Getting Started With Rails Testing

Michael Denomy

Boston Ruby Study Group

January 15, 2013

About Me

- Original background in robotics and medical device software
- Introduced to TDD and XP in 2004
- Senior Developer at Cyrus Innovation
  - Web consulting company specializing in Agile and Extreme Programming (XP) practices
  - www.cyrusinnovation.com
- mdenomy on Twitter and github
- Blog: www.mdenomy.wordpress.com

Agenda

- Test Driven Development Basics
- Red-Green-Refactor loop
- Overview of Rails test environment
  - What are the different types of tests
  - What tools are available
- Write and run a few simple tests

Why Test?

Kent Beck on Test Driven Development

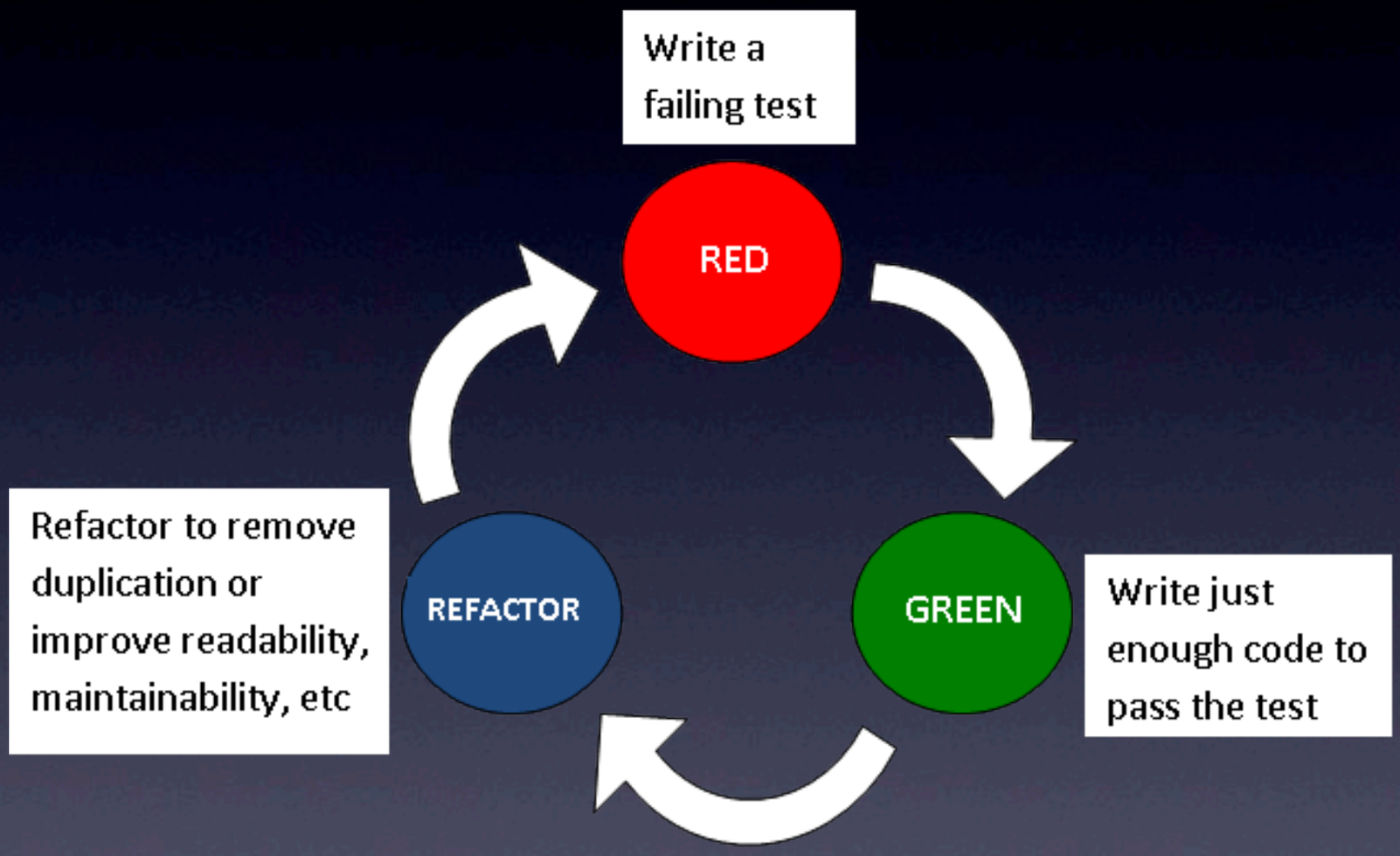
-“Test-driven development is a way of managing fear during programming.”

TDD Workflow

- Write a test that captures a desired behavior
  - Test fails, because not implemented yet
- Write the *smallest amount* of code to make the test pass.
  - Run the test. It passes
- Refactor to remove duplication, improve readability, modularize, etc.
  - Run the test, it better still pass.

Test Driven Development (TDD)

- Red-Green-Refactor cycle



- Repeat this cycle over and over in small increments

Rails Test Types

- Unit Tests
  - Used to test *models*
- Functional Tests
  - Used to test *controllers*
  - Also allows for *view* testing by looking for key HTML elements
- Integration Tests
  - Tests the *interactions* between different controllers and application *workflow*.

Anatomy of a Test

- Arrange-Act-Assert Pattern
  - Arrange
    - Set up any pre-conditions and data
  - Act
    - Run the code under test
  - Assert
    - Verify that the code did what you expected
- Keep tests simple
  - One “logical assertion” per test
    - Multiple asserts OK, but should be related

Testing Frameworks

- Test-Unit
  - TDD style framework, built into Rails
- RSpec
  - BDD style testing framework
  - More expressive syntax than Test-Unit
- MiniTest
  - Lightweight, supports Test-Unit and RSpec style
  - Built in to Ruby 1.9
- Capybara
  - Integration testing framework, simulates browser actions
- FactoryGirl
  - Create data for tests

Find the tools that work for *you*

Expense Tracker

- Sample code to show TDD workflow
  - Using branches and tags in git to simulate TDD cycle and Red-Green-Refactor
  - Developer workflow is the focus (not the app)
- Not using scaffolding generator
  - Add code as you need it
- Repository available at

  https://github.com/mdenomy/expense_tracker

  - After cloning, do “git branch -a” to see all branches
  - See Readme.md for description of branches

Resources

- Michael Hartl's Rails Tutorial
  - Teaches Rails in TDD fashion
  - Great version control practices with git
  - http://ruby.railstutorial.org/
- Rails Guides
  - http://guides.rubyonrails.org/testing.html
- Rails Test Prescriptions - Noel Rappin
  - http://pragprog.com/book/nrtest/rails-test-prescriptions
- Test Driven Development - Kent Beck
  - Addison-Wesley 2003

Summary

- Work in small steps
- Red-Green-Refactor
  - Don’t mix the steps
  - Version control is your friend
- Find the tools that work for you
  - And use them
- Avoid using the scaffolding
  - OK for prototyping

You *will* struggle to start, but stick with it

Getting Started With Rails Testing

Michael Denomy