

Test Driven Development and Puppet

Berlin User Group Meetup
April 10, 2014

Johan De Wit
(johan@open-future.be)
<http://puppet-be.github.io/>

Whoami

- Open Source Consultant @ open-future
- Organiser Belgian Puppet User Group
- A Sys-Admin
- A very poor developer (but working on it)
- Love riding :
 - Bikes
 - horses



Do you test your modules ?

I did !!

- 1: write a module
- 2: puppet parser validate ?
- 3: puppet-lint ?
- 4: puppet apply tests/init.pp (smoke)
- 5: puppet agent -t --noop
 - You use vagrant – right ??



Start over again

- Me => dev **ops**
- Should be **DEVops**
 - Yes, we write code to manage our infrastructure.
 - Learn from Developers
 - UNIT TESTING
 - INTEGRATION TESTING
 - ACCEPTANCE TESTING
 -

Testing is great

- Confidence changing things
- Discover breaking things before deploy
- Test against # puppet & # ruby versions
- Test many os'es without deploying them
- Test early - fast feedback
- Prevent regression of old problems

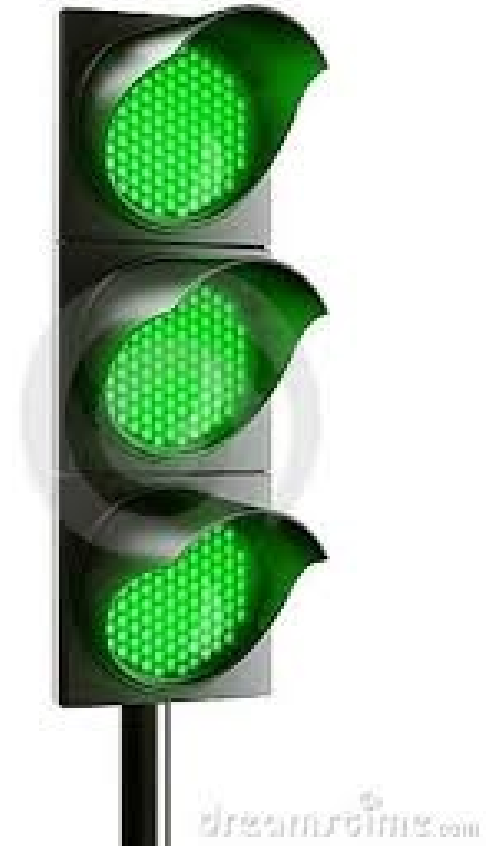
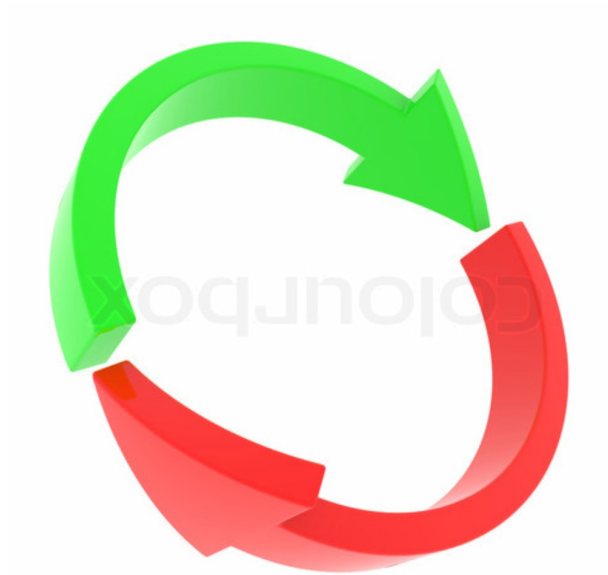
First thing first

- Unit testing
 - Rspec-puppet
 - Start with testing, then coding
- It is the beginning of ..
 - Integration testing (beaker)
 - Travis
 - Jenkins
 -

Think colors

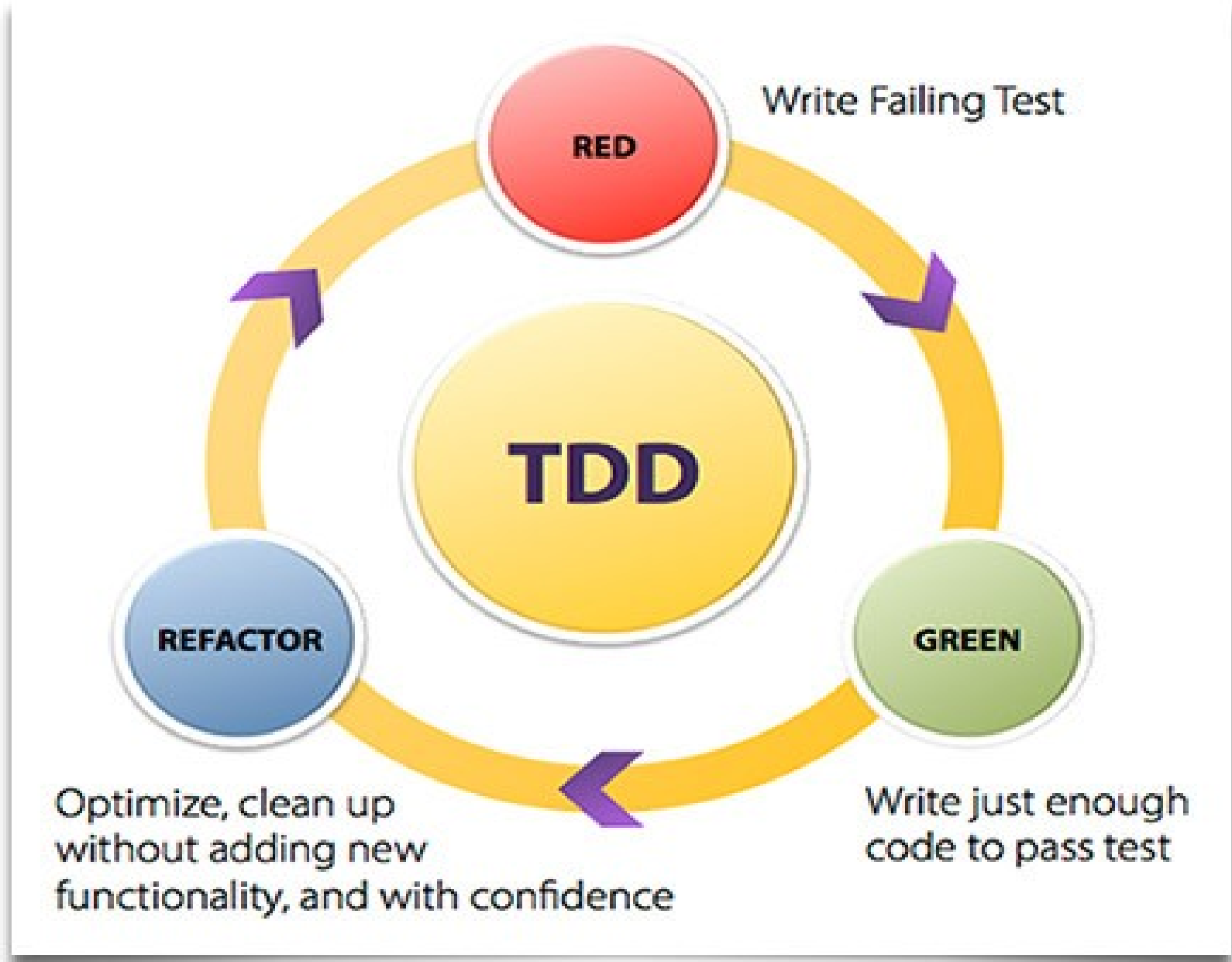


Test



Puppet
Code

GoTest Driven Development



Benefits of TDD

- Test case from the beginning
- Better code coverage
- Tests are maintained during life cycle
- Focus on the needed functionality, step by step
- Encourage simple design (avoid over engineering)
- First step in test automation (unit testing)

TDD does not

- Replace integration testing
- Replace compliance testing
-

Great ... but



Getting Practical

TDD and puppet modules

- Write the docs first (README.md)
 - Explain your parameters
 - Describe the defaults
 - What is the function of your module
 - What is its intended behaviour
- Write the first test from the README
- Run the tests, all should **fail**
- Write **just enough** code to **pass** the test
- Refactor and reiterate the process
(Ted Arroway: Small moves, Ellie, small moves. [Contact])

The right tools for the right job

- <http://www.rvm.io>
 - Switch easily between ruby version
- Rspec-puppet
 - Written by tim :
 - <http://rspec-puppet.com>
 - <https://github.com/rodjek/rspec-puppet>
- Puppet module skeleton
 - <https://github.com/ghoneycutt/puppet-module-skeleton>
 - <https://github.com/garethr/puppet-module-skeleton>
 - ...

TDD and rspec-puppet

- Testing against the compiled catalog
 - Are the right resources in the catalogue
 - With the right attributes
- Is the rspec a duplicate of the manifest code
 - When you start – yes, because we start simple
 - But we can copy/paste ? Right !!
 - Refactoring a basic module shows already the benefits.
 - Adding parameters
 - Adding logic (eg. Support for multiple OS)
 - ...
- Puppet modules with proper rspec test are better candidates
- It should/will become common to do PR including rspec tests

Hands on TDD

- Based on the TDD tutorial Garrett Honeycutt
 - <https://github.com/ghoneycutt/learnpuppet-tdd-vagrant>
 - <https://github.com/ghoneycutt/puppet-module-skeleton>
- Why ?
 - Followed the TDD session on LOADays
 - Everything is configured out of the box
 - Easy to start doing it the right way
 - Garrett learned me puppet

Hands on TDD – the setup

- The module directory tree

```
[root@puppet motd]# tree -a
.
```

- |-- **.fixtures.yml**
- |-- Gemfile
- |-- .gitignore
- |-- LICENSE
- |-- **manifests**
 - | `-- **init.pp**
- |-- Modulefile
- |-- Rakefile
- |-- README.md
- |-- **spec**
 - | |-- **classes**
 - | | `-- **init_spec.rb**
 - | `-- **spec_helper.rb**
- `-- .travis.yml

Hands on TDD – the setup

- puppet generate module witjoh-motd
- mv witjoh-motd motd
- Rakefile

```
[root@puppet motd]# rake -T
rake build          # Build puppet module package
rake clean          # Clean a built module package
rake coverage       # Generate code coverage information
rake help           # Display the list of available rake tasks
rake lint           # Check puppet manifests with puppet-lint / Run
puppet-lint
rake spec         # Run spec tests in a clean fixtures directory
rake spec_clean  # Clean up the fixtures directory
rake spec_prep   # Create the fixtures directory
rake spec_standalone # Run spec tests on an existing fixtures
directory
rake validate   # Validate manifests, templates, and ruby files
```

Hands on TDD – the setup

- `spec_helper.rb`
 - Code that is run before your spectest
 - Configures your spec testing environment

```
[root@puppet spec]# cat spec_helper.rb
require 'rubygems'
require
'puppetlabs_spec_helper/module_spec_helper'
```

Hands on TDD – the setup

- `.fixtures.yml`
 - Catalog dependencies are taken care off
 - Resolves dependencies to other modules
 - Creates symlink to own module
 - (does not support `metadata.json` from forge modules)

```
[root@puppet motd]# cat .fixtures.yml
fixtures:
  repositories:
    stdlib:
      repo: 'git://github.com/puppetlabs/puppetlabs-stdlib.git'
      ref: '3.2.0'
  symlinks:
    motd: "#{source_dir}"
```

A Simple TDD Session workflow

- Write README first
 - Explain the function of your module
 - Parameters
 - Default values
 - Valid values
- Write the test based on the readme
- Write the code
 - Just enough code to pass the test
- Refactor and add more stuff
 -

Hands on TDD – the test

- First test
 - `<module >/spec/classes/init_spec.rb`

Rake validate

```
[root@puppet classes]# rake validate
(in /root/demos/motd)
puppet parser validate --noop
manifests/init.pp
ruby -c spec/classes/init_spec.rb
Syntax OK
ruby -c spec/spec_helper.rb
Syntax OK
[root@puppet classes]#
```

Hands on TDD – init_spec.rb

```
require 'spec_helper'
describe 'motd' do

  context 'with defaults for all parameters' do
    it { should contain_class('motd') }

    it {
      should contain_file('motd').with({
        'ensure' => 'file',
        'path'   => '/etc/motd',
        'owner'  => 'root',
        'group'  => 'root',
        'mode'   => '0644',
        'content' => nil,
      })
    }
  end
end
```



```
[root@puppet motd]# SPEC_OPTS="--format documentation" rake spec
HEAD is now at elf2a93 Update Modulefile, CHANGELOG for 3.2.0
/usr/bin/ruby -S rspec spec/classes/init_spec.rb --color
```

```
motd
```

```
with defaults for all parameters
```

```
should contain Class[motd]
```

```
should contain File[motd] with owner => "root", mode => "0644", group => "root", content defined, ensure => "file" and path => "/etc/motd" (FAILED - 1)
```

```
Failures:
```

```
1) motd with defaults for all parameters should contain File[motd] with owner => "root", mode => "0644", group => "root", content defined, ensure => "file" and path => "/etc/motd"
```

```
Failure/Error: })
```

```
expected that the catalogue would contain File[motd]
```

```
# ./spec/classes/init_spec.rb:15
```

```
Finished in 0.27044 seconds
```

```
2 examples, 1 failure
```

```
Failed examples:
```

```
rspec ./spec/classes/init_spec.rb:7 # motd with defaults for all parameters should contain File[motd] with owner => "root", mode => "0644", group => "root", content defined, ensure => "file" and path => "/etc/motd"
```

```
/usr/bin/ruby -S rspec spec/classes/init_spec.rb --color failed
```

```
[root@puppet motd]#
```

Hands on TDD – The code

```
# == Class: motd
#
# Module to manage motd
#
class motd {

  file { 'motd':
    ensure => 'file',
    path   => '/etc/motd',
    owner  => 'root',
    group  => 'root',
    mode   => '0644',
    content => undef,
  }
}
```

Hands on TDD – The test

```
[root@puppet manifests]# SPEC_OPTS="--format documentation" rake spec
(in /root/demos/motd)
Initialized empty Git repository in /root/demos/motd/spec/fixtures/modules/stdlib/.git/
remote: Reusing existing pack: 4681, done.
remote: Total 4681 (delta 0), reused 0 (delta 0)
Receiving objects: 100% (4681/4681), 931.86 KiB | 958 KiB/s, done.
Resolving deltas: 100% (1745/1745), done.
HEAD is now at elf2a93 Update Modulefile, CHANGELOG for 3.2.0
/usr/bin/ruby -S rspec spec/classes/init_spec.rb --color

motd
  with defaults for all parameters
    should contain Class[motd]
    should contain File[motd] with content defined, mode => "0644", owner => "root",
ensure => "file", path => "/etc/motd" and group => "root"

Finished in 0.27287 seconds
2 examples, 0 failures
[root@puppet manifests]#
```

More rspec

```
describe 'with path specified' do
  context 'as a valid path' do
    let(:params){ { :path => '/usr/local/etc/motd' } }
    it {
      should contain_file('motd').with({
        'path' => '/usr/local/etc/motd',
      })
    }
  end

  context 'as an invalid path' do
    let(:params) { { :path => 'invalid/path' } }
    it 'should fail' do
      expect {
        should contain_class('motd')
      }.to raise_error(Puppet::Error)
    end
  end
end
end
end
```

More rspec

```
['666', '66666', 'invalid', true].each do |mode|
  context "as invalid value #{mode}" do
    let(:params) { { :motd_mode => mode } }

    it 'should fail' do
      expect {
        should contain_class('motd')
      }.to
      raise_error(Puppet::Error, /^motd::mode must be
a four digit string./)
    end
  end
end
```

```
# package
it {
  should contain_package('ntp_package').with({
    ... })
}
# file
it {
  should contain_file('ntp_config').with({
    ...
    'require' => 'Package[ntp]',
  })
}

# service
it {
  should contain_service('ntp_service').with({
    ...
    'subscribe' => 'File[ntp_config]',
  })
}
```

More rspec

check for a specific line

```
it { should contain_file('ntp_conf').with_content(/^tinker  
panic 0$/) }
```

Check that some content is not include

```
it { should_not contain_file('ntp_conf').with_content(/^tinke  
panic 0$/) }
```

More rspec

```
context 'with default values for parameters on EL 6' do
  let(:facts) do
    { :osfamily => 'RedHat',
      :lsbmajdistrelease => '6',
    }
  end
end
end
```


More rspec – defined resources

```
# spec/defines/mkdir_p_spec.rb
require 'spec_helper'
describe 'common::mkdir_p' do
  context 'should create new directory' do
    let(:title) { '/some/dir/structure' }

    it {
      should
      contain_exec('mkdir_p-/some/dir/structure').with({
        'command' => 'mkdir -p /some/dir/structure',
        'unless' => 'test -d /some/dir/structure',
      })
    }
  end
end
```

More rspec – defined resources

```
context 'should fail with a path that is not absolute' do
  let(:title) { 'not/a/valid/absolute/path' }
  it do
    expect {
      should contain_exec('mkdir_p-
not/a/valid/absolute/path').with({
        'command' => 'mkdir -p
not/a/valid/absolute/path',
        'unless'    => 'test -d
not/a/valid/absolute/path',
      })
    }.to raise_error(Puppet::Error)
  end
end
end
end
```

What should be tested

- All resources should be in the catalog
 - 100% code coverage
- Parameters
 - Proper defaults
 - Setting params, does that work ?
 - Logic of params
 - Parameter validation

What should be tested

- Module logic
 - Based on facts (eg: `::osfamily`)
 - Multiple os support
- Dynamic content
 - Test your templates

Unit testing is the beginning

- Integration testing
- Acceptance testing
-



TDD by a developer

- Test driven development for puppet – Puppetcamp Londen
by Gareth Rushgrove
- <https://speakerdeck.com/garethr/test-driven-development-for-puppet>
- Test Drive Development and Puppet Tutorial
LOADays
by Garrett Honneycutt
- <http://www.slideshare.net/gh/20140406-loa-daystddwithpuppettutorial>

Belgian Puppet User Group

Home Members Sponsors Photos Pages Discussions More

Group tools



Change photo

Brussels,
Belgium

Welcome!

+ SCHEDULE A NEW MEETUP

Upcoming 1

Suggested 0

Past

Calendar

FEATURED MEETUP

Round Table Puppet Infrastructure Setup

Combelle NV
Skaldenstraat 121, Gent (map)



Tue Feb 25
7:00 PM

I'M ATTENDING

What's

NEW MEM
Stefan
joined
Yester

NEW COM
Martin
comme
Fosden
group s
2 days

NEW COM
Bram L
comme