

Near Future of Automated Software Testing

Learn about the specific DOD automated testing challenges,
some of the solutions and areas for improvement in near
future

Presenter:

Elfriede Dustin, IDT, email: edustin@idtus.com



Intro

Presenter:

- Elfriede Dustin, IDT – edustin@idtus.com – see bio here
<http://amazon.com/author/elfriededustin>
- IDT (www.idtus.com) specializes in the design, development, and implementation of Automated Software Testing solutions
 - Patent(s) Pending: 13/605,182 ; 61/531,769 ; 61/577,298 ; 61,564,009



Agenda

- Top 3 Automated Software Testing Challenges
 - Challenges and Solutions
 - What should the near future of automated software testing look like
 - What are we still missing?
- Demo
- Q&A



TOP 3 AUTOMATED TESTING CHALLENGES



Top 3 Automated Testing Challenges

1. Understanding your testing problem before choosing a tool
2. Knowing which tools are available
3. Understanding your Tool Requirements



Automated Software Testing Challenge 1

UNDERSTANDING YOUR TESTING PROBLEM BEFORE CHOOSING A TOOL



Challenge: Understanding your testing problem before choosing a tool

First evaluate the testing problem we are trying to solve:

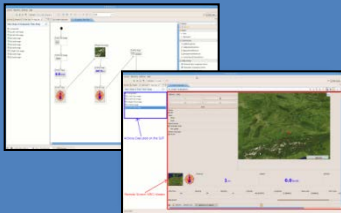
What is your Testing Problem?



Challenge: Understanding your testing problem before choosing a tool (cont)

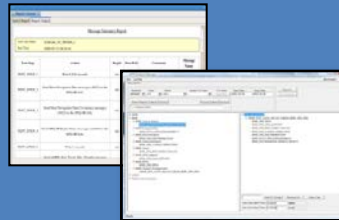
- Functional GUI testing
- Data Analysis
- Information Assurance
- Performance Testing
- Other?

Test Manager



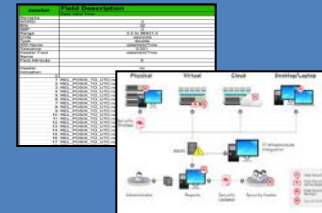
- Test Execution / Scenario Control
- Message Generation and Receive
- GUI Capture and Playback

Analysis Manager



- Event Reconstruction
- Requirements Verification
- Performance Analysis

Information Security Manager

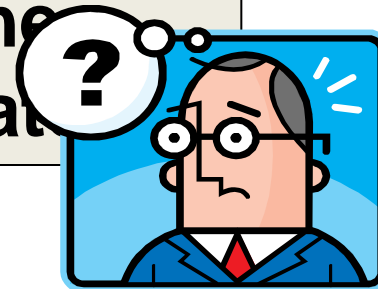


Challenge: Understanding your testing problem before choosing a tool

Once you understand the testing problem

**Develop your automated
testing strategy**

**For some the tool is the
automated testing strategy**



Automated Test Strategy

- Define the scope and objectives
- Assess tests for automatability (not all tests lend themselves to automation) – identify the test automation requirements
- Define automated test environment
- Identify automated test approach
- Know tools available and your tool requirements (see next challenges)
- Implement and track
- Reporting



Automated Software Testing Challenge 2

KNOWING WHAT TOOLS ARE AVAILABLE

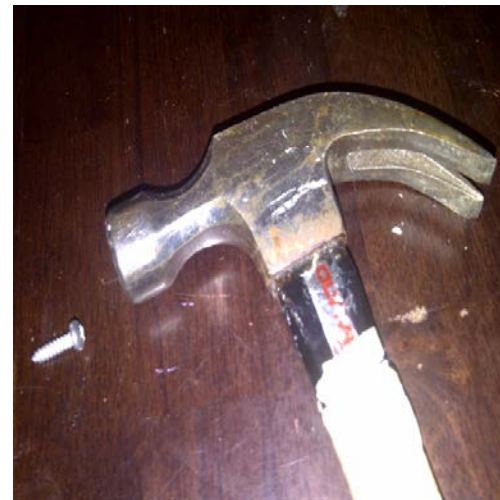


Challenge – Not knowing what tools are available

- Once you've defined the automated testing strategy – Know the tools available

**Using a hammer as
your only tool**

If the only tool you have is a hammer, you tend to see every problem as a nail



Challenge – Not knowing about tools available (cont)

- Choosing/Using a tool for the wrong reason
 - Based on market share
 - Based on somebody's opinion
 - that's the only tool they know
 - that's the tool we've always used



Automated Software Testing Challenge 3

UNDERSTANDING YOUR TOOL REQUIREMENTS



Understanding your Tool Requirements

- For example, here are our tool requirements to implement our test strategy:
 - 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
 - Needs to be DOD approved



Understanding your Tool Requirements

- For example, here are our tool requirements to address various automated testing challenges (cont):
 - Non-developers should be able to use the tool
 - Automated test case maintenance should be low
 - Low license fee
 - Be able to handle large amounts of data analysis



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)
 - Systems cannot be fielded if tested with a modified configuration
 - We had to cross off our list tools such as IBM Rational Functional Tester, Hewlett-Packard's Mercury QuickTest Pro or SmartBear's TestComplete.

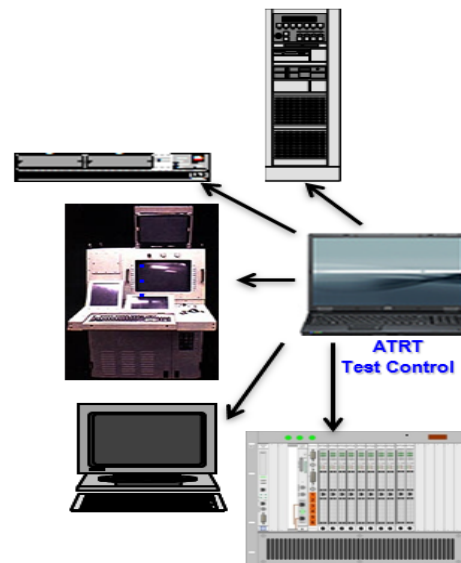


DOD Tool Requirements (cont)

- *Can't be intrusive to system under test*

ATRT's solution:

- ATRT uses Remote Desktop Protocol (RDP) or Virtual Network Computing (VNC) to connect to the SUT



DOD Tool Requirements (cont)

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)

- ***OS and Platform independent***

ATRT's solution:

Since various VNC or RDP versions exist for most OSs, we were able to meet the “OS independent” requirement



DOD Tool Requirements (cont)

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, i.e. 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 Stingray grids, etc.).



DOD Tool Requirements (cont)

- ***GUI technology independent***

ATRT's solution:

Via VNC and RDP we can interact with all GUI elements of the SUT as images, independent of the GUI technology used.



DOD Tool Requirements (cont)

Handle 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).

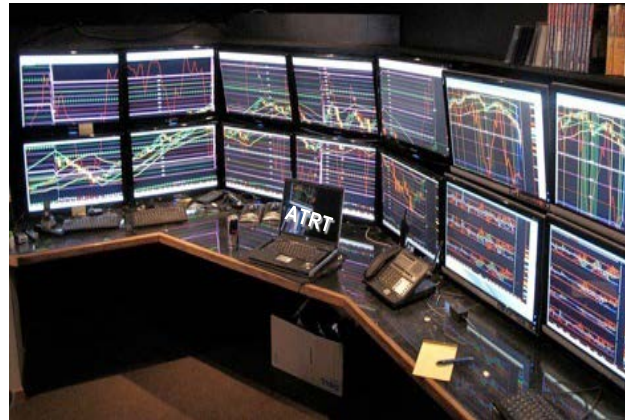


DOD Tool Requirements (cont)

- *Handle systems of systems environments*

ATRT's solution:

- ATRT 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.



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.
- While separate tools are available that can handle either GUI based automation or message based automation we couldn't find one that did both out of the box



DOD Tool Requirements (cont)

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

ATRT's solution:

The typical DOD systems we test use different protocols such as TCP/IP, User Datagram Protocol [UDP], Common Object Request Broker Architecture [CORBA], http, DDS, proprietary, and more, and all use different message data formats.

We developed an approach whereby all protocols and data can be tested via our ATRT message base



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

*ATRT is
"strong
enough for a
developer
but made for
a tester"*

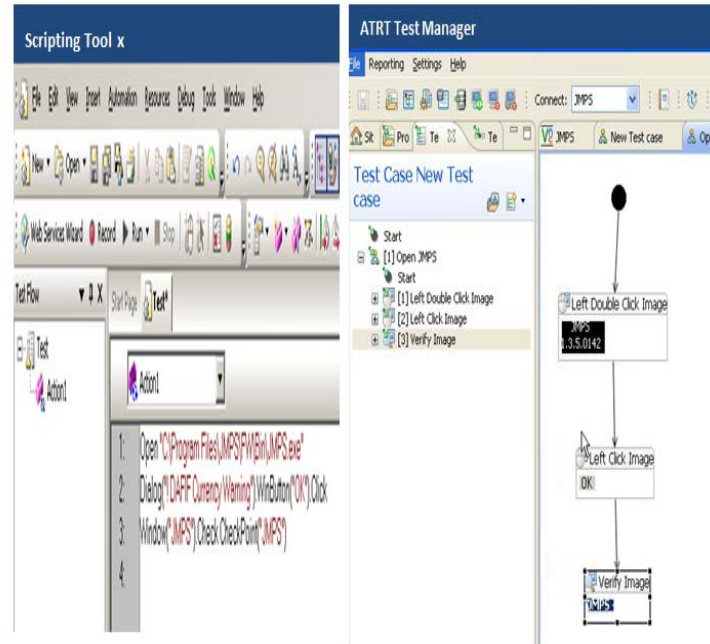


DOD Tool Requirements (cont)

- ***Non-developers should be able to use the tool:***

ATRT's 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



DOD Tool Requirements (cont)

Automated test case maintenance should be simple

- One of the biggest challenges of automated testing is test case maintenance
 - If maintenance is too time consuming automated tests will become outdated and shelveware
 - 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***

ATRT's 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



DOD Tool Requirements (cont)

Automated test case maintenance should be simple

ATRT's solution (cont):

Functions

- The most maintainable tests are the ones that are easily modifiable.
- ATRT allows you to create modular test cases with the use of functions, so if you have to make a changes to steps repeated in other test cases, you can do that in one place in the function.
- Functions simply allow you to reuse test actions in various test cases.
- ATRT's Function manager will tell you where the functions are being reused



Tool Requirements (cont)

Be able to handle large amounts of data analysis

- SUTs can produce millions of data outputs which makes it impossible for a human to evaluate all in an efficient or timely 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*

ATRT's solution:

ATRT Analysis Manager allows for:

- Data Management
- Event Reconstruction
- Requirements Verification
- Performance Analysis



Tool Requirements

Traceability Matrix and Reporting

- The tool needed to provide a requirements and test case traceability capability
- Needed to provide various reporting features



Tool Requirements

- ***Traceability Matrix and Reporting***

ATRT's solution:

- ATRT allows for requirements and test cases import
- Allows for traceability of various test artifacts
- Provides various pre-execution test reports and post-execution test reports, which 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*

ATRT's solution:

ATRT uses the 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



Demo

- Demo



ATRT

WHAT ARE WE MISSING?

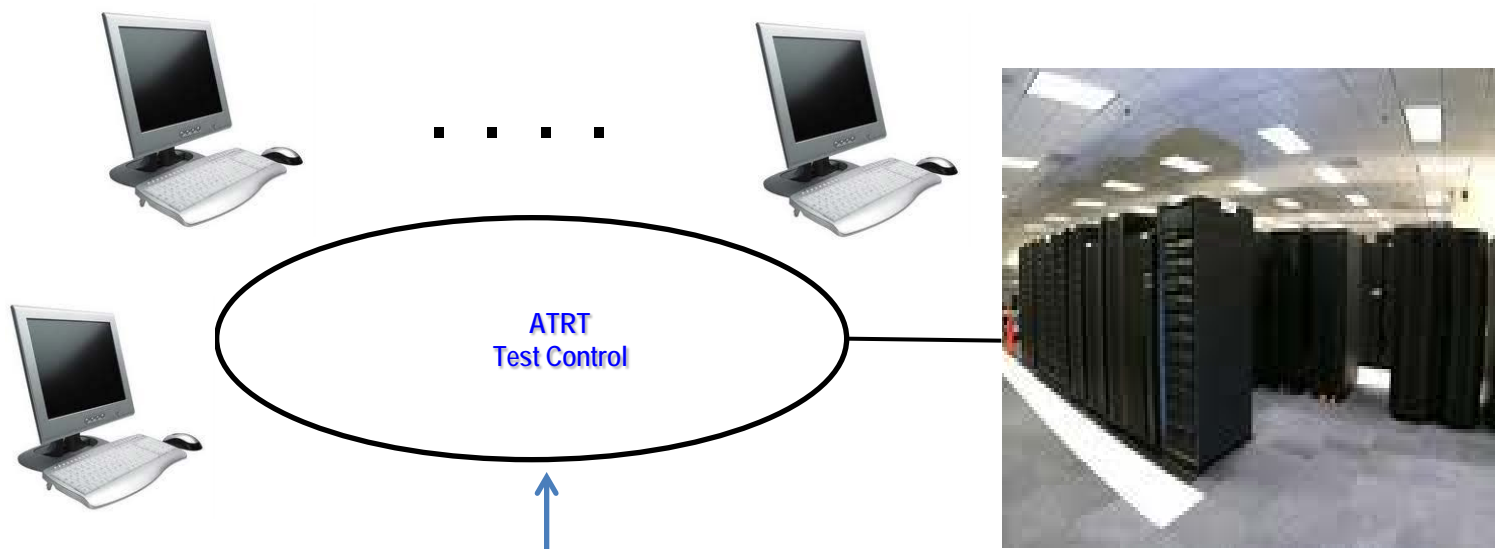


Near future

- Tool needs to be “smarter”
 - Self-healing – probes that ship with the software
 - Determines an area that’s buggy and help develop additional test cases
 - Can detect sounds/ashes
 - Image detection less sensitive
 - Know which areas of code changed only run those test cases applicable to that changed area at the system level




ATRT Technology Suite

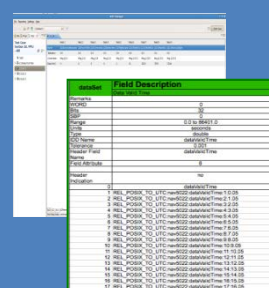


ATRT

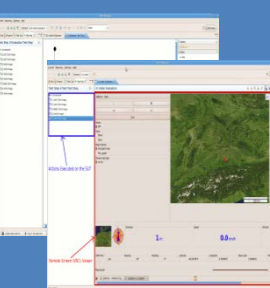
Test Manager



Scenario Control




Message Generation
And Receive

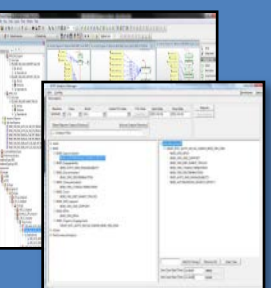


GUI Capture
and Playback


Analysis Manager



Reporting and
Traceability



Use Case Based
Analysis



Questions

?



Closing

- Today we've discussed and demo'ed few of the ATRT features
 - Many more features to be shown
 - Quarterly webinars
 - ATRT Summit September 2013 in Arlington, VA
- Contact us for more information or to request an evaluation copy
 - edustin@idtus.com
 - Phone: 7037253051
 - <http://idtus.com/contact/>

