From Test Legacy to Model-Based Testing
How to refactor an existing test repository into an MBT model?

UCAAT 2014
September 16/18 - Munich

Arnaud Bouzy – Bruno Legeard
{arnaud.bouzy, bruno.legeard}@smartesting.com
Agenda

狻 Introduction
   – Our vision of MBT
   – MBT models
   – From test legacy to MBT models motivation
狻 Refactoring test legacy
狻 Case study
狻 Lessons learned
狻 Conclusion
Our Vision of MBT

Model Assets for Automated Test Generation

Targeted domain: Enterprise IT
CertifyIt supports Business Process Modeling using BPMN (Business Process Modeling Notation)
Decision Tables (Business rules)

- Decision tables are used to represent business rules
- For ex: Create a User,
  - User should have valid user name
  - User’s password and retype password should be same
  - User should have first name and last name.
Abstract Test data – Modeling using UML Enumerations

- Equivalence classes are modeled as UML “enumeration classes” and enumeration “literals”
- Enumeration are used as the types of most attributes and parameters
Model-Based Testing - Process

1. **Business Processes**
   - Import

2. **Specifications**
   - Referenced by

3. **Requirements**
   - Import

4. **Business Processes for MBT**
   - Import

5. **Modeling Functional Behavior and Business Rules**
   - Smartesting CertifyIt Test Generation
   - Export

6. **Test Objective Charter**
   - Test Analyst

7. **Validation**: 
   - Scenarios
   - Generated Tests
   - Project Manager

8. **Reporting**: 
   - Project Metrics
   - Progress Tracking
   - Coverage

9. **Test Plan**
   - Test-Requirement traceability

10. **Scripts for Automation**

11. **Test Analyst**

12. **Smartesting CertifyIt**

13. **Business Analyst**

14. **Smartesting**

**Test Management & Test Automation Tools**

- Tester & Test Automation Engineer
From Test Legacy to MBT

**Motivations**

1. Addressing existing testing projects with MBT (not only the new one)
2. Stop creating the MBT Model from scratch: (partial) automated generation of the MBT model from existing test cases
3. MBT as an enabler to refactor and better structure test legacy for maintainability issues

**Challenges**

1. How to factorize “equivalent” test steps in existing test cases?
2. How to increase the abstraction level (without losing test accuracy) of test steps / test cases?
3. How to generate exploitable MBT models from test legacy?
Agenda

- Introduction
- Refactoring test legacy
  - Project process
  - Uploading test cases
  - Refactoring test steps
  - Abstraction
  - Model inference
- Case study
- Lessons learned
- Conclusion
Refactoring test legacy

Full process & tools named Impulse

Needs:
- A mix of people with business and technical skills
- A mix of specific and standardized tools

Some iterations will have to be made to optimize models progressively
Uploading test cases

Objective: import test cases from legacy environment into Impulse tooling

Characteristics:
- Ad hoc engine, based on reusable components
- Technical transformation activity
Refactoring test steps

- **Objective:** transform natural language content into so called ‘Action Words’ later seen as operations in modeling

- **Characteristics:**
  - Business skills needed (identification of common actions whatever the textual presentation is)
  - Effort highly depends on the quality and consistency of legacy text
  - Progressive activity, based on tool refactoring capability.
Abstraction

- **Objective:** Equalize level of abstraction of operations used later in modeling

- **Characteristics:**
  - Business skills needed (level of abstraction must be business consistent)
  - Progressive activity, based on tool refactoring capability.
Model inference

 geniş

- Objective: Transfer assert into modeling and implement business rules
- Characteristics:
  - Business and modeling skills needed
  - Iterations will be probably necessary to produce a productive model
Agenda

- Introduction
- Refactoring test legacy
- Case study
  - From legacy test repository to model
- Lessons learned
- Conclusion
Imported test cases

- Alternative with customer creation - Employed over 18 - category domestic appliance

   - Description
   - Enter description here

   - Tags
   - Add new tag

   - Datatable
   - Definition

   1. « Connect to application »
   2. « Check home page is correctly displayed »
   3. « Define credit duration: 24 months »
   4. « Check credit eligibility »
   5. « Define first name in ‘personal data’ »
   6. « Define last name »
   7. « Define gender »
   8. « Define birth date »
   9. « Save form »
   10. « Check submission is completed »
   11. « Enter contact information and preference (mail, e-mail, phone) »
   12. « Eligible customer »
   13. « Attach contract to customer file »
   14. « Save contract »

© SMARTESTING 2010-2013 – This document is the property of Smartesting. It may not be reproduced in whole or in part
Refactoring

Promote steps as action word

Enter personal information

Steps
1. « Define first name in 'personal data' »
2. « Define last name »
3. « Define gender »
4. « Define birth date »
5. « Save form »
6. « Check submission is completed »
7. « Enter contact information and preference (mail, e-mail, phone) »

Sources
- Alternative with customer creation - Student over 18 - category leisure
- Alternative with customer creation - Student over 18 - category domestic appliance
- Alternative with customer creation - retired over 18 - category domestic appliance
- Alternative with customer creation - Retired over 18 - category leisure
- Alternative with customer creation - Employed over 18 - category leisure
- Alternative with customer creation - Senior Employed - category leisure

Cancel  Promote
Import as model
Ready to go
Agenda

- Introduction
- Refactoring test legacy
- Case study
- Lessons learned
  - Automated and manual tasks
  - Eligible projects
- Conclusion
Automated and manual tasks

- Transfer from one format to another can be automated
- Can be supported … but not fully automated:
  - Identify actions which are semantically equivalent
  - Define the right level of abstraction
  - Implement the business rules in the model
- With these human decisions, you can produce an efficient MBT model
Eligible projects

⇒ Criteria:
   – Test repository level of abstraction
   – Organization of the test repository
   – Availability of people with the required skills

⇒ Type of transformation:
   – Transfer of knowledge
   – Accelerate test design through effective MBT
Agenda

- Introduction
- Refactoring test legacy
- Case study
- Lessons learned
- Conclusion
  - Summary
  - Benefits
Summary

Legacy Test Repository → Test Refactoring → Refactored Test Repository → Model Inference

Test Analyst
Business specialist

Test Analyst

MBT Model

Test Generation

Maintained Test Repository
Benefits

- Accelerates MBT modeling with initial context
- Accelerates business knowledge acquisition for testers
- Provides comparison between actual model and legacy repository
Thank you for your attention