CONTINUOUS DELIVERY:
TALES FROM WINDOWSLAND

Rachel Laycock (@rachellaycock)
ThoughtWorks
Just use linux...
Continuous Delivery

Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
releasing frequently

feedback from users
reduce risk of release

John Allspaw: “Ops Metametrics” http://slidesha.re/dsSZIr
releasing frequently

feedback from users
reduce risk of release
real project progress
CD is BIG!

Organisational Alignment

Release Management

Architecture  Quality Assurance  Continuous Integration  Configuration Management  Data Management  Environments & Deployment
We will talk about...

- Build
- Deploy
- Provisioning Environments
Build and Deployment Pipeline

Here Be Dragons!

You are a pretty rubbish dragon slayer
<table>
<thead>
<tr>
<th>#</th>
<th>Revision</th>
<th>Date</th>
<th>Status</th>
<th>Environment</th>
<th>Tests</th>
<th>Date</th>
<th>Status</th>
<th>Environment</th>
<th>Tests</th>
<th>Date</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>42</td>
<td>e3091759c691...</td>
<td>3 days ago</td>
<td>Triggered by changes</td>
<td>Setup_Environment</td>
<td>Unit_Tests</td>
<td>Integration_Tests</td>
<td>Functional_Tests</td>
<td>Deploy_to_Staging</td>
<td>Setup_Environment</td>
<td>Unit_Tests</td>
<td>Integration_Tests</td>
</tr>
<tr>
<td>41</td>
<td>50de3d56e371...</td>
<td>6 days ago</td>
<td>Triggered by changes</td>
<td>Setup_Environment</td>
<td>Unit_Tests</td>
<td>Integration_Tests</td>
<td>Functional_Tests</td>
<td>Deploy_to_Staging</td>
<td>Setup_Environment</td>
<td>Unit_Tests</td>
<td>Integration_Tests</td>
</tr>
<tr>
<td>40</td>
<td>4bcb9678075a...</td>
<td>3 days ago</td>
<td>Triggered by changes</td>
<td>Setup_Environment</td>
<td>Unit_Tests</td>
<td>Integration_Tests</td>
<td>Functional_Tests</td>
<td>Deploy_to_Staging</td>
<td>Setup_Environment</td>
<td>Unit_Tests</td>
<td>Integration_Tests</td>
</tr>
<tr>
<td>39</td>
<td>23685b4abf82...</td>
<td>4 days ago</td>
<td>Triggered by anonymous</td>
<td>Setup_Environment</td>
<td>Unit_Tests</td>
<td>Integration_Tests</td>
<td>Functional_Tests</td>
<td>Deploy_to_Staging</td>
<td>Setup_Environment</td>
<td>Unit_Tests</td>
<td>Integration_Tests</td>
</tr>
<tr>
<td>38</td>
<td>6fabad234df...</td>
<td>4 days ago</td>
<td></td>
<td>Setup_Environment</td>
<td>Unit_Tests</td>
<td>Integration_Tests</td>
<td>Functional_Tests</td>
<td>Deploy_to_Staging</td>
<td>Setup_Environment</td>
<td>Unit_Tests</td>
<td>Integration_Tests</td>
</tr>
</tbody>
</table>
Basic deployment pipeline

Source code

**Commit stage**
- Compile
- Commit tests
- Assemble
- Code analysis

*Develop*
See code and test

*metadata*
*binaries*
*reports*

**Commit**

**Capacity stage**
- Configure environment
- Deploy binaries
- Smoke test
- Run capacity tests

**Production**
- Configure environment
- Deploy binaries
- Smoke test

*Version control*

**Operations**
- Perform push-button releases

*Testers*
Self-service deployments

*Developers*
See code metrics and test failures

Wednesday, 2 October 13
Story time
A little story about TFS...

SCM:
- Server side merging & no local commits

Team Build:
- no pipeline support
- difficult to adjust build step definitions
Compile and unit test

XML == Yuk!
Compile and unit test

PowerShell

PSake

```powershell
# Task definitions

# Default task
task default -depends Full

# Full task
task Full -depends Backup, ListFiles, Commit

# Backup task
task Backup {
    Write-Host "Backup"
    gci d:\temp\codebak | ? { !$_._.PsI:\container } | copy-item -destination d:\temp\codebak
}

task ListFiles {
    Write-Host "ListFiles"
    gci d:\temp\codebak | select -exp FullName | sc d:\temp\codebak\files.txt
}

task Commit {
    Write-Host "Commit"
    Start-Process GitExtensions.exe -ArgumentList commit, d:\temp\codebak
}
```

Code == yay!
Package
Deployment

Provisioning boxes

deployments
### Build and Deployment Pipeline

<table>
<thead>
<tr>
<th>Revision</th>
<th>Status</th>
<th>Revision</th>
<th>Status</th>
<th>Integration Tests</th>
<th>Status</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>42</td>
<td>Passed</td>
<td>e3091759c691</td>
<td>Passed</td>
<td>Passed</td>
<td>Passed</td>
<td>Passed</td>
</tr>
<tr>
<td>41</td>
<td>6 days ago</td>
<td>50de3d56e371</td>
<td>3 days ago</td>
<td>Passed</td>
<td>Passed</td>
<td>Passed</td>
</tr>
<tr>
<td>40</td>
<td>3 days ago</td>
<td>4bcb9678075a</td>
<td>4 days ago</td>
<td>Passed</td>
<td>Passed</td>
<td>Passed</td>
</tr>
<tr>
<td>39</td>
<td>4 days ago</td>
<td>23685b4abf82</td>
<td>4 days ago</td>
<td>Passed</td>
<td>Passed</td>
<td>Passed</td>
</tr>
<tr>
<td>38</td>
<td>4 days ago</td>
<td>6fabad234df</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Basic deployment pipeline

Artifact repository

Source code

Commit stage
- Compile
- Commit tests
- Assemble
- Code analysis
- Reports
- Binaries
- Metadata

Acceptance stage
- Configure environment
- Deploy binaries
- Smoke test
- Acceptance tests

Capacity stage
- Configure environment
- Deploy binaries
- Smoke test
- Run capacity tests

Production
- Configure environment
- Deploy binaries
- Smoke test
- Run acceptance tests

Operations
- Perform push-button releases
- Reports
- Metadata

Developers
- See code metrics
- Test failures

Testers
- Self-service deployments

**Deploy**

Wednesday, 2 October 13
Deployment

- MSDeploy
- PowerShell
- New Kids on the Block!

XML == Yuk!

Code == yay!

Tools == yay!
Deployment Pain

- One way to package to rule them all!
- So I have to install and configure this through the GUI?
- cmd line - c’mon!
- Package installs - where are you?
**MOAR PAIN**

- SQL Server & logic in your stored procedures

Provisioning boxes

deployments

Database migrations
Provision Environments

Provisioning boxes
Snowflake server
Story time
Provision Environment
Virtual Machine Images

- Development
  - Visual Studio
  - SQL Server
  - .NET 4
  - GIT

- CI
  - Go
  - .NET 4
  - GIT

- Prod-like
  - .NET 4
  - SQL Server
  - 3rd Party software
virtualization

great for creating production-like test envs, highly parallel testing

cloud is great for utility computing and scaling on demand

most real systems will be heterogeneous

use virtualization with Puppet, Chef & PowerShell
Infrastructure as Code
Infrastructure Pain

- Proprietary Software
- Windows Versions
- Remoting
- Registries
- Windows Updates
- ISO’s
- Cost

It looks like you are trying to configure some servers...
Now it’s your turn...

- Don’t be afraid of non-ms tools
- Leverage existing languages
- Contribute to OpenSource ALT.NET projects
- Just do it!
Happy endings?

• It can only get better...
• New tools all the time
• Maybe we can self select away from proprietary software...
• Figuring out a problem is the first step to fixing it
Questions?
Resources

• Continuous Delivery, Dave Farley & Jez Humble
• Continuous Integration, Paul Duvall
• Beautiful Builds, Roy Osherove
• Release IT, Michael Nygard