Behavior Driven Development and Testing of Qt and QML applications

Qt Developer Days 2014

by Reginald Stadlbauer
About me

- Name: Reginald Stadlbauer
- Company: froglogic GmbH
- Position: co-founder and CEO
- Worked as Software Engineer at Trolltech and the KDE project
About froglogic

- HQ: Hamburg
- Founded: 2003
- US presence since 2008
- Product focus on Squish
  - Squish GUI Tester (Cross-Platform/Cross-Technology GUI Test Automation)
  - Squish Coco (C, C++ and C# Code Coverage)
- More than 3,000 customers world-wide
Overview

- What is BDD and TDD
- Automating a Behavior Driven Test
- Live Demo & Conclusion
What is BDD / BDT?

“BDD is a second-generation, outside-in, pull-based, multiple-stakeholder, multiple-scale, high-automation, agile methodology. It describes a cycle of interactions with well-defined outputs, resulting in the delivery of working, tested software that matters.” - Dan North


OR...
What is BDD / BDT?

- Based on **Test Driven Development**
  - Write (failing) test
  - Implement feature until test passes
  - Unit-Test level granularity (inside-out)

- But
  - Focus on application's behavior and specification
  - Description in a human-readable DSL (e.g. Gherkin)
  - Less focus on implementation details
Versatile usage of Feature Files

- User story / feature specification
- Communicate with customer / users
- Documentation of acceptance test
- Sequence to walk through for manual tests
- Storyboard for automation of tests
Why BDD/BDT

- “Test first” development on a higher level
- Clearly separate test logic from implementation
- Allow non-programmers to define features & tests
- Have a common, single language two communicate
What is BDD / BDT – Unit Conversion

![QML Unit Converter](image)

The first conversion from 378.9 cm to 3.789 km results in a success.

The second conversion from 378.9 cm to 3.789 g results in an error: "Unit type mismatch: Length vs. Weight."
**Feature:** Valid conversion

**Scenario:** Convert meter in centimeter

- **Given** the Unit Converter is running
- **When** I enter 378.9
- **And** choose to convert from "m"
- **And** choose to convert to "cm"
- **And** click Convert
- **Then** 37890 should be displayed in the result field
What is BDD / BDT – Unit Conversion

Feature: Invalid conversion

Scenario: Mix units
  Given the Unit Converter is running
  When I enter 378.9
  And choose to convert from "m"
  And choose to convert to "kg"
  And click Convert
  Then ERROR should be displayed in the result field
  And "Unit type mismatch: Length vs. Weight." should be displayed in red
What is BDD / BDT – The Process

Write a failing Feature Test

Make Feature Test pass

Implement/Refactor Code

Test Manually Automated
Automating a Behavior Driven Test

- **Requirements**
  - BDT framework
  - testing framework (unit, GUI, ...)
  - Glue between both
BDT Framework – Generate Skeletons

- Parse feature files
- Generate step definition skeletons (functions and annotations) in preferred language

**Test.feature**

Feature: Valid conversion

Scenario: Convert meter in centimeter
  Given the Unit Converter is running
  When I enter 378.9
  ....

**Test.py**

```python
@Step("Given the Unit Converter is running")
def step(context):
    test.warning("Implement me")

@Step("When I enter 378.9")
def step(context):
    test.warning("Implement me")
```
BDT Framework – Run Feature Files

- Parse feature files
- Execute feature files by mapping to steps to step definitions (functions)
- Reporting

```python
@Step("Given the Unit Converter is running")
def step(context):
    [...]

@Step("When I enter 378.9")
def step(context):
    [...]
```

```feature
Feature: Valid conversion

Scenario: Convert meter in centimeter
  Given the Unit Converter is running
  When I enter 378.9
  ....
```
Testing framework

- Support the specific programming language
- Support the specific UI technology of AUT
- Scripting support / integration options
- Tooling for convenient test creation, maintenance and debugging

Test.py

```python
@Step("Given the Unit Converter is running")
def step(context):
    startApplication("UnitConverter")

@Step("When I enter 378.9")
def step(context):
    click("FromField")
    typeText("378.9")
```
Integrating BDT and test frameworks

- Need to “talk the same language”
- Reporting
- Debugging
BDT Frameworks

- Cucumber
- JBehave
- Behave
- SpecFlow
- RSpec
- Lettuce
- Squish GUI Tester
- ...

Unit Testing Frameworks

- QtTestLib
- Qt Quick Test
- CppUnit
- GoogleTest
- xUnit
- NUnit
- JUnit
- ...
GUI Testing Frameworks

- Squish GUI Tester
- HP QTP
- Rational Functional Tester
- Selenium
- ...
Live Demos

- QML Unit Converter
  - Non-GUI test (backend)
  - GUI test (frontend)
Questions? Visit our booth or email sales@froglogic.com
Free and supported trial of Squish at http://www.froglogic.com/evaluate
About Squish GUI Tester

- Cross-Platform / Cross-GUI-Technology Test Automation
  - Windows, Linux, Mac OS X, Unix, RTOSes, Mobile
  - Java (Swing/AWT, SWT/RCP, JavaFx), Qt/QML/QtQuick, Web, MFC, WinForms, WPF, iOS, Cocoa, Carbon, Android, Tk, Flex, ...
- Object-based GUI object identification
- Record & replay
- Powerful scripting (JavaScript, Python, Ruby, Tcl, Perl)
- Eclipse-based IDE
- Built-in BDD framework and support

- Batch-testing via command-line tools
- Remote/distributed testing architecture
- Integrations: Microsoft ALM, HP QC/ALM, Rational RQM, Seapine TCM, SpiraTest, MKS, XStudio, Jenkins, Hudson, TeamCity, Bamboo, Robot Framework, JUnit, Maven, …