



Test at the Speed of Business White Paper

Introducing Axe – Code generation is the “Next Generation” for automated testing

Introduction

This white paper introduces Axe from Odin Technology and the code generation approach to test automation it employs which can truly be described as “Next Generation”. The paper is intended to illustrate the differences between the code generation approach against traditional approaches to automation, highlighting the benefits and illustrating where traditional methods fail to meet expectations.

Software test automation has been a good example of where in principle the promises and expectations rarely meet the reality. Much of this is in the expectation that simply buying software tools will solve a problem, we all know that is not true but there are further dimensions which need to be considered for success to follow. One glaring omission here is that the skills required for successful test automation are actually software engineering and development related; at Odin we believe that this is the single major factor in failure of test automation projects.

On numerous occasions we have met organisations who have invested major sums in test automation products which simply do not give the expected ROI and have often become “shelf-ware”. In many cases the problem has been compounded further by talented technicians starting to develop additional bespoke software around the tools in order to “un-shelf” them, often these are “keyword frameworks” or similar. This leads to the additional burden of development and maintenance of bespoke toolsets in the hope of achieving some return on investment.

At Odin technology we have taken a step back from the technical sophistication required of automation tools and looked at how a highly productive and tester-centric environment can be created. One where testers can develop and automate more sophisticated test cases using the power of the automation tools but without the requirement to acquire proprietary technical skills in the process. This is achieved by using a design environment they are all conversant with, namely Excel. Odin has taken this further by not only combining best practice in testing and test automation but also code generation and software engineering principles to create an unrivalled platform for harnessing the power of automation.

In the following sections of this white paper we will attempt to illustrate how this has been achieved and how it can benefit all automated testing projects irrespective of underlying toolsets.

Why could Axe be considered “Next generation”?

The answer is almost apparent in the question! Axe is unique in being a code generation platform for automated testing. Axe enables the design of test scenarios in a simple Excel spread sheet layout which can be processed by Axe to create both an automated test script (in the required execution tool language) and the related manual documentation for the scenario.

This “code and document generation” approach differentiates Axe from most approaches to automation. The manual documentation is synchronised with the automation script 100% of the time, every time. This therefore avoids the first level of potential defect entry in automation, there is no time

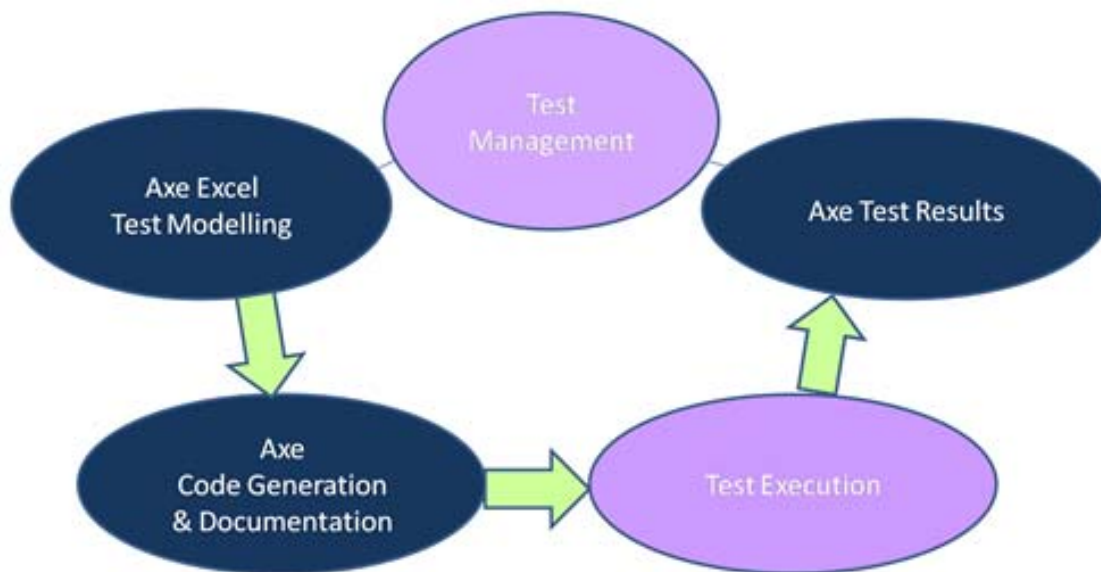


Test at the Speed of Business White Paper

delay or manual intervention between documenting the detailed steps of a scenario and producing the automated script code.

Furthermore Axe is cognizant of the fact that the test automation world is a complex one and that there are numerous tools performing different tasks that need to be integrated. In the “Axe world” nearly all assets are stored in portal data formats, primarily XML, which allows smooth integration not only with all leading commercial and open source execution tools but also with test management tools which provide the control room for testing effort. But here is the “twist”, Axe is unique in its ability to integrate multiple vendors tools together, thus solving the problem most organisations have of the fact that tools are often purchased at different times to meet different business needs.

As such Axe provides a holistic approach to automation which can truly be classed as “Next Generation” the diagram below illustrates this:



An explanation of this is as follows:

- Using Excel as a means to define test scenarios, Axe unleashes the capabilities of Excel combined with the tester’s knowledge to quickly create test scenarios. These scenarios can be easily replicated and changed to increase the coverage of test cases against an application under test. Axe leverages the functionality of Excel by being able to use built-in features (such as date handling and manipulation) to create inputs for the Axe code and document generator.
- Axe employs a small action set and simple spread sheet layout which can be easily learnt with one day of practical exercise based training. This consists of five core actions: Set, Get, Val, Load & Save, but can be easily extended to create new actions to enhance the function of the generated scripts or overcome automation difficulties.
- Once test scenarios have been defined, Axe can generate documentation, object use statistics (coverage) and the coded automation scripts required. For leading test management and version control tools, Axe can create test items through the tool APIs and upload generated code.



Test at the Speed of Business White Paper

- Scripts can then be executed against the AUT either controlled directly from Axe, the test management tool or a scheduler. Alongside execution tool logs, Axe produces a complementary “English” log of what happened on a step by step basis.
- Finally following the completed execution of the required test scenarios, Axe can upload results to the test management tool to complete the whole automation process.

How does the “Axe” approach differ from a “Traditional” approach?

There are a number of dimensions to answering this, the most obvious relate to productivity, on-going maintenance, the skills required and the resulting ROI.

In terms of productivity, the benefits of the Axe code generation approach are immense. In our case studies and experience, the benefits of Axe in terms of productivity over traditional approaches are at least a factor of four and often many more times. This is even more impressive from a maintenance perspective in that Axe has been proven to improve maintenance by a factor of five.

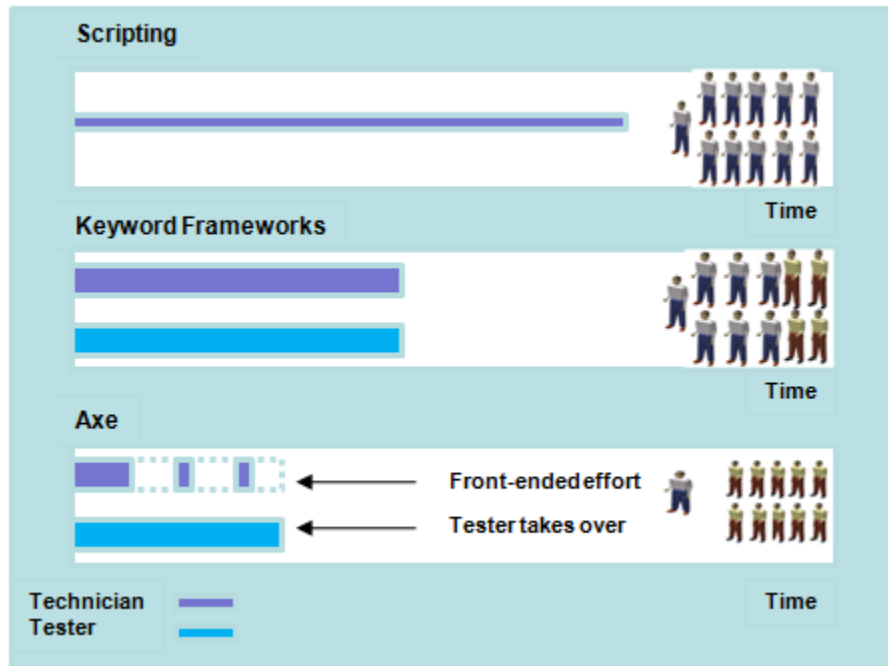
Axe is used in complex automation projects where the test scenarios required to carry out an adequate regression test is in the thousands rather than traditional automation approaches which usually fail to scale over a few hundred. The largest implementation of Axe currently has over 22,000 test scenarios defined and is still growing!

From a skills perspective Axe wins “hands down”! Axe was developed with the specific purpose of allowing testers to automate test scenarios, and as such it uses the tools a tester would use (such as Excel – 96% of testers use this daily) therefore avoiding testers being exposed to the proprietary interface and programming languages of automation tools. Testers are not required to become technicians (most testers do not have or want to learn programming skills). Where technical input is required, at project inception or to solve automation interaction problems, an Axe technician with the appropriate programming skills fills the gap.

The diagram below shows the change in skill base from traditional approaches to automation, through the keyword framework evolution to the Axe “Code Generation” approach. In the “Axe world”, ten testers can be supported by a single technician and on an “as required basis”, thus enabling them to service a much wider automation effort.



Test at the Speed of Business White Paper



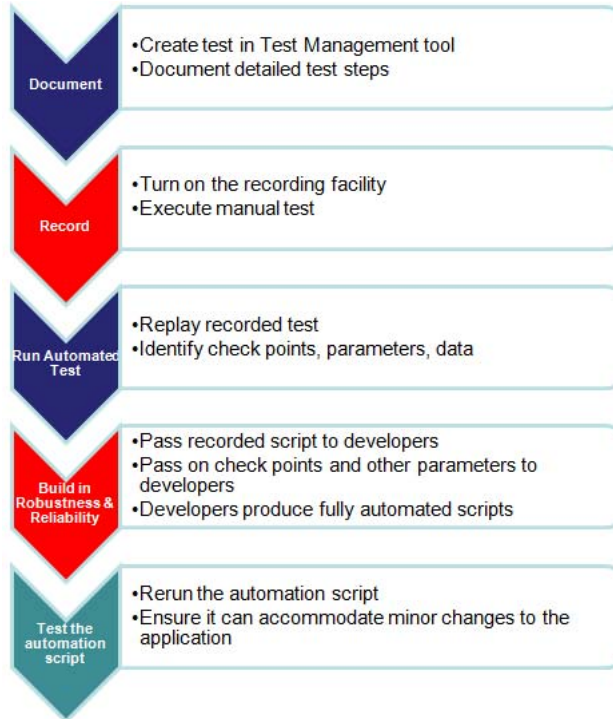
From a process perspective there are also significant differences, the Axe approach is simpler in the number of steps to the process are lower, and from a qualitative perspective Axe produces more robust, higher quality outputs than the traditional approach. Added to this is the fact that, unlike traditional and keyword framework approaches, Axe gives much more flexibility in terms of supported execution and test management tools therefore “future proofing” the deliverables against tool changes. For example if the underlying execution tool being used is changed, migration can be easily managed with minimum changes required. This is because the test scenario definitions are abstracted, stored in Excel spread sheets with Axe generating the documentation and coded scripts automatically. Changing the tool simply means changing the code language being generated from the scenarios and any supporting assets.

The diagram below shows how the Axe automation approach is a shorter process and thus less prone to introducing defects into the test scenarios:

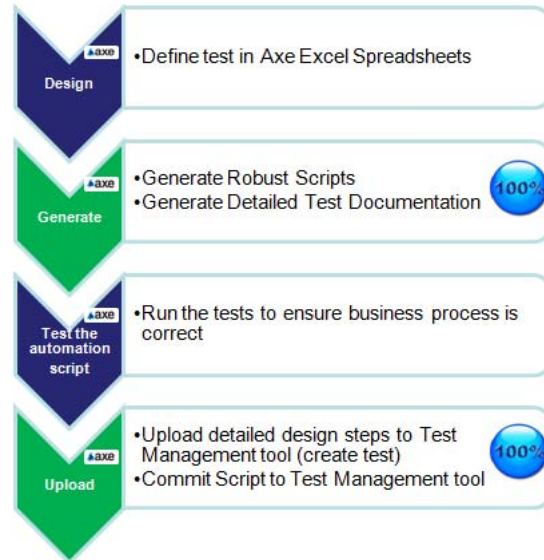


Test at the Speed of Business White Paper

Traditional Approach



Axe Approach



What are the Business Benefits of Axe?





From a business perspective these can be easily be defined as follows:

- Improved Team Collaboration** – A traditional coded/scripted approach relies on technical expertise throughout the project lifecycle to hand craft tests in the code like scripting languages used by the execution tools. The emergence of “Keyword Frameworks” split the work between Testers and Technicians. A Technician is still required to implement keywords in code modules with Testers defining the sequence of modules and the required data for tests. Axe however, empowers Testers to define all elements of a test scenario, business workflow and data. A technician is only required to setup and support a project thus lowering costs.
- Reduced Cost of Implementation** – Axe significantly reduces the cost of implementation giving a much quicker ROI. By reducing the level of technical expertise required and by using more readily available and often lower cost testing staff, the cost is immediately reduced. Furthermore with a four-fold or above productivity advantage and a five-fold saving on maintenance, Axe clearly gives a large cost advantage and far quicker ROI.
- Scalable** – Keyword frameworks and the traditional approaches to automation rarely scale beyond hundreds of test scenarios. The business requirement is often thousands of tests to give a reassuring level of coverage. Scripting is effectively software development prone to all the issues that presents. Axe has regularly been used on projects where thousands of test scenarios need to be automated. By removing the need to write code like scripting languages, the issue of



Test at the Speed of Business White Paper

scalability becomes a simple matter of organising spread sheets and re-using components to quickly deliver new scenarios.

-  **Proven ROI** – Axe leverages a company's current investment in automation tools by cutting the cost of implementation and maintenance making automation affordable and more predictable than via traditional approaches. In addition, Axe can be used with open-source automation tools in many cases further reducing costs and increasing the ROI.
-  **Easy to use** – Because Axe has been developed for testers to use, this means automation across the application lifecycle for non-technical staff becomes a reality. The reliance on expensive technical resource is thus removed and the application knowledge of the testers is captured for re-use in the test assets ensuring much higher quality automation as a result.
-  **Automate throughout the Application Lifecycle and beyond!** – Axe has been used not only in traditional “Waterfall” environments but extensively in Agile projects too. It can be used at all testing phases of the lifecycle and more importantly across both User Interface (UI) and non-UI components delivering true integration testing. All of this is achieved using the same simple Excel based definition of scenarios irrespective of whether the application component is UI or non-UI.
-  **Future-Proofing** – Axe works with all leading proprietary automation tools as well as a number of open-source execution tools. This gives a business more flexibility on selection of vendors and the use of offshore partners. In the case of off-shoring, as the assets are held in a readily understandable and documented Excel structure, these can be returned to the client and off-shored elsewhere to maintain or increase the business benefit of outsourcing as a business tool. The business is no longer constrained by technical skills or the choice of execution tool.

Conclusion

We have attempted in this white paper to convey how Axe in its “Next Generation” guise rewrites the rules of automation. Businesses need not now worry about choosing the right tool for automation but can now move forward on the principle that automation has massive business benefit that can now be achieved in a much simpler fashion than tradition would leave us to believe.

Furthermore Axe although “Next Generation” is not unproven with a large number of blue chip clients, some of which are shown below.



If you have any questions about this white paper please contact the author at david.tracey@odintech.com