

Software Engineering Methodology

Lecture 13 - Behavior Driven

Gregory S. DeLozier, Ph.D.

gdelozie@kent.edu

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>
- Read this
- Work through examples
- Try some examples of your own

*** THIS IS A HUGELY VALUABLE TECHNOLOGY

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

DEMO TIME