Unit Testing in Groovy

Jeremy Jarrell www.JeremyJarrell.com @jeremyjarrell





Why Should We Unit Test?

Finds problems **early** in development

Preserves **key** behavior of the system

Documents the code

What Makes a Good Unit Test



Fast

Isolated

Repeatable

Self-verifying

Timely

Test-driven development (TDD)
Writing your unit tests
before you write the
production code.

Red-Green-Refactor



Watch the test Fail

Red-Green-Refactor





Watch the test Fail

Write just enough code for the test to Pass

Red-Green-Refactor







Watch the test Fail

Write just enough code for the test to Pass

Refactor the code

Results in clean APIs

- Results in clean APIs
- Testable code is flexible code

- Results in clean APIs
- Testable code is flexible code
- Results in much less production code

- Results in clean APIs
- Testable code is flexible code
- Results in much less production code
- No need to schedule 'unit-testing' time

Disadvantages of TDD

Increases initial development time

Disadvantages of TDD

- Increases initial development time
- Has a significant learning curve

Disadvantages of TDD

- Increases initial development time
- Has a significant learning curve
- Difficult without full team buy-in

Unit Testing Support in Groovy

JUnit is built **directly** into the Groovy runtime

Mocking APIs are included in the Groovy platform

Advanced testing frameworks are also available

Groovy Is Dynamic

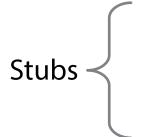
The compiler doesn't enforce typechecking

Unit tests become your safety net Especially on a large team

Mock Object

A simulated object that mimics the behavior of a real object in a controlled way.

Mocks Versus Stubs



- Can be used as stand-in objectsWill return canned values when asked



- Supports all the functionality of stubs
- Can verify a member was called a certain number of times
- Can assert a specific interaction occurred

Summary

| | Unit testing | provides a | multitude d | of benefits | s to softwar | e develo _l | pers |
|--|---------------------|------------|-------------|-------------|--------------|-----------------------|------|
|--|---------------------|------------|-------------|-------------|--------------|-----------------------|------|

Summary

- Unit testing provides a multitude of benefits to software developers
- Functionality for unit testing is built directly into the Groovy platform

Summary

- Unit testing provides a multitude of benefits to software developers
- Functionality for unit testing is built directly into the Groovy platform
- Mocking allows you to isolate and control the dependencies of your code