Automate Operational Readiness and Validation Testing

of SQL Server with Powershell and Pester

mikefrobbins.com

Get-Help –Name 'about_Presenter'

- Mike F Robbins

- Microsoft MVP

- SAPIEN Technologies MVP

- Leader & Co-Founder of Mississippi PowerShell User Group

- Co-Author of Windows PowerShell TFM 4th Edition

- Author of Chapter 6 in the PowerShell Deep Dives book

- Winner of the Advanced Category in the 2013 Scripting Games

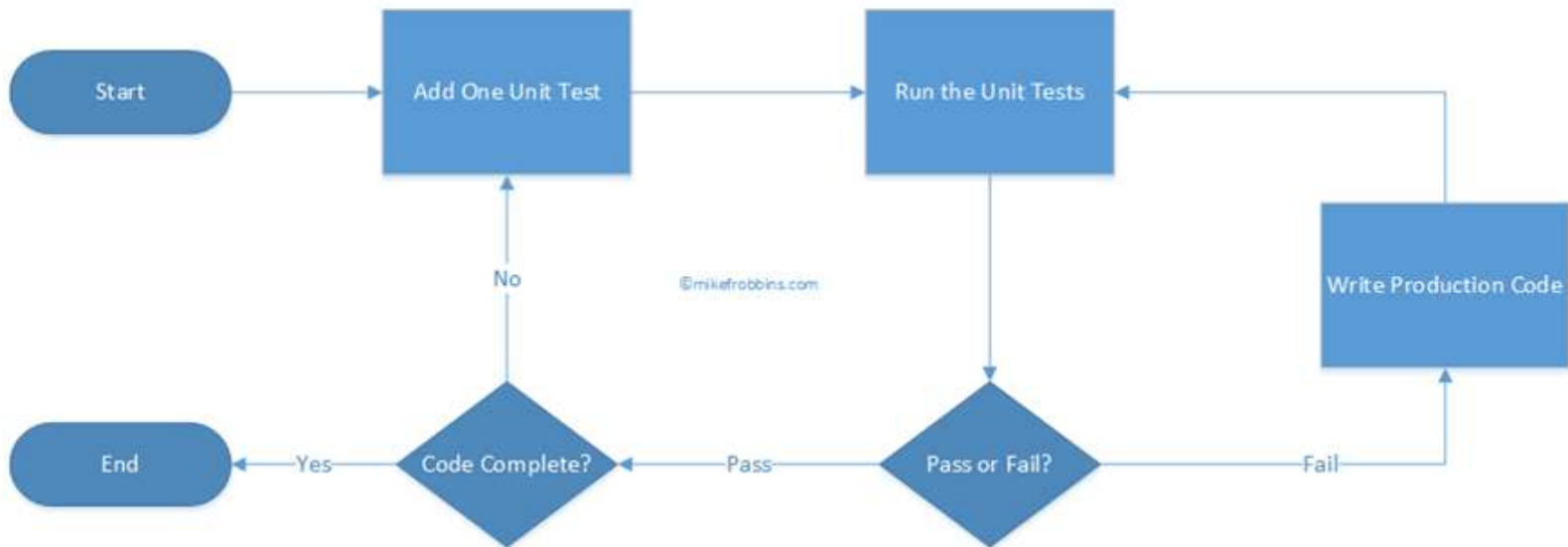- Learn more about me @ mikefrobbins.com

Pester

- Unit Test Framework for PowerShell

- Assertions
  - Should

- Mocking

- Isolated File Operations
  - TestDrive

- Unit Testing within Modules
  - Public Functions
  - Private Functions

For more info about Pester, see: https://github.com/pester/Pester/wiki

Test-Driven Development (TDD)

Test Driven Development Workflow

Start → Add One Unit Test → Run the Unit Tests

Add One Unit Test ← No ← Code Complete?

©mikefrobbins.com

Write Production Code

End ← Yes ← Code Complete? ← Pass ← Pass or Fail? ← Fail ← Write Production Code

Operational Validation Testing

- PowerShell

- Pester

- TSQL

- WMI

- Operation Validation Framework

- Other third party products

Start-Process 'Demo'

Demo Code: github.com/mikefrobbins/Presentations

Select-Object –Property 'Resources'

- [Pester Wiki on GitHub](#)

- [Operation Validation Framework](#)

- [PowerShell.org](#)

- [PowerShell Magazine](#)

- [Hey, Scripting Guy! Blog](#)

- [User Groups](#)

- [Twitter](#)

- [Blogs](#)

- [Slack](#)

- [Trello](#)

Get-Contact –Identity 'Presenter'

- Blog: mikefrobbins.com (or MrPowerShell.com)

- Twitter: @mikefrobbins

- LinkedIn: www.linkedin.com/in/mikefrobbins

- E-Mail: See mikefrobbins.com/about/

- User Group: mspsug.com (or MsPowerShell.com)