Model-Based Testing of a Financial Application – A Case Study

Bruno Legeard,
smartesting,
France
Model-Based Testing of a Financial Application Case Study

Prof. Bruno Legeard
Smartesting
Agenda

1. Model-based testing to industrialize the functional testing process
2. Financial application – Experience report
3. Summary and discussion
Agenda

Model-based testing to industrialize the functional testing process

Financial application – Experience report

Results and discussion
Test Industrialization – Key issues

“Functional testing industrialization based on a seamless process from Business requirements to Test Assets”

– Business Requirements and Use cases as drivers
– Automatic Generation of Test Cases and Executable Scripts
– Implementation of Test Management and Execution tools.
– Creation of Competence Centers and Test Services for functional testing
Functional testing – manual process

Test Design

Test results

Requirements & specifications

Test Execution Environment

Manual execution or test automation

Test Script

Test plan construction

Test repository

Traceability management

Coverage matrix
Model-based testing – Smartesting® Process

- Automation Layer
- Test Management Environment
- Test Results metrics
- Keywords-based testing automation
- Subject Matter Expert
- Test Architect
- Smartesting Test Designer
- Iterative Process
- Smartesting Test Publisher
- Test plan & Test cases
- Coverage matrix
- Inspection
- Automatic generation
- Executable Test scripts
- Requirements
- Test Models

- Model-based testing – Smartesting® Process

Keywords: based testing, automation, Automation Layer, Test Management Environment, Environment, Test Results, Subject Matter Expert, Test Architect, Smartesting Test Designer, Iterative Process, Test plan & Test cases, Coverage matrix

Inspection
Test Generation example

- Tests & Scripts Generation
- Business Requirements
  Traceability along with Test Process

Specifications

Behavior modeling with IBM RSM or Borland Together

Test Generation with Smartesting Test Designer

Tests Publishing in HP QC/QTP, IBM Rational…
Agenda

Model-based testing to industrialize the functional testing process

Financial application – Experience report

Results and discussion
BNP-Paribas Case Study

Smarttesting on a Business Application of BNPP
Confirmation/chasing application
– Matching deals, Reporting, Litigation, …

Case study as presented at "atelier BNPP", the external BNPP forum (www.atelier.fr 26/03/08)
Scope of the project:

- Migration of technology Borland-Delphi → Java - J2EE Framework
- Off-shore Development in India (Internal team - Mumbai)
- Functional qualification of the application on site BNPP Paris
- Iterative approach (agile project approach)
- Functional qualification with Smartesting solution and automation of the non-regression tests

Testing objectives: Ensure functional qualification and certification of off-shore deliveries
Smarttesting on a Business Application of BNPP

**Tooling:**
- UML modeling: Rational Software Modeler 7.0
- Tests generation: Test Designer 3.2
- Test Management: HP-Mercury Quality Center 8.0
- Test Automation: HP-Mercury Quick Test Professional 8.2

**Methodology:**
- Iterative approach
- 6 deliveries to test
- Systematic regression testing
Smartesting on a Business Application of BNPP

Project Typology:
- Financial application (deal matching, chasing and litigation)
- Web application

Project Context:
- Migration from Borland-Delphi technologies → Java - J2EE Framework
- Development off-shored in India (Internal team)
- Functional validation at customer site – Paris

Testing objectives:
- Guarantee the non-regression of the new web based application compare to the Legacy code
Smarttesting – *Step-by-step*

**Test Environment**
(HP/Mercury, Borland, Compuware, Rational, etc.)

**Results analysis**

**Requirements**

**Translation**

**UML Specification**

**Test Plans & Test cases**

**Publisher for HP QC/QTP**

**Smarttesting Test Designer v3.2**

**Automatic Generation**

**Smarttesting**
## Starting Point: Requirements and Legacy code

<table>
<thead>
<tr>
<th>Pre-Matching</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>PreMatching Search</td>
<td>Multi-criteria</td>
</tr>
<tr>
<td>PreMatching Search Warning</td>
<td></td>
</tr>
<tr>
<td>Confirmation Details</td>
<td>Confirmation Details (Load Pair Confirmation and Full Detail)</td>
</tr>
<tr>
<td>PreMatching Display Deal</td>
<td></td>
</tr>
<tr>
<td>PreMatching Export results</td>
<td></td>
</tr>
<tr>
<td>XLS</td>
<td></td>
</tr>
<tr>
<td>Warning PreMatching Export Results XLS</td>
<td></td>
</tr>
</tbody>
</table>
Test modeling

Result Analysis

Test execution
Environment HP QTP

Test Scripts

Test Results

Requirements

Test Design

Test Model

Smartesting Test Designer v3.2

Automatic Generation

Test Plan & Test cases

Publisher for HP TD/QTP

Automatic Generation

Test modeling
Model: Class Diagram

- Business key entities
- Point of control and observation
Model: Dynamic behaviour

- Layered UML State Machine
- Express the dynamic of the application
Model - Test Data

Test generation based of self-defining data context

- display

- matching_mod

- SUT : ARC ORCA

Deal1 : Deal
- dealMatchingStatus = AMS_LITIGATION
- dealEvent = NEW
- dealApplicationSite = FF
- dealNumber = DEAL_KNOWN_1
- dealInstrument = FXD

Deal2 : Deal
- dealMatchingStatus = AMS_PENDING
- dealEvent = AMEND
- dealApplicationSite = FF
- dealNumber = DEAL_KNOWN_2
- dealInstrument = FXD
Check the Model

Test Designer™ Plug-In
- Type Checking
- Export model
Test Generation

- Test environment (HP/Mercury, Borland, Compuware, Rational, JUnit, ...)
- Requirements
- Translation
- UML Specification

Smarttesting Test Designer
- Automatic Generation
- Test Plan & Test Cases
- Publisher for HP TD / QTP
- Test Scripts
- Test Result

Automatic Generation
### Test Generation

#### Test list/

<table>
<thead>
<tr>
<th>Test name</th>
<th>Model element</th>
<th>UserNote: Save Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>save_the_note_67</td>
<td>One deal selected</td>
<td></td>
</tr>
<tr>
<td>save_the_note_66</td>
<td>One deal selected</td>
<td></td>
</tr>
<tr>
<td>save_the_note_65</td>
<td>One deal selected</td>
<td></td>
</tr>
<tr>
<td>save_the_note_64</td>
<td>One deal selected</td>
<td></td>
</tr>
<tr>
<td>save_the_note_63</td>
<td>One deal selected</td>
<td></td>
</tr>
<tr>
<td>save_the_note_62</td>
<td>One deal selected</td>
<td></td>
</tr>
<tr>
<td>save_the_note_61</td>
<td>One deal selected</td>
<td></td>
</tr>
<tr>
<td>save_the_note_60</td>
<td>One deal selected</td>
<td></td>
</tr>
<tr>
<td>save_the_note_59</td>
<td>One deal selected</td>
<td></td>
</tr>
<tr>
<td>save_the_note_58</td>
<td>One deal selected</td>
<td></td>
</tr>
<tr>
<td>save_the_note_57</td>
<td>One deal selected</td>
<td></td>
</tr>
<tr>
<td>save_the_note_56</td>
<td>One deal selected</td>
<td></td>
</tr>
<tr>
<td>save_the_note_55</td>
<td>One deal selected</td>
<td></td>
</tr>
<tr>
<td>save_the_note_54</td>
<td>One deal selected</td>
<td></td>
</tr>
<tr>
<td>save_the_note_53</td>
<td>One deal selected</td>
<td></td>
</tr>
<tr>
<td>save_the_note_52</td>
<td>One deal selected</td>
<td></td>
</tr>
<tr>
<td>save_the_note_51</td>
<td>One deal selected</td>
<td></td>
</tr>
<tr>
<td>save_the_note_46</td>
<td>One deal selected</td>
<td></td>
</tr>
<tr>
<td>save_the_note_45</td>
<td>One deal selected</td>
<td></td>
</tr>
<tr>
<td>save_the_note_44</td>
<td>One deal selected</td>
<td></td>
</tr>
<tr>
<td>save_the_note_43</td>
<td>One deal selected</td>
<td></td>
</tr>
<tr>
<td>save_the_note_42</td>
<td>One deal selected</td>
<td></td>
</tr>
<tr>
<td>save_the_note_41</td>
<td>One deal selected</td>
<td></td>
</tr>
<tr>
<td>save_the_note_40</td>
<td>One deal selected</td>
<td></td>
</tr>
<tr>
<td>save_the_note_39</td>
<td>One deal selected</td>
<td></td>
</tr>
<tr>
<td>save_the_note_38</td>
<td>One deal selected</td>
<td></td>
</tr>
<tr>
<td>save_the_note_37</td>
<td>One deal selected</td>
<td></td>
</tr>
<tr>
<td>save_the_note_36</td>
<td>One deal selected</td>
<td></td>
</tr>
<tr>
<td>save_the_note_35</td>
<td>One deal selected</td>
<td></td>
</tr>
<tr>
<td>save_the_note_34</td>
<td>One deal selected</td>
<td></td>
</tr>
<tr>
<td>save_the_note_33</td>
<td>One deal selected</td>
<td></td>
</tr>
<tr>
<td>save_the_note_32</td>
<td>One deal selected</td>
<td></td>
</tr>
<tr>
<td>save_the_note_31</td>
<td>One deal selected</td>
<td></td>
</tr>
<tr>
<td>save_the_note_30</td>
<td>One deal selected</td>
<td></td>
</tr>
<tr>
<td>save_the_note_29</td>
<td>One deal selected</td>
<td></td>
</tr>
<tr>
<td>save_the_note_28</td>
<td>One deal selected</td>
<td></td>
</tr>
<tr>
<td>save_the_note_27</td>
<td>One deal selected</td>
<td></td>
</tr>
<tr>
<td>save_the_note_26</td>
<td>One deal selected</td>
<td></td>
</tr>
<tr>
<td>save_the_note_25</td>
<td>One deal selected</td>
<td></td>
</tr>
<tr>
<td>save_the_note_24</td>
<td>One deal selected</td>
<td></td>
</tr>
<tr>
<td>save_the_note_23</td>
<td>One deal selected</td>
<td></td>
</tr>
<tr>
<td>save_the_note_22</td>
<td>One deal selected</td>
<td></td>
</tr>
<tr>
<td>save_the_note_21</td>
<td>One deal selected</td>
<td></td>
</tr>
<tr>
<td>save_the_note_20</td>
<td>One deal selected</td>
<td></td>
</tr>
<tr>
<td>save_the_note_19</td>
<td>One deal selected</td>
<td></td>
</tr>
<tr>
<td>save_the_note_18</td>
<td>One deal selected</td>
<td></td>
</tr>
<tr>
<td>save_the_note_17</td>
<td>One deal selected</td>
<td></td>
</tr>
<tr>
<td>save_the_note_16</td>
<td>One deal selected</td>
<td></td>
</tr>
<tr>
<td>save_the_note_15</td>
<td>One deal selected</td>
<td></td>
</tr>
<tr>
<td>save_the_note_14</td>
<td>One deal selected</td>
<td></td>
</tr>
<tr>
<td>save_the_note_13</td>
<td>One deal selected</td>
<td></td>
</tr>
<tr>
<td>save_the_note_12</td>
<td>One deal selected</td>
<td></td>
</tr>
<tr>
<td>save_the_note_11</td>
<td>One deal selected</td>
<td></td>
</tr>
<tr>
<td>save_the_note_10</td>
<td>One deal selected</td>
<td></td>
</tr>
<tr>
<td>save_the_note_9</td>
<td>One deal selected</td>
<td></td>
</tr>
<tr>
<td>save_the_note_8</td>
<td>One deal selected</td>
<td></td>
</tr>
<tr>
<td>save_the_note_7</td>
<td>One deal selected</td>
<td></td>
</tr>
<tr>
<td>save_the_note_6</td>
<td>One deal selected</td>
<td></td>
</tr>
<tr>
<td>save_the_note_5</td>
<td>One deal selected</td>
<td></td>
</tr>
<tr>
<td>save_the_note_4</td>
<td>One deal selected</td>
<td></td>
</tr>
<tr>
<td>save_the_note_3</td>
<td>One deal selected</td>
<td></td>
</tr>
<tr>
<td>save_the_note_2</td>
<td>One deal selected</td>
<td></td>
</tr>
<tr>
<td>save_the_note_1</td>
<td>One deal selected</td>
<td></td>
</tr>
<tr>
<td>save_the_note_0</td>
<td>One deal selected</td>
<td></td>
</tr>
</tbody>
</table>

#### Equivalence classes

- Test case
- Test Data

---

**Script Example:**

```python
# Example script for test case

# Import necessary libraries

# Define test scenario

# Initialize test case

# Define test steps

# Execute test steps

# Verify test conditions

# Log test results
```
Publication into HP QualityCenter & QuickTestPro

Keywords:
- based testing
- automation

Flowchart:
1. **Test Result**
2. **Requirements**
3. **Test Design**
4. **Test model**
5. **Test Scripts**
6. **Publisher for HP TD / QTP**
7. **Test Plan & Test Cases**
8. **Smartesting Test Designer**
9. **Automation Generation**

**Test Management Environment**
**Automation Layer**

Keywords:
- Keywords-based testing automation
Publication in HP Quality Center
Automation: Implementing modeled actions

Automated translation of Test cases into test scripts
Implementation of automation libraries

```plaintext
function cancel_the_save_of_the_Note(self)
    functionObjective = "Cancel the adding of a note"
    Browser(orcaBrowserIdentificationMedium).Page(matchingPageIdentificationMedium).WebButton("html id:=Cancel").Click
    Browser(orcaBrowserIdentificationMedium).Page(matchingPageIdentificationMedium).Sync
    reporter.ReportEvent micPass, functionObjective, "click on 'Cancel' item"
end function
```
**Project Metrics (1)**

<table>
<thead>
<tr>
<th>Function</th>
<th>Number of tests</th>
<th>Number of steps</th>
<th>Number of checkpoints</th>
<th>Workload (person.day)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Matching</td>
<td>206</td>
<td>2134</td>
<td>7771</td>
<td>30</td>
</tr>
<tr>
<td>Pre-Matching</td>
<td>89</td>
<td>584</td>
<td>2462</td>
<td>13</td>
</tr>
<tr>
<td>Chasing up</td>
<td>183</td>
<td>2076</td>
<td>2876</td>
<td>10,5</td>
</tr>
<tr>
<td>Setup Criteria</td>
<td>88</td>
<td>704</td>
<td>4349</td>
<td>4,5</td>
</tr>
<tr>
<td>Warning</td>
<td>180</td>
<td>1945</td>
<td>3142</td>
<td>11</td>
</tr>
</tbody>
</table>

**Workload:**

- Modeling & test generation → 48%
- Automate key-word and test execution → 52%
## Project Metrics (2)

<table>
<thead>
<tr>
<th>Category</th>
<th>Number of check-point / tests</th>
<th>Number of tests / person.day</th>
<th>Number of check-points / person.day</th>
</tr>
</thead>
<tbody>
<tr>
<td>Matching</td>
<td>37,7</td>
<td>6,8</td>
<td>259</td>
</tr>
<tr>
<td>Pre-Matching</td>
<td>27,6</td>
<td>6,8</td>
<td>189,3</td>
</tr>
<tr>
<td>Chasing up</td>
<td>15,7</td>
<td>17,4</td>
<td>274</td>
</tr>
<tr>
<td>Setup Criteria</td>
<td>49,4</td>
<td>19,6</td>
<td>966</td>
</tr>
<tr>
<td>Warning</td>
<td>17,4</td>
<td>16,4</td>
<td>286</td>
</tr>
<tr>
<td><strong>Average</strong></td>
<td><strong>29,59</strong></td>
<td><strong>13,4</strong></td>
<td><strong>395</strong></td>
</tr>
</tbody>
</table>
Results summary

Test results

- 746 test cases generated with Test Designer 3.2 and fully automated with Quick Test Professional v.8.2
- Complete test suite execution time: 4 hours → makes it possible to test each deliverable from India team (each week in release period to ensure non-regression testing)

Savings:

- Project delivery on time
- Systematic non-regression testing for each release (and after bug corrections)
- Smartesting process covers 95% of functional testing project needs
Agenda

Model-based testing to industrialize the functional testing process

Financial application – Experience report

Summary and discussion
Roles and actions

- **System Matter Expert**
  - Test automation and execution
  - Test data management
  - Results analysis

- **Test Architect**
  - Create and maintain functional requirements
  - Manage domain knowledge

- **Test Designer**
  - Test case design
  - Requirements traceability
  - Ensure functional coverage

- **Tester and Test Automation eng.**
  - Test automation and execution
  - Test data management
  - Results analysis

- **Quality Center / Quick Test Pro**
  - HP
Benefits of model-based testing

Quality: test design starts early in the development cycle
   - The test model is developed in coordination with the requirements analysis and tests are generated iteratively (agility)

Productivity
   - ROI from the third test execution
   - Development of reusable test assets

Improvement of the functional test coverage
   - Systematic coverage of the application expected behavior

Easier and faster maintenance of the test suite
   - At every functional change, only the model have to be updated. Then re-generation of the test suite
THANK YOU

Name: Bruno Legeard
Company: Smartesting
18 rue Alain Savary
25000 Besançon - France
E-Mail: legeard@smartesting.com
Phone: +33 3 81 25 53 63
Web: www.smartesting.com