

# The Craft of Developing Ruby Applications

## Random Ideas

- Behavior/Test Driven Development (B/TDD)
  - Test::Unit introduction to show how it is not easy to write/organize tests
  - RSpec is great for writing unit tests
  - Define unit tests( RSpec ) vs functional/Integration tests ( Cucumber )
  - Write tests first or at least as you write the code
- Continuous Integration
  - Benefits
  - HowTo
- User Stories
  - user stores and their benefit to the development process
  - User Stories and writing cucumber tests
  - Using Pivotal Tracker
  - Product Backlog Management
- Social Coding
  - Benefits
  - Using git and github
  - Introduce social coding and its benefits as part of the craft of developing ruby applications
- Team Dynamics
  - Single Developer
  - Pair Programming
  - Large Team Decomposition
- Refactoring
  - avoid writing another language in ruby
  - Improving code you just wrote
  - turning good code into extensible code
  - Input validation

- Custom Exceptions
- Meta-Programming
  - Go over the example of adding ActiveRecord like sugar to the data table syntax
  - method missing
  - class methods (define\_exception)
- Profiling
  - introduce benchmark
  - cover the simple example of converting 2 arrays into a hash
- Static Code Analysis
  - using reek to uncover code smells
  - flay
  - flog
  - heckle
- Example Applications
  - Report Framework
  - DataTable
  - DefineException
  - VR::Script written more generically