Automated Software Testing for DoD: Challenges and Solutions

Learn about the specific DOD automated testing challenges and how IDT's Automated Test and Retest (ATRT) capabilities solves these challenges, along with a revolutionary approach to automated test cases update and maintenance.

Elfriede Dustin, IDT
edustin@idtus.com
http://amazon.com/author/elfriededustin
Agenda

• About IDT
• DOD requirements of an automated testing solution
• How does Automated Test and Re-Test (ATRT) meet these specific requirements
• Demo
• Q&A
About IDT

Information technology business headquartered in Arlington, Virginia

Our primary objective is to enable new software capabilities to be deployed with higher quality, faster and more affordably.

The solutions and services we provide:

• DOD approved integrated automated testing solution – Automated Test and Re-test (ATRT)
• Automated testing strategies
• Help improve your test program
• Training
• Certification
• Summit?
DOD REQUIREMENTS OF AN AUTOMATED TESTING SOLUTION
DoD requirements of an automated testing solution

– Can’t be intrusive to system under test (SUT)
– Needs to be OS and Platform independent
– Needs to be GUI technology independent
– Needs to be able to handle a multi-computer, systems of systems environment
– Needs to be able to handle display and non-display centric automation
DOD requirements of an automated testing solution (cont)

- Non-developers should be able to use the tool
- Automated test case maintenance should be low
- Zero license cost
- Be able to handle large amounts of data analysis
- Traceability Matrix and Reporting
- Needs to be extensible
- Many more
DOD Tool Requirements

• Can’t be intrusive to system under test (SUT)
  – Customer wanted a tool that does not affect the SUT configuration in any way (required to maintain pristine SUT configuration)
  – Tool would get blamed for any issues if installed on SUT during testing/uninstalled after delivery
  – We had to cross off our list tools such as IBM Rational Functional Tester, Hewlett-Packard’s Mercury QuickTest Pro or SmartBear’s TestComplete.
DOD Tool Requirements (cont)

Needs to be OS and Platform independent

– Customer wanted a tool that would be compatible with any OS or platform imaginable
  
  • e.g. SUTs run on any Linux variant (Centos, etc.), Windows platform (Win7, XP, etc.), Solaris, Mac, IPad, laptop, desktop, system of systems, etc.

– While there are tools available that cover some of the OSs listed here or some of the platforms, again we had to cross off our list tools such as IBM Rational Functional Tester, Hewlett-Packard’s Mercury QuickTest Pro or SmartBear’s TestComplete
DOD Tool Requirements (cont)

Needs to be GUI technology independent

- Our tool solution should be able to handle any type of GUI technologies written in any type of language, i.e. Motif, C#, and so on, and to handle any type of third-party non-custom GUI control.

- Many of the current vendor-provided automated software testing tools are GUI technology dependent. That means that if proprietary programming languages or third-party controls are used in the SUT GUI, the automated testing tool often is not compatible, which presents automated testing problems (e.g. problems with RFT and Stingray grids, etc.).
DOD Tool Requirements (cont)

Needs to be able to handle a multi-computer, systems of systems environment

– SUTs consist of networked computers (i.e., multiple servers, systems of systems, multiple monitors and displays interconnected to form one SUT).
DOD Tool Requirements (cont)

Requirements:
• Can’t be intrusive to system under test
• Needs to be OS independent
• Needs to be GUI technology independent
• Needs to be able to handle a multi-computer, systems of systems environment

Our Solution to these requirements:
– A tool that uses Remote Desktop Protocol (RDP) and Virtual Network Computing (VNC) to connect to the SUT

• VNC is a graphical desktop sharing system that uses the RFB protocol to remotely control another computer. It transmits the keyboard and mouse events from one computer to another, relaying the graphical screen updates back in the other direction, over a network
DOD Tool Requirements (cont)

Requirements:
• Can’t be intrusive to system under test
• Needs to be OS independent
• Needs to be GUI technology independent
• Needs to be able to handle a multi-computer, systems of systems environment

Our Solution to these requirements (cont):
Since various VNC versions exist for most OSs, we were able to meet the “OS independent” requirement through our VNC solution.
Via VNC we can interact with all GUI elements of the SUT as images, independent of the GUI technology used.
DOD Tool Requirements (cont)

Requirements:
• Can’t be intrusive to system under test
• Needs to be OS independent
• Needs to be GUI technology independent
• Needs to be able to handle a multi-computer, systems of systems environment

Our Solution to these requirements (cont):

• Our solution can handle distributed and concurrent testing over a network: Automated tests can be executed concurrently over a network for the test case where various GUI- or message-based outputs are dependent on each other over a network or have to run in parallel.
• As long as a VNC server is installed on any of the networked computers, ATRT (which now also serves as a VNC client) can connect to them.
DOD Tool Requirements (cont)

Needs to be able to handle display and non-display centric automation

– Customer wanted a tool that could test the SUT via the GUI interface (to reproduce and automate operator actions), plus be able to test the various backend (non-GUI) interfaces using various protocols, such as messages being sent after the GUI buttons had been pushed, etc.
Needs to be able to handle display and non-display centric automation

Our Solution:

Each SUT used a different protocol (TCP/IP, User Datagram Protocol [UDP], Common Object Request Broker Architecture [CORBA], proprietary, and more), and all used different message data formats. We developed an approach whereby all protocols and data could be tested via our ATRT framework.
DOD Tool Requirements (cont)

Non-developers should be able to use the tool:

• Testers often are subject matter experts but not necessarily software developers who could use an automated testing tool efficiently.

• Non-developers generally don’t want to be bothered with developing automated testing scripts; they want to be able to use the tool with the simple click of a button.

• Organizations don’t want their expensive developer staff spending time developing automated test cases when they can develop features

• If tool is not usable it can become shelfware
DOD Tool Requirements (cont)

Non-developers should be able to use the tool:

Our Solution

- Model based approach to developing automated testing.
  - Allows the testers to drag the action they want to take on a “canvas” to develop their automated tests in a test flow form

- no scripting is involved
  - the tester can click on an icon that matches any mouse feature (left click, right click, middle click, mouse down, etc.) to create the test action

Hint: The purpose of automation is to serve testing. This can best be done by tailoring the automation so that the testers find it as easy as possible to use.
Non-developers should be able to use the tool (cont)
Automated test case maintenance should be simple

Functions

– One of the biggest challenges of automated testing is test case maintenance

– Much time is spent maintaining the automated tests with each new system build or SUT release

– If maintenance is too time consuming automated tests will become outdated and shelfware
DOD Tool Requirements – simple Test Case maintenance (cont)
DOD Tool Requirements – simple Test Case maintenance (cont)

• For Automated Test Case maintenance:
  • We want the red reset button
    – Reset to previously working state
    – Reset and circumvent the error
    – Reset...
DOD Tool Requirements (cont)

*Automated test case maintenance should be simple*

Our Solution

– For automated GUI Tests we came up with an approach to near self-healing automated tests.

– Automated GUI Tests that can adapt to the many GUI changes with the simple click of a button to reduce maintenance efforts. A feature we call automated recapture
Zero License Cost

– Our customer wanted a tool solution with minimal license cost

*Our Solution:*

Develop this solution under an SBIR – no license cost to Government
Tool Requirements (cont)

Be able to handle large amounts of data analysis

– SUTs can produce millions of data outputs which makes it almost impossible for a human to evaluate all in an efficient manner

– With an automated data driven testing feature one test scenario can be reused over and over with different data values and data combinations allowing for increased data scenario coverage, resulting in analysis requirements of large amounts of data.
Tool Requirements (cont)

Be able to handle large amounts of data analysis
Our solution:
ATRT Analysis Manager which allows for:
• Data Management
• Event Reconstruction
• Requirements Verification
• Performance Analysis
Tool Requirements

Traceability Matrix and Reporting – Our Solution:

• ATRT provides various pre-execution test reports and post-execution test reports
• Reports can be exported into html; pdf; Word; or Excel
Tool Requirements

Tool needs to be extensible

- Customer wanted a tool that could integrate with their existing requirements management and defect tracking tools.
- Customer wanted to be able to extend the tool themselves, as needed
Tool Requirements (cont)

Tool needs to be extensible

Our Solution:

ATRT uses the open source Eclipse development environment. We chose the Eclipse rich client platform because it could simply be expanded on the base environment for ATRT to allow for configurability/extensibility via a powerful plug-in framework.

ATRT also provides open APIs to allow for ease of integration

*Hint: Use a flexible development environment so that you can tailor the tools and utilities to your specific customer and SUT testing needs.*
ATRT Technology Suite
What discriminates ATRT?

• Non-intrusive to System Under Test (SUT)
• Operating environment independent
• Scalable from components to System of Systems
• Developed to support system integration, test, and certification of mission critical systems:
  – All programs have reported at least 50% productivity gain
• Graphical User I/F for test automation, no scripting or software development required
• Provides an integrated solution for the automated execution of message based interface tests and operator tests with post test analysis
What is Different?

<table>
<thead>
<tr>
<th>Current Testing Approach</th>
<th>Testing with ATRT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Repetitive human labor intensive testing</td>
<td>Automated computer based testing</td>
</tr>
<tr>
<td>Time constrained</td>
<td>24/7/365 testing</td>
</tr>
<tr>
<td>Limited permutation testing</td>
<td>Unlimited permutation testing</td>
</tr>
<tr>
<td>Evaluation of test results days or months after test event</td>
<td>Test reports available at completion of test event</td>
</tr>
</tbody>
</table>
Demo

• Demo
Info

For more info, please contact me at edustin@idtus.com

To download an eval copy, please go here http://idtus.com/products/atrt-test-manager/