Staging Drupal

Change Management Strategies for Drupal

DrupalCamp CT 2010

Introductions

Erich Beyrent

http://twitter.com/ebeyrent

http://drupal.org/user/23897

My Modules

- Permissions API
- Crowd SSO
- LDAP Extended Groups
- Search Lucene Biblio
- Search Lucene Attachments
- Search Lucene OG
- Visual Search API

Agenda

- Playing well with others
- Managing code changes
- Managing database changes
- Deployment strategies

The only thing that's constant is...

Staging? What is that?

Staging? What is that?

"Staging" is the process of delivering changes from one environment to another.

DO NOT WANT

- I develop on the live server.
- I work alone.
- Backups?

The Manual Process

- Point.
- Click.
- Wait.
- Rinse and repeat.

TOTAL FAIL

Why it fails

- It's tedious.
- It's time-consuming.
- It's error-prone.
- It's risky.

A traditional approach

- Use source control

Use source control

- Essential to the development and staging process
  - Creates a flow from Dev → QA → Production
- Manages changes to the code over time
- Use a standard repository layout consisting of “tags, branches, trunk”
- Use multiple repositories to separate core code from project code

What goes into source control

- Code, configuration, theme-based files
- Use source control templates
- What about database snapshots?
- DON'T manage user-generated content in source control

A traditional approach

- Use source control
- Manage changes in code

Manage database changes

- Export your views, panels, and content types to code
- Use exportables and ctools to export other data
- Manage configurations with Strongarm
- Use Permissions API for roles and permissions
- Use Features!

Why should we do this?

- Multiple environments need to be updated
- Allows for a phased approach to change management
- Saves time and money
- Is fully testable and reproducible
- Minimizes downtime and helps manage expectations

Source Control and Update Scripts: Friends with Benefits

- All changes can be viewed, compared, and reverted in version control, which helps with debugging
- Deployment processes are reproducible, and become part of QA
- Changes become portable and can be easily replicated in multiple environments

Choose Wisely

- How many changes are there?
- How long will it take to write the update code?
- How many environments need to be updated?

A traditional approach

- Use source control
- Manage database changes in code
- Use deployment tools

Use deployment tools

- Source control as a deployment tool
  - SVN update, post-commit hooks
  - GIT push
- Make, rsync
- Aegir, Phing/Ant, Capistrano, Hudson

The process

- Import current database snapshot into your sandbox
- Update your codebase
- Develop, commit, update
- Promote changes to QA environment, test
- Tag and release
- Drink beer

The specifics

- Changes are coded in a module
- The module implements `hook_update_N()`
- *BUT WAIT, THERE'S MORE*
  - Installing a module does NOT invoke any `hook_update_N()` implementations
  - `hook_install()` and `hook_update_N()` must be in sync

The specifics

- Use the APIs (node, user, etc) instead of writing queries
- Use smaller update functions for maintainability

The specifics

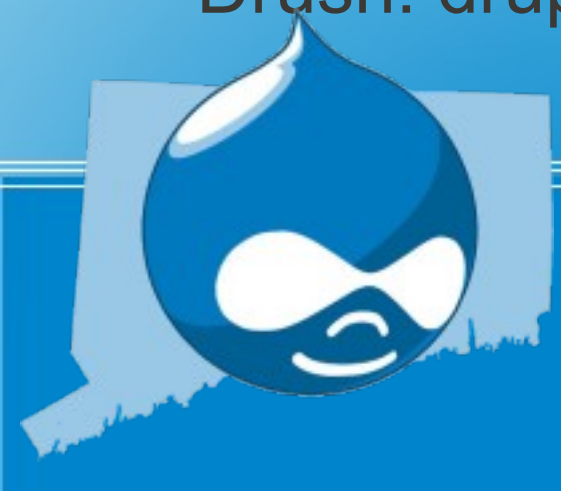
- Use the APIs (node, user, etc) instead of writing queries
- Use smaller update functions for maintainability

The Problems

- Inconsistent and incomplete export functionality
- Primary key issues with content
- Dependencies
- Requires technical proficiency

Drupal Tools

- Devel: Generate: drupal.org/project/devel
- Demonstration Site: drupal.org/project/demo
- Node Export: drupal.org/project/node_export
- Permissions API: drupal.org/project/permissions_api
- Views Export: drupal.org/project/views
- CCK: Content Copy: drupal.org/project/cck
- Drush: drupal.org/project/drush



More Drupal Tools

- Variable Dump: drupal.org/project/variable_dump
- Exportables: drupal.org/project/exportables
- Transformations: drupal.org/project/transformations
- Migrate: drupal.org/project/migrate
- Deploy: drupal.org/project/deploy
- Features: drupal.org/project/features
- Strongarm: drupal.org/project/strongarm

Questions?