation through a global workforce of more than 2,000 dedicated professionals across strategy, delivery, business services and operations, all dedicated to helping clients grow profitably.

About Accenture

Accenture is a leading global professional services company, providing a broad range of services and solutions in strategy, consulting, digital, technology and operations. Combining unmatched experience and specialized skills across more than 40 industries and all business functions—underpinned by the world's largest delivery network—Accenture works at the intersection of business and technology to help clients improve their performance and create sustainable value for their stakeholders. With more than 375,000 people serving clients in more than 120 countries, Accenture drives innovation to improve the way the world works and lives. Visit us at www.accenture.com.

16-3019