# Bulletproof PowerShell

Siarhei Shchahrykovich, Platform Engineer at dotmailer

# Problem

# Best practices

- Unit testing
- Static code analyzer

# Unit testing

The goal of unit testing is to isolate each part of the program and show that the individual parts are correct

- Finds problems early
- Facilitates change (Refactoring)
- Simplifies integration
- Adds documentation
- Improves design

https://en.wikipedia.org/wiki/Unit_testing

# Clean Code By Uncle Bob

https://cleancoders.com/landing

# What is Pester?

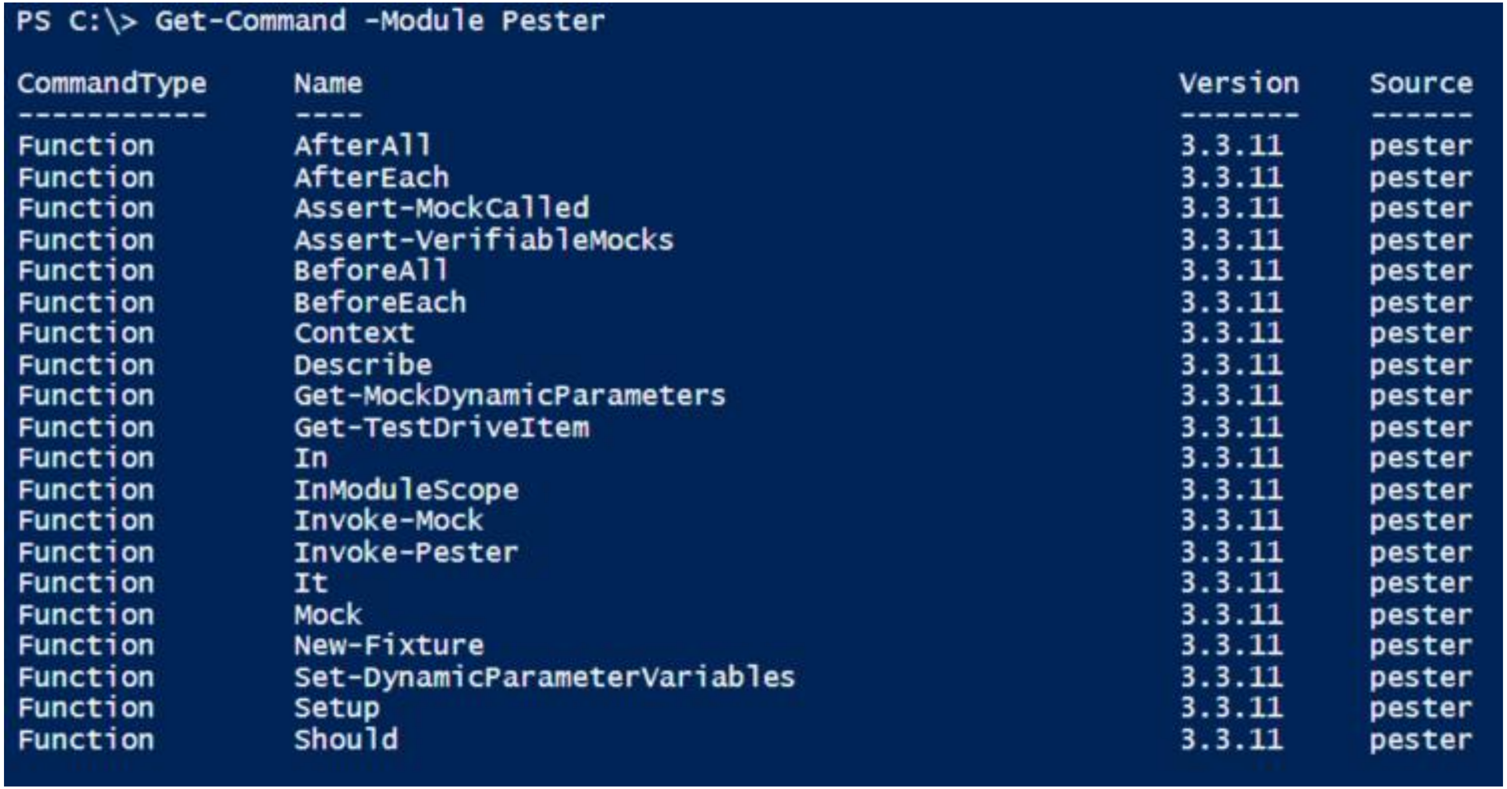
- Pester is a BDD based test runner for PowerShell
- Pester provides a framework for running Unit Tests
- Pester also contains a powerful set of Mocking Functions
- Code coverage

https://github.com/pester/Pester/

# Why Pester?

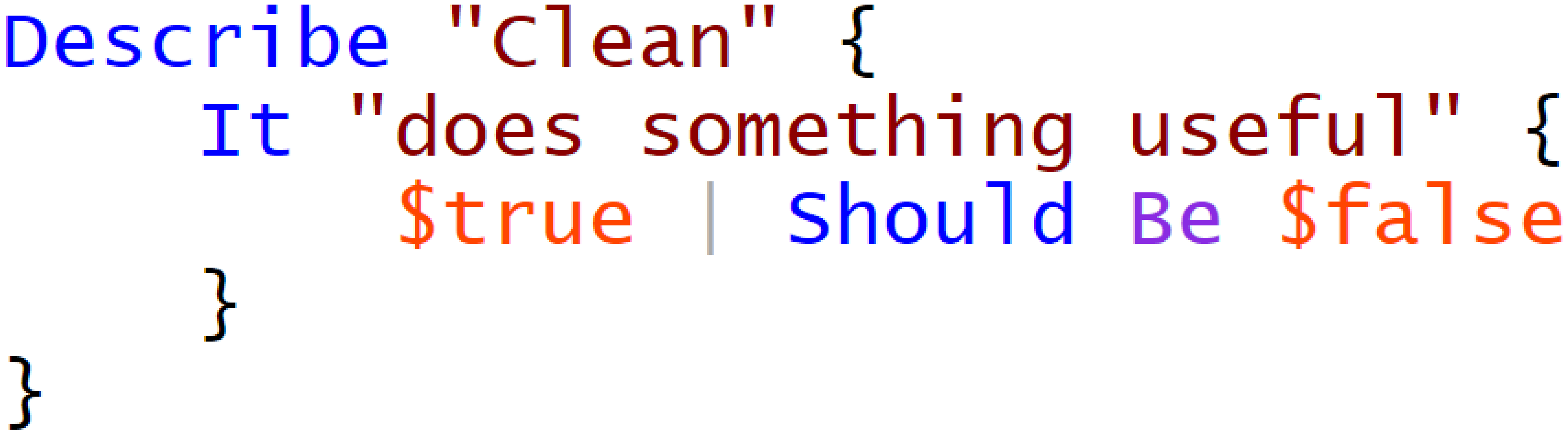
- Part of Windows 10
- Pester based tests for testing PowerShell
  https://github.com/PowerShell/PowerShell-Tests
- PSUnit https://github.com/knutkj/pstest
- PsTest https://psunit.codeplex.com/

# Pester commands



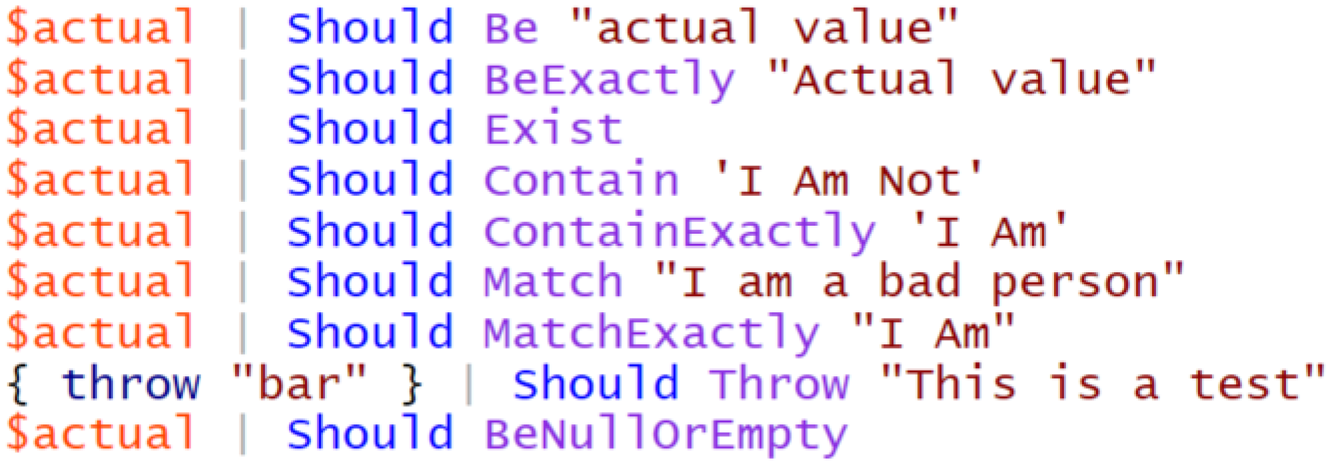
```
PS C:\> Get-Command -Module Pester

CommandType     Name                                               Version    Source
-----------     ----                                               -------    ------
Function        AfterAll                                           3.3.11     pester
Function        AfterEach                                          3.3.11     pester
Function        Assert-MockCalled                                  3.3.11     pester
Function        Assert-VerifiableMocks                             3.3.11     pester
Function        BeforeAll                                          3.3.11     pester
Function        BeforeEach                                         3.3.11     pester
Function        Context                                            3.3.11     pester
Function        Describe                                           3.3.11     pester
Function        Get-MockDynamicParameters                          3.3.11     pester
Function        Get-TestDriveItem                                  3.3.11     pester
Function        In                                                 3.3.11     pester
Function        InModuleScope                                      3.3.11     pester
Function        Invoke-Mock                                        3.3.11     pester
Function        Invoke-Pester                                      3.3.11     pester
Function        It                                                 3.3.11     pester
Function        Mock                                               3.3.11     pester
Function        New-Fixture                                        3.3.11     pester
Function        Set-DynamicParameterVariables                      3.3.11     pester
Function        Setup                                              3.3.11     pester
Function        Should                                             3.3.11     pester
```

# First Test



```
Describe "Clean" {
    It "does something useful" {
        $true | Should Be $false
    }
}
```

# Should



```
$actual | Should Be "actual value"
$actual | Should BeExactly "Actual value"
$actual | Should Exist
$actual | Should Contain 'I Am Not'
$actual | Should ContainExactly 'I Am'
$actual | Should Match "I am a bad person"
$actual | Should MatchExactly "I Am"
{ throw "bar" } | Should Throw "This is a test"
$actual | Should BeNullOrEmpty
```

# TestDrive – Isolating File Operations

Pester creates a PSDrive inside the user's temporary drive that is accessible via:

- "TestDrive:\test.txt"
- $TestDrive

https://github.com/pester/Pester/

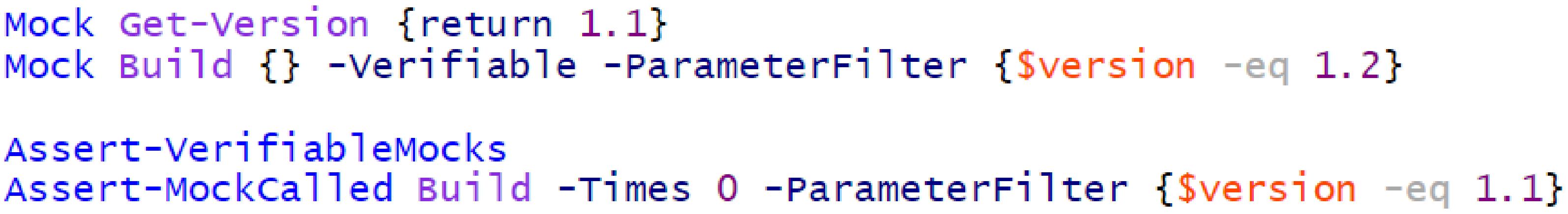
# Mocks

Simulated objects that mimic the behavior of real objects in controlled ways:

- Mock the behavior of ANY powershell command
- Verify that specific commands were (or were not) called
- Verify the number of times a command was called with a set of specified parameters

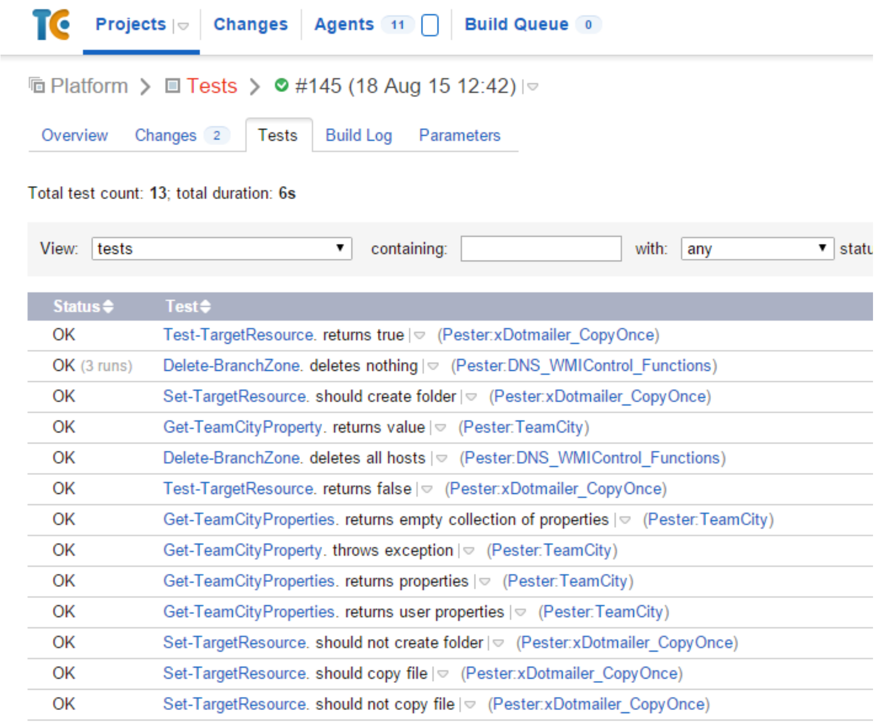
https://github.com/pester/Pester/

# Mocks



```
Mock Get-Version {return 1.1}
Mock Build {} -Verifiable -ParameterFilter {$version -eq 1.2}

Assert-VerifiableMocks
Assert-MockCalled Build -Times 0 -ParameterFilter {$version -eq 1.1}
```

# TeamCity integration via NUnit xml file



Total test count: **13**; total duration: **6s**

View: tests containing: with: any statu

| Status | Test |
|---|---|
| OK | Test-TargetResource. returns true \| (Pester:xDotmailer_CopyOnce) |
| OK (3 runs) | Delete-BranchZone. deletes nothing \| (Pester:DNS_WMIControl_Functions) |
| OK | Set-TargetResource. should create folder \| (Pester:xDotmailer_CopyOnce) |
| OK | Get-TeamCityProperty. returns value \| (Pester:TeamCity) |
| OK | Delete-BranchZone. deletes all hosts \| (Pester:DNS_WMIControl_Functions) |
| OK | Test-TargetResource. returns false \| (Pester:xDotmailer_CopyOnce) |
| OK | Get-TeamCityProperties. returns empty collection of properties \| (Pester:TeamCity) |
| OK | Get-TeamCityProperty. throws exception \| (Pester:TeamCity) |
| OK | Get-TeamCityProperties. returns properties \| (Pester:TeamCity) |
| OK | Get-TeamCityProperties. returns user properties \| (Pester:TeamCity) |
| OK | Set-TargetResource. should not create folder \| (Pester:xDotmailer_CopyOnce) |
| OK | Set-TargetResource. should copy file \| (Pester:xDotmailer_CopyOnce) |
| OK | Set-TargetResource. should not copy file \| (Pester:xDotmailer_CopyOnce) |

# Pester demo

# PowerShell ScriptAnalyzer

Static code checker for Windows PowerShell modules and scripts. 40 built-in rules:

- Avoid Empty Catch Block
- Avoid Global Vars
- Avoid Using File Path
- Use PS Credential Type
- Custom rules

https://github.com/PowerShell/PSScriptAnalyzer

# ScriptAnalyzer demo

# Installation

- WMF 5.0* http://www.microsoft.com/en-us/download/details.aspx?id=48729
- Pester https://github.com/pester/Pester/
- PSScriptAnalyzer https://github.com/PowerShell/PSScriptAnalyzer

# QA

Thanks!

Email: shchegrikovich@gmail.com

Twitter: https://twitter.com/shchegrikovich

Github: https://github.com/shchahrykovich