

To Automate Or Not To Automate?

Kelly Adams
Brian LeSuer



Overview

- Why automate?
- How to prioritize what to automate
- Survey of automation projects



Why Automate?

- Productivity - reduces labor-some task of manual testing
- Repeatability - same tests get executed each time
- Efficiency – shortens test cycles
- Effectiveness – increases test coverage
- Flexibility - late changes can be accepted



Why Automate?

- Stability – simulate usage over long period of time
- Complacency – over time groups tend to short-cut test cycles
- Environment – provides a means of testing when a component outside application changes
- Accountability – can produce a record of executed test procedures for audit
- Preservation – tester expertise is captured in the automated tests



Prioritizing What To Automate

■ Good candidates

- Short or simple transactions
- Many data combinations
- Expected results are stable or easy to generate at runtime
- Tests that are executed regularly
- Tasks that are difficult to do manually
- Highest priority features

■ Poor candidates

- Long or complex transactions
- One-offs
- Unstable or difficult to predict results



Survey of Automation Projects

■ Project Description

■ Challenges

■ Solutions



Point of Sale/Cash Register

- Drive Cash Register to perform regression of cashier functions: sales, returns, exchanges
- Challenges
 - Multiple verification points
 - POS won't allow other foreground apps
- Solutions
 - Predict expected results based on inputs
 - Capture Virtual Receipt, Electronic Journal, Credit Logs, Write all data to ini files and compare
 - Suspend transaction for manual steps



Web Retail

- Verify large retail web site functionality such as searching for items and placing orders
- Challenges
 - Dynamic data from external sources
 - Site structure and navigation data dependent
- Solutions
 - Base actions on positional items
 - Capture data early in tests and use later for verification of results



Multi-User/Store Server

- Simulate multiple terminals in a store or office to ensure proper handling of concurrent transaction
- Challenge
 - Synchronize multiple terminals to perform concurrent transactions
- Solution
 - Use test tool features that allow spawn and rendezvous so that each machine can be controlled independent of the others
 - Create synchronization points to simulate race conditions



Multiple Server Verification

- During offline hours, confirm that all servers are functioning as required
- Challenges
 - Multiple servers must be in same state
 - Limited time to test all systems
- Solutions
 - Use “smoke” test which tests all components which could have startup or server implications
 - Raise alarms (e.g. emails) if critical failures seen on any server



Single User Performance

- Measure elapsed time to respond to a user request in a GUI application
- Challenge
 - Measure amount of time that it takes to respond to a user request
- Solution
 - Create and start a timer
 - Perform the request
 - Watch for cue from GUI that processing is complete
 - Enable State of a control
 - Disappearance of an hourglass cursor or progress bar
 - Stop the timer and measure elapsed time
 - Generate timer report with O/S resources



Load Test Component

- In conjunction with performance test, drive user scenarios unsuitable for performance tools
- Challenges
 - Non-web client / protocol affects servers
 - Functional test should be synchronized
- Solutions
 - Build function to control start time and duration of test
 - Log performance type data in common location
 - Arrange testing for machine availability



Non-GUI: Folder Monitoring

- Measure performance of a server operation that processes batches of files
- Challenge
 - Determining when the process is complete
- Solution
 - Stop service on remote servers via Computer Management
 - Copy given number of files to listener location
 - Poll the folder locations to determine when all files were consumed
 - Publish report at the conclusion of each run



Heavily Graphical Application

- Test drawing tool which works within a custom canvas which is not visible to automation tools
- Challenge
 - Verify resulting image with tool which cannot “see” it
- Solution
 - Calculate critical points of figures and verify with cursor type changes
 - Verify or click regions on screen by capturing 1x1 pixel bitmaps
 - Click and drag predicted points to verify resizing



Graphical Elements

- Test that a PDF conversion process maintains integrity of files
- Challenge
 - Verification of converted files – many aspects
- Solution
 - Generate baselines with production app
 - Harness third-party tool to convert baseline files to TIFF
 - Harness third-party tool to compare TIFF files
 - Able to test thousands of files with minimal manual intervention



Web Page Size Validation

- Verify that all pages on site are within size requirements and will load quickly
- Challenges
 - Site contains massive number of pages
 - Size of pages dynamic based on multiple components
- Solutions
 - Build set of test URLs using tool if possible
 - Allow specification of depth levels to test
 - Measure number of objects and total page size
 - Log exceptions to spreadsheet and sort at end



Handheld Device

- Test a Windows CE application on a handheld device
- Challenge
 - Creating an environment where test tool can interface with the application under test
- Solution
 - CE provides a Windows representation of the application on a PC interfaced with handheld device
 - Drive the Windows application on the PC which in turns sends commands to the device
 - Allows user to test the application as if it were running on PC



System Installations

- Install new software on systems daily
- Challenge
 - Driving and rebooting a remote machine
 - Verification of registry and file distribution
- Solution
 - Drive from remote system using agent
 - Delay reboot to allow time to disconnect
 - Configure remote system to log on and load agent
 - Put controller in wait loop for reboot period
 - Execute registry edit application to verify installation



Multiple Implementations

- Some application dev tools can generate an app for more than one deployment platform
- Challenge
 - Create a single set of tests to validate java and browser version of same application
- Solution
 - Create a single test plan and a single set of test functions
 - Create dual sets of application definition where all objects have same name as twin
 - Can be challenging to maintain same level for each object



Complex Navigation Tree

- Application includes large navigation tree, with simple testing required at each node
- Challenge
 - Build test plan and data to navigate tree
- Solution
 - Capture navigation tree in structured format
 - Write conversion program to convert tree to test plan and build navigation data for each node
 - With navigation accomplished, same simple test executed for large sets of nodes



Other Uses Of Automation

- Creating test case data
- Performing routine tasks
- Keeping a secure machine from logging off
- Executing setup steps
- Cleaning up after manual or automated testing
- Capturing monitor data on system



Thank you!
Questions?

