Experience Report:
Visual Test Design for Test Automation in Agile of a Large-Scale IT Systems

Presented by Elodie Bernard
Statement
Problem and workflows
Problem statement

• Complexity in use of the current MBT approaches
• Introduction of a Lightweight MBT: a visual test design approach
• Simplify the modeling notation
• Facilitate the maintenance of test cases during and through sprints
Workflows statement

Typical MBT modeling approach

Visual test design

Behavioral modeling
Modeling concepts
Example of visual test design with Yest® (from Smartesting)

- Limited number of modeling artifacts
- High modelling capability
- Ability to represent simple as well as complex business processes
Acceptance Test Driven Development
With a visual test design approach
ATDD concept with a visual test design approach

We have new:
- User stories
- Business rules
- Acceptance criteria

We update the graphical representation to be in line with business rules

We generate acceptance scenarios

Start of a new sprint

Features are developed and made accessible on a test environment

End of the sprint

Changes in product backlog occur in the sprint and will be developed

We test the new features
Lessons learned from using a visual test design approach

• Helps to easily update the test assets
• Allows to quickly generate tests that required an update
• Improves communication and work between project stakeholders
Test automation
Overview
Test automation overview

• Keyword-driven-testing
• Java Selenium add-on
• Data set management

Keywords table with Yest

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Class</td>
<td>Keyword</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>com.test.Automation</td>
<td>Login</td>
<td>id</td>
<td>password</td>
</tr>
<tr>
<td>3</td>
<td>com.test.Automation</td>
<td>CheckData</td>
<td>correctOrNot</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>com.test.Automation</td>
<td>OpenThePage</td>
<td>page</td>
<td></td>
</tr>
</tbody>
</table>
Test automation process

- The visual representation of the test
- The abstract scenario and the corresponding automated test script
- The test script in java/Selenium with the use of dataset

The visual representation of the test automation process.

The abstract scenario and the corresponding automated test script:

**Test script**

```
public class test_log_in_ok {
    public class TestLogInOk {
        private final String id;
        private final String password;

        public test_log_in_ok(String id, String password) {
            this.id = id;
            this.password = password;
        }

        @Test
        public void execute() {
            Login(id = Pebl5216r, password = piPyds-4515)
            CheckData(correctOrNot = true)
            OpenThePage(page = HomePage)
        }
    }
}
```

The test script in java/Selenium with the use of dataset.
### Test automation process

#### Data set collection

<table>
<thead>
<tr>
<th>Name of data set</th>
<th>a</th>
<th>password</th>
<th>id</th>
</tr>
</thead>
<tbody>
<tr>
<td>DataSet 1</td>
<td>pPyds-4515</td>
<td>Pcl/c5216r</td>
<td></td>
</tr>
<tr>
<td>DataSet 2</td>
<td>pQjkm-78</td>
<td>Pehgt687r</td>
<td></td>
</tr>
<tr>
<td>DataSet 3</td>
<td>z785_ufigf</td>
<td>Drt451251</td>
<td></td>
</tr>
</tbody>
</table>

#### The scenario to automate

<table>
<thead>
<tr>
<th>Actions</th>
<th>Expected results</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Connect to the application with a correct identifier and a correct password. The identifier and the password are provided.</td>
</tr>
<tr>
<td>2</td>
<td>Check the data. The data are correct.</td>
</tr>
<tr>
<td>3</td>
<td>Access to the home page. Access validated.</td>
</tr>
</tbody>
</table>

#### The scripting details

- **Test campaign selection**
- **Configuration and scripting details**
- **Test script**
  - **Login**
    - `id = id`
    - `password = password`
- **Define common initialization and finalization scripts**
- **Automation script of the selected test action**
  - **Initialization script**
  - **Test script**

**Create a data set collection**

**Link the keywords to the test actions**

**Link data sets to the future automated script**
Lessons learned about the automation process

- Our experiences have shown that:
  - Having a visual link between the manual and automated test assets is beneficial
  - Documentation of automated test cases is directly accessible to all project stakeholders through a visual approach
  - The ability to transcribe test cases easily via the keyword-driven system provides visibility and improves maintenance management
Conclusion and future works
What is new in our approach?

• The approach dramatically simplified the MBT approach
• Maintain a short learning curve and good usability by functional testers
• Be in line with iterative and incremental development approaches
• Supporting both scenario-based and automated test
Futur works

• To experiment the training of functional testers
• Continue to develop an add on in Yest
• To define good practice to facilitate and improve MBT approach, visual test design
• Apply new approaches and methodologies to a group of IT projects