# Introduction to Testing

Jason Coposky & Terrell Russell

# Rough Sketch

- What’s the Point?
- Kinds of Software Testing
- Unit Testing
- Regression Testing
- Feature Testing
- Test Driven Development
- Putting it all to Work

# What’s the Point?

- Demonstrate that a Product actually works
- Verify that a Product meets requirements
- Provide a safety net for making major changes
- Ensure that old bugs do not arise once again

# Types of Software Testing

- Unit Testing – testing a single function or module
- Feature Testing – demonstrate that a particular behavior is correct
- Regression Testing – run tests that expressed old bugs
- Acceptance Testing – verify requirements are met for delivery
- Others – Performance, Usability, Stress, and many more

# Unit Testing

- Small executable tests to demonstrate behavior of Components: functions, classes, modules
- Typically written in a Framework for a given language: JUnit, RUnit, CppTest
- Allows for the creation odd and difficult execution environments for the software – much harder at the feature level
- Best approach for increasing code coverage
- Ground Up Testing – show that the smallest Components work first

# Feature Testing

- Ensure success at the high level
- Show that real world inputs give correct answers
- Throw bad data at the system and watch it fail gracefully, or not
- Generally a home grown or Commercial System – we use python’s built-in testing capabilities

# Regression Testing

- Bugs Happen – capture them in a test
- Commit the test alongside the fix
- Implemented through the Feature Testing framework
- An excellent candidate for automation – run regularly to ensure you don’t go backwards
- Customarily run as a separate suite in automation

# Advanced Maneuver - Test Driven Development

- Given a framework for testing – write the test first!
- Use tests to describe the behavior of your system
- Once tests are in place make them pass
- Builds out Feature Test suite
- Helps inform expected behavior of the System

# Examples: revisiting getPercent

```
getPercent <- function( value, pct ) {
    # how do we implement this?
}

result <- getPercent( 10, 110 )
# we expect 11 here

result <- getPercent( 10, -2 )
# we expect INVALID_PARAMETER here
```