

QuickCheck for C Users

This course spans four days, during which participants will learn how to build software models in Erlang, QuickCheck's modelling language, how to write QuickCheck specifications for various kinds of system in an effective way, and how to connect QuickCheck to a C API under test.

... Day 1

Modelling Software in Erlang I

QuickCheck uses Erlang to model the system-under-test in a declarative way. This module introduces the most important Erlang features, and shows how to use them for modelling.

Properties and generators

Introduces the basic elements that QuickCheck adds to Erlang: test data generators, and the properties that use them. Shows how to run tests, and collect statistics on the results. QuickCheck's rich API for constructing generators is described.

... Day 2

Modelling Software in Erlang II

Teaches some of the more advanced features of Erlang, and their applications to modelling systems under test.

Symbolic Test Cases

In many cases QuickCheck test cases are symbolic: this module teaches how to construct and use symbolic test cases--and how to know when to do so.

Positive and Negative Testing

Testing usually starts with positive cases; this module teaches participants to build negative QuickCheck testing on top of positive testing, reusing the work already done. Covers QuickCheck's fault injection API.

... Day 3

Testing Stateful Systems

Introduces QuickCheck's state machine formalism, which allows operations to be specified in terms of state transitions, and pre- and post-conditions, then finds minimal sequences that violate the spec. Takes the Erlang process registry as an example.

Minimizing counterexamples

The usefulness of a failing test case is inversely proportional to its size and complexity. This module discusses methods to find as small failing test cases as possible, and teaches how to configure QuickCheck's simplification strategy to best effect.

... Day 4

Protocol Testing

QuickCheck is often used to test protocols. This module discusses protocol testing using QuickCheck state machines, and includes an extended example of testing a bank server.

Testing C with QuickCheck

Testing APIs in other languages than Erlang requires connecting them to QuickCheck. This module teaches simple ways of making the connection between QuickCheck and C.

We do continually improve our courses, so the content of your course may vary a little from the above.

... Contact

sales@quviq.com • Telephone +46 (0)70 438 8567

QuviQ AB • Bergshamravägen 4 • S-433 60 SÄVEDALEN • Sweden • www.quviq.com