

# TESTING PROPERTIES WITH QUICKCHECK

@steshaw

# “CHOOSE YOUR OWN ADVENTURE”

Lightning talk, [Brisbane Software Testers](#)

# QUICK OVERVIEW

- Test properties of your programs.
- Randomised test inputs.

Demo (basics)

# APPEAL TO AUTHORITY

*Better than unit tests—Michael Nygard  
(2013)*

# TESTING UI

- Choose your own adventure.
- randomised user actions.

Demo (arbitrary)

# TESTING CONCURRENT SYSTEMS

... or ...

- Concurrent
- Testing
- Systems

:P

# CONCURRENT SYSTEMS (2)

- Google “[Erlang QuviQ](#)” to find reports and talks.
- Distributed Systems are Concurrency Systems that span multiple nodes.
  - i.e. these ideas apply to Distributed Systems.
  - i.e. web

# REFERENCES

- Original QuickCheck paper from 2000—Claessen and Hughes (PDF)
- QuickCheck homepage and manual
- Testing and quality assurance—Chapter 11 of Real World Haskell
- Introduction to QuickCheck on Haskell Wiki
- <https://github.com/steshaw/quickcheck-examples>



# REFERENCES (2)

## TESTING CONCURRENT SYSTEMS

- Erlang QuickCheck
- Testing Erlang Concurrency with QuickCheck (PDF)
- Finding Race Conditions in Erlang with QuickCheck and PULSE (PDF)

# REFERENCES (3)

## TESTING DISTRIBUTED SYSTEMS

- [Testing Telecoms Software with Quviq QuickCheck \(PDF\)](#)
- [Modeling eventually consistent databases with QuickCheck \(Video\)](#), Jon Meredith, Basho.

# SEE ALSO

- [SmallCheck](#) for exhaustive checking of your properties (rather than QuickCheck style random samples)
- [Comparison of QuickCheck and SmallCheck](#)