Mendix Application Test Suite
Expert Webinar - September 30 - 2016

Clyde Waal - Expert Services Consultant
Eduard de Bruijn - Solution Consultant
Agenda

- Introduction to ATS
- Demo ATS
- Adopting ATS
- Roadmap
- Q&A
Introduction to ATS
Mendix Provides the Fast Track for Digital Innovation

Mode 1
Well Defined & Low Rate of Change

Digital Applications

Fluid & High Rate of Change

Mode 2
Traditional Process

Core Systems

Small Cross Functional Team

Rapid Iteration

Unified Innovation Platform

Traditional Development Team

Traditional Process

Traditional Tools and Infrastructure

Defined Requirements

Rate Of Change
Testing in Mode 2

- QA critical success factor in Mode 2
  - Make quality an integral part of the development process
  - Minimize regressions
  - Provide feedback to developers as quickly as possible
  - Maximize efficiency

- This requires a test & performance management framework that is fully embedded in the ALM cycle
  - Simple, easy to use and highly automated
  - Fitted for small cross-functional teams (DIY, Do It Yourself)
  - Boosts the DevOps experience
Mendix Application Test Suite

- A cloud service offered by Mendix in partnership with Mansystems to automate functional testing of Mendix applications.

- Built as add-on to Selenium (in Mendix):
  - Cross-browser functional testing based on keywords
  - Recording of test scenarios
  - Supports scheduling and parallel testing
  - Seamless compatibility with Mendix platform version
Key Benefits

- Reduce cost & effort of testing
  - Less effort spent on testing thanks to automation
  - Less rework for developers thanks to testing early-on in the project
  - Less incidents / tickets after go-live

- Contributes to shorter Time to Market

- Contributes to ‘First Time Right’ delivery

- Leads to higher customer satisfaction
Adopting ATS in the enterprise
Mendix Provides the Fast Track for Digital Innovation

**Mode 1**
Well Defined & Low Rate of Change

- Defined Requirements
- Traditional Development Team
- Traditional Process
- Traditional Tools and Infrastructure

**Digital Applications**

- Rapid Iteration
- Fluid & High Rate of Change
- Unified Innovation Platform

**Core Systems**

- Small Cross Functional Team
Preserving agility when adopting ATS

Mode 1 - Traditional:
• Emphasizing safety & accuracy
• Quality safeguarded by formal testing process
• Traditional testing stages (V-model)
• Developers & testers not in same team

Key success factor: well-implemented process

Mode 2 - Innovation:
• Emphasizing agility & speed
• Quality safeguarded by active product owner
• Development & testing done simultaneously
• Developers & testers in same (Scrum) team

Key success factor: effective day-to-day product ownership

Although ATS supports both modes, you need the appropriate governance to achieve full benefits.
Effective agile product ownership

- Takes active responsibility for testing based on business goals
  - Is in close dialogue with business stakeholder that has a stake in app quality
- Uses agile requirements and specification practices
  - E.g. impact mapping, specification-by-example

**Impact mapping:**

**Given** (precondition) field service engineer is viewing the list of scheduled service locations

**When** (actor+action) field service engineer clicks on one of the service locations

**Then** (observable result) field service engineer will be able to see the service location on a map
Example Impact map

Why?
- **GOAL**
  - Lower field maintenance costs
  - Metric: Lower costs by 25% before end 2016

Who?
- **ACTOR**
  - Service engineers

How?
- **IMPACT**
  - Drive more efficiently
    - Metric: Avg. no of KM per appointment
  - Waste last time searching for instructions
    - Metric: Avg. no. of minutes spent on searching for instructions per appointment
  - Waste less time doing administration
    - Metric: Avg. no. of minutes spent on filling out administration

What?
- **EPIC**
  - Optimized route based on appointments
  - Searchable electronic maintenance instructions
  - Ability to directly order replacement parts on location

Broken up into user stories
Effective agile product ownership

- Takes active responsibility for testing based on business goals
  - Is in close dialogue with business stakeholder that has a stake in app quality
- Uses agile requirements and specification methodologies
  - Impact mapping, user stories, specification-by-example

**Impact mapping:**

**Why?**
- Lower field maintenance costs
  - Metric: Lower costs by 25% before end 2018

**Who?**
- Service engineers

**How?**
- Drive more efficiently
  - Metric: Avg. no of KM per appointment
- Waste last time searching for instructions
  - Metric: Avg. no. of minutes spent on searching for instructions per appointment
- Waste less time doing administration
  - Metric: Avg. no. of minutes spent on filling out administration

**What?**
- Optimized route based on appointments
- Searchable electronic maintenance instructions
- Ability to directly order replacement parts on location

**Given** (precondition)
- Field service engineer is viewing the list of scheduled service locations

**When** (actor+action)
- Field service engineer clicks on one of the service locations

**Then** (observable result)
- Field service engineer will be able to see the service location on a map

**Specification-by-example:**
- Given
  - Field service engineer is viewing the list of scheduled service locations
- When
  - Field service engineer clicks on one of the service locations
- Then
  - Field service engineer will be able to see the service location on a map
**User story:**

As a field service engineer I want to view my scheduled service locations on a map

<table>
<thead>
<tr>
<th>Specification-by-example:</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Given</strong></td>
</tr>
<tr>
<td><strong>When</strong></td>
</tr>
<tr>
<td><strong>Then</strong></td>
</tr>
</tbody>
</table>
Effective agile product ownership

- Takes active responsibility for testing based on business goals
  - Is in close dialogue with business stakeholder that has a stake in app quality
- Uses agile requirements and specification methodologies
  - Impact mapping, user stories, specification-by-example

**Impact mapping:**

- **Why?** (GOAL) Lower field maintenance costs
  - Metric: Lower costs by 25% before end 2018
- **When?** (ACTOR) Service engineers
- **How?** (IMPACT)
  - Drive more efficiently
  - Metric: Avg. no of km per appointment
  - Waste last time searching for instructions
  - Metric: Avg. no. of minutes spent on searching for instructions per appointment
  - Waste less time doing administration
  - Metric: Avg. no. of minutes spent on filling out administration
- **What?** (EPIC)
  - Optimized route based on appointments
  - Searchable electronic maintenance instructions
  - Ability to directly order replacement parts on location

**Specification-by-example:**

<table>
<thead>
<tr>
<th>Given (precondition)</th>
<th>field service engineer is viewing the list of scheduled service locations</th>
</tr>
</thead>
<tbody>
<tr>
<td>When (actor+action)</td>
<td>field service engineer clicks on one of the service locations</td>
</tr>
<tr>
<td>Then (observable result)</td>
<td>field service engineer will be able to see the service location on a map</td>
</tr>
</tbody>
</table>
Getting started with testing your app

1. Define main functional flow
2. For this flow, create a test script
3. Duplicate this test script for re-use
4. Modify these duplications, as needed, by:
   1. Inserting new steps in these duplications
   2. Modify test data used in these duplications
5. Execute test scripts
6. Examine failed test scripts
# Organizing your team for testing

<table>
<thead>
<tr>
<th>SCRUM role</th>
<th>Testing responsibilities</th>
</tr>
</thead>
</table>
| **Scrum Master**       | • Ensure registration of the project in ATS  
                          • ATS account management                                                               |
| **Product Owner**      | • Define hierarchy for test scripts  
                          • Examine test outcomes with business  
                          • Schedule automatic execution of test scripts for regression purposes                |
| **Team member (junior)** | • Define individual test scripts  
                               • Manual test execution  
                               • Evaluate test results and report to product owner                                   |
| **Team member (senior)** | • All responsibilities of a junior member defined above  
                                  • Create actions for custom widgets                                                   |
Roadmap
ATS roadmap

July
- Usability
- Recording test scripts
August
- Import/export test data
- Improved dashboard and reporting
September
- Custom widget actions
- Actions Howto
October
November
December
- Public ATS multi-tenant
Thank You