Software Testing 2017
TDD & BDD Overview

Gregory S. DeLozier, Ph.D.

gdelozie@kent.edu

Topics

- Purpose of Testing
- Test Driven Development
- Behavioral Driven Design (BDD)

Software Testing...

- Is an empirical
- technical
- investigation
- conducted to provide stakeholders
- with *information*
- about the *quality*
- of the product or service under test

(Cem Kaner)

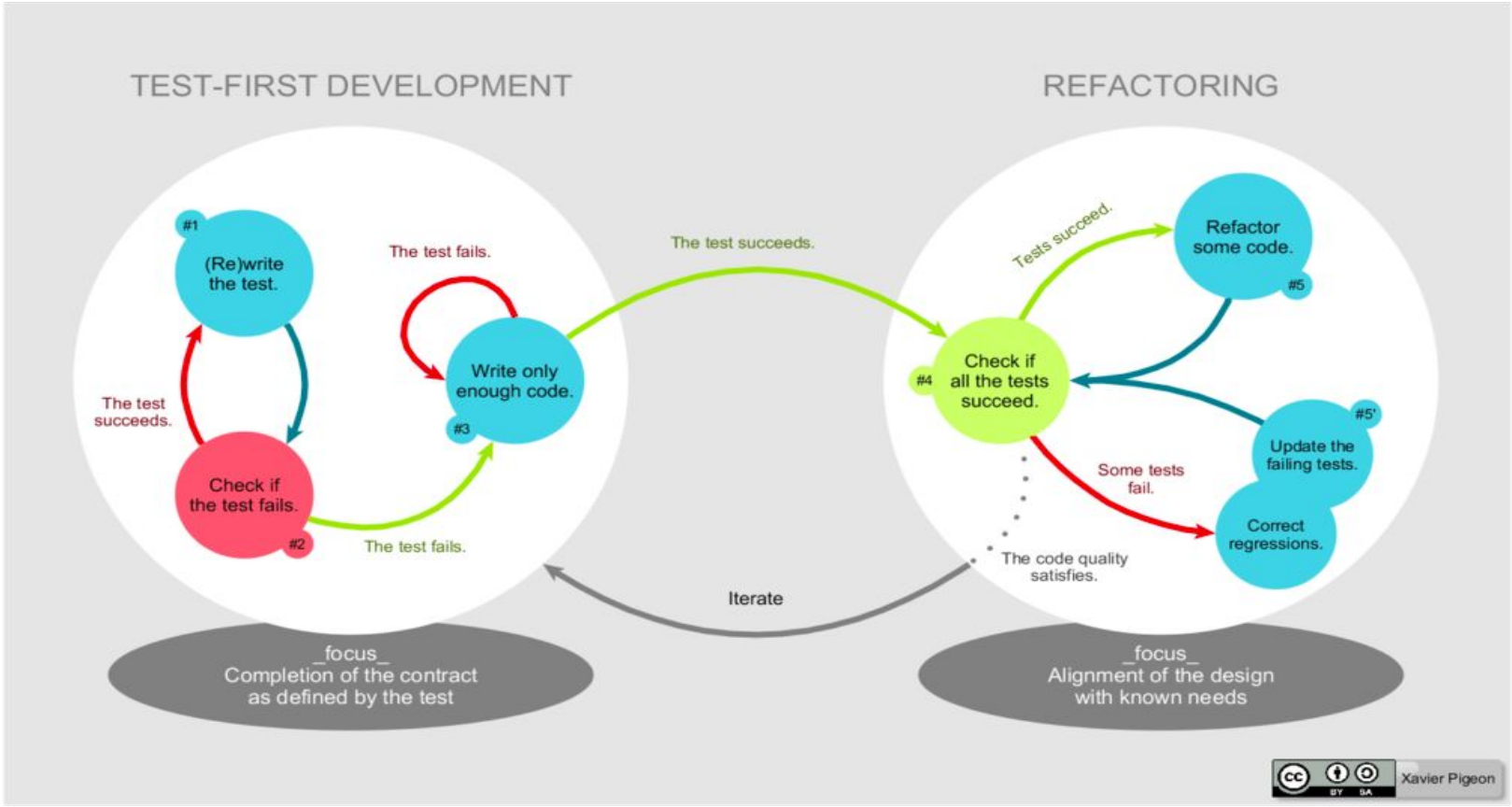
Test Driven Development

Test Driven Development

Repeat while employed

- Understand a requirement
- Write a test
- Observe it fail ← permission to code
- Write code to make the test run
- Observe it working
- Repeat

The TDD Cycle



https://en.wikipedia.org/wiki/Test-driven_development

The Unittest Framework

Available under many languages

Unit test suites -- collections of tests

Unit tests -- individual tests

Framework to organize the tests

Python/Unittest -

https://docs.python.org/3/library/unittest.html

Module Testing

Link directly to module being tested.

Call procedures and methods of module

Can test module exhaustively

Difficult to simulate effects of environment

Web Service Testing

Use something like "requests" library

Used to make HTTP/HTTPS requests to web servers.

Results are examined to determine test results.

http://docs.python-requests.org/en/master/

Web Application Testing

Use Python WebDriver module

Ability to connect to web GUI directly

Remotely operate a web browser

Can connect to lots of languages

http://selenium-python.readthedocs.io/

Behaviour Driven Development

Behavior Driven Development

- A continued refinement of TDD
- Emphasize collaboration with stakeholders

- Unit tests are about specific features

- BDD tests start out with requirements
  - Emphasizing business value
  - Stated in terms of user experience

A BDD Requirement

- Example

In order to keep track of stock
As a store owner
I want to add items back to stock when they're returned.

- Includes
  - Purpose or benefit
  - Who wants it
  - What needs to happen

A BDD Criteria or Scenario

- Example

Scenario 1: Refunded items should be returned to stock
Given that a customer previously bought a black sweater from me
And I have three black sweaters in stock.
When he returns the black sweater for a refund
Then I should have four black sweaters in stock.

- Includes
  - Initial condition
  - Event or action
  - Outcome

Python 'behave' Scenario

```
Feature: showing off behave

  Scenario: run a simple test
     Given we have behave installed
      when we implement a test
      then behave will test it for us!
```

```
...this is in a _feature_ file.
```

Python 'behave' Implementation

```
from behave import *

@given('we have behave installed')
def step_impl(context):
    pass

@when('we implement a test')
def step_impl(context):
    assert True is not False

@then('behave will test it for us!')
def step_impl(context):
    assert context.failed is False
```

'behave' module

- $ pip install behave
- create file structure
  - feature file
  - ./steps/xxx.py implementation files
  - example:

```
features/
features/everything.feature
features/steps/
features/steps/steps.py
```

- $ behave

'behave' Results

```
% behave
Feature: showing off behave # tutorial/tutorial.feature:1

  Scenario: run a simple test        # tutorial/tutorial.feature:3
    Given we have behave installed   # tutorial/steps/tutorial.py:3
    When we implement a test         # tutorial/steps/tutorial.py:7
    Then behave will test it for us! # tutorial/steps/tutorial.py:11

1 feature passed, 0 failed, 0 skipped
1 scenario passed, 0 failed, 0 skipped
3 steps passed, 0 failed, 0 skipped, 0 undefined
```

The 'behave' Tutorial

- http://pythonhosted.org/behave/tutorial.html
- Work through examples with me in class
- Try some examples of your own

*** THIS IS A HUGELY VALUABLE TECHNOLOGY

- There are others: cucumber, jbehave, rspec, etc.
- The ideas are very similar

Closing Concepts

* Unit tests (i.e. unit-test style tests) verify that systems behave the way that developers believe they should behave.
* Behaviour tests verify that systems behave the way that all participating stakeholders believe they should behave.
* Readable tests make that happen.