Verifying Changes to Existing Scripts with Pester Tests

Robert C. Cain, MVP

OWNER, ARCANE TRAINING AND CONSULTING

@arcanecode www.arcanecode.com

Module 1

- Hortense, PodcastSight, No Agenda Show, Pester Basics

Module 2

- Describe, Context, It. Module validation.

Module 3

- Testing functions, Mock Data, Scoping

Module 4

- More function testing, InModuleScope, Mock Functions

Pipeline Get-PodcastImage

Pipeline Get-PodcastMedia

Modify both to not download files already present

Where to start?

Existing Tests!

Full integration test, when possible

Tests for functions being modified, at a minimum

Unit Tests vs Acceptance Tests

Small Company

Single developer authors unit tests

Same developer creates acceptance tests

Both tests in same PS1 file

```
$here = Split-Path -Parent $MyInvocation.MyCommand.Path

    Get-Module Podcast-NoAgenda | Remove-Module -Force
      Import-Module $here\Podcast-NoAgenda.psm1  -Force
```

Large Company

Developer creates unit tests

Dedicated tester writes acceptance tests

Tests in separate, distinct PS1 files

```
Import-Module Podcast-NoAgenda -Force
```

Summary

Ran tests before beginning changes

Altered unit tests for first requirement, changed code, tested

Altered unit tests for second requirement, changed code, tested

Updated acceptance after all unit tests were passing