

Drupal: Challenges, and Techniques for Managing Change

CPOSC '09 - Introduction

- Christian Pearce
- Work for xforty technologies
- Long time FOSS advocate
- Husband and Dad

Goals



- Cover different types of change
- Compare and contrast solutions to deal with change (some)
- A showcase of what is available

Change

- Why so much change?
 - Release Cycle
 - Site Configuration (Dev-> Production)
- Is this a good thing?
 - Fresh Technology
 - Bug and Security Fixes
 - No Code Baggage
 - Superior Code Base



Challenge



- N Teir
- Dev -> Production Env
- Databases
- Code
- Local files

Type of Change

- Code Changes (Developer)
- Configuration Changes (Site Administrator)
- Content Changes (Editor)



Code Changes

- Drupal gets EOLd about every 12-24 months
- They keep one previous version active
- <http://drupal.org/node/65922>
- Security and Bugs fixes, for modules and core
- Plus your own

Configuration Changes

- Changes to functionality
- Usually performed by the developer, or site administrator
- Consists of site building and configs
- Menus, Content Types, Views, Blocks, etc

Content Changes

- Changes to the content (everything else)
- Managed by Editor
- Not always possible to provide a UI, for editor, so handled by Developer
- Allow for online or offline processing
- Keeping Development in sync with Production

What can we do about it?

- Traditional Tools
- Drupal Tools



Characteristics

- Difficulty
- Audience
- Type of solution (ie code change, content change, etc)
- Maturity/Fragility
- Completeness/Limitations
- Technical Solution (Form API, or array playbacks, Custom Code)
- Usage Statistics

Pragmatism



- Decide for yourself

Traditional Tools...

Source Code and Shell Scripts (Code)



- http://www.davidgrant.ca/maintaining_v
- Put upstream code in vendors branch
- Put your code in separate branch
- Edit your code, run merges

Manual Process (Code, Config, Content)

- Build script
- Record your changes on paper
- Complete solution, very manual



Sync dev to prod using Toad (Config, Content)

- <http://codebaboon.com/method-synchronising-database-changes>
- Even and Odd numbers for dev and prod
- Hack code (5.x), mod mysql variables
- Toad for MySQL -- diff/patch
- Tough problem, creates bizarre solutions

Drupal Tools...

Import/Export Utilities (Config,Content) (Demo)

- Improves efficiency, reliability of Conventional methods
- Import Export API
- Individual Import Exports
 - Views
 - Content Types (CCK)
 - Stable
 - Other modules might have similar functionality

Drush (Code, Config)

- Drush dl *module_name*
- Drush enable *module_name*
- Drush sync

Multisite (Code)

- Put all your eggs in one basket
- Keeps modules and core to a single source
- Good for simple sites
- Sets the stage for AEGir in the future

AEGir (Code)(Demo)

- <http://groups.drupal.org/aegir-hosting-system>
- <http://developmentseed.org/blog/2009/may/20/aegir-scratch-installing-aegir-02-rc-1>



Deploy (Config, Content)(Demo)

- Deployment plans, Straight Deploy
- What can you push? Content, Types, Node References
- It does creates/updates
- Solves some Config and most Content Change

AutoPilot (Code,Config,Content)

Simplify
drupal build
management.

(autopilot makes it easy)

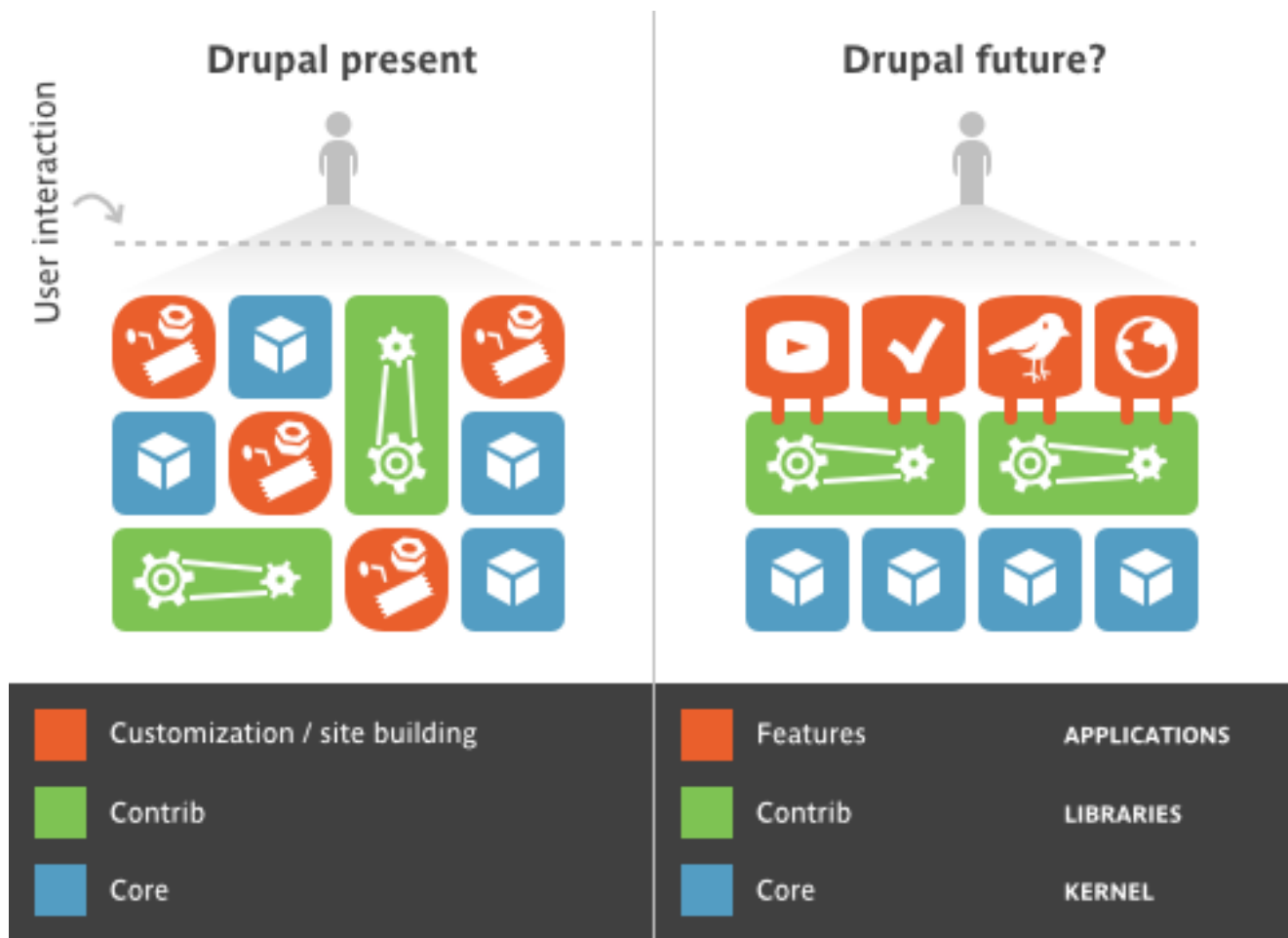


Context (Config)

- <http://developmentseed.org/blog/2008/apr/09/context-ui>
- Create sections,
- Evolved UI for blocks, menus, panels

Features (Config)(Code)(Demo)

- <http://developmentseed.org/blog/2009/may/29/making-and-using-features-drupal>



Spaces (Config)

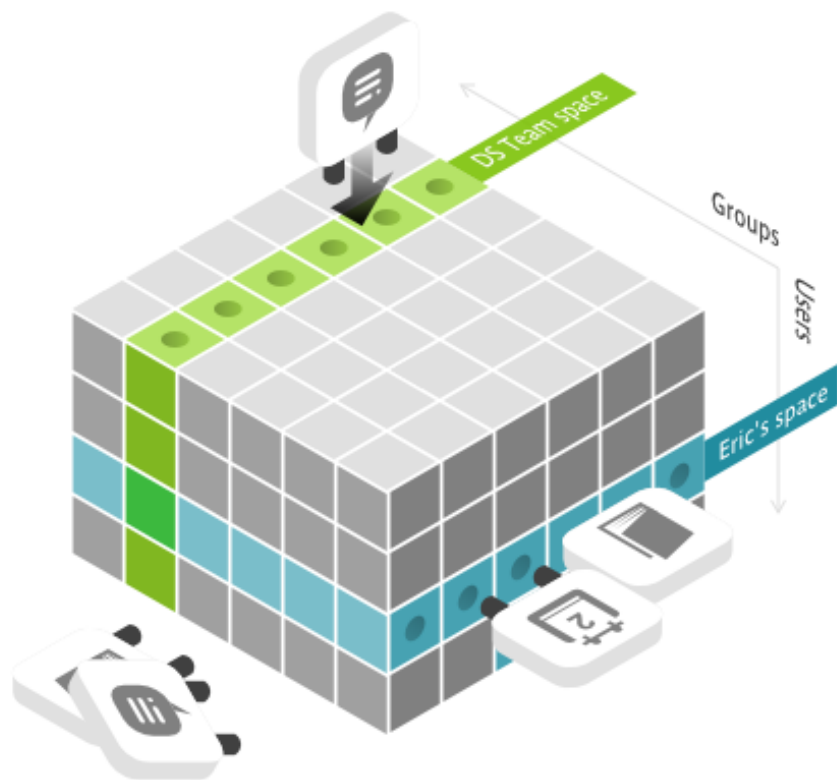


Fig 2. Spaces features can be used interchangeably between different space types on a single site.

- <http://developmentseed.org/blog/2009/j>
- Carve your site up into zones
- Turn on or off Features per Space using local overrides

So much more...

- project/migrate
 - <http://dc2009.drupalcon.org/session/migration>
- project/node_import
- Just go out and read, point is lots of solutions need to DYOD.

Why all the options? Why bother?

- There is no silver bullet
- Different people have different requirements
- By hand sucks after a while
- It is error prone



What do I use?



- Traditional Methods
- Looking at Aegir, and Contexts/Features (over Patterns)
- Keep in mind most of what I showed is cutting edge

Resources

- <http://drupal.org/taxonomy/term/64>
- <http://drupal.org/node/417192>
- <http://groups.drupal.org/change-management-systems>
- <http://groups.drupal.org/packaging-deployment>

Other bits

- Local Meetup
- <http://groups.drupal.org/central-eastern-pennsylvania>

Questions?