Lessons Learned in Test Automation

Goran Pop-Jordanov

Presentation Overview

- About me
- Why automating?
- What to automate?
- Agile Development
- Case studies
- Resources
- ROI

About me

- BSEE, Computer Engineering (SW/HW)
- Software Developer
- Quality Manager
- Test & Verification Engineer
- Test Automation Developer
- Test Automation Lead (Technical)
- Project Leader
- Certified Psychotherapist (Germany)

About me (cont.)

- Energoinvest, Belgrade, Yugoslavia
- PC College, Belgrade, Yugoslavia
- Siemens, Vienna, Austria
- European Space Operation Center, Darmstadt, Germany
- Management Information Systems, Darmstadt, Germany
- Bank Austria, Vienna, Austria
- IBM, Cologne, Germany
- Klocwork, Ottawa, Canada
- RIGPA, Lodeve, France
- IBM, Ottawa, Canada
- Nav Canada, Ottawa, Canada

Why automating?

- All dream about it
- For fun
- Higher product confidence
- Shorter delivery cycles
- Late changes
- Cost savings
- Like to program

What to automate?

- Routine tasks (download, install, setup, reporting)
- Smoke tests
- Continuous integration
- Static analysis
- Performance
- Unit test
- Feature test (integration, system, regression)

Technology Levels

- Infrastructure
- GUI
- API (command layer)
- CLI
- Domain language

Case Study: Transaction Processing

- Sneak-in activity
- System level test: feeding input – verifying output
- Extending existing manual TCs (HTML) with embedded shell scripts
- Fair coverage of base functionality

Transaction Processing - continued

- Performance testing:
  - logging the transactions in production
  - re-playing in test bed in exactly same time pattern
- Performance improvement in new releases
- Performance problem analysis

Case Study: Logging, Reporting...

- Logging the customer calls
  - moving from paper & pencil (!)
  - automatic timestamps
  - linking to related problems
  - search
- What is installed on which machine
- Test results overview

(we are talking pre-wiki age)

Case Study: Command Line

- Again - created in an afternoon out of fustration
- Smoke test: command line options
- Using tables of frequently changed option names
- Runnable on many different operating systems (Java)

Case Study: Public API

- Using special test clients to expose all of the API
- Reusing the existing automation framework

Case Study: Difficult to Verify

- Use the automation to run the tests
- Unsure what the correct results are
- Support for manual verification:
  - diff
  - visual synchronisation

Case Study: SUT Scripting

- Client/Server
- Client scripting capability used to simulate multiple parallel users (say 10) each using separate physical machine (pre-virtualization age)
- Load/Performance testing

Case Study: GUI Test 1

- Complex graphical editors with embedded browsers and an application server
- Object based GUI recognition
  - not click(23, 56) but
  - Button.click("OK");
- Extending the tool object recognition capabilities
- Task layer (not record/playback)

GUI Test 1 - continued

- Business Object layer
  - Button.select(“OK”), translates into
  - Dialog.accept();
- Same tests for both HTML/Java clients

Case Study: API driven GUI

- Model View Controller pattern
  - Initiate the “user like” actions through model “commands”
- Challenging current SUT architecture
- Initiated product wide developer action to provide the appropriate test API
- This API could be used for live demos and even exposed to customers

Case Study: Infrastructure Automation

- Download, install, setup, run, report, cleanup
- Poor build system interface
- Endless count of temporary solutions created by coop students
- Daily waste of time
- Never delivered “proper” solution promised by the build team

Case Study: GUI Test 1b

- Outsourced team extension
- Home made extension of same base tool
- Taken off the team after 6 months “to firefight higher priorities”
- Effort thrown away

Case Study: GUI Test 2

- Fixtures for Easy Software Testing (FEST)
- Fluid interface: window().button().click();
- Swing: Generic Matcher extensibility
- Assert Module
- Reflection Module
- Mock Object Module
- http://fest.easytesting.org/wiki/pmwiki.php

Case Study: Test Driven Development

- Test first development strategy
- Both GUI and unit test level
- FEST, JUnit
- Fortyfying refactorings
- to be continued...

Agile Development

- Ideal for test automation
- Weekly iterations
- Easy to count story points from test cases
- Velocity = test cases per iteration
- Giving a good example to development team

Resources

- Test automation is software development
- Dedicated unchanging team
- Chronic lack of resources
- Fear of waste

Return On Investment

- Perceived (advertised) vs real ROI
- Effectiveness of test automation depends on SUT quality
- Bug filing responsiveness
- Exploring the SUT from different perspective
- Better testability - better quality
- ROI after 3 releases

Thanks :)