# OVF - checklisty są sexy!

Mateusz Czerniawski

Arcontar

mczerniawski@arcon.net.pl

@Arcontar

arconnetblog.wordpress.com

15.03.2018 Wroclaw

# Agenda:

1. Pester – what, how and why
2. PPoSh OVF – the Story, the Concept
3. Baseline your environment
4. „Describe” your needs
5. Test it! Monitor it! Report it!
6. What’s next?!
7. Q&A

# Pester – „the new black” (in WinOps world)

- https://github.com/pester/Pester - PowerShell TDD style testing framework
- http://www.powershellmagazine.com/2014/03/12/get-started-with-pester-powershell-unit-testing-framework/ - PowerShell unit testing framework (year 2014)
- https://blogs.technet.microsoft.com/heyscriptingguy/2015/12/14/what-is-pester-and-why-should-i-care/ (year 2015)
- https://leanpub.com/pesterbook/c/4ti1DbCOBQux (25-06-2017)

## Pester – powered by community

- Format-Pester https://github.com/equelin/Format-Pester
- PSCribo - https://github.com/iainbrighton/PScribo
- PoshSpec - expand the Pester DSL to test infrastructure - https://github.com/Ticketmaster/poshspec
- ReportUnit - https://mcpmag.com/articles/2016/12/01/create-a-simple-pester-test-report.aspx
- NUnit output – integration with CI tools - https://github.com/pester/Pester/wiki/Showing-Test-Results-in-CI-(TeamCity,-AppVeyor)

  http://www.dexterposh.com/2015/06/powershell-pester-jenkins-ci.html

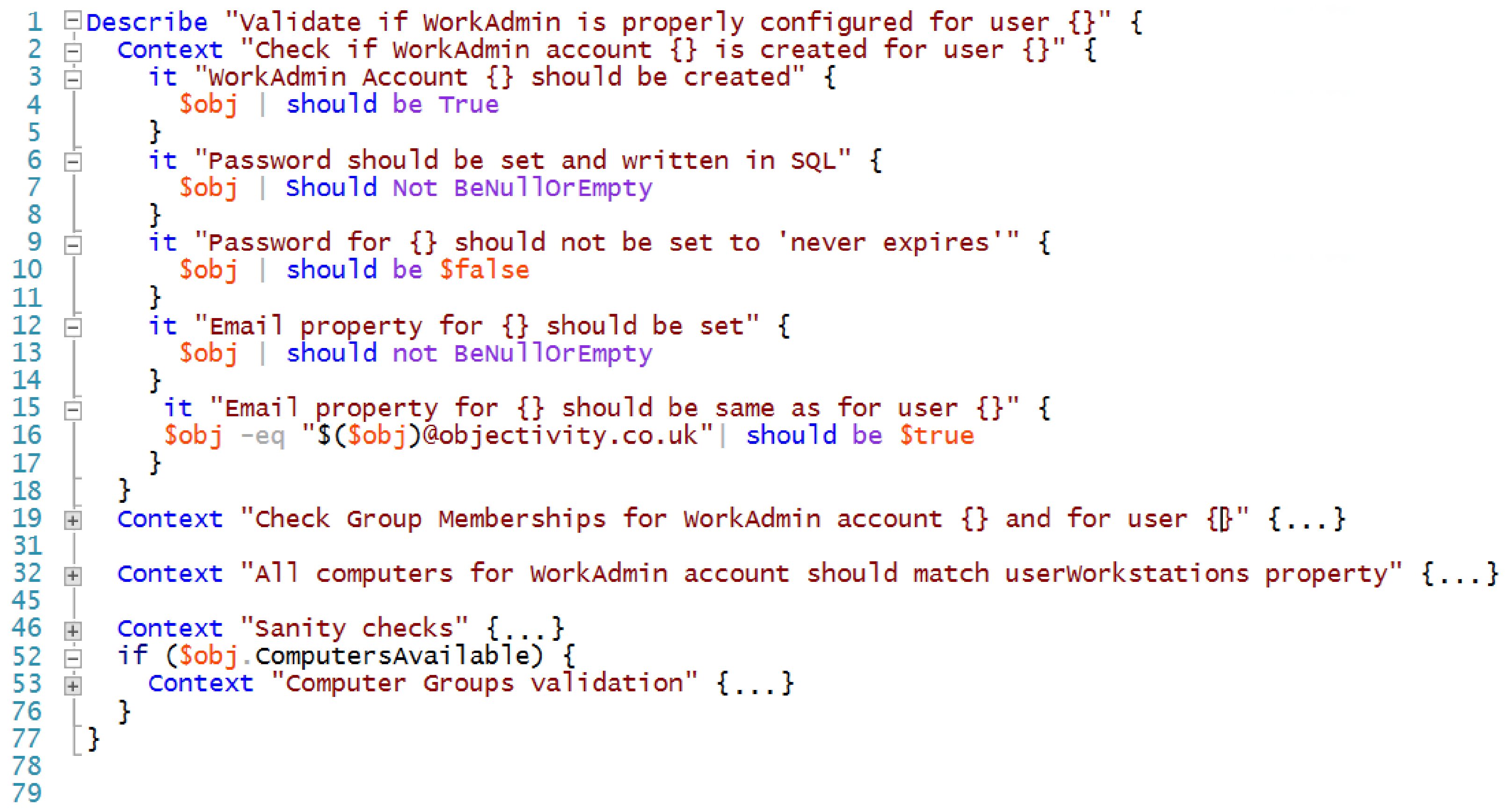
## Community OVF modules

- OVF - https://github.com/PowerShell/Operation-Validation-Framework
- DBAChecks - https://dbatools.io/introducing-dbachecks/
- Vester - https://github.com/WahlNetwork/Vester

# PPoSh OVF – the Story

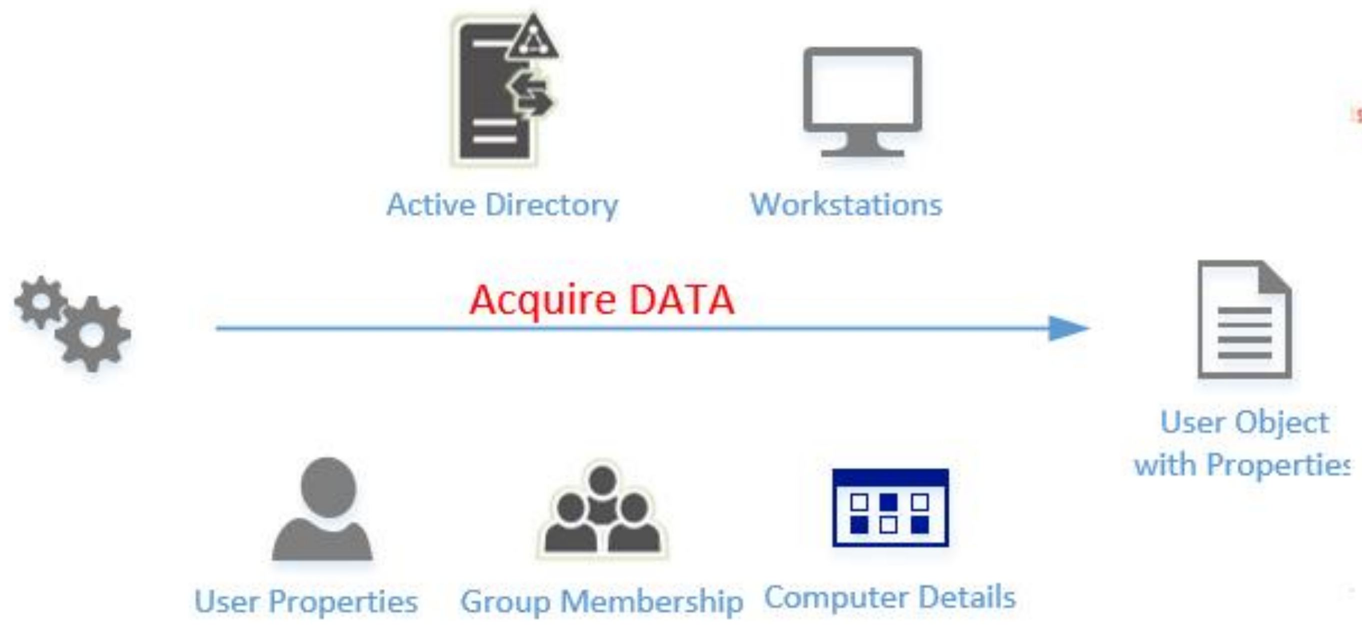
- Why OVF?
- Why not „Microsoft.PowerShell.Operation.Validation”?
- Why PPoSh OVF and PPoSh OVF Diagnostics?

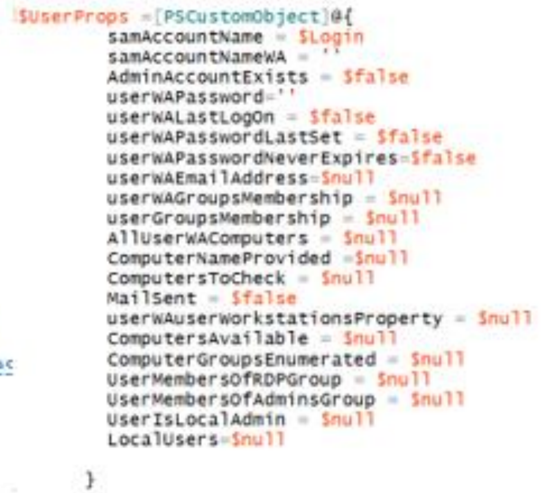
# The Story - The beggining



```
Describe "Validate if WorkAdmin is properly configured for user {}" {
  Context "Check if WorkAdmin account {} is created for user {}" {
    it "WorkAdmin Account {} should be created" {
      $obj | should be True
    }
    it "Password should be set and written in SQL" {
      $obj | Should Not BeNullOrEmpty
    }
    it "Password for {} should not be set to 'never expires'" {
      $obj | should be $false
    }
    it "Email property for {} should be set" {
      $obj | should not BeNullOrEmpty
    }
     it "Email property for {} should be same as for user {}" {
     $obj -eq "$($obj)@objectivity.co.uk"| should be $true
    }
  }
  Context "Check Group Memberships for WorkAdmin account {} and for user {}" {...}

  Context "All computers for WorkAdmin account should match userWorkstations property" {...}

  Context "Sanity checks" {...}
  if ($obj.ComputersAvailable) {
    Context "Computer Groups validation" {...}
  }
}
```

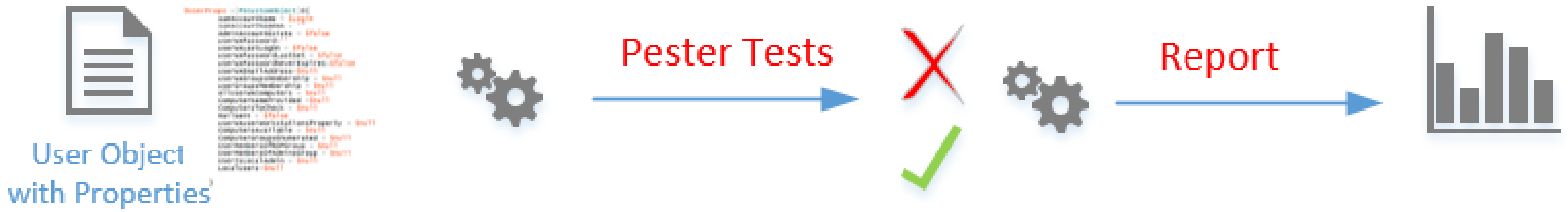
# The Story - The beggining



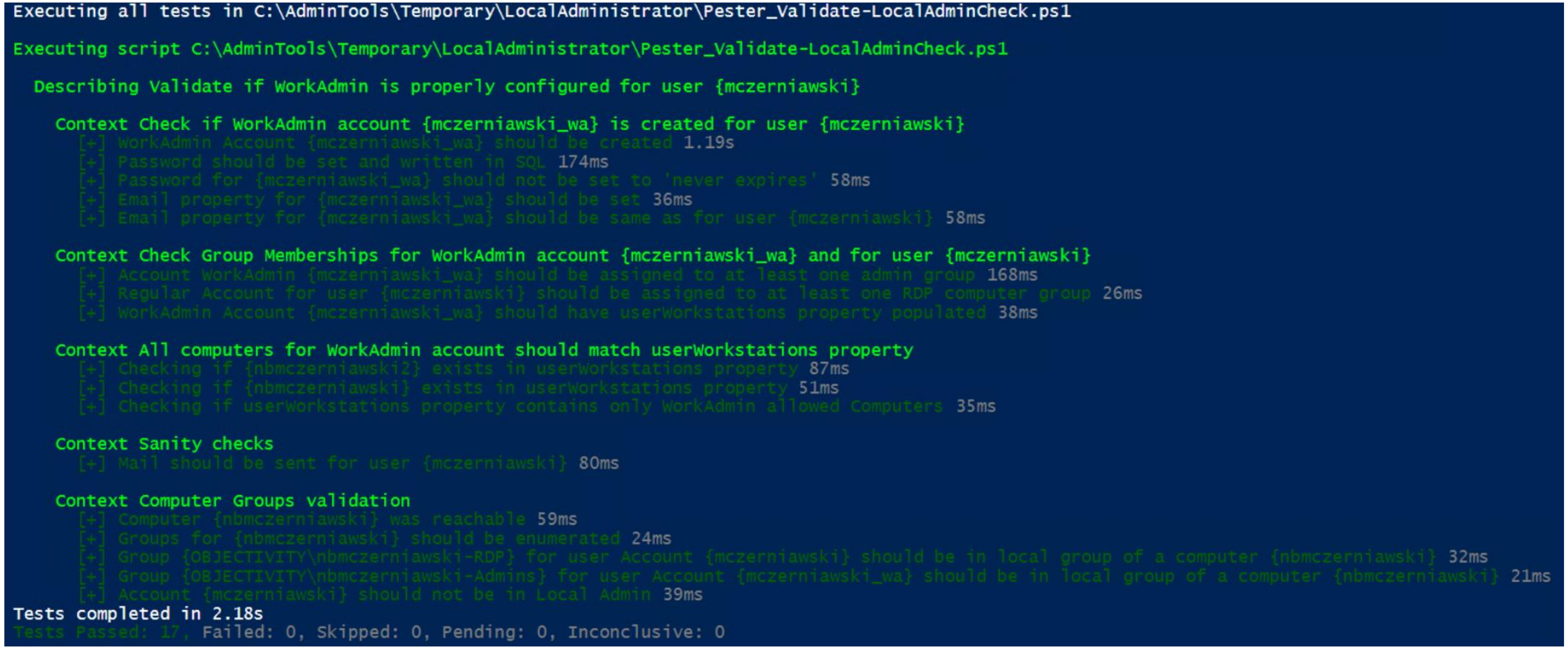


```
$UserProps =[PSCustomObject]@{
        samAccountName = $Login
        samAccountNameWA = ''
        AdminAccountExists = $false
        userWAPassword=''
        userWALastLogOn = $false
        userWAPasswordLastSet = $false
        userWAPasswordNeverExpires=$false
        userWAEmailAddress=$null
        userWAGroupsMembership = $null
        userGroupsMembership = $null
        AllUserWAComputers = $null
        ComputerNameProvided =$null
        ComputersToCheck = $null
        MailSent = $false
        userWAuserWorkstationsProperty = $null
        ComputersAvailable = $null
        ComputerGroupsEnumerated = $null
        UserMembersOfRDPGroup = $null
        UserMembersOfAdminsGroup = $null
        UserIsLocalAdmin = $null
        LocalUsers=$null

      }
```

## The Story - The beggining

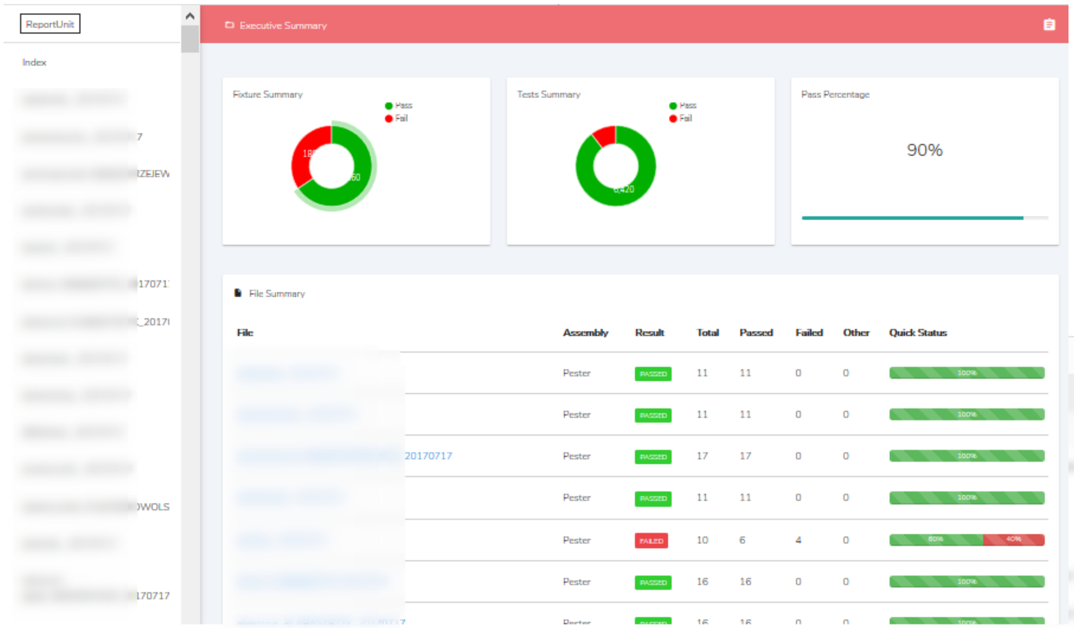


## The Story - The beggining

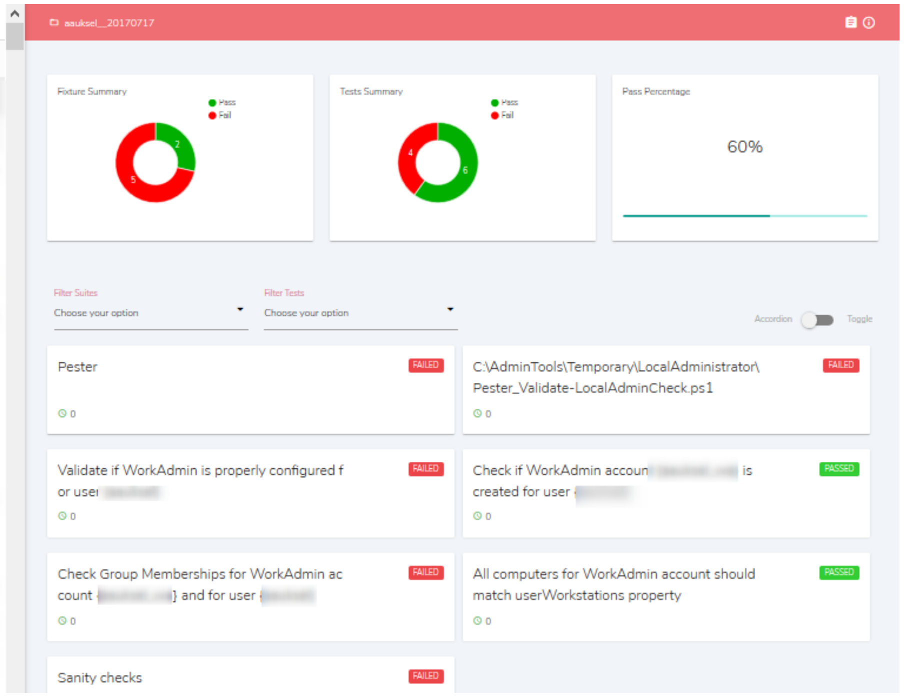


```
Executing all tests in C:\AdminTools\Temporary\LocalAdministrator\Pester_Validate-LocalAdminCheck.ps1

Executing script C:\AdminTools\Temporary\LocalAdministrator\Pester_Validate-LocalAdminCheck.ps1

  Describing Validate if WorkAdmin is properly configured for user {mczerniawski}

    Context Check if WorkAdmin account {mczerniawski_wa} is created for user {mczerniawski}
      [+] WorkAdmin Account {mczerniawski_wa} should be created 1.19s
      [+] Password should be set and written in SQL 174ms
      [+] Password for {mczerniawski_wa} should not be set to 'never expires' 58ms
      [+] Email property for {mczerniawski_wa} should be set 36ms
      [+] Email property for {mczerniawski_wa} should be same as for user {mczerniawski} 58ms

    Context Check Group Memberships for WorkAdmin account {mczerniawski_wa} and for user {mczerniawski}
      [+] Account WorkAdmin {mczerniawski_wa} should be assigned to at least one admin group 168ms
      [+] Regular Account for user {mczerniawski} should be assigned to at least one RDP computer group 26ms
      [+] WorkAdmin Account {mczerniawski_wa} should have userWorkstations property populated 38ms

    Context All computers for WorkAdmin account should match userWorkstations property
      [+] Checking if {nbmczerniawski2} exists in userWorkstations property 87ms
      [+] Checking if {nbmczerniawski} exists in userWorkstations property 51ms
      [+] Checking if userWorkstations property contains only WorkAdmin allowed Computers 35ms

    Context Sanity checks
      [+] Mail should be sent for user {mczerniawski} 80ms

    Context Computer Groups validation
      [+] Computer {nbmczerniawski} was reachable 59ms
      [+] Groups for {nbmczerniawski} should be enumerated 24ms
      [+] Group {OBJECTIVITY\nbmczerniawski-RDP} for user Account {mczerniawski} should be in local group of a computer {nbmczerniawski} 32ms
      [+] Group {OBJECTIVITY\nbmczerniawski-Admins} for user Account {mczerniawski_wa} should be in local group of a computer {nbmczerniawski} 21ms
      [+] Account {mczerniawski} should not be in Local Admin 39ms
Tests completed in 2.18s
Tests Passed: 17, Failed: 0, Skipped: 0, Pending: 0, Inconclusive: 0
```

# The Story - The beggining



| File | Assembly | Result | Total | Passed | Failed | Other | Quick Status |
|---|---|---|---|---|---|---|---|
| [redacted] | Pester | PASSED | 11 | 11 | 0 | 0 | 100% |
| [redacted] | Pester | PASSED | 11 | 11 | 0 | 0 | 100% |
| [redacted] 20170717 | Pester | PASSED | 17 | 17 | 0 | 0 | 100% |
| [redacted] | Pester | PASSED | 11 | 11 | 0 | 0 | 100% |
| [redacted] | Pester | FAILED | 10 | 6 | 4 | 0 | 60% 40% |
| [redacted] | Pester | PASSED | 16 | 16 | 0 | 0 | 100% |
| [redacted] | Pester | PASSED | 16 | 16 | 0 | 0 | 100% |



Pester FAILED 0

C:\AdminTools\Temporary\LocalAdministrator\Pester_Validate-LocalAdminCheck.ps1 FAILED 0

Validate if WorkAdmin is properly configured f or user [redacted] FAILED 0

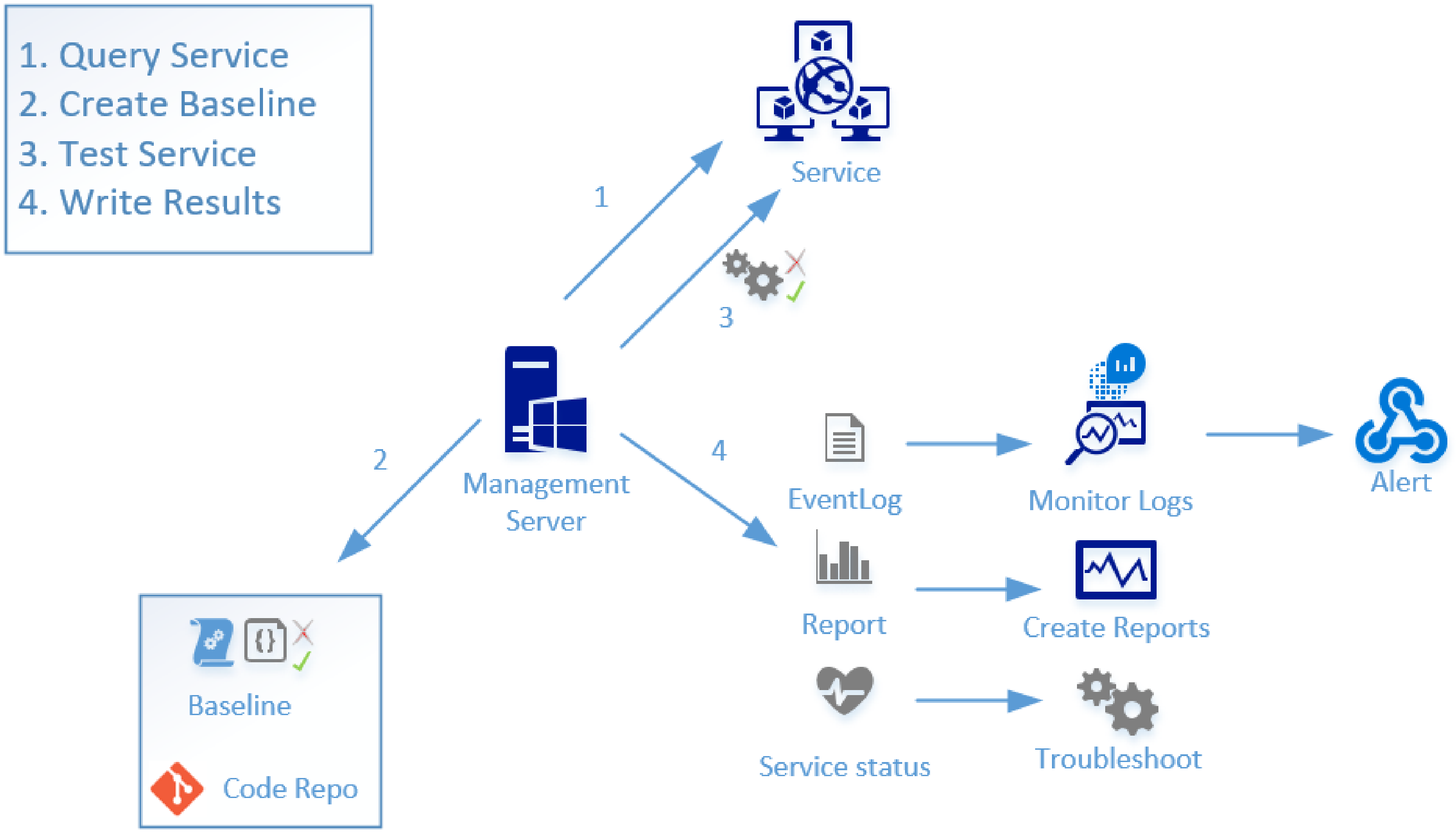
Check if WorkAdmin accoun [redacted] is created for user [redacted] PASSED 0

Check Group Memberships for WorkAdmin ac count [redacted]} and for user [redacted] FAILED 0

All computers for WorkAdmin account should match userWorkstations property PASSED 0

Sanity checks FAILED

# PPoSH OVF – The Concept



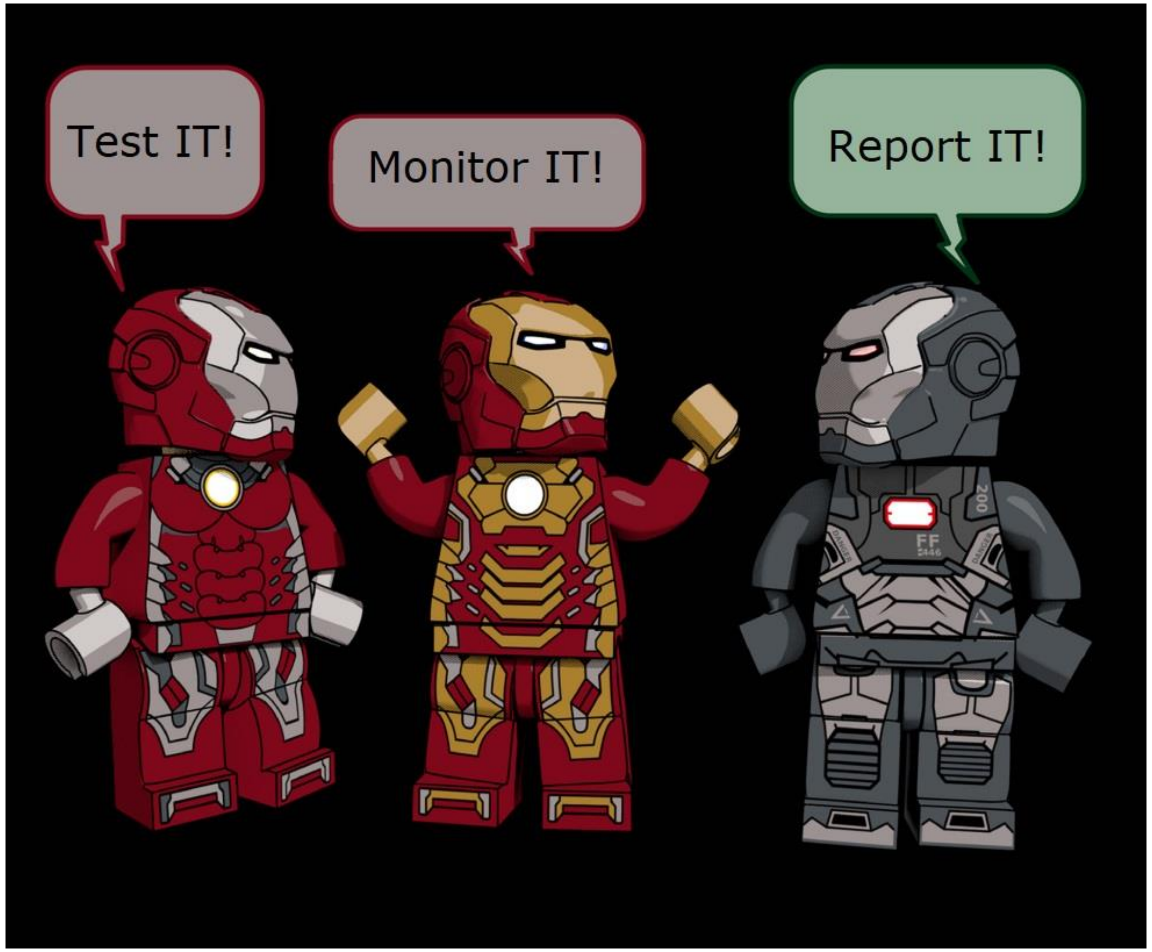
## Baseline your environment

Demo

## „Describe” your needs

Demo

## Test it! Monitor it! Report it!



Demo

# PPoSh OVF and PPoSh OVF Diagnostics

- PPoSh OVF:
- Invoke Pester Tests
- Write EventLog Entries
- Invoke Report Unit

- PPoSh OVF Diagnostics:
- Create Baseline Configuration
- Simple and Comprehensive Tests, Tagging, Node selection

## What’s next?!

- Update Help and Examples
- Split Diagnostics module
- Publish to PSGallery with AppVeyor
- Cert sign the code

- In the meanwhile add new tests and refactor closed-sourced

- Security Best Practice related tests

# Questions?

15.03.2018 Wroclaw

SysOps/DevOps Polska PPOSH