AQUENT

A dotCMS Build and Deployment Lifecycle

How Aquent approaches development with dotCMS

AQUENT

Who am I

Christopher Falzone - cfalzone@aquent.com

https://www.facebook.com/chris.falzone
https://www.linkedin.com/in/cfalzone

Aquent - http://aquent.com

AQUENT

Part 1 - Website Development

dotCMS Environments

- Development Server
  - Developer Playground
  - Features developed and tested here
  - Will be replaced in the future with local dev environments
- Staging Server
  - Finished features deployed here for QA
  - Content Authored here
    - Push Publishing to Dev and Production
- Production Servers
  - Released Features here

AQUENT

Source Control

- All VTL/SASS/JS in Git on Github
  - Parse Code Widget
- Environment Branches:
  - Stag / Master (Production)
- Feature Branches:
  - Encapsulated Work
  - Pull Requests = Code Review
- Releases / Hotfixes

Repo Layout:

- src/
  - aquent.com/
    - js/app.js
    - sass/**.scss
    - pages/**.html
    - bower_components
  - vitamintalent.com/
- dist/ (mimics what is in webdav)
  - aquent.com/
  - vitamintalent.com/

AQUENT

Local Preview

- Static HTML w/ Live Reload
  - Currently Powered by Panini [LINK]
  - Allows Devs to have live preview of CSS/JS/HTML changes
  - Less than ideal - double development, html gets out of date if not updated
- Plans to move to a local dotCMS instance

AQUENT

Building Source Files

- Toolchain
  - Node.js via NVM
  - NPM
  - Gulp
  - Bower
  - Foundation
- SVG and SASS Processing
- CSS Combining and Minification
- JS Combining and Minification

AQUENT

Deploying Code

- Dev Server
  - webdav
- Jenkins
  - Github webhook trigger
  - Deploys files over Rest API
  - Gotcha #1: Finding a File’s ID
  - Gotcha #2: Deleting A File

AQUENT

Demo Time

AQUENT

Issues

- Can't release when something fails QA on Staging
- What is Merged is Deployed
- Not currently Handling Moved/Renamed files well
- A 5 minute fix takes more than 5 minutes

Part 2 - Plugin Development

AQUENT

Source Control

- Deja Vu - Git and Github
- Git Flow (A successful git branching model) [LINK]
- All work done in develop branch
- Release Branches into Master
- Releases and Snapshots stored in Artifactory Repo

Building the Plugin

- Gradle Jar
- Dependency Jars are included in the OSGI jar automatically

AQUENT

Deploying the Plugin

- Github webhook triggers Jenkins Job
- Build Pipeline:
  - Build Dev
  - Deploy Dev
    - Restart OSGI Framework
  - Deploy Stag
    - Restart OSGI Framework
  - Build Master
  - Deploy Production
    - Restart OSGI Framework

AQUENT

Demo Time