Hybrid Keyword Data Driven Automation Frameworks
Test Automation as a Service (TaaS)
Real world case study

What value did this case study deliver?
- Complete “Automation Lifecycle Management” (ALM) solution;
- Providing “Automated Testing as a Service” in the Cloud;
- “Pay As You Go” flexible billing (no expensive tool licensing costs);
- Distributed continuous build integration with “Application Under Test”;
- Shorter market deployment times through quicker Automated Testing;
- Shorter environment deployment times through Automated Health Check;
- Better quality testing freeing specialist test resource to concentrate on non-repetitive test tasks;
- Easy interface for non-technical testers to create their own “user stories”;
- Capability Maturity Model Integration (CMMI) through Business Process Modelling (BPM) comprising of Business Level Keywords (BLK);
- Introduction of “Full” Agile automation testing and “First” day automation.

What approach?
- Manual Approach;
- Traditional Automated Approach:
  - Vendor Tool (Record & Playback) Limited requirements Limited output

Hybrid Approach

Summary

<table>
<thead>
<tr>
<th>Manual</th>
<th>Traditional (Vendor Tool)</th>
<th>Hybrid (NET Development)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time</td>
<td>Vendor Tool</td>
<td>Hybrid Tool (NET)</td>
</tr>
<tr>
<td>Cost</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Effort</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Risk</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Framework Approach

- Hybrid utilising the best technologies and resources
- Business level keywords represent user stories
- Dynamic use of sanitised business data
- Modularity: modularity of test procedures to provide reliable processing of the framework components

Vendor Tool (Ranorex IDE)

Traditional Tools Modularity Data Keyword BPT BPM Approach

- Limited Support
- Full Support

Single Technology Stack (SPOF)
- Tool Vendor Dependency
- C# Project incompatible with Visual Studio
- Technology Blocks (Custom Flex Objects/Classes)
- Limited non-technical tester interface

Vendor Support (maintenance subscription)
- Object Repository (object-based Record / Replay)
- Constraints Advantages

Hybrid (NET Development)

- In-house framework development & maintenance effort.
- Additional skills / knowledge resources
- Azure Platform Subscription
- Object Repository (limited Record / Replay support)
- Constraints Advantages


Integration with development (Visual Studio 2010)
Integration with deployment (TeamCity v5.1)
Integration with testing (Quality Center v11)

Return on Investment

<table>
<thead>
<tr>
<th>Number of Executions</th>
<th>Investment in Effort</th>
<th>Return on Investment</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

MCT = 3
MCT + HC = 3.4
Traditional = 3.7

Leverage Hybrid (Framework)

\[
\text{Return on Investment} = \frac{\text{Revenue} - \text{Cost}}{\text{Cost}}
\]

\[
\text{Number of Executions} = \frac{\text{Revenue}}{\text{Cost}}
\]
Automation Goals

- Maintainable: To reduce the amount of test maintenance effort
- Reusable: Modularity of test procedures & functions
- Manageable: Effective test design, execution and traceability
- Robust: Providing object/event/error handling & recovery
- Flexible: Independent of test environment and be scalable
- Measurable: Customisable reporting & test execution progress

Business Process Testing (Process)

Automation Lifecycle Management (Process)

Business Level Keywords (Testing)

Technology Stack (TaaS)
Unified Test Reporting (UTR)

Unified Test Reporting – Real-time Log

Process Improvement

- Agile Planning – Team Foundation Server linked to Task Board (TFS v2);
- Coding Standards – Code reviews (internal/external to team), coding tools, unit test standards, agile hints, logging etc etc;
- Testing – Internal peer reviews, unit tests developed, exploratory testing for each story as a task, regression test story in each sprint;
- Wiki – Knowledge capture including online framework documentation (screenshot);
- Technical Notes Debt – on project wiki (intranet SharePoint).

Key Achievements

- Centralised - management of test assets to provide open connectivity as a generic source of test scenario data;
- Dynamic - generation of sanitised input test data validated against business rules along with an object repository based on a combination of descriptive programming containing XPATH regular expressions;
- Legacy - support for static data types plus full/partial fall back support for manual test execution;
- Unified reporting - using funnel virtualisation techniques to analyse the quality of builds, assisted by diagnostic data adapters;

SUMMARY

- Hybrid – utilising the best technologies and resources to do the job;
- Keyword – creating simple and robust tests scenarios written in business level keywords combined with natural language;
- Data – effective use of dynamic business data to provide an input source;
- Driven – processes the test component actions, objects and events seamlessly;
- Automation – that is collaborative, distributed and scalable;
- Frameworks – independent of application or environment under test.