The Four T’s of Test Automation

Presented by:

David Dang
Questcon Technologies

Presented at:
STAREAST Software Testing Analysis & Review
May 5-9, 2008; Orlando, FL, USA
David Dang

David Dang joined Questcon Technologies in 1999. David is a Mercury Interactive Certified Instructor (CI) for QuickTest Professional, WinRunner, and QualityCenter. As director of automation practices, David works with clients to assess, define, and implement automation strategies to maximize the clients’ returns on investment for automation tools and minimize automated script maintenance. David is the author of the QuestAssured® Test Automation Methodology and the lead test automation instructor at the Questcon headquarters in Greensboro. David has been a featured speaker on test automation and related topics at local and national QA and Testing conferences including Quality Assurance Institute (QAI), StarEast, STP Con and PSQT.

David received his B.S. in Management Information Systems and B.A. in Psychology from the University of Buffalo.
The 4 T’s of Test Automation:
Your Planning Guide to Success

Presented By:
David Dang
Questcon Technologies

May 9, 2008
Let’s review the typical test automation process . . .

Test Automation: Saves time and Increases Coverage

Tool Evaluation

Training

Team starts using the tool
Typical Automation Project Outcome

Adopted by QA with success at the beginning

Over subsequent release cycles, maintenance time and costs increase

Final outcome, the automation suite becomes a maintenance nightmare and is used by a few users or becomes shelfware
The primary objective of this presentation is to demonstrate how proper planning for test automation will yield greater success. Attendees will learn:

- Potential benefits of a successful automation project
- Key factors to consider in the planning phase of test automation
- Proper alignment of automation tools to technology under test
- Evaluation criteria for manual test cases
- Required skill sets for test automation
- Timeline impacts of using different automation approaches
While there are multiple considerations that must be addressed during the planning phase of a test automation project, these four factors have the largest impact on the overall success.
Create an inventory of the current technology that you must test:
- Platforms
- Programming Languages
- Databases
- Third-party controls

Determine what technology changes may occur within two years.

Examples of technology under test (TUT):
- Platform – Mainframe, Client/Server, and .Net (long-term)
- Programming Language – Cobol II, VB 6, C++, and C# (long-term)
- Database – DB2, Sybase, and MySQL (long-term)
- Third-party controls – VSFlexGrid and WebServices (long-term)
Considerations

- Level of support for TUT
- Amount of customizations required for non-supported TUT
- Ease of use
- Skill sets required to work with tool
- Common languages
- Tool cost
- Tool maintenance cost
- Integration with other tools
- Reporting

Open Source Tools
- FitNesse
- Selenium
- Ruby
- JUnit

Packaged Tools
- QuickTest Pro
- Functional Tester
- TestPartner
- SilkTest
- MS TeamTest
Matching TUT to the automation tool ensures faster automation script development and decreased maintenance time by:

- Reduction in custom coding
- Reduction in object mapping
- Increased flexibility in framework design
- Easier debugging
- Better reporting

Additionally, this approach reduces learning curve of automation tool

- Increases the level of testing
- Provides better cost justification
### Test Case

<table>
<thead>
<tr>
<th>Test Case Identifier:</th>
<th>Test case 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test Case Objective:</td>
<td>Make sure create order works</td>
</tr>
<tr>
<td>Pre-requisite:</td>
<td>None</td>
</tr>
<tr>
<td>Priority: 1, 2, 3, 4, or 5</td>
<td>1</td>
</tr>
<tr>
<td>Input Variable:</td>
<td>Use existing data</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step #</th>
<th>Description</th>
<th>Expected Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>STEP1</td>
<td>Input new order</td>
<td>Order is created</td>
</tr>
</tbody>
</table>

Is this a good test case?

Why?
Test cases contain detailed levels of information:
- Test case name
- Test case objective
- Pre-requisites
- Test case priority
- Input and output data
- Step name
- Step description
- Expected result

Determine:
- Frequency of execution over a yearly period
- Level of testing required for validation
- Complexity of the test cases
- Manual only steps, e.g. loading a backup tape
- Automation priority based on risk factors
<table>
<thead>
<tr>
<th>Step #</th>
<th>Description</th>
<th>Expected Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>STEP1</td>
<td>From Order application main screen, click on New Order icon</td>
<td>New order window appears</td>
</tr>
<tr>
<td>STEP2</td>
<td>Execute query:</td>
<td>One customer number is return</td>
</tr>
<tr>
<td></td>
<td>Select top 1 cust_num</td>
<td></td>
</tr>
<tr>
<td></td>
<td>From db_customer</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Where cust_active = “Y”</td>
<td></td>
</tr>
<tr>
<td>STEP3</td>
<td>Input customer number into Customer Number field</td>
<td>Customer Number field is populated</td>
</tr>
<tr>
<td>STEP4</td>
<td>Input today date into Order Date field.</td>
<td>Order Date field is populated</td>
</tr>
<tr>
<td></td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>Validation</td>
<td>Verify message on status bar</td>
<td>Message “Your order was successfully created. Your order number is 999999.”</td>
</tr>
</tbody>
</table>

**Validation**

<table>
<thead>
<tr>
<th>Validation</th>
<th>Description</th>
<th>Expected Results</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Verify order is created in database. Execute query:</td>
<td>Compare inputted values (customer number, date, materials, quantity, shipping address) with values return from query. All the values should match.</td>
</tr>
<tr>
<td></td>
<td>Select *</td>
<td></td>
</tr>
<tr>
<td></td>
<td>From db_order</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Where Ord_Num = 999999</td>
<td></td>
</tr>
</tbody>
</table>
Benefits

- Scope identification of the automation effort
- Prioritization of automation effort based on risk factors
- Elimination of manual test cases that were out-dated
- Identification of manual test cases that provide little to no return on investment
- Understanding of test case factors in creating an automation framework
Required Skill Sets

- QA experience in various levels of testing such as smoke, regression, and functional
- Solid understanding of business flows, business rules, and data variations in the application
- Experience with multiple data strategies
  - Creating “new” data
  - Gathering existing data
  - Building data
- Experience with test automation concepts
- Understanding of computer programming
- Grasp of application architecture
- Comprehension of Object-oriented programming (OOP)
Optional Skill Sets

- Experience in programming using Java, VB, or C languages
- Experience with multiple automation approaches
- Experience in SQL
- Experience in writing classes and methods
- Experience in writing functions and sub-procedures
- Experience in using or interacting with window APIs

Project Management
- Timelines
- Deliverables
- Resources
- Metrics
- Budget
Establishes a good baseline for skill sets required to implement test automation

Can be used to determine QA members that will work on test automation
- Most companies struggle in test automation when they have the QA group perform test automation
- Recommend creating an automation team

Reduction in trial and error approach

Automation approach designed around skill set level available
Automation Projects Typically Mirror SDLC

- The same steps that occur in the SDLC should occur in an automation project.
- Most automation projects only account for Implementation and Test.
- Timeline should take every step in the SDLC into consideration.
Factors that Impact Timeline

- Resource availability
  - Most companies assign 30 - 50% of QA member time to test automation
  - The reality is QA members cannot effectively perform both manual and automation testing
  - This greatly increases the timeline for creating and maintaining test automation suite

- Resource skill sets

- Selected automation tool

- Testing environment

- Data strategy

- Manual test case condition
Factors that Impact Timeline

- Automation framework
- There are five common automation frameworks
  - Record/Playback
  - Data-driven
  - Modular
  - Action-based/Keyword
  - Database
- Automation frameworks are not mutually exclusive
- Automation frameworks can be combined, e.g. Action-based and database approaches
- Appropriate automation framework dependant on many factors including the company goal for test automation
Timeline

Case Study

Graph showing trends in different areas:
- Development
- Maintenance
- Maintenance/Year @ 10 Releases/Year

Areas listed on the x-axis:
- Record/Playback
- Data-Driven
- Modular
- Action-based/Keyword
- Database

Y-axis values range from 0 to 1800.

Graph indicates changes over time in the mentioned areas.
Benefits

- Sets clear expectations for management
- Identified timeline for automation development life cycle (ADLC)
- Maintenance time factored into the automation framework design
- Clear understanding of milestones and deliverable
- Easier measurement of the cost and benefit of test automation
- ROI calculation for break-even point for test automation
The most important aspect of any test automation project is planning. With the right strategy, you will save time and money, and set clear expectations for your organization.

During planning, remember the 4 Ts:

- Technology
- Test Cases
- Talent
- Timeline