Test Automation and Keyword-driven testing

Brian Nielsen,

bnielsen@cs.aau.dk
3. Script-Based Testing

+/- test impl. = programming
+ automatic execution
+ auto regression testing

fragile tests break easily?
(dependents on abstraction)

- ad-hoc coverage
- no coverage measurement
4. Keyword-Driven Testing

+ abstract tests
+ automatic execution
+ auto regression testing
  - robust tests
  - ad-hoc coverage
  - no coverage measurement
Script-based Testing

Test inputs and observations (verdict assignment) *programmed* in scripts in dedicated or general-purpose languages.

+ Repeatable (for regression testing)
+ Data driven testing: Fixed scripts, each parameterized with different data
  – Mixes test harness and logical test cases
  – Must be updated when specs (or impl) change
  – Very lengthy
  – Good programmers may write well-structured reusable test code but *"it is just test code"*
Keyword Driven Testing

• Script based automation where test case design is **separated** from automation
  – Focus on actions user/environment can do on objects in SUT (at different abstraction levels)
    • ”Actions” appear in scripts,
    • ”Action-code” implements the action
• aka “Action-word” testing (or, table driven testing), < 1994
• Black-box, subsystem, accept tests
• Automated execution using a “framework”.
Keyword Driven Testing

+ Concise, flexible, maintainable,
+ Read-/writeable by non-programmers

+ Expressiveness of a scripting language
  – Control structures and complex computations
  – (Branching in test cases, complex data, matching of expected results)
# Keyword Driven Test

## Eg. Sequence

<table>
<thead>
<tr>
<th>Object</th>
<th>Keyword</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>LoginDialog</td>
<td>Login</td>
<td>&quot;badUserId&quot;, &quot;GoodPass&quot;</td>
</tr>
<tr>
<td></td>
<td>VerifyLogin</td>
<td>&quot;Login failed&quot;</td>
</tr>
<tr>
<td>LoginDialog</td>
<td>Login</td>
<td>&quot;goodUserId&quot;, &quot;badPass&quot;</td>
</tr>
<tr>
<td></td>
<td>VerifyLogin</td>
<td>&quot;Login failed&quot;</td>
</tr>
<tr>
<td>LoginDialog</td>
<td>Login</td>
<td>&quot;goodUserId&quot;, &quot;GoodPass&quot;</td>
</tr>
<tr>
<td></td>
<td>VerifyLogin</td>
<td>&quot;welcome&quot;</td>
</tr>
<tr>
<td>WelcomeDialog</td>
<td>changeLogin</td>
<td>&quot;newUserName&quot;, &quot;newpwd&quot;</td>
</tr>
<tr>
<td></td>
<td>confirm</td>
<td>&quot;goodUserId&quot;, &quot;newpwd&quot;</td>
</tr>
<tr>
<td></td>
<td>confirmChange</td>
<td>&quot;Change Failed&quot;</td>
</tr>
</tbody>
</table>
KWD Process

Test Expert

Receiving Test Basis (Business Req./Detailed Design)

Defining additional keywords needed

Creating Automated Scripts

Automation Expert

Developing Infra

Debug + Full execution on Dev. environment

Execution on Test Environment

Transfer selected TCs to the regression repository

Automation Fixture

```
Open(Scripts)
While(!EOF){
  Keyword, data=readFile();
  Case(keyword){
    Login: widget->submit(data);
    verifyLogin: ...
  }
  Close()
}
```
Tools

- SAFS, open source
- EMOS Framework, open source
- FIT/FitNesse, open source
- Certify, Worksoft
- Unified TestPro, SDT
- TestFrame, LogicaCMG
- TestArchitect, LogiGear
- Tools with built-in keyword support:
  - TestQuest Pro,
  - QuickTest Professional
Example

http://localhost:8080/FitBookExamples.Chapter4TestingActions.Fig6TestChatServer
## Comparison

<table>
<thead>
<tr>
<th>Criteria</th>
<th>Manual</th>
<th>Script</th>
<th>KWD</th>
<th>Model-Based</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cost Development</td>
<td>L</td>
<td>M</td>
<td>M</td>
<td>L</td>
</tr>
<tr>
<td>Maintenance</td>
<td>H?</td>
<td>H</td>
<td></td>
<td>L</td>
</tr>
<tr>
<td>Tools/equip</td>
<td>L</td>
<td>M</td>
<td>M</td>
<td>H</td>
</tr>
<tr>
<td>Time</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Time to first test</td>
<td>L</td>
<td>M</td>
<td>M</td>
<td>M-H</td>
</tr>
<tr>
<td>Re-execution</td>
<td>H</td>
<td>L</td>
<td>L</td>
<td>L</td>
</tr>
<tr>
<td>Maintenance</td>
<td>L</td>
<td>H</td>
<td>H</td>
<td>L</td>
</tr>
<tr>
<td>Learn-ability</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Skills</td>
<td>L</td>
<td>H</td>
<td>M*</td>
<td>H*</td>
</tr>
<tr>
<td>Maturity</td>
<td>L</td>
<td>M</td>
<td>M-H</td>
<td>H</td>
</tr>
<tr>
<td>Readability</td>
<td>L</td>
<td>L</td>
<td>M</td>
<td>H*</td>
</tr>
<tr>
<td>Interest / Challenging</td>
<td>L</td>
<td>M</td>
<td>M</td>
<td>H</td>
</tr>
<tr>
<td>Effectiveness</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bug detection</td>
<td>H</td>
<td>L</td>
<td>L</td>
<td>H</td>
</tr>
<tr>
<td>Tracability</td>
<td>L</td>
<td>L</td>
<td>M</td>
<td>H</td>
</tr>
<tr>
<td>Coverage</td>
<td>L</td>
<td>L</td>
<td>M</td>
<td>H</td>
</tr>
</tbody>
</table>
References

• http://en.wikipedia.org/wiki/Keyword-driven_testing
• http://www.cs.waikato.ac.nz/~marku/mbt/
• http://www.onestoptesting.com/automation-framework/keyword-driven-testing/
• Mark Fewster and Dorothy Graham, Software Test Automation, 1999 (Chapter 22)
• M. Utting Practical Model-based testing