PASSION FOR TESTING

Shift Left: Test Automation in Agile Environments
Unlock the Secrets
Shift Left - Test Automation Agile Environments

AGENDA

• Today we will Cover
  • Challenges around Agile Testing
  • Shift-Left – Specification by Example
  • The Agile test-pyramide
  • What are the benefits of Shift-Left & early automation?
    • Agile customers - case studies
    • How do developers need to contribute?

• Summary

• Q&A
Agile Methods are on the Rise

More than 2/3 of new software developments are now being set up with an agile method.

...here they want to get to ...

**Waterfall**

...here is where they are today

**Agile**

© Copyright by TRICENTIS®, all rights reserved
Costs increase constantly (often linearly) over the duration of the project.

The customer will not see the expected value until much later on.
Costs and experienced value | conventional (II)

Project aborted due to shortage of resources or the project scope needs to be changed significantly

And what now?
Short “sprints” generate **deliverable versions early on.**

The most important thing first – strict **value/risk-oriented procedure**
TOSCA | Waterfall-View

- Requirements
- Test Case Design
- Manual Test Cases
- Automated Test Cases
- Execution Lists
- Reporting

© Copyright by TRICENTIS®, all rights reserved 7
Agile Development

Shift from the **should** to the **must**

Shift **to the left**

**Specification by Example**

**Early** test automation
Shift to the Left!
Specification by Example
When we were children we drew…
V-Model

- Requirement definition
- Specification
- Techn. design
- Implementation
- Unit/Module testing
- (System) integration testing
- Acceptance testing (UAT)
From the waterfall to the agile method…

The size of the birds changes!

Birds remain birds!

Many small birds are considerably more maneuverable (more flexible) than large ones.

Each of them can fly on its own!
Specification by Example - Key to Success

- Requirement definition
- Specification
- Techn. design
- Implementation
- Specification by Example
- Acceptance testing (UAT)
- (System) integration testing
- Unit/Module testing
Test Case Design is …

- sufficiently substantiated
- flexible if changes occur
- lean and very cost-effective (compared with the specification of test cases)
Shift to the Left!
Early Test Automation
Practice: **main focus of efforts is on the E2E test!**

- Poor cost efficiency
- Limited test coverage

- Availability of all systems in the test?
- Consistent test data in all systems?
Objective: reducing the number of E2E tests

**“Slow and Brittle: Replacing End-to-End Testing”**

... *Our goal for this workshop is to generate serious, practical alternatives to end-to-end testing ...*
Test Pyramid – Target State

- Powerful, automated Unit und Acceptance Tests (system tests)
  - Test access: non-UI
- Tests via GUI only for:
  - Testing the display on the screen
  - Testing the connection to the underlying layers
  - User Acceptance Testing
E2E – System internal und Crossover Tests

System internal Tests

E2E Tests

System_1  ...  ...  ...  System_n
Testing up to the Interfaces

- System-internal tests extended **right to the interfaces**
- Coupled systems are „**stubbed**“ or „**mocked**“
  - Share of functionality that can be tested without coupled systems (= system-internally) is sufficiently extended
Testsequence UI and nonUI - Example

Example: Create Customer, Customer Project TRICENTIS, via Browser-UI

Example: Create Customer, Customer Project TRICENTIS, via WebService

# of test steps: Save 75%!

Maintenance: Save 75%!

Test duration: Save 90%!
Test streams & Test automation | traditional

Functionality to be tested

Releasetest

Regressiontest

... One-time (manually)
... Test of new functionality with regressive error potential (manually)
... Regressiontest (automated)
Regression test – automated:
- High automation degree from the start
- Dominated by nonUI access

Functionality to be tested

- … One-time (manually)
- … Test of new functionality with regressive error potential (manually)
- … Regression test (automated)
Testing Challenges:

- Move towards Agile software development
- Cost and scheduling pressure
- All previous attempts at automation have failed

Solution:
Model-based test automation.

Results:

- Core regression suite implemented in **less than 3 months**
  - 1,009 test cases
  - 33,797 test actions
  - 35 end-to-end business process scenarios, multiple day scenarios
  - 2,511 verification points
- Execution effort down by **80%** (from 535 h to less than 100 h).
- Annual cost reduction exceeds **50%**.

GBST Shares
Settles close to 50% of all trades on the Australian Stock Market, ~ AU$ 2.5 billion every day
Testing Challenges:

- Agile development with continuous integration
- Very short development cycles
- Multiple interfaces (Browser, Smartphone, rich client)

Solution:

Development provides test-accessible nonUI interface (WebServices, REST) for every single use-case!

... enabling model-based test automation, early nonUI test automation

German Payback System vendor, hosting payback systems for Lufthansa, Deutsche Bahn, etc.
How Developers need to support …

Agile testing requires …

- early
- automated
- nonUI

… testing.

Early automation won‘t work via UIs!
Agile developers need to provide …

… appropriate hooks:
- Comprehensive, public interface for each use-case …
- … allowing to fully test business logic …
- … avoiding „noisy“ communication
- … being maintained!
Summary

- Agile Testing: Volatile Backlog, Short Sprints, Value Management and Autonomous Teams
- Quality remains the Number 1 Concern
- Quality results from testing the right things and improving test coverage and is held back by the lack of testing efficiency
- Shift Left: establish specification by example as mandatory part of your user stories
- Shift Left: apply testing literally from Day 1 of the project and nonUI test automation is the KEY
- Agile developers need to provide and maintain nonUI interfaces allowing for gapless testing of business logic.
Thank You!

WWW.GOTOSCA.COM