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

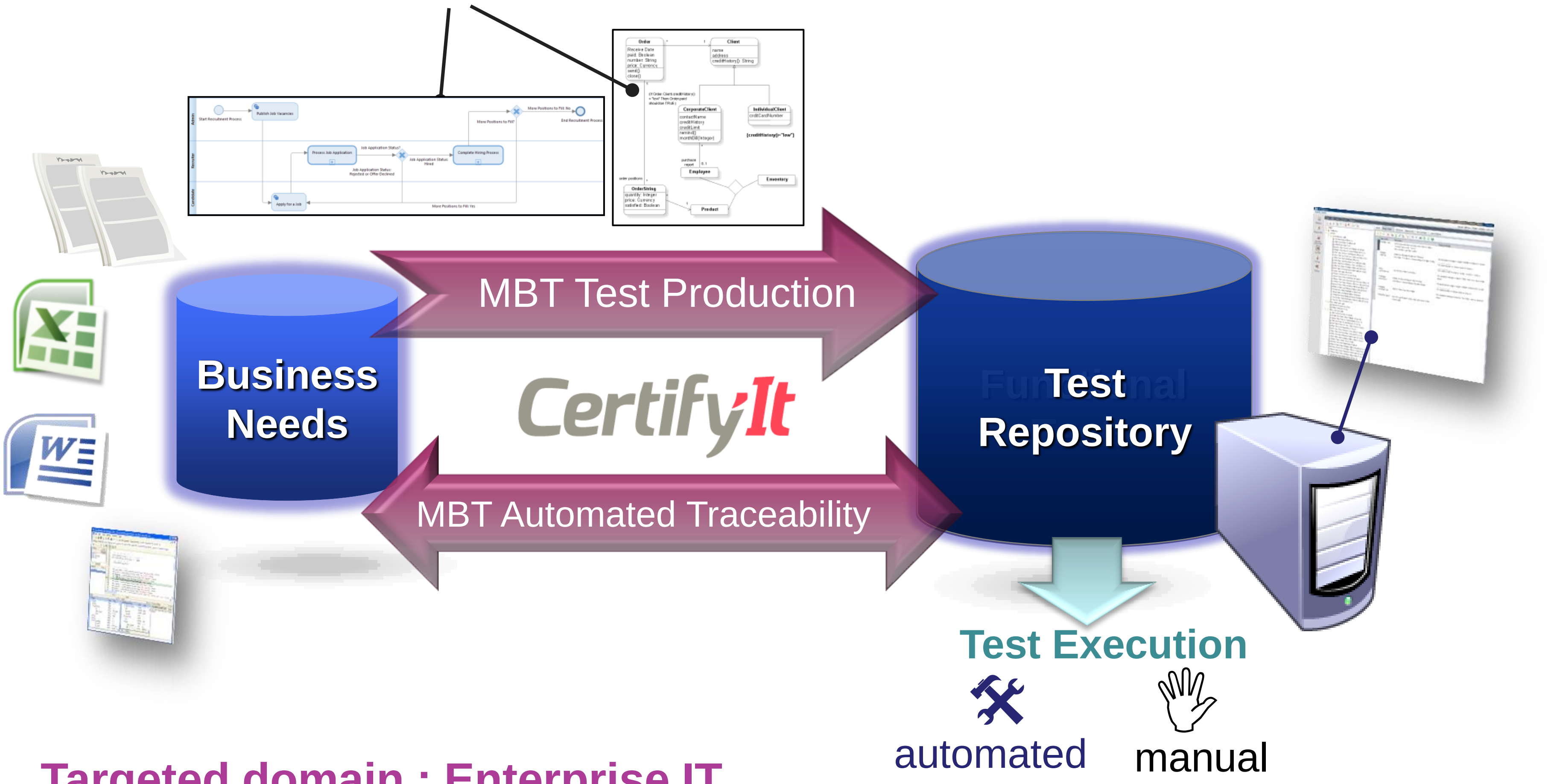
Agenda

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

Our Vision of MBT

smartesting
Optimize your Test Center

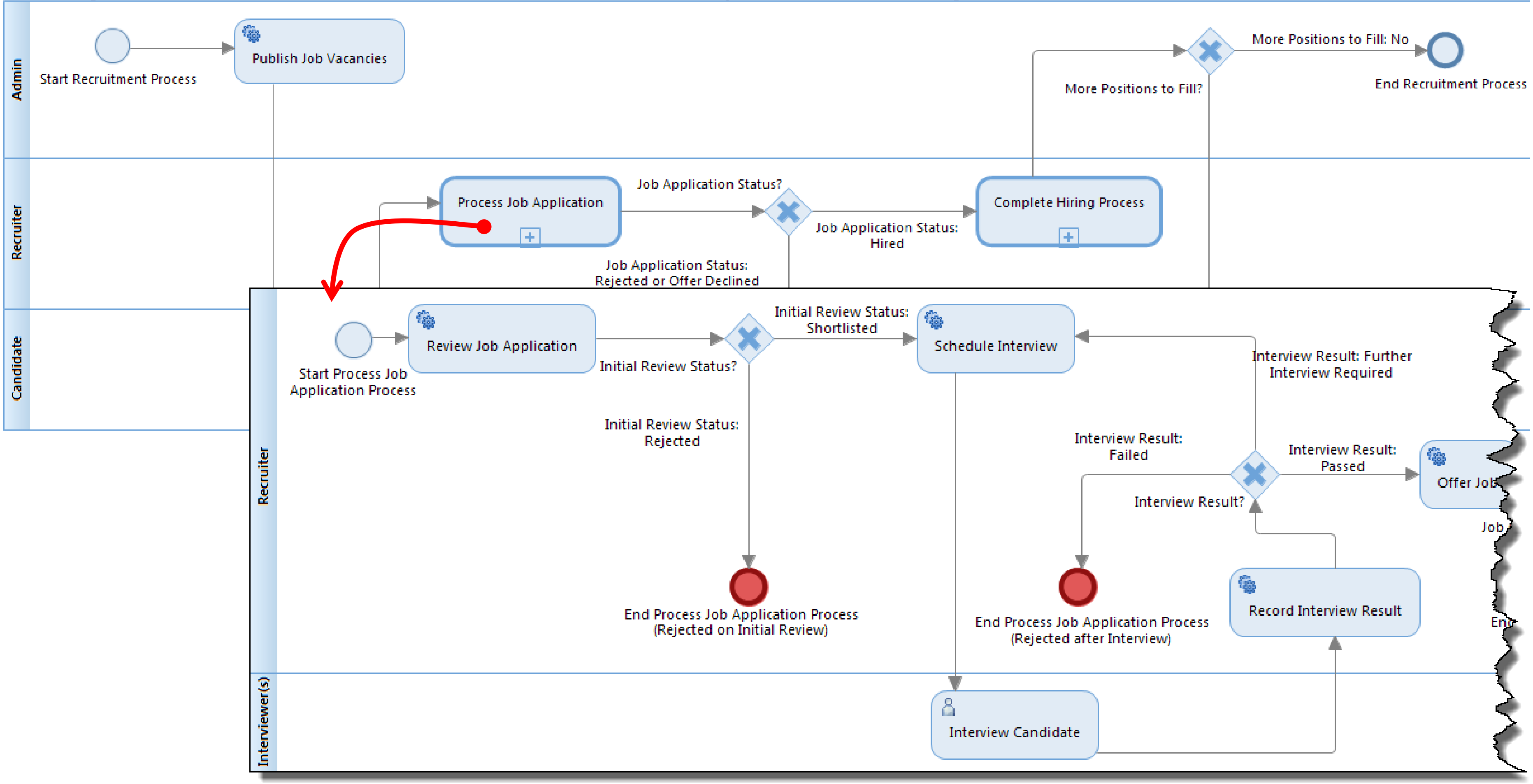
Model Assets for Automated Test Generation



Targeted domain : Enterprise IT

MBT Models

- CertifyIt supports Business Process Modeling using BPMN (Business Process Modeling Notation)


Admin
Start Recruitment Process
Publish Job Vacancies
More Positions to Fill: No
End Recruitment Process
More Positions to Fill?
Recruiter
Process Job Application
Job Application Status?
Complete Hiring Process
Job Application Status: Hired
Job Application Status: Rejected or Offer Declined
Candidate
Recruiter
Start Process Job Application Process
Review Job Application
Initial Review Status?
Initial Review Status: Shortlisted
Schedule Interview
Interview Result: Further Interview Required
Initial Review Status: Rejected
Interview Result: Failed
Interview Result: Passed
Offer Job
Interview Result?
Job
End Process Job Application Process (Rejected on Initial Review)
End Process Job Application Process (Rejected after Interview)
Record Interview Result
En
Interviewer(s)
Interview Candidate

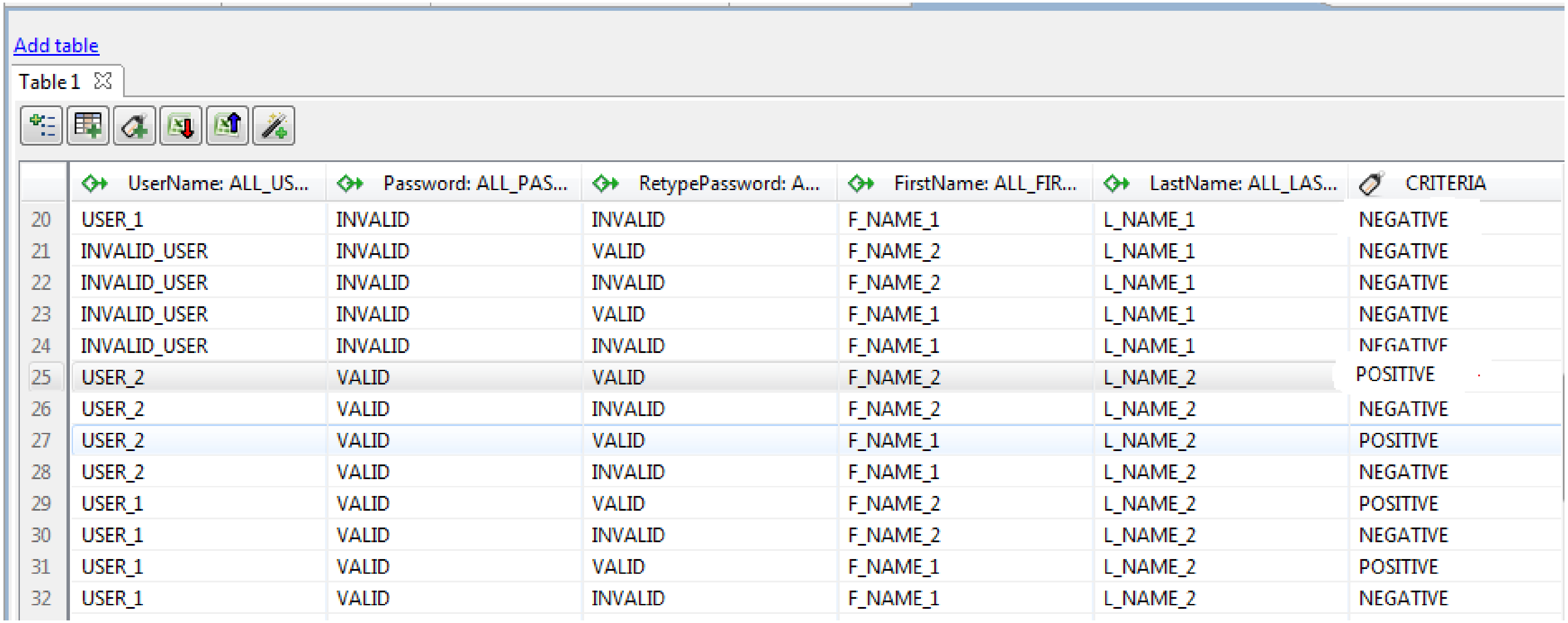
Decision Tables (Business rules)

- **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

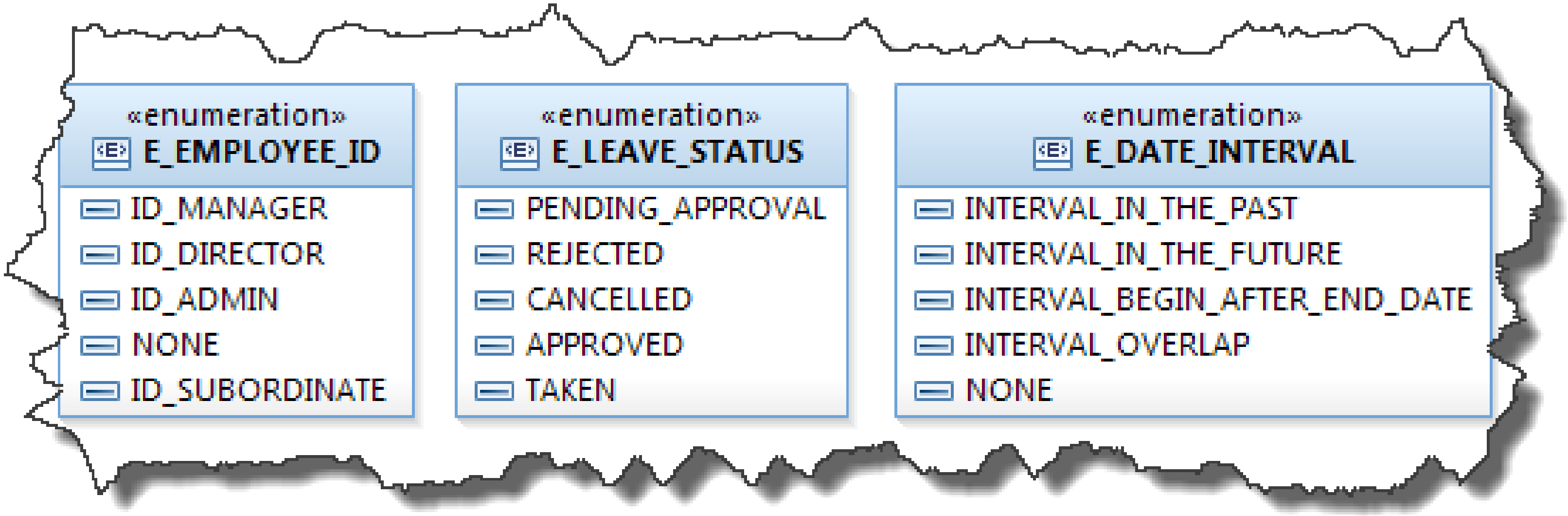
  User should have first name and last name.



	UserName: ALL_US...	Password: ALL_PAS...	RetypePassword: A...	FirstName: ALL_FIR...	LastName: ALL_LAS...	CRITERIA
20	USER_1	INVALID	INVALID	F_NAME_1	L_NAME_1	NEGATIVE
21	INVALID_USER	INVALID	VALID	F_NAME_2	L_NAME_1	NEGATIVE
22	INVALID_USER	INVALID	INVALID	F_NAME_2	L_NAME_1	NEGATIVE
23	INVALID_USER	INVALID	VALID	F_NAME_1	L_NAME_1	NEGATIVE
24	INVALID_USER	INVALID	INVALID	F_NAME_1	L_NAME_1	NEGATIVE
25	USER_2	VALID	VALID	F_NAME_2	L_NAME_2	POSITIVE
26	USER_2	VALID	INVALID	F_NAME_2	L_NAME_2	NEGATIVE
27	USER_2	VALID	VALID	F_NAME_1	L_NAME_2	POSITIVE
28	USER_2	VALID	INVALID	F_NAME_1	L_NAME_2	NEGATIVE
29	USER_1	VALID	VALID	F_NAME_2	L_NAME_2	POSITIVE
30	USER_1	VALID	INVALID	F_NAME_2	L_NAME_2	NEGATIVE
31	USER_1	VALID	VALID	F_NAME_1	L_NAME_2	POSITIVE
32	USER_1	VALID	INVALID	F_NAME_1	L_NAME_2	NEGATIVE

Abstract Test data (Allowed Values)

- 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



«enumeration»
E_EMPLOYEE_ID

- ID_MANAGER
- ID_DIRECTOR
- ID_ADMIN
- NONE
- ID_SUBORDINATE

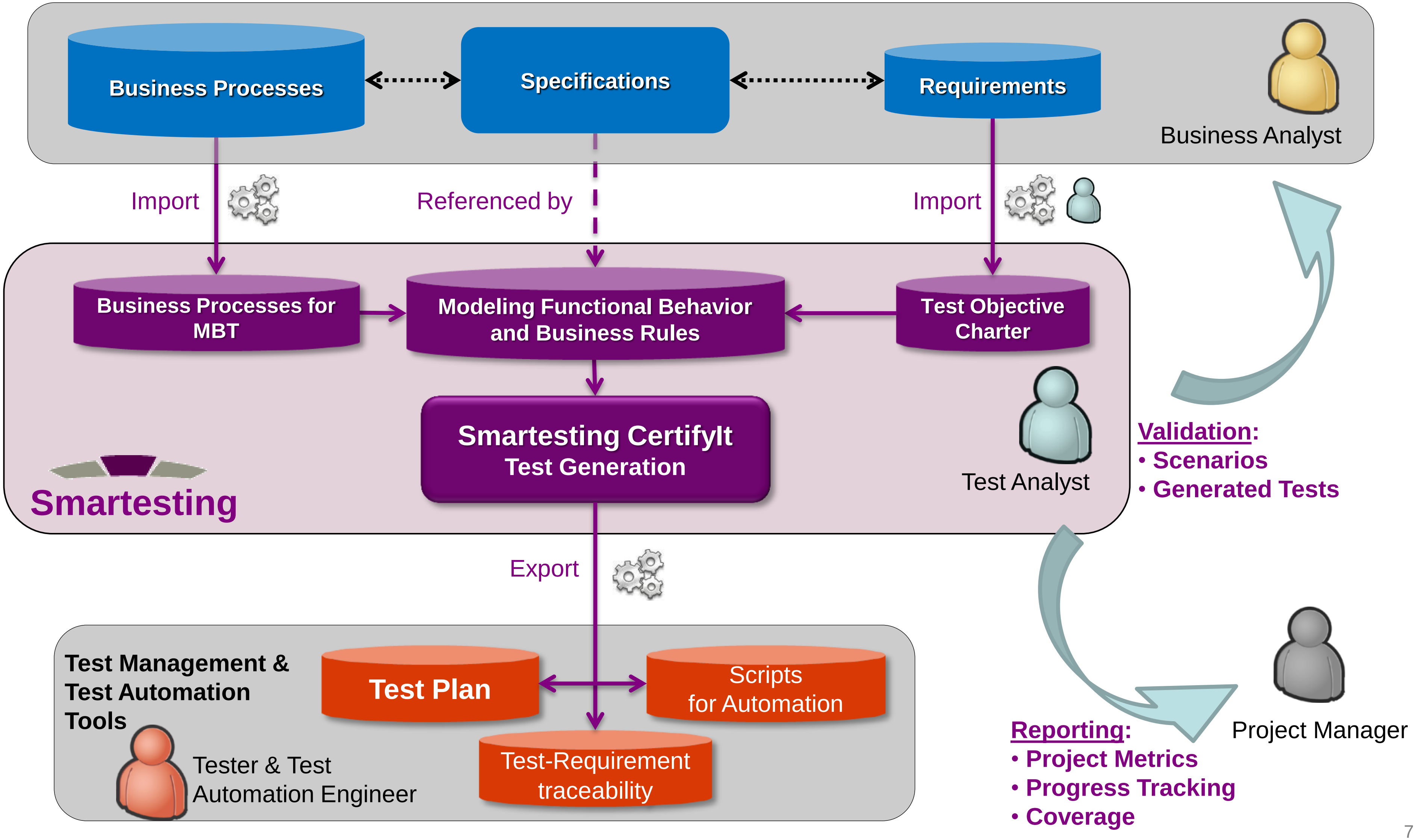
«enumeration»
E_LEAVE_STATUS

- PENDING_APPROVAL
- REJECTED
- CANCELLED
- APPROVED
- TAKEN

«enumeration»
E_DATE_INTERVAL

- INTERVAL_IN_THE_PAST
- INTERVAL_IN_THE_FUTURE
- INTERVAL_BEGIN_AFTER_END_DATE
- INTERVAL_OVERLAP
- NONE

Model-Based Testing - Process



Validation:
- **Scenarios**
- **Generated Tests**

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

From Test Legacy to MBT

➲ Motivations

1. Adressing 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 exisiting test cases
3. MBT as an enabler to refactor and better structure test legacy for maintenability issues

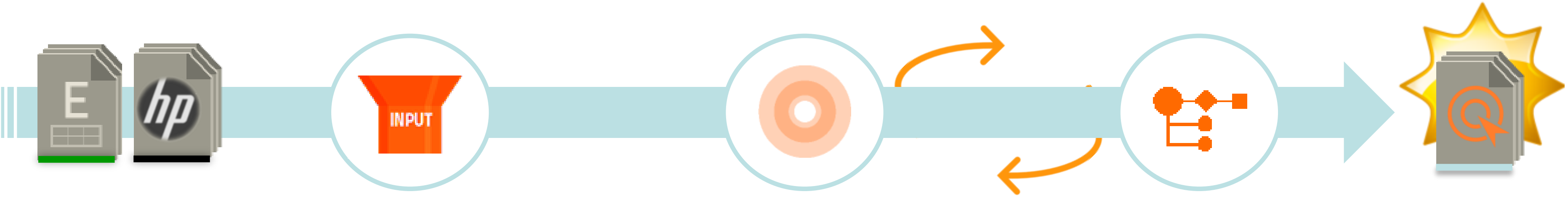
➲ Challenges

1. How to factorize “equivalent” test steps in existing test cases?
2. How to increase the abstraction level (without loosing 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



Existing Repository

1 Repository upload

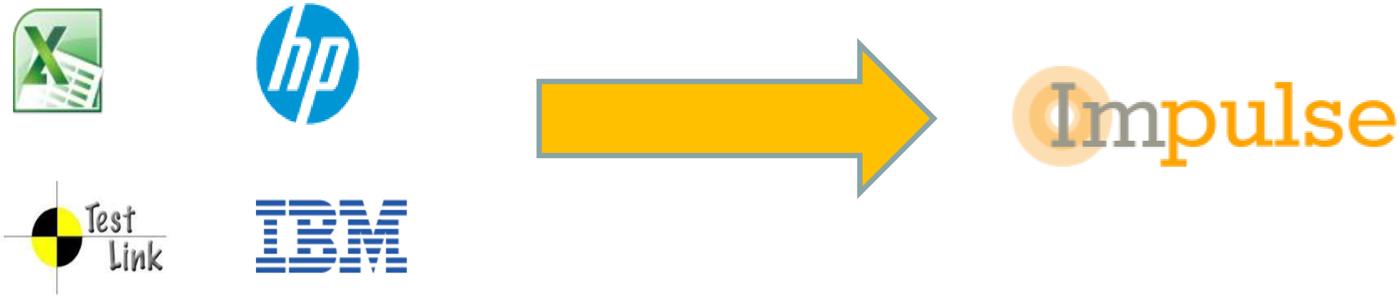
2 Test content optimization

3 Model inference

Usable Model

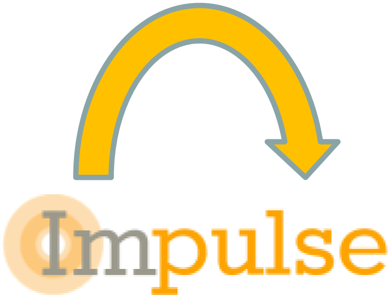
- Full process & tools named Impulse
- Needs:
  - A mix of people with business and technical skills
  - A mix of specific and standardized tools
- Some iterations will have to be made to optimize models progressively

Uploading test cases



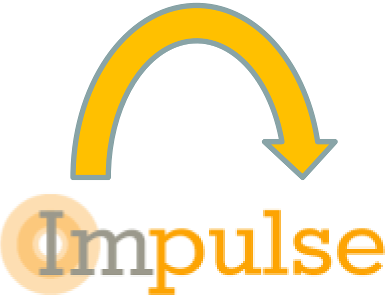
- Objective: import test cases from legacy environment into Impulse tooling
- Characteristics:
  - Ad hoc engine, based on reusable components
  - Technical transformation activity

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.

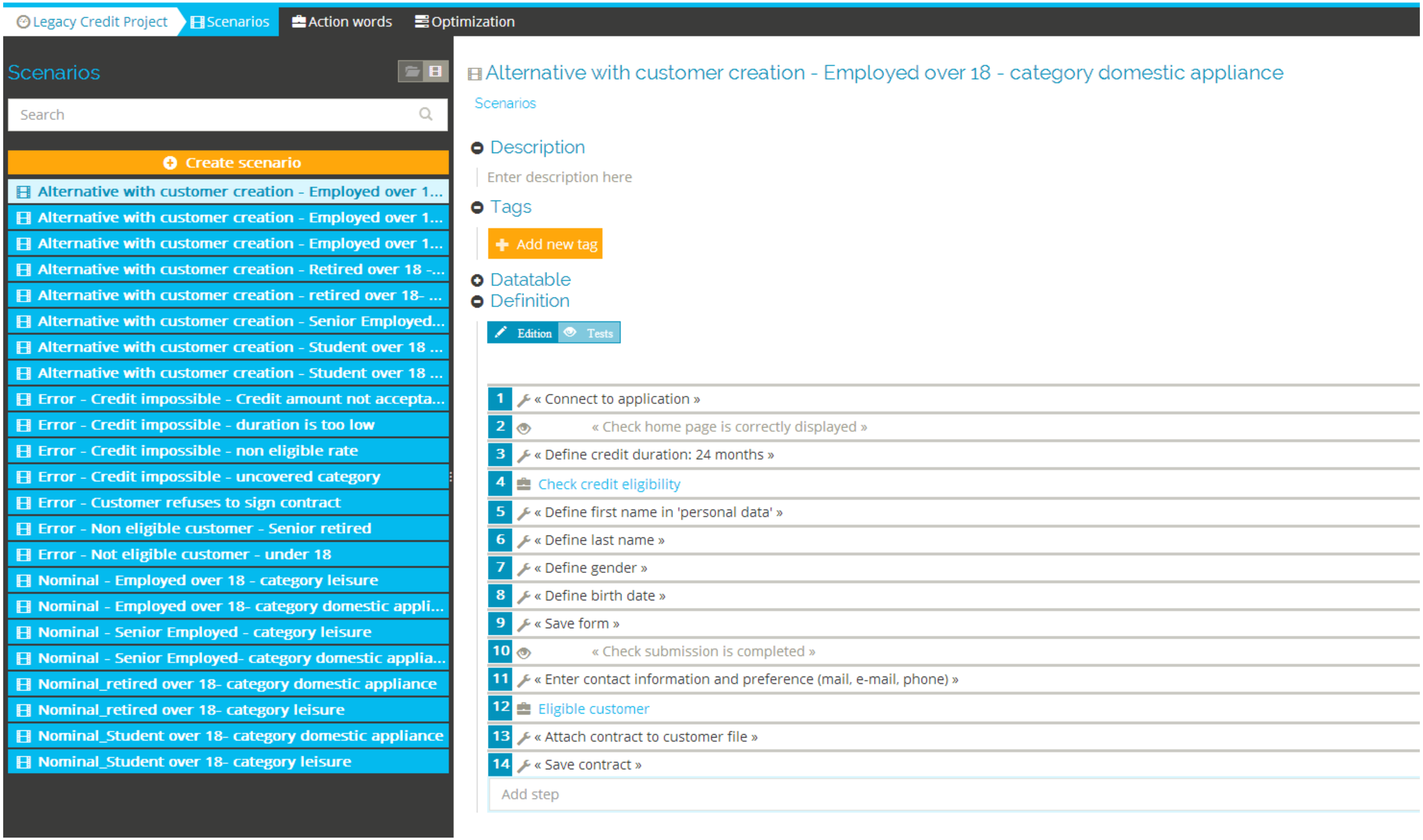
Model inference

- ➲ 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



Scenarios

Search

Create scenario

- Alternative with customer creation - Employed over 1...
- Alternative with customer creation - Employed over 1...
- Alternative with customer creation - Employed over 1...
- Alternative with customer creation - Retired over 18 -...
- Alternative with customer creation - retired over 18- ...
- Alternative with customer creation - Senior Employed...
- Alternative with customer creation - Student over 18 ...
- Alternative with customer creation - Student over 18 ...
- Error - Credit impossible - Credit amount not accepta...
- Error - Credit impossible - duration is too low
- Error - Credit impossible - non eligible rate
- Error - Credit impossible - uncovered category
- Error - Customer refuses to sign contract
- Error - Non eligible customer - Senior retired
- Error - Not eligible customer - under 18
- Nominal - Employed over 18 - category leisure
- Nominal - Employed over 18- category domestic appli...
- Nominal - Senior Employed - category leisure
- Nominal - Senior Employed- category domestic applia...
- Nominal_retired over 18- category domestic appliance
- Nominal_retired over 18- category leisure
- Nominal_Student over 18- category domestic appliance
- Nominal_Student over 18- category leisure

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

Scenarios

Description

Enter description here

Tags

Add new tag

Datatable

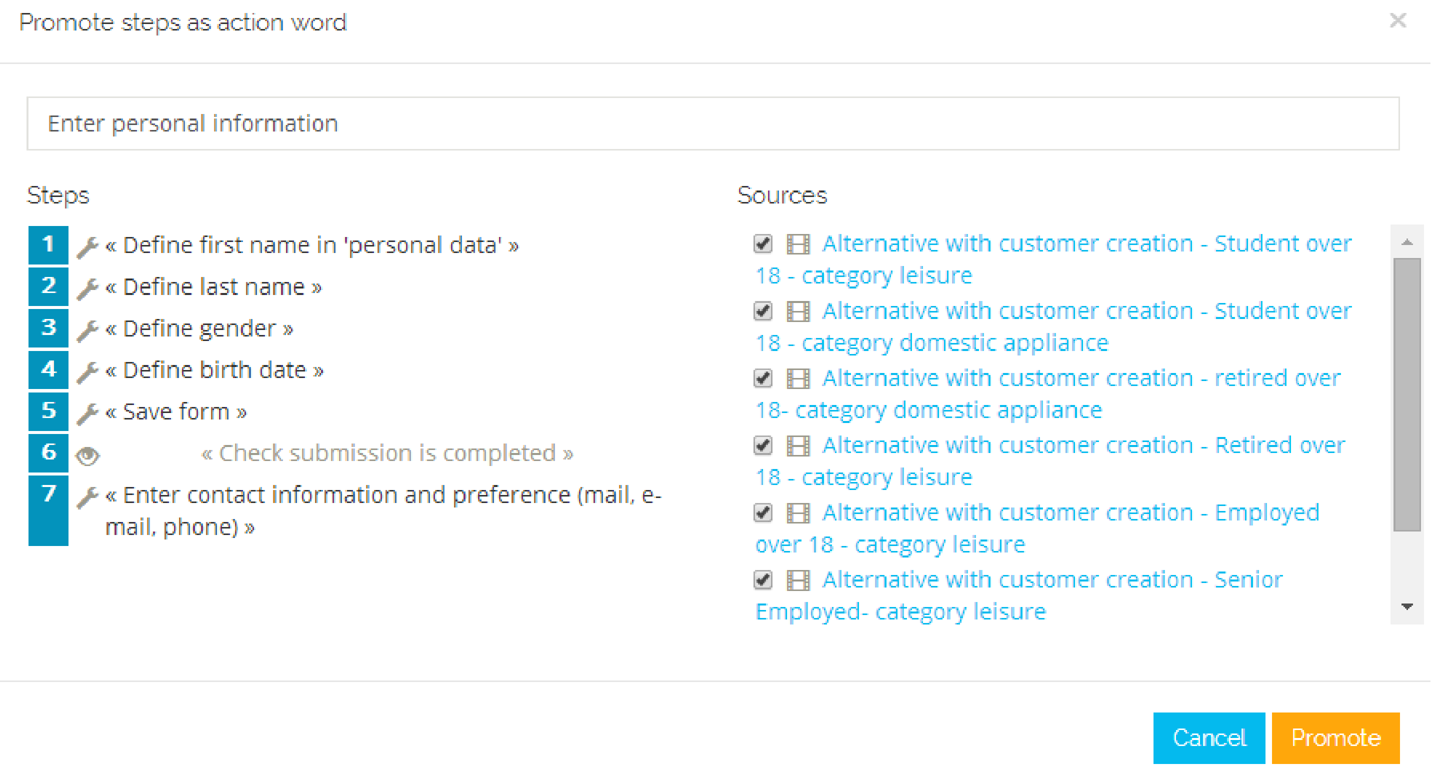
Definition

Edition Tests

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 »

Add step

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

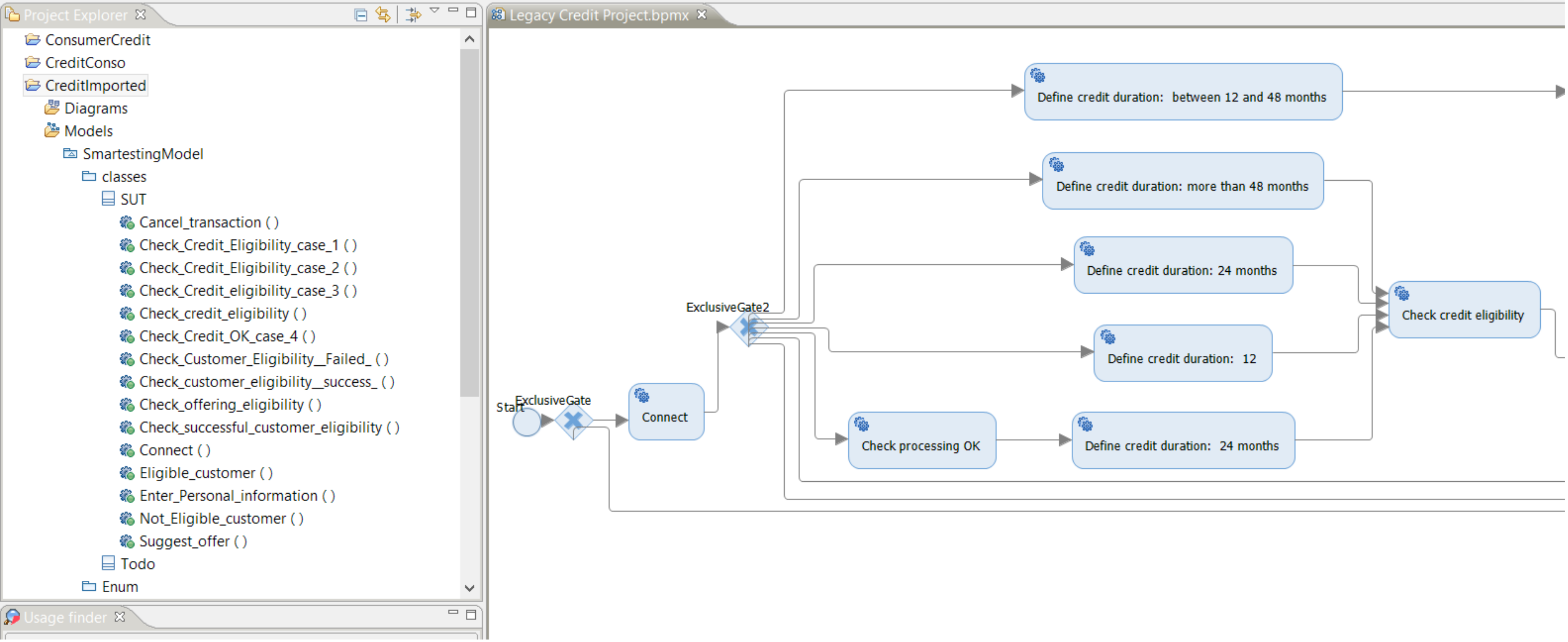
☑ Alternative with customer creation - retired over 18- category domestic appliance

☑ 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

Import as model



Project Explorer ☒

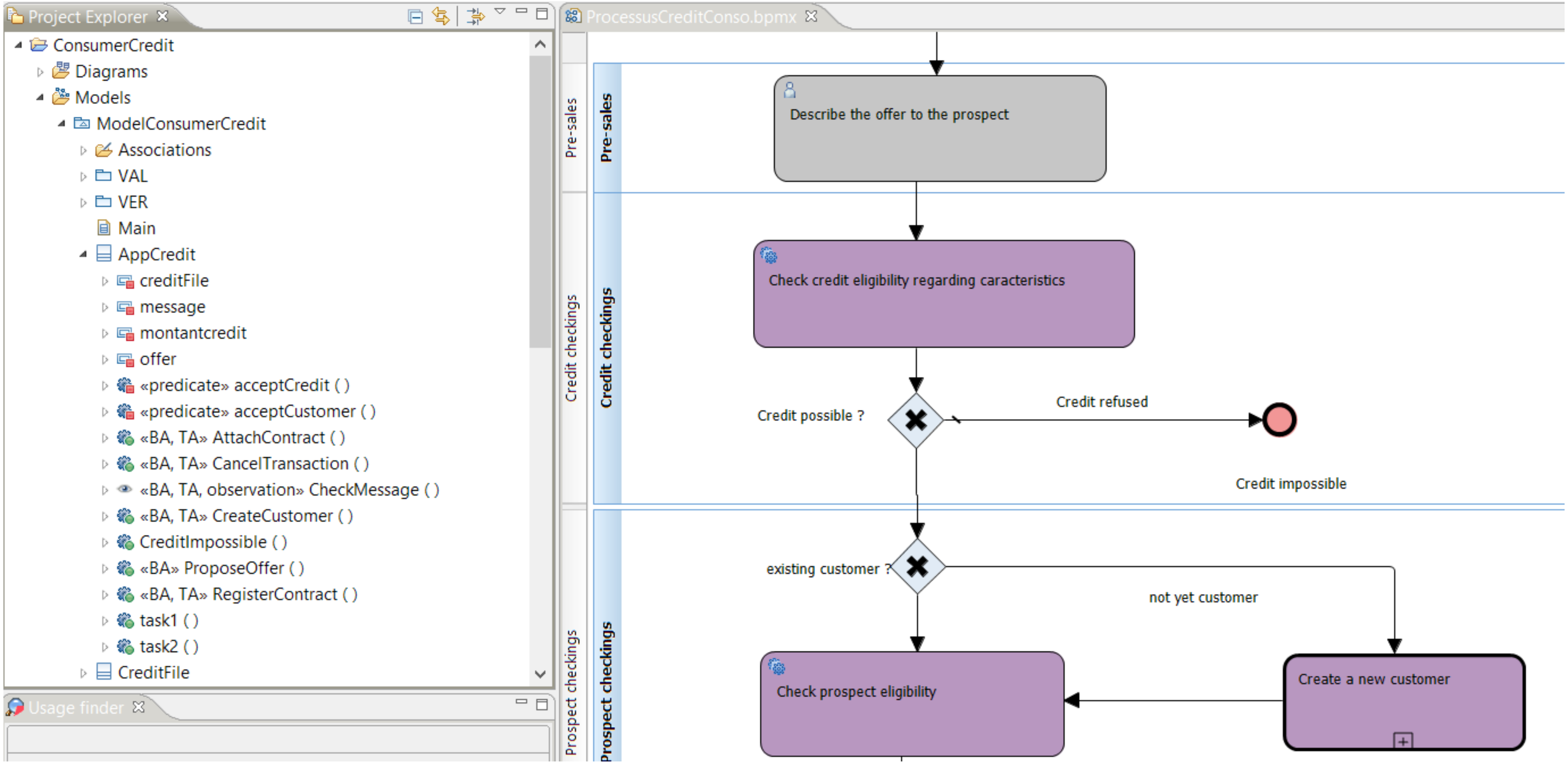
- ConsumerCredit
- CreditConso
- CreditImported
  - Diagrams
  - Models
    - SmartestingModel
      - classes
        - SUT
          - Cancel_transaction ( )
          - Check_Credit_Eligibility_case_1 ( )
          - Check_Credit_Eligibility_case_2 ( )
          - Check_Credit_eligibility_case_3 ( )
          - Check_credit_eligibility ( )
          - Check_Credit_OK_case_4 ( )
          - Check_Customer_Eligibility__Failed_ ( )
          - Check_customer_eligibility__success_ ( )
          - Check_offering_eligibility ( )
          - Check_successful_customer_eligibility ( )
          - Connect ( )
          - Eligible_customer ( )
          - Enter_Personal_information ( )
          - Not_Eligible_customer ( )
          - Suggest_offer ( )
        - Todo
      - Enum

Usage finder ☒

Legacy Credit Project.bpmx ☒

Start
ExclusiveGate
Connect
ExclusiveGate2
Define credit duration: between 12 and 48 months
Define credit duration: more than 48 months
Define credit duration: 24 months
Define credit duration: 12
Check processing OK
Define credit duration: 24 months
Check credit eligibility

Ready to go



- ConsumerCredit
  - Diagrams
  - Models
    - ModelConsumerCredit
      - Associations
      - VAL
      - VER
      - Main
      - AppCredit
        - creditFile
        - message
        - montantcredit
        - offer
        - «predicate» acceptCredit ( )
        - «predicate» acceptCustomer ( )
        - «BA, TA» AttachContract ( )
        - «BA, TA» CancelTransaction ( )
        - «BA, TA, observation» CheckMessage ( )
        - «BA, TA» CreateCustomer ( )
        - CreditImpossible ( )
        - «BA» ProposeOffer ( )
        - «BA, TA» RegisterContract ( )
        - task1 ( )
        - task2 ( )
      - CreditFile

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

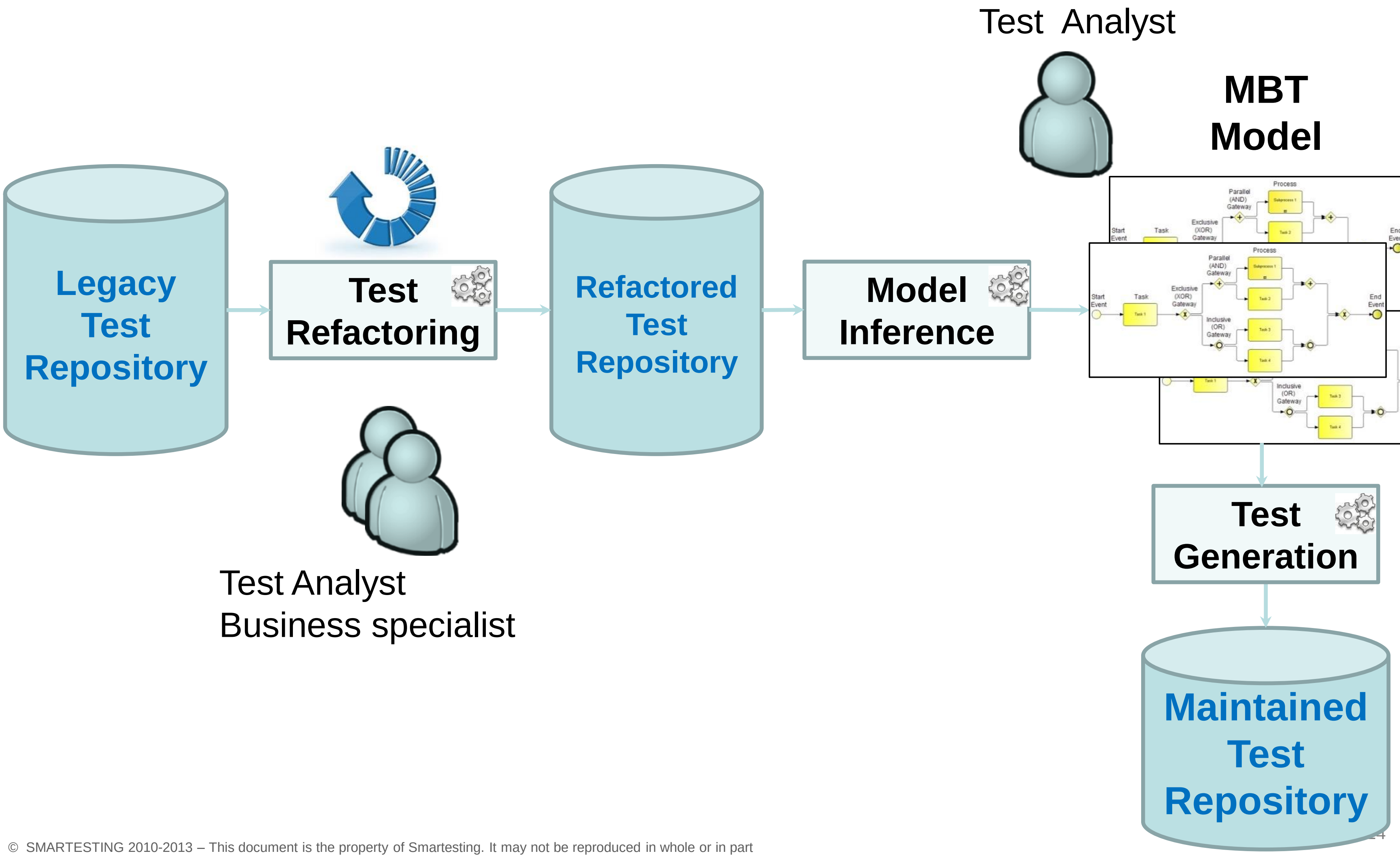
Eligible projects

- ➲ 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



Test Analyst

MBT Model

Legacy Test Repository

Test Refactoring

Refactored Test Repository

Model Inference

Test Generation

Maintained Test Repository

Test Analyst
Business specialist

Benefits

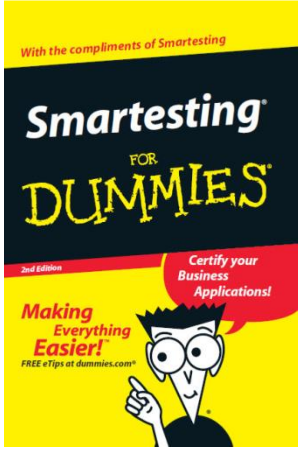
- Accelerates MBT modeling with initial context
- Accelerates business knowledge acquisition for testers
- Provides comparison between actual model and legacy repository

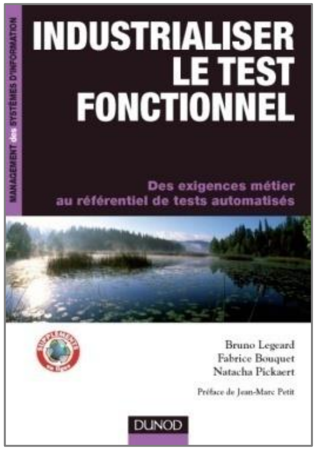
Thank you
for your attention

twitter

twitter.com/smartesting

Linked in
MBT Group


With the compliments of Smartesting
Smartesting®
FOR
DUMMIES®
2nd Edition
Certify your Business Applications!
Making Everything Easier!™
FREE eTips at dummies.com®


MANAGEMENT des SYSTÈMES D'INFORMATION
INDUSTRIALISER LE TEST FONCTIONNEL
Des exigences métier au référentiel de tests automatisés
Bruno Legeard
Fabrice Bouquet
Natacha Pickaert
Préface de Jean-Marc Petit
DUNOD