Property Based Testing Hands on

ACCU 2016

Bristol, April 2016

Marc Evers – marc@qwan.eu

Rob Westgeest – rob@qwan.eu

Willem van den Ende – willem@qwan.eu

willem@qwan.eu

rob@qwan.eu

marc@qwan.eu

QWAN

Quality Without A Name

www.qwan.eu

What we do

Increase business value from software development

and

helping others do it

through

Mentoring

Training

Developing

Organizing conferences

Our learning vision

Learn by doing

Session Objectives

Learn and **explore** property based testing, by doing & reflecting

Learn what property based testing is, how it can be useful and how it influences your design

Property Based Testing

- Roots: QuickCheck in Haskell
- Property based testing:
  - define properties (invariants) of code
  - generate lots of random input data
  - run code with input, verify invariants hold for all inputs
  - Formal Methods, applied!

Contrast with example based testing:
define small number of samples that express intent

Property Based Testing

- Useful/promising for
  - characterization testing, understand existing libs
  - testing validation logic
  - testing mappings & adapters
  - finding pesky corner cases
  - when example based testing gets repetitive
  - when your examples don’t express intent very well, no matter how hard you try
- We're still learning about it

Short demo

Workshop structure

- Self-contained exercises in Haskell and JavaScript (choose what you like)
- Exercises start “painting by numbers” and get more exploratory along the way
- Work in pairs, at your own pace
- We'll walk around for questions and discussion

Environment & exercises

github.com/qwaneu/property-based-tutorial

Environment: Cyber-dojo

Get Tutorial from repo:

- exercises/js/property-based_js.pdf (JavaScript)
- exercises/hs/quickcheck-workbook.pdf (Haskell)

Cyber-Dojo

- Get on the WIFI: qwan / engineering
- Go to 192.168.5.4
- Enter a practice session
- Enter the 6 digit hex code
  - Javascript / JSVerify: @@@
  - Haskell / Quickcheck: @@@
- Choose 'start'

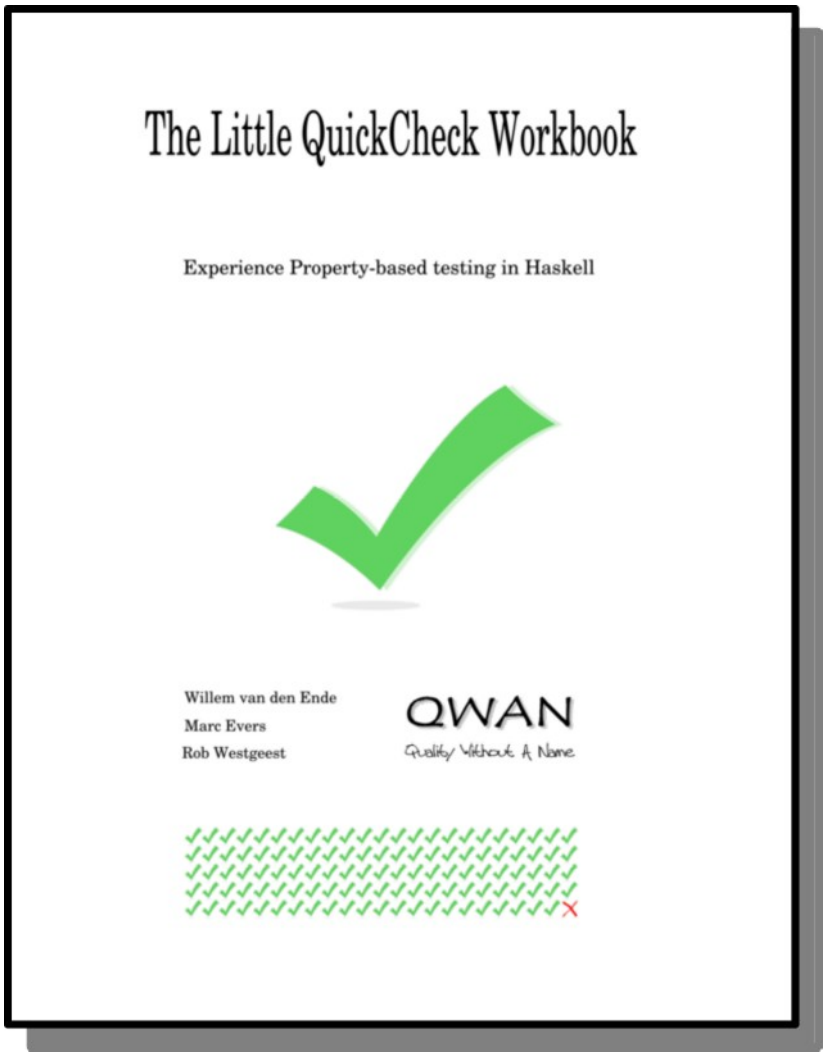
Go!

github.com/qwaneu/property-based-tutorial

github.com/jsverify/jsverify

Get our book from Leanpub

- leanpub.com/quickcheckworkbook
- Discount code: ACCU2016



What we've found

- Make invariants explicit: constraints & edge cases
- Defining generators: preconditions
- Test order seems to matter
- Let test fail first (like TDD)
- TDD-as-if-you-meant-it approach useful
- Baby steps!

What we've found

- 2 kinds of properties:
  - Consistent state: constraining behaviour
  - Invariants on state changes: drive behaviour
- Strong typing can replace some properties
- *Mistake proofing* your code: Primitive Obsession hurts

Issues & Puzzles

- How do you know your test is correct?
- What to do with a property that reflects the actual production code?
- Do properties belong to production code? And generators?

More information

Property based testing is available in (almost) all your favourite programming languages!

- Diamond Kata - TDD with only Property-Based Tests - www.natpryce.com/articles/000807.html
- Uncovering Defects with Property-Based Testing Using ScalaCheck – boldradius.com/blog-post/VVNb0igAANCifV5z/uncovering-defects-with-property-based-testing-using-scalacheck
- John Hughes - Testing the Hard Stuff and Staying Sane (YouTube video)
- Choosing properties for property-based testing fsharpforfunandprofit.com/posts/property-based-testing-2/

Keep an eye on:

github.com/qwaneu/property-based-tutorial