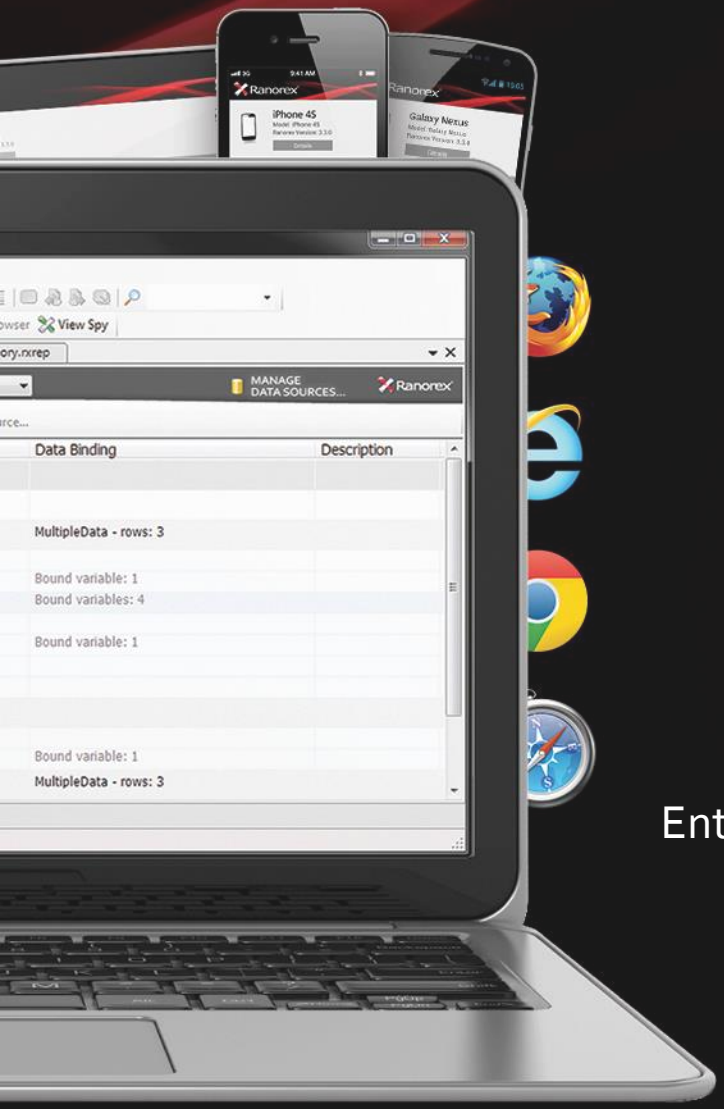


Why Test Automation Fails

in Theory and in Practice

Jim Trentadue
Enterprise Account Manager- Ranorex
jtrentadue@ranorex.com
Thursday, January 15, 2015

Agenda

Agenda

- Test Automation Industry recap
- Test Automation Industry outlook
- Adoption challenges with Test Automation; common myths and perceptions
- Why are Test Automation adoptions failing?
- Why are Test Automation implementations failing?
- Correcting the Test Automation approach, concepts and applications
- Session recap

Test Automation Industry recap

Test Automation Industry recap

Ranorex

- First Record \ Playback tools appeared about 20-25 years ago
- The primary objective was to test desktop applications. Some tools could test broken URL links, but not dig into the objects
- The tester was required to have scripting or programming knowledge to make the tests run effectively, even in Record \ Playback
- Below is an evolution of Test Automation Frameworks:

Record \ Playback → **Structured Testing (invokes more conditions)** → **Data Driven** → **Keyword Driven** → **Model / Object Based** → **Actions Based** → ***Hybrid*** **(combines 2 or more of the previous frameworks)**

Test Automation Industry outlook

Test Automation Industry outlook

- Number of test automation tools have increased to over 25-30, primarily focused on taking the scripting / programming out of the equation for the manual testers
- Technical complexities have increased considerably to cover the following:



- License models now offer a Test Execution only component, apart from the license to create and maintain the automated tests and object repository
- Tools have the capability to integrate with various Continuous Integration tools and Test Management solutions

Adoption of Test Automation; common myths and perceptions

Test Automation adoption challenges

Ranorex

- New challenges below have slowed the test automation productivity:
  - Rapid deployment times to production
  - Shifts from a Waterfall SDLC to an Agile methodology
  - How Test-Driven Development impacts when to automate tests
  - How component testing such as: Database testing, Data Warehouse testing, Web-Service testing or Messaging testing impacts what tests to automate
- To ensure success, many organizations have employed a Proof of Concept study

Tool Evaluation Criteria - Tables

Table 1: Definition of Tool Features Metric

	Definition
Easy to install	Determine whether or not an install is required to use software, e.g. install required or cloud based
Knowledge of scripts required	Determine if test scripts can be recorded with recorded user clicks or is programming knowledge required
Programming language	Determine what programming languages the tool will with work
Access to code	Determine if code will be generated during recording and if access is allowed and if code can be modified
Tool customization	Determine if fields be added or deleted in the tool
System requirements	Determine what operating system the tool will run on, what software requirements does the tool necessitate, and what hardware requirements the tool imposes
Platform Support	Determine if the tool can support configuration testing, e.g. tests can be run on different hardware and software configurations, across the network
Testing environments	Determine if the testing environment is a command prompt, Windows GUI, or part of the development environment
Multiuser Access	Determine the tool support for multiple users, support usage via Citrix, support usage via VPN
Defect Tracking integration	Determine if the tool comes with an integrated defect-tracking feature or can be integrated to other defect tracking applications (and how easy can this be done)
User friendly interface	Determine how easy or difficult it is to use software and if the features in the tool are easy to understand
Version Control	Determine if the tool comes with integrated version control capability or can the tool be integrated with other version control tools (and how easy can this be done)
Test Planning and Management	Determine if the tool comes with or can be integrated with test planning and requirements management tool, does the tool follows specific industry standard on testing process (such as SEI/CMM, ISO), e.g. supports planning, managing, and analyzing testing efforts, reference test plans, matrices, and product specifications to create traceability

Common myths and perceptions for automation

Why is the Test Automation adoption failing in organizations?

Case Study 1: Struggling with (against) Mgmt

Ranorex

Excerpts from an Austrian test engineer / manager and his interactions with management

"It Must Be Good, I've Already Advertised It"

Management's intention was to reduce testing time for the system test. There was no calculation for our investment, we had to go with it because manual testing was no longer needed. The goal was to automate 100 % of all test cases.

"Testers Aren't Programmers"

My manager hired testers not be to programmers so he did not have more developers in the department. Once told that some tools require heavy scripting / coding, the message was relayed that testers need to do "Advanced Scripting", rather than "Programming"

"Automate Bugs"

An idea by one manager was to automate bugs we received from our customer care center. We were told to read this bug and automate this exact user action. Not knowing the exact function, we hard-coded the user data into our automation. We were automating bugs for versions that were not in the field anymore.

"Impress the Customers (the Wrong Way)"

My boss had the habit of installing the untested beta versions for presentations of the software in front of the customer. He would install unstable versions and then call developers at 5:30am to fix immediately. We introduced automated smoke tests to ensure good builds.

Experiences of Test Automation – Case Studies of Software Test Automation: Graham & Fewster

Test Automation adoption questions

Aligned into categories for questions that may not have been answered…or asked

Who	What	When	Where	Why	How
• are the right people to create, maintain test cases and object repository information? • may execute automated tests instead creating? • will be the dedicated go to person(s) for the tool?	• is our organization's definition of automated testing: what it is and what it is not? • are the test automation objectives? • is the expected testing coverage sought after with test automation?	• are testing resources available for this initiative? • can training happen in relation to the procurement? • are there project deadlines that compete against test automation?	• will the test automation take place; dedicated test environment? • are the associates located? This will decide whether a training should be remote on onsite • will a central repository reside?	• start a test automation initiative – what are the specific issues? • run a vendor POC program; what do we want from this assessment? • start with a top-down management approach?	• do our test automation objectives differ from our overall testing objectives? • does manual testing fit into a test automation framework? • do existing tools such as (CI, CM, TM) work with the automation solution?

Test Automation adoption failures

Ranorex

Reviewing some not so best practices and assumptions around test automation adoption

Test Automation automates manual test cases

- Manual Test Cases may not have been written for automation with entry points, data and exit criteria
- Error handling is not considered as part of manual testing; separate from defect detection
- Modularity is not a key consideration for manual test case planning, as it is for test automation

Test Automation must cover 100% of the application

- Testers should just go through all of the entire regression test library and strive to have that all automated
- New application changes should just be added to the automation library of tests as it is complete
- This full automation run should happen overnight and any discrepancy should be logged as an application defect

Testers can automate their tests during a release

- Replace the time required for manual testing to do automated testing the first time and each subsequent run will be shorter
- Every manual tester we have can automate their own tests, they have the testing knowledge to do so
- This does not have to be a separate initiative, there is ample time during each project release

Why is the Test Automation implementation failing in organizations?

Case Study 2: Chosing the Wrong Tool

Ranorex

Background for the Case Study	• The goal was to automate the testing of major functionality for web development tools at a large internet software development company
Pre-existing Automation	• Most of the functional tests were executed manually. There were some unmaintained automation scripts that could potentially serve useful if restored
New Tool or Major Maintenance Effort?	• The GUI had gone through major changes for the application. The automation tool had to be flexible enough to automate the changes and be easy to maintain
Moving Forward with existing tool?	• The consensus was that the tool and coding required was difficult for manual testers to maintain. Additionally, the type of object recognition provided did not work
What was done after the tool was replaced?	• Clearly understand how automated tools use objects and realize that identification by image was not adequate. Research was done to find the best tool to identify objects
Conclusion	• Closely examine the tools that work best with the controls in your environments. Topics like maintainability and error-handling/debugging should be critical

Experiences of Test Automation – Case Studies of Software Test Automation: Graham & Fewster

Test Automation implementation failures

Ranorex

What could go wrong when trying to use the purchased solution

The test automation tool won't work on my application

- The application(s) under test uses technologies that do not work well with the test automation tool
- When I attempt to playback the test I just recorded, the recording doesn't playback at all
- I have to use one test automation solution for one technology (Web), another for Mobile, yet another for legacy desktop apps

Screens & Objects in the AUT are changing; automate?

- The automated test case(s) needs to be re-recorded in most times of any GUI changes, causing heavy maintenance
- It's not worth automating newer applications; automation is really only for regression testing of legacy applications
- It's said that automation is key with Agile Development, but automation can't work when deployed in increments

Automation tools are too complex for manual testers

- We don't have time to work with the test automation solution with overly aggressive project schedules and over-allocation
- Test Automation software is primarily marketed to testers with a programming and development skillset / background
- This initiative doesn't have the cross-department support that it should to be effective

Correcting the test automation approach

Test Automation adoption corrections

Ranorex

Changing the mindset concerning test automation

Test Automation automates manual test cases

- Automated Test Cases are separate from manual test cases with entry points, data and exit criteria
- Error handling is an essential part of automated test cases; separate from defect detection
- Modularity is the approach on how automated test cases are developed, to ensure reusability for each automated test step

Test Automation objectives aim for the highest ROI

Test Automation must cover 100% of the application

- Testers start automation with a regression test library, but focus on tests that are time-consuming and have a high ROI
- Applications changes need to be planned for in a Master Test Plan and automated during a maintenance window
- Any automation failure needs to be analyzed whether the defect is in the automated test case or the system under test

Test Plan includes automated and manual test cases

Testers can automate their tests during a release

- The initial cycle of creating and executing automated tests will be longer than manual tests, with each cycle runtime reduced
- Manual testers will require training on the test automation tool, approach and best practices
- Test automation should be a separate initiative, defined with its own goals and objectives - apart from an application release

Separate goals & timelines for Test Automation

Case Study 3: Develop a Mutli-Solution Strategy

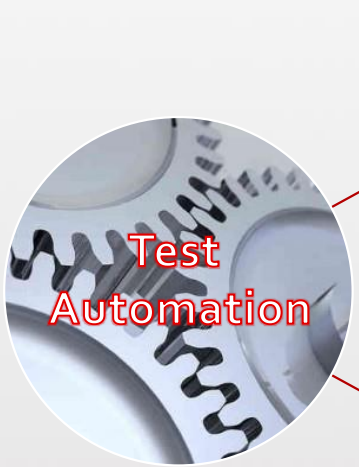
Ranorex

Background for the Case Study

- A leading electronic design automation company had a family of products that was comprised of over 120 programs. This was GUI-based and ported to all leading industry workstation hardware platforms

Current Testing Activities

- Three distinct test phases: Evaluation (done by end-user), Integration and System
- Integration and System was done manually and consumed 38 person-weeks of effort for every software release on every platform



- Completed automation in just under a year
- *First trial*: 2-person weeks manual; automated: 1-person week – but not everything could be automated
- *Without automation*, manual: 10 person-weeks on every platform; automate: 3 person-weeks

- Automate in two phases: *breadth and depth*
- Manual testing took 8 person-weeks per platform; automation took 4 person-weeks
- Total tests automated was approximately 79%

Proposed Solution

- Evaluate and decide between purchasing a solution or building in-house
- Hire an outside consultant to validate the tool selection prior to purchase, then train testing staff on selected tool
- An in-house solution was built

Key Benefits

- Test effort per platform cut in half
- Consistent regression testing
- Less dependent on application SME's
- Better use of testing resources

Software Test Automation: Effective use of test execution tools – Fewster & Graham

Test Automation implementation corrections

Ranorex

Changing the approach to putting test automation into practice

The test automation tool won't work on my application	• Develop a set of criteria that matches your portfolio of applications to automate and request the vendors for a POC • Start to request the object names from development as they turn over new builds. Very important for test automation • Investigate software packages that contain everything in one; vendor management and procurement will thank you!	**Successful POC will prove the tools works on all apps**
Screens & Objects in the AUT are changing; automate?	• Find a tool that you don't have to re-record. If a new control was added, you wouldn't rewrite your manual test case • Consider starting your testing with automated tests to start. The chances of these tests staying current is much higher • Test automation should be a key element in sprint planning; for new functionality and existing maintenance	**Understand the object landscape and prepare the test plan**
Automation tools are too complex for manual testers	• Schedule a just-in-time training session with the tool shortly after procurement, with staff not allocated to a critical project • Layout the resource plan on how every tester contributes. Plenty of roles are needed for this initiative than just coding • Testing management should work with the IT leaders to discuss the time required from a PM, BA, Dev and DBA	**Develop a resource plan for all required associates**

Case Study 4: Automating Government Systems

Ranorex

Background for the Case Study

- The goal was to speed up the Department of Defense (DoD), software delivery to the field. It was leading-edge technology, but lengthy testing and certification processes were in place before software could be released

Can't Be Intrusive to System Under Test (SUT)
- The tool has to be independent of the System Under Test; no modules should be added

Must Be OS Independent
- The solution(s), had to go across Windows, Linux, Solaris, Mac, etc.

Must Be Independent of the GUI
- The tool needed to support all languages that the applications were written in

Must Automate Tests for Both Display & Non-Display Interfaces
- The system should be able to handle operations through the GUI and backend processes

Must Work in a Networked Multicomputer Environment
- The tool needs to test a system of systems: multiple servers, monitors and displays interconnected to form one SUT

Non-Developers Should Be Able to Use the Tool
- One of the main requirements from the DoD. Testers were SME's in the application, but not software developers writing scripts.
- The steps should be action driven: Projects – Test Cases – Test Steps - Actions

Must Support an Automated Requirements Traceability Matrix
- Needed a framework that could allow for test management activities, including documenting a test case in the Automated Test and Re-Test (ATRT) solution, along with a hierarchical breakdown of test cases into test steps and sub-steps

Test Automation ROI

- Solutions selected spanned across technologies
- Reduced some test cycles from 5 testers to 1
- After initial resistance, testing department now worked on automated test cases, both development and execution
- Additional processes for Continuous Integration and Test Management were now enabled

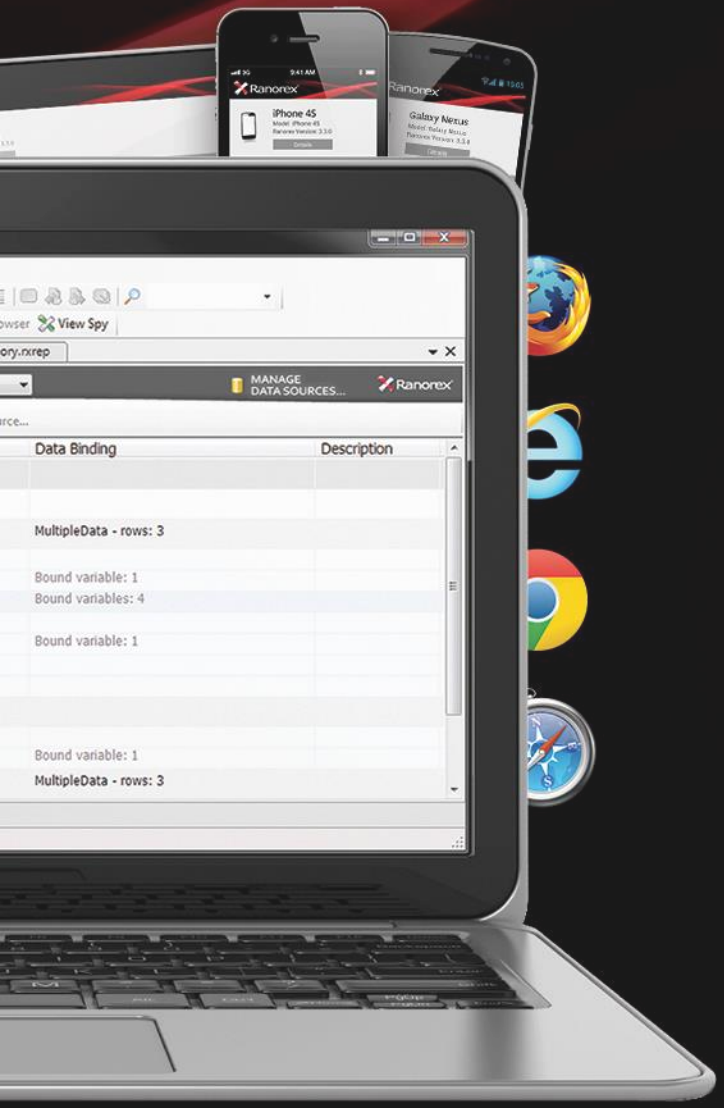
Experiences of Test Automation – Case Studies of Software Test Automation: Graham & Fewster

Session Recap

Recap of the presentation

Reviewing the main points of the presentation

- The Test Automation industry focused on desktop, but is driving to multi-technology stacks to cover a large amount of platforms
- Primary skillset for those working in test automation was Programming or Scripting, but solutions are available for those without these skills
- Common perceptions and myths will exist, review the list of questions to be asked and use that to drive your strategy
- Present some of the real-life case studies forward to management, to ensure this does not occur during your research and implementation
- Review some common failures in the theory and implementation and compare if any of those are problematic at your organization
- Make the corrective action for each failure step accordingly. Incorporate test automation as part of your overall test strategy



Thank you for attending!

Jim Trentadue
Enterprise Account Manager- Ranorex
jtrentadue@ranorex.com
Thursday, January 15, 2015