
Automated Deployment With Umbraco

Mark Bowser



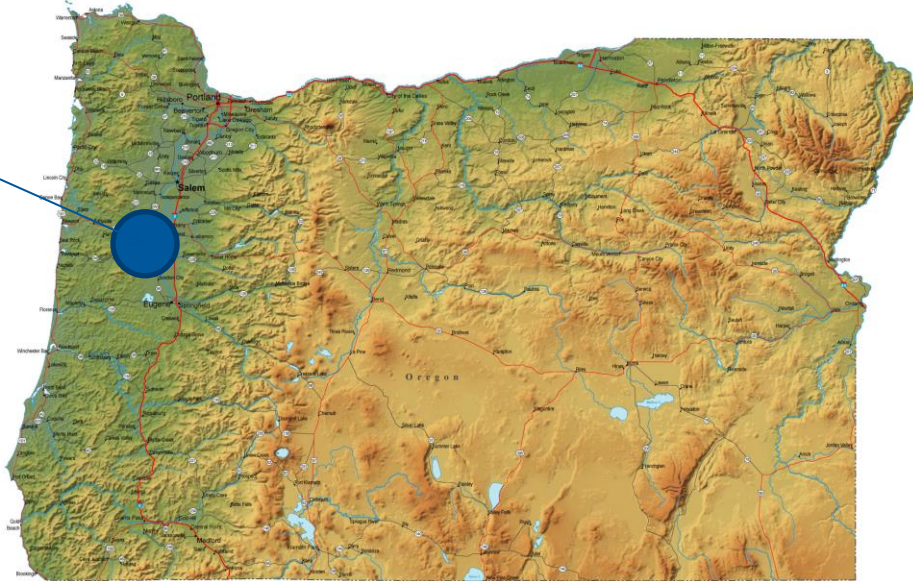
uWestFest '16 - San Diego, CA



Mark Bowser

Full Stack Web Developer

Corvallis



Goals

Not go too deep into technical side

Show how the pieces work together

Inspire people to try it out

What does it take?

Branching strategy

TeamCity

Octopus Deploy

uSync?

MORE !?

Why bother?



Consistency

Reinforces good habits

Gateway to automated testing

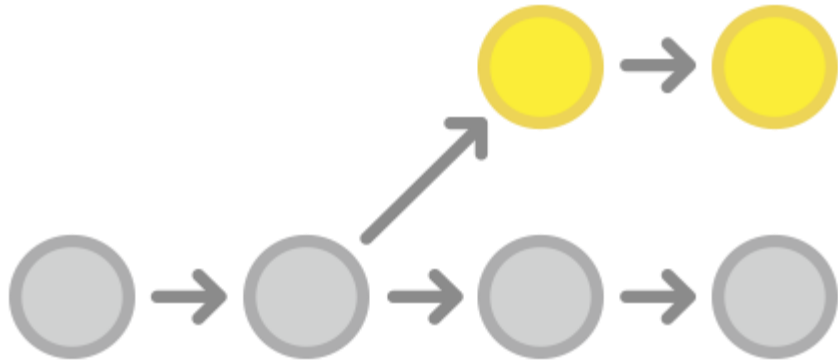
Speedy deploys

Easy setup*



Branching Strategy

Feature Branches



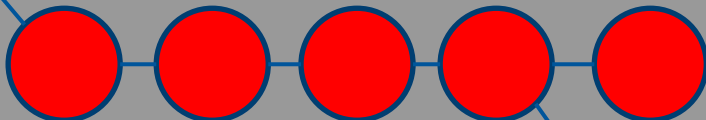
Solves problems with:

- Large features
- Unapproved features

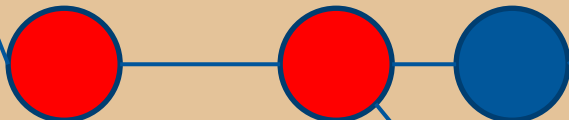
master



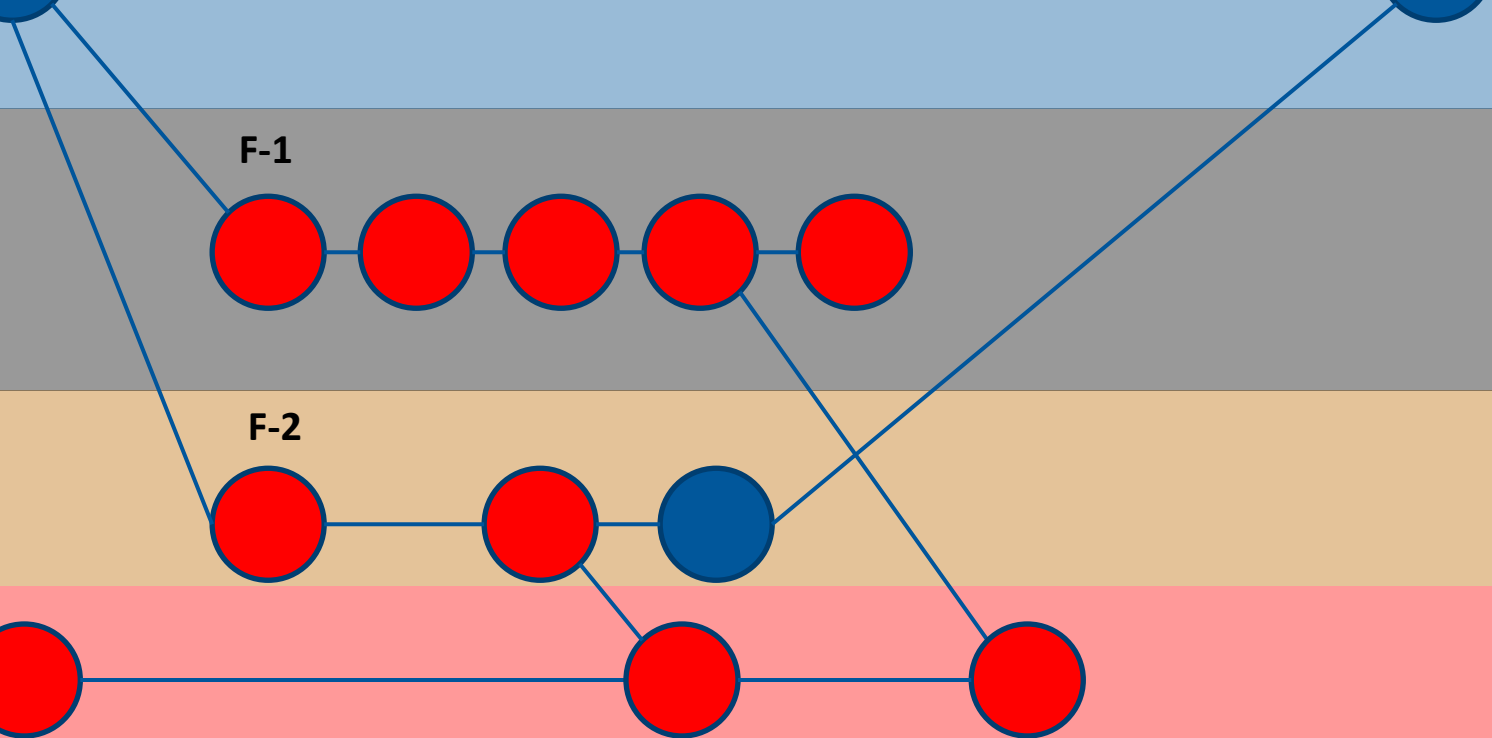
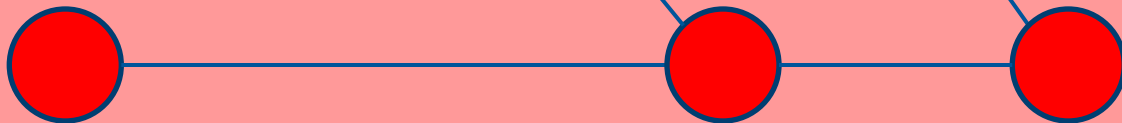
F-1



F-2



dev



☹ Branch Naming

- 🕒 [] Check to see why Home Page header his different than all other pages Mark Bowser 24 Feb 2016 8
- 🕒 ***** Home Page Small Redesign for Alan ***** 1
- 🕒 [CR] Marquee image changes (6-7) Alan Ballard 2
- 🕒 [CR] Reposition Promo Boxes (2-4) Alan Ballard 2
- 🕒 [CR] Footer map changes (4) Alan Ballard 3
- 🕒 [CR] Advantage box becomes footer accordion in mobile (4) Alan Ballard 3

homePageRedesign	6	2016-02-25
responsive	20 30	2016-02-12
brochureGate	23	2016-02-12
siteSearch	29 1	2016-02-10
inconsistentHeader	32 1	2016-02-10

😊 Branch Naming

- 🕒 E-24 Pull CSS out of the content nodes and into SASS 💬 1
- 🕒 E-25 Replace uQuery and NodeFactory with new Umbraco API
- 🕒 E-26 Implement "Examine" for blog search functionality

E-2	19	2016-02-03
B-16	14	2016-02-02
B-17	14	2016-02-02
B-3	15	2016-02-01

Build Server



Build Server

Pulls from source control

Builds project

Runs tests

Deploy?



Projects and Build Configurations

▶  Planar

▼  ProWorks ProWorks Internal Projects

▼  Main Site

 dev

 live

▼  Proworks Internal Dashboard

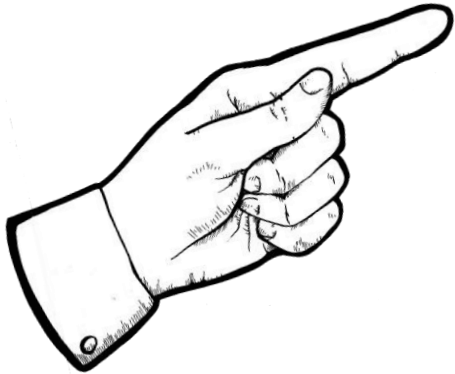
 dev

Nested Project Structure

Build Configurations

- Version Control Settings
- Build Triggers
- Build Steps

Version Control Settings



Git

Mercurial

TFS

Subversion

CVS

Other...

Build Triggers



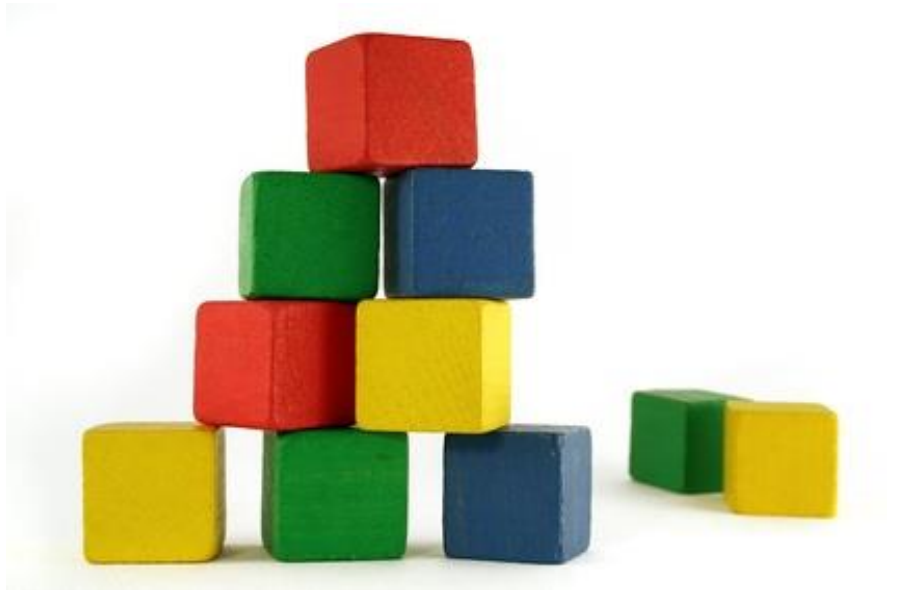
Manual (no trigger)

On commit

Scheduled

Other?

Build Steps



MSBuild

Run Unit Tests

Run Commandline?

**So how do we actually
deploy stuff?**



MSBuild + WebDeploy

```
/p:Configuration=Debug
/p:VisualStudioVersion=12.0
/p:DeployOnBuild=True
/p:DeployTarget=MSDeployPublish
/p:MsDeployServiceUrl=123.456.789/msdeployagents
ervice
/p:AllowUntrustedCertificate=True
/p:MSDeployPublishMethod=RemoteAgent
/p:CreatePackageOnPublish=True
/p:DeployIisAppPath="mywebsite.proworks.com"
/p:IgnoreDeployManagedRuntimeVersion=True
/p:FilesToIncludeForPublish=AllFilesInProjectFolder
/p:SkipExtraFilesOnServer=True
/p:ExcludeFoldersFromDeployment="App_Data\Logs;App_Data\preview;media"
/p:IncludeSetAclProviderOnDestination=False
/p:AutoParameterizationWebConfigConnectionString=False
```

Enter additional command line parameters to MSBuild.exe.

1. Install Web Deploy on server
2. Deal with windows service has permissions
3. Create users on source and dest machines
4. Deal with mostly vague TeamCity errors

I no longer recommend Web Deploy

Complex. Easy to miss little things.

Vague error messages

No easy rollback

There is a better way!

Octopus Deploy





Octopus Deploy



Not a build server

Consumes nuget feeds

Handles deployment

Why Octopus?

Easier than Web Deploy

Good at rolling back

Very good documentation

What does an Octopus deploy look like?

1. Consumes nuget feed
 2. Creates Release
 3. Deploys to new folder
 4. Points IIS to that folder
 5. Fixes permissions
-

TeamCity + Octopus Deploy Implementation

**With our powers
combined!**



Four Steps

1. Umbraco project configured to be packaged into NuGet
 2. Media set up in Virtual Directory
 3. TeamCity = NuGet Server
 4. Octopus deploys NuGet packages
-

Preparing Umbraco for NuGet packaging

1. Nuget OctoPack
2. Create nuspec file

“Install-Package OctoPack”

Umbraco Media in Virtual Directory

1. New SharedMedia directory outside of umbraco
2. New Virtual Directory pointing to SharedMedia

TeamCity NuGet Server

1. Install Octopus Deploy Integration plugin
 2. Enable NuGet Server
 3. OctoPackEnforceAddingFiles=true
 4. MSBuild step should check “Run OctoPack”
 5. New “Octopus Deploy: Create Release” build step
-

Prepare Octopus Deploy

1. Add an “External Feed” for TeamCity Nuget Server
 2. Script template for fixing Umbraco permissions
 3. New “Deploy Package” step
 4. New “Fix Permissions” step
-

TeamCity + Octopus VS UaaS?

Pros of UaaS:

- Really easy to spin up new sites
- Handles content deployment as well as code deployment

Pros of TeamCity + Octopus:

- More control
 - Probably supports more exotic setups
-

Questions?

Contact me at:

mark@proworks.com

More on Octopus

Big pieces of Octopus:

- Tentacles
 - Releases
 - Environments
 - Lifecycles
 - Projects
 - Channels
-