T9
Test Automation
Thursday, May 3rd, 2018
11:15 AM

Not Your Parents’ Test Automation: Application of Non-Traditional Automation

Presented by:

Paul Grizzaffi
Magenic

Brought to you by:

350 Corporate Way, Suite 400, Orange Park, FL 32073
Paul Grizzaffi
Magenic

Paul Grizzaffi is a principal automation architect at Magenic. His career focus has been on the creation and deployment of automated test strategies, frameworks, tools, and platforms. Paul has created automation platforms and tool frameworks based on proprietary, open source, and vendor-supplied tool chains in diverse product environments of telecom, stock trading, e-commerce, and healthcare. A Certified ScrumMaster, he is on the Industry Advisory Board of the Advanced Research Center for Software Testing and Quality Assurance at UT Dallas. Contact Paul via LinkedIn or Twitter and read his blog.
Not Your Parents’ Automation

Who Is This Guy?
Paul Grizzaffi
Principal Automation Architect at Magenic
“Software Pediatrician”
Career focused on automation
Advisor to Software Test Professionals (STP)

paulg@magenic.com
http://www.linkedin.com/in/paulgrizzaffi
@pgrizzaffi
http://responsibleautomation.wordpress.com
What’s Coming

Context
Examples
Non-Traditional Tools

What Comes To Mind?
What Comes To Mind?

- Scripts
- Saves Time
- Tests Cases
- A Lot Of Work
- Inconsistent
- Regression Testing
- I'm Getting Laid Off
- Brittle

OH! I'M SORRY

DID I BREAK YOUR AUTOMATION?
Traditions Can Be Important

- Traditional automation
  - Detect behavior changes
  - Reduce effort on smoke and regression testing
  - Earlier execution, earlier alerts
  - Scheduled execution

My previous organizations have this and it’s valuable

Is there something else?
What If We Think Differently?

“Let’s help the humans”
What makes us more efficient or more effective?
What’s valuable?
What hurts?
Instead of automation, how about assistance?
Automation Assist
Umbrella term for non-traditional automation
“Words mean things” – Patrick Amaku
Changing meanings is difficult
New vocabulary for new concepts
Things that increase the value of manual effort
“Off label” tool usage
New tools, applications, scripts
Tools not traditionally used in automation

Lets Look At Some Examples

High Volume Automated Testing
High Volume Automated Testing

“a family of testing techniques that enable the tester to create, run and evaluate the results of arbitrarily many tests”

Workshop on Teaching Software Testing (WTST) 2013

High Volume Automated Testing

Aka HiVAT
Dr. Cem Kaner, J.D., Ph.D.: http://kaner.com/?p=278
Harry Robinson: http://wtst.org/?page_id=169
Interesting facets for the organization
  Many executions
  Random execution
  Results vetted by humans
Scud

The premise
- Product is large, complex, aging
- Not feasible to enumerate and follow all paths

The solution
- Random menu clicker – “Scud”
- Looking for things that “don’t seem right”
- Selenium-based Python script: 32 hours of effort
- Found four issues in the first week of use

Why that solution?
- Value shown in previous organization
- Selenium was the right kind of tool kit
- Broad license usage and not competing with other tool
ACH File Checker

The premise
- Need to test that ACH files are created correctly
- Funding was cut; project to end early
- Development would not be completed

The solution
- Create a command line app to perform part of the testing
- C# program: ~5 hours of effort
- Eliminated hours of testing effort

Why that solution?
- The code was there, but provided no current benefit
- Cheap, fast way to enable value
- Something’s better than nothing
The premise
“Regression” testing between software versions
Volatile stock market data
Takes about 8 hours of effort

The solution
Comparison tool
Connect to both servers and compare
Time to test reduced 1 minute
C++ program: “ROI”: 6 weeks

Why that solution?
Traditional test scripts not appropriate
Existing API into the product
C++ primary development language
Product bug found during first execution
Account Creator

The premise
- User account creation is a pain in the...well...you know...
- UI is a slow, lots of typing
- Multiple tables to update

The solution
- Script created in Selenium IDE
- Effort: “a couple of days”

Why that solution?
- Saved ~20 minutes per account entry
- Easy to automate via UI
- Hard to automate via database interaction
- Coding was not forte
Zero Remover

The premise
- Thousands of “golden files”
- In some cases DB now returns NULL instead of 0.00
- Manual effort estimate: 4 – 6 weeks

The solution
- A program to do the file transformation
- C# program: 5 hours

Why that solution?
- Disposable
- Windows/.Net shop
- Easy to distribute executable
Some Words About Effort

Effort calculation is a funny thing
ROI can be difficult to calculate
http://blog.smartbear.com/testing/there-aint-no-roi-in-software-testing/

Dawn Haynes

“Opportunity cost” can be easier and as useful
Think about value
Dammit Jim... I’m a Tester, Not A Programmer

So?
The command line is your friend
Built in text and file processing commands
Chainable commands
Example: Incomplete transaction due to timeout
Example Log File

So Close
The Pipe

A way of chaining commands together

<command a> | <command b> | ...

Result of <command a> is input to <command b>

Supported on most command prompts/shells

That’s The Stuff!
Let’s Be Flexible

Powershell & Unix/Linux/MacOs
Is that not automation assist?

Is that not a script?

Didn’t we just program?

Tools

| (aka pipe)

Search-String, findstr, grep

Get-Content –Wait, tail –f

wget

Unxutils

Cygwin

Bash on Windows!
Takeaways

This is different
Scripting tests is OK, but it’s an implementation
Situational – the “knowns” help guide
Ecosystem – running, distributing
Life span – disposable or long term
Notice anything?

Questions?

- paulg@magenic.com
- http://www.linkedin.com/in/paulgrizzaffi
- @pgrizzaffi
- http://responsibleautomation.wordpress.com