



SEPARATION OF CONCERNS IN DEVOPS

ChesterBurbidge.com @burbdude



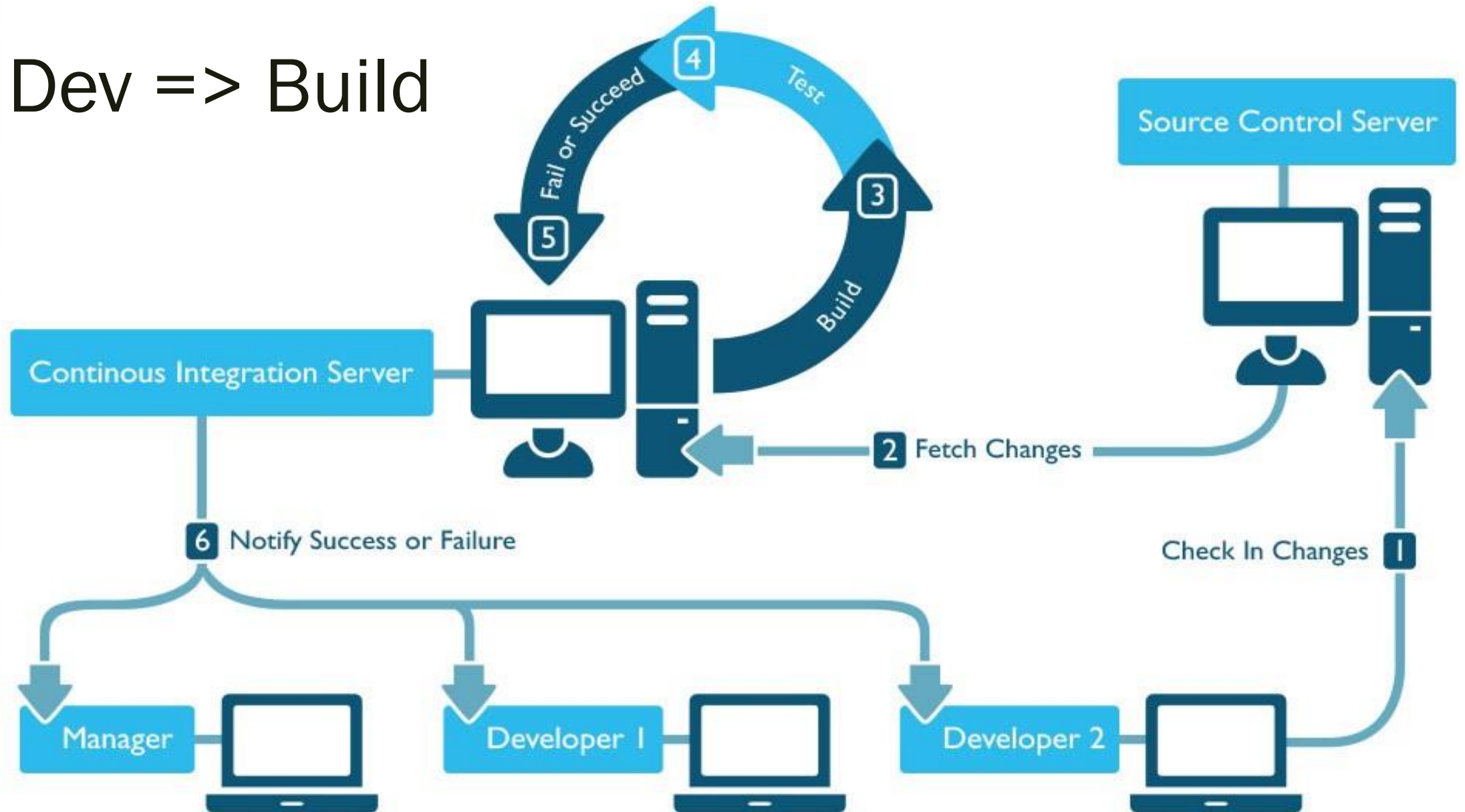
Hi

- Software Engineer at Abide Financial
- Recently performing SRE role
- Re-assessed build and deployment processes

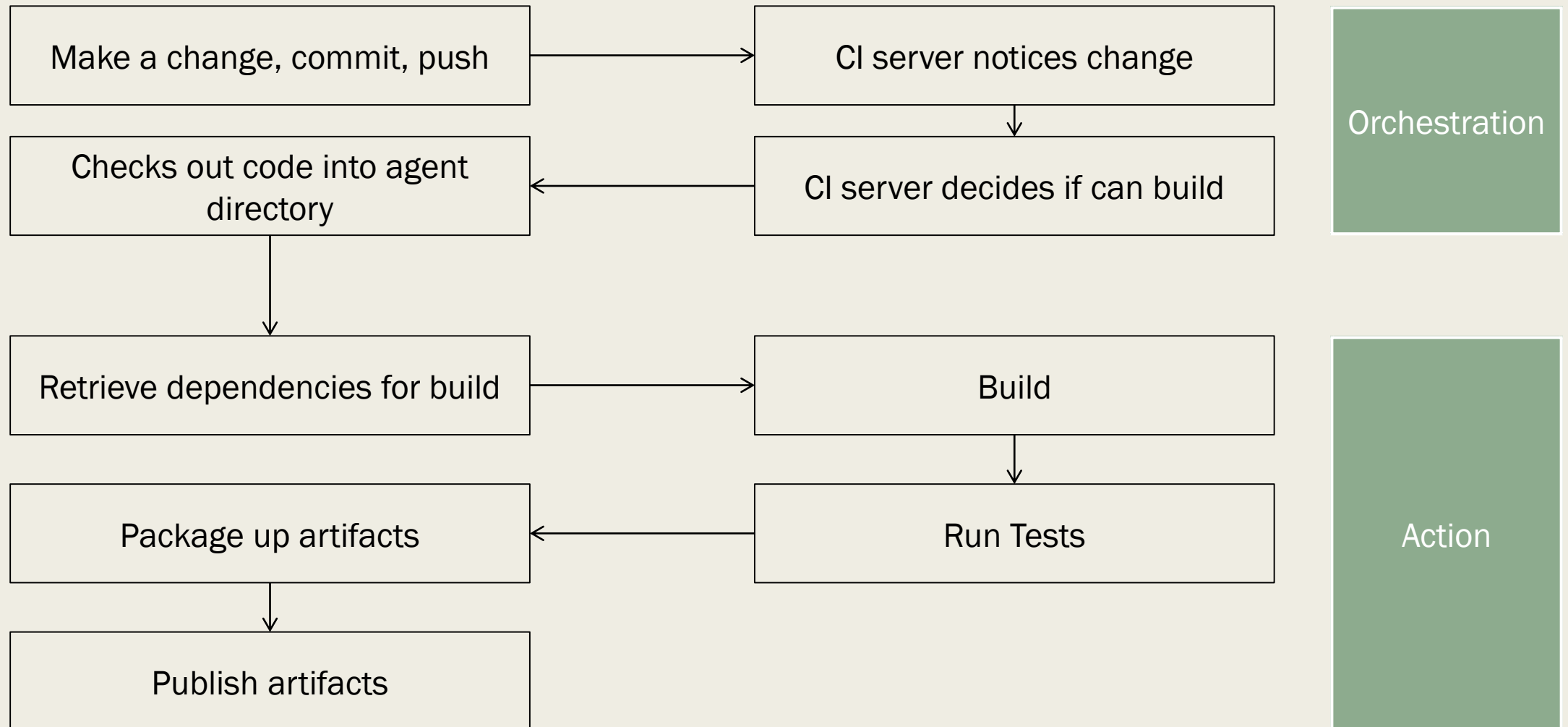
This talk

- Originates from mis-using Teamcity and Octopus deploy
- Need to separate concerns of orchestration and action in build and deployment.
- Dev => Deployed journey

Dev => Build



Dev => Build



What's the problem?

- Lack of source control
- Inability to run locally
- Long development feedback loop
- Type of logic
- Unclear dependencies
- Vendor lock-in
- Complex workflow leads to hacks

What to do?

- Separate 'action' build logic from Teamcity
 - *Programming in your CI/CD tool*
Thoughtworks tech radar – HOLD - Nov 2015
- Create build script as described by continuous integration
- Use something like RAKE, FAKE or ...

Cake



Cake





Cake

```
1 #tool nuget:?package=NUnit.ConsoleRunner&version=3.4.0
2
3 var target = Argument("target", "Default");
4
5 Task("Clean").Does(() => {
6     CleanDirectories(".**/bin/Release");
7 });
8
9 Task("Build").IsDependentOn("Clean").Does(() => {
10     MSBuild("./UsefullLibrary.sln",
11         settings => settings.SetConfiguration("Release")
12             .SetMSBuildPlatform(MSBuildPlatform.x64)
13             .SetPlatformTarget(PlatformTarget.MSIL));
14 });
15
16 Task("Run_Tests").IsDependentOn("Build").Does(() => {
17     NUnit3("./**/bin/Release/*.Tests.dll");
18 });
19
20 Task("Run_Long_Tests").IsDependentOn("Build").Does(() => {
21     NUnit3("./**/bin/Release/*.LongRunningTests.dll");
22 });
23
24 Task("Default").IsDependentOn("Run_Tests").Does(() => {});
25 RunTarget(target);
```

Terminal

```
0 Warning(s)
0 Error(s)

Time Elap
Finished

=====
Copy-File
=====
s outputs
=====
Executing
Finished
sts.csproj
Done build
=====
Package Build succ
=====
Skipping
=====
Time Elaps
Finished e
Package
=====
Executing task: Package
Finished executing task: Package
=====
Default
=====
Executing task: Default
Finished executing task: Default
=====
Task
-----
Task Duration
-----
Clean 00:00:00.0166879
Restore-NuGet-Packages 00:00:00.2460337
Patch-Assembly-Info 00:00:00.0074312
Build 00:00:03.4555588
Run-Unit-Tests 00:00:31.1255977
Copy-Files 00:00:00.0504735
Zip-Files 00:00:00.2058103
Create-NuGet-Packages 00:00:00.8015971
Package 00:00:00.0012023
Default 00:00:00.0009088
-----
Total: 00:00:35.9113013

patrik@pa
patrik@val
```

example

Target CoreCompile:
Skipping target "CoreCompile" because its outputs are up-to-date.

Windows PowerShell

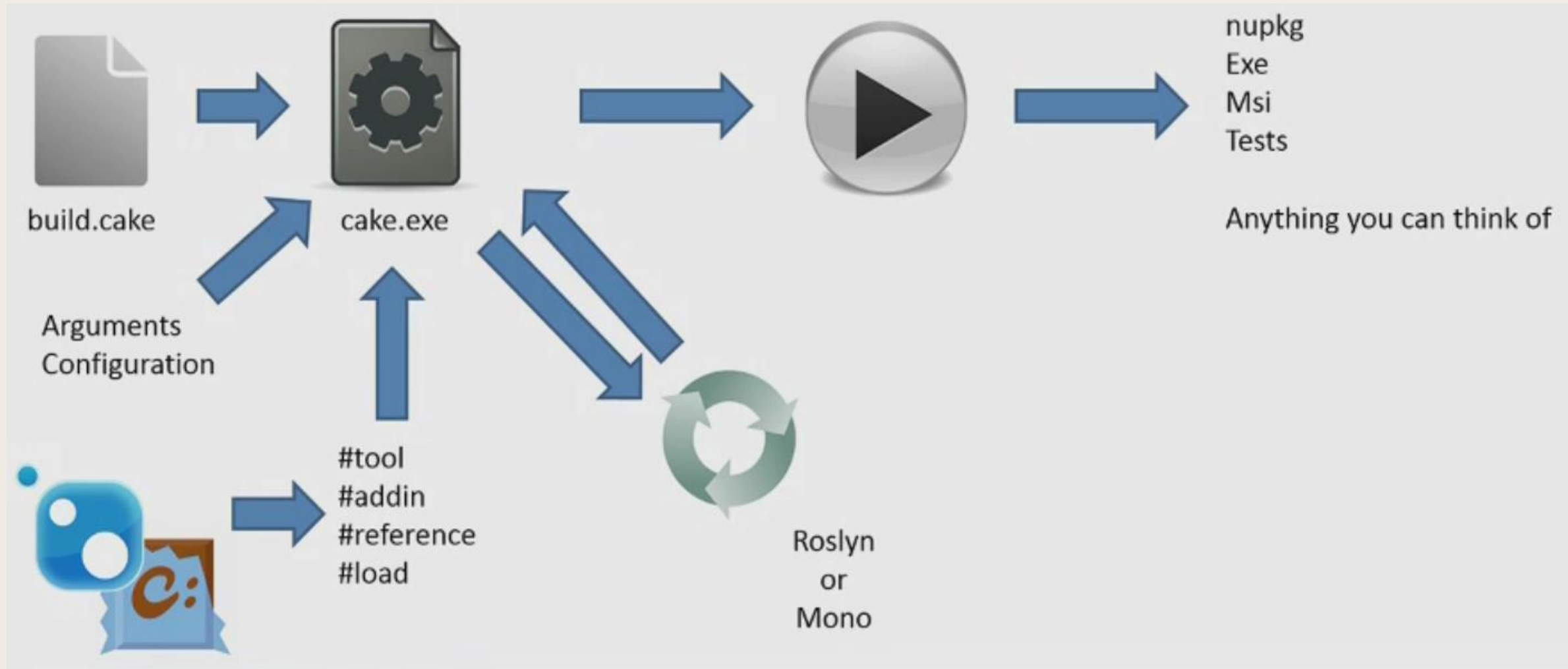
```
nupkg'.
Parsing nuspec...
Transforming nuspec...
Writing temporary nuspec...
Executing: C:/Source/Git/cake-build/cake/tools/nuget.exe pack -Version "0.4.2" -BasePath "
C:/Source/Git/cake-build/cake/build/v0.4.2/bin" -OutputDirectory "C:/Source/Git/cake-build
/cake/build/v0.4.2/nuget" "C:/Source/Git/cake-build/cake/nuspec/Cake.Core.temp.nuspec"
Attempting to build package from 'Cake.Core.temp.nuspec'.
Successfully created package 'C:/Source/Git/cake-build/cake/build/v0.4.2/nuget\Cake.Core.0
.4.2.nupkg'.
Finished executing task: Create-NuGet-Packages

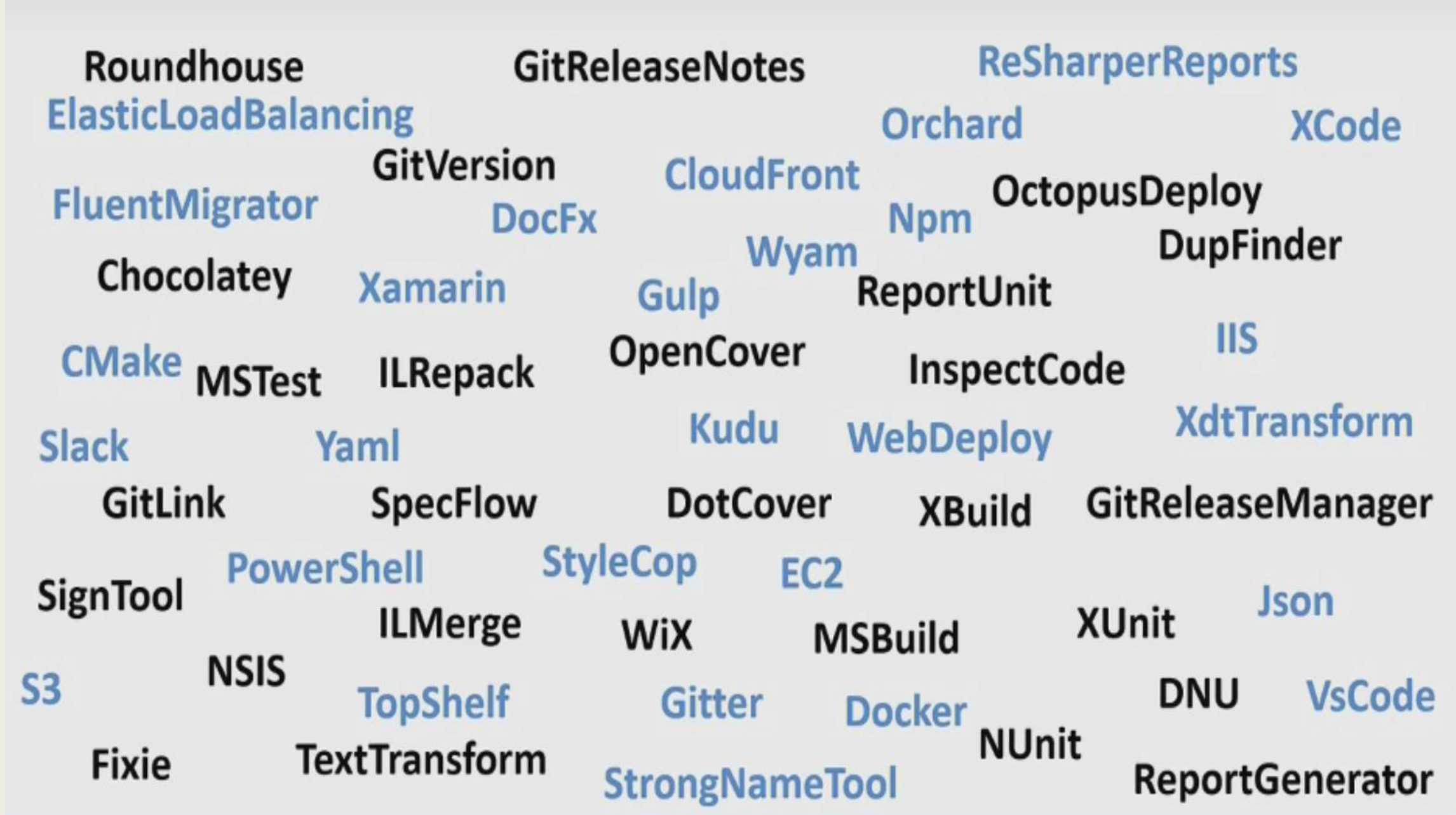
=====
Executing task: Package
Finished executing task: Package
=====
Default
=====
Executing task: Default
Finished executing task: Default
=====
Task
-----
Task Duration
-----
Clean 00:00:00.0166879
Restore-NuGet-Packages 00:00:00.2460337
Patch-Assembly-Info 00:00:00.0074312
Build 00:00:03.4555588
Run-Unit-Tests 00:00:31.1255977
Copy-Files 00:00:00.0504735
Zip-Files 00:00:00.2058103
Create-NuGet-Packages 00:00:00.8015971
Package 00:00:00.0012023
Default 00:00:00.0009088
-----
Total: 00:00:35.9113013

PS C:/Source/Git/cake-build/cake> .\build -Verbosity Diagnostic
```

Cake

- Cake (C# Make) is a build automation system with a C# DSL to do things like compiling code, copy files/folders, running unit tests, compress files and build NuGet packages.





A Piece of Cake - C# powered cross platform build automation - Gary Ewan Park
<https://www.youtube.com/watch?v=zZlyEn4jF2U>

Demo...

Cake – in practice

- Versioned Nuget package

common.csx – functions to #load into build.cake

build.cake – basic build logic

build.ps1 – standard cake file

install.ps1 – copy files into correct place

semver with common.csx public functions as API

- Import functions with overloads

public void runNUnit2Tests(

*string testFilePattern = "./**/bin/**/*Tests.dll",*

string outputFilePath = "./.build/TestResults.xml")

- Work with abstractions, put implementations in Nuget package

Separation Boundary

Action



CAKE

 `build.ps1`

Orchestration

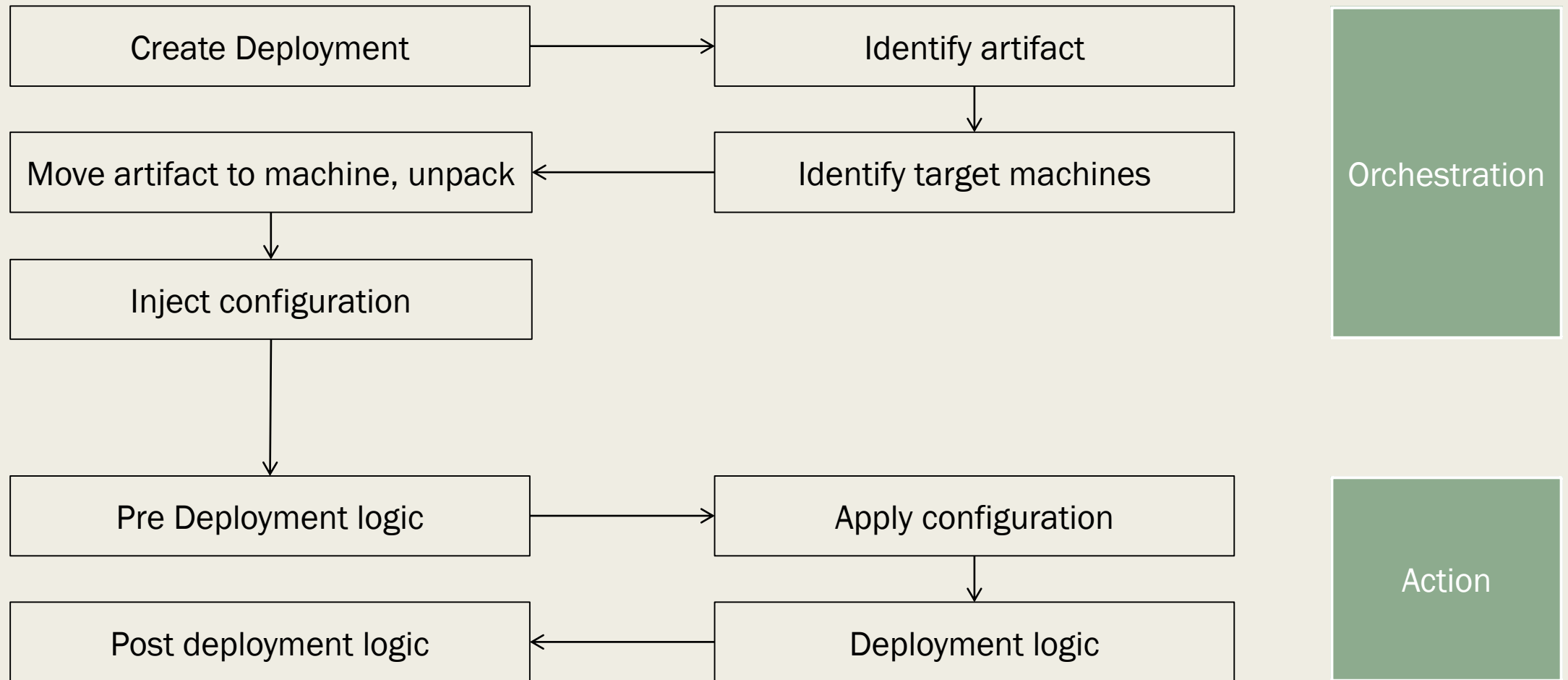


TeamCity

Build configs now very simple, single build step –
`.\build.ps1`

Other Tasks, longer running tests etc
`.\build.ps1 –target “Run_Integration_Tests”`

Build Artifact => Deployed



What's the problem?

- Lack of source control
- Inability to run locally
- Long and dangerous development feedback loop
- Application only configured on servers
- Vendor lock-in

Solution?

- Separate deployment action logic into code alongside application
- Language? Consider deployment platform
 - *Powershell, python, go*
- Version/Share code

EVERY PROBLEM



**NEEDS THE POWERSHELL
HAMMER**

A solution

- Nuget package (– what else?)
 - *Octopus packages on project level, suits nuget*
 - *Guide consumer, not restrict*
- Mimic deployment orchestration tool
 - *script for each deployment event*
 - Pre-Deployment.ps1
 - Deployment.ps1
 - Post-Deployment.ps1
 - *Configure.ps1 – XML transform conventions*
 - *Variables.json - file in place of octopus variables*

Deploy becomes

- Octopus orchestrates, powershell deploys
- Separation boundary is Variables.json and powershell scripts with environment argument
- Octopus template across projects
 - *If pre-deployment.ps1 exists, run*
 - *Substitute values into Variables.json*
 - *Run Configure.ps1, perform XML transforms etc*
 - *If Deployment.ps1 and/or Post-Deployment.ps1 exist, run*

Conclusion

- Teamcity and Octopus deploy are great orchestration tools.
- Important to separate build and deploy code from CI/Deployment tool
- Cake is great for build logic.
- Many deploy logic options.

Thank you

Any questions?