

Modern Drupal Development

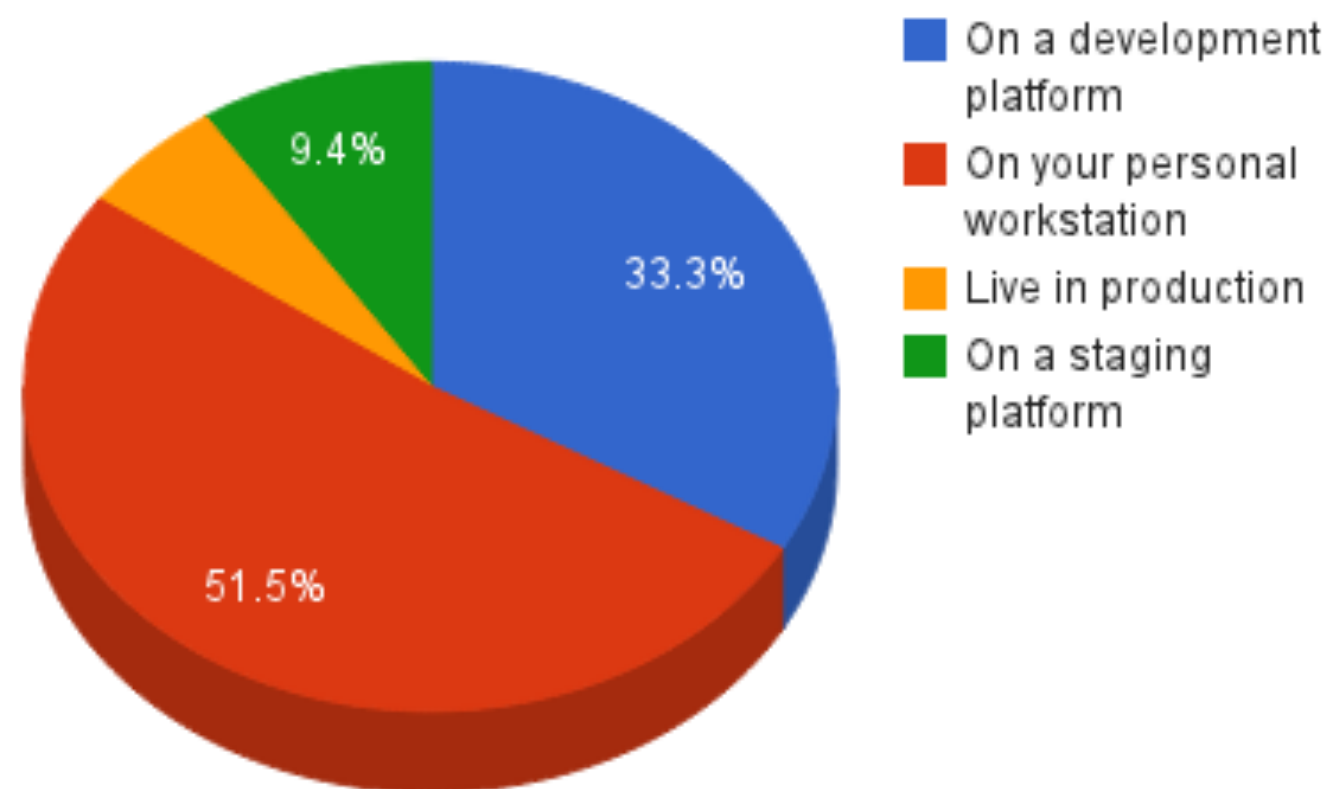
Virtualization & Version Control

Obsolescence

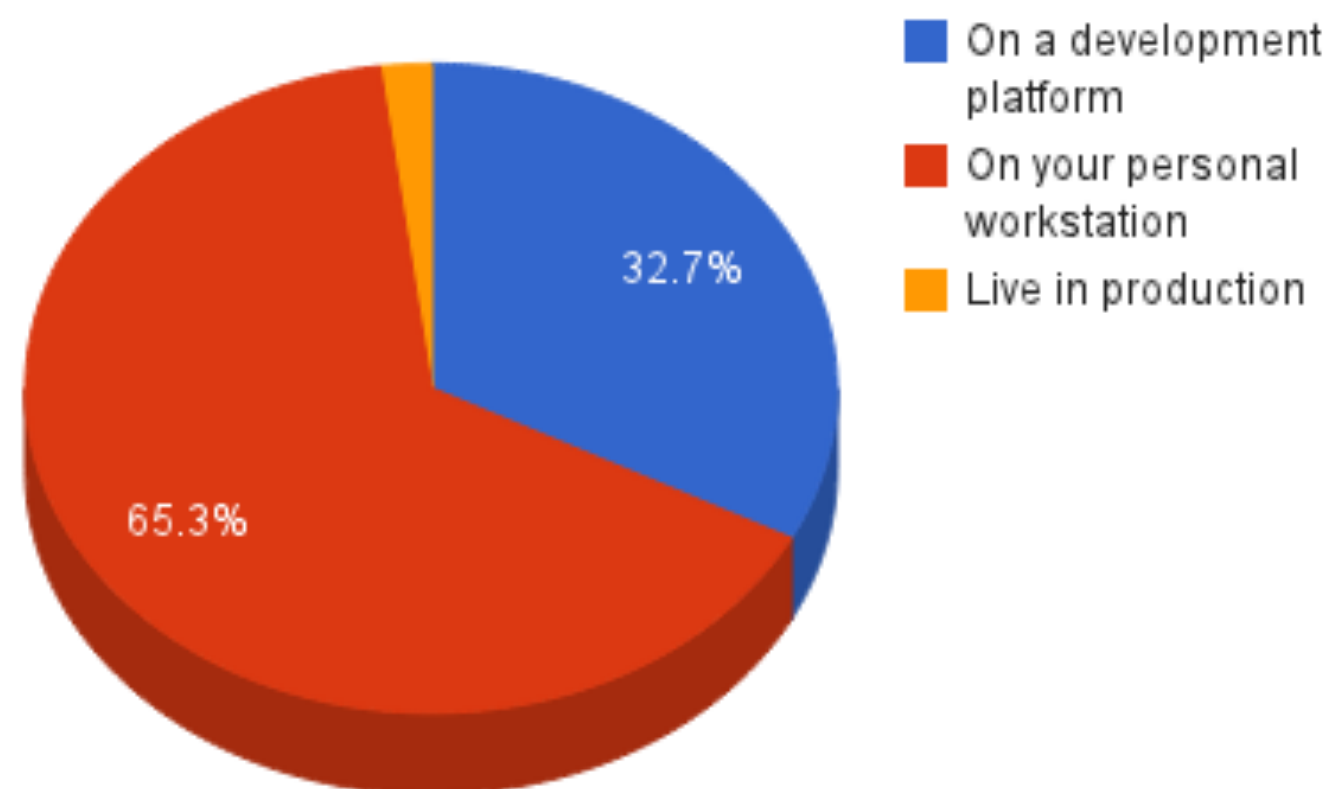
- CESD is falling behind community standards
 - Feature development
 - Version control
 - Collaboration

What do others do?

2012 Where do you develop your site usually



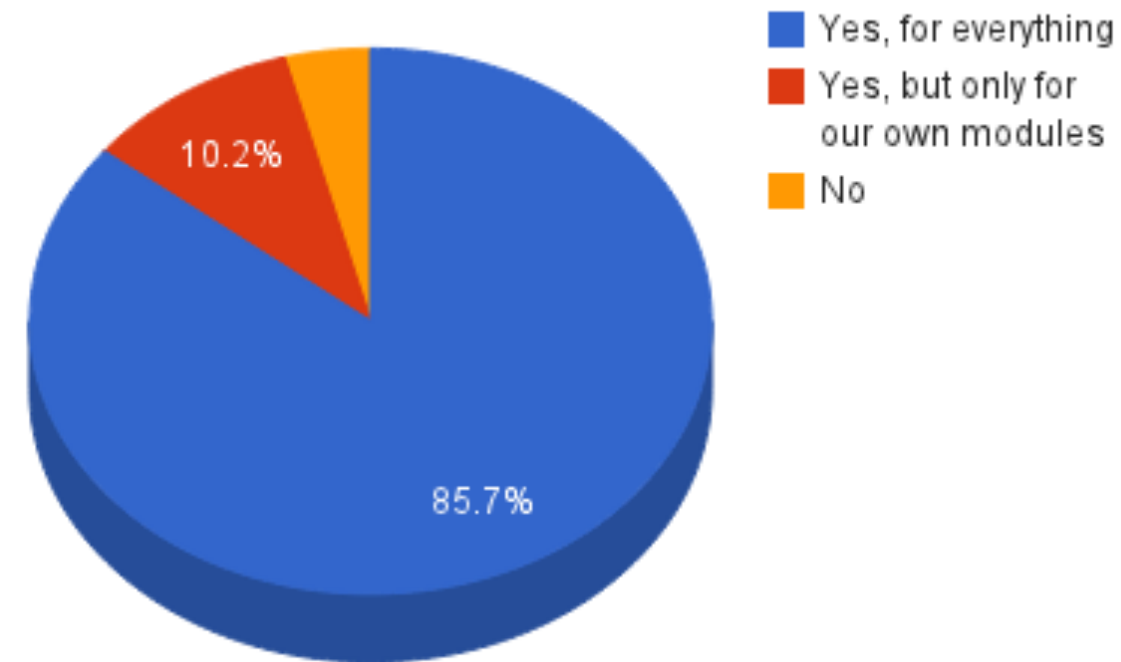
2014 Where do you develop your site usually



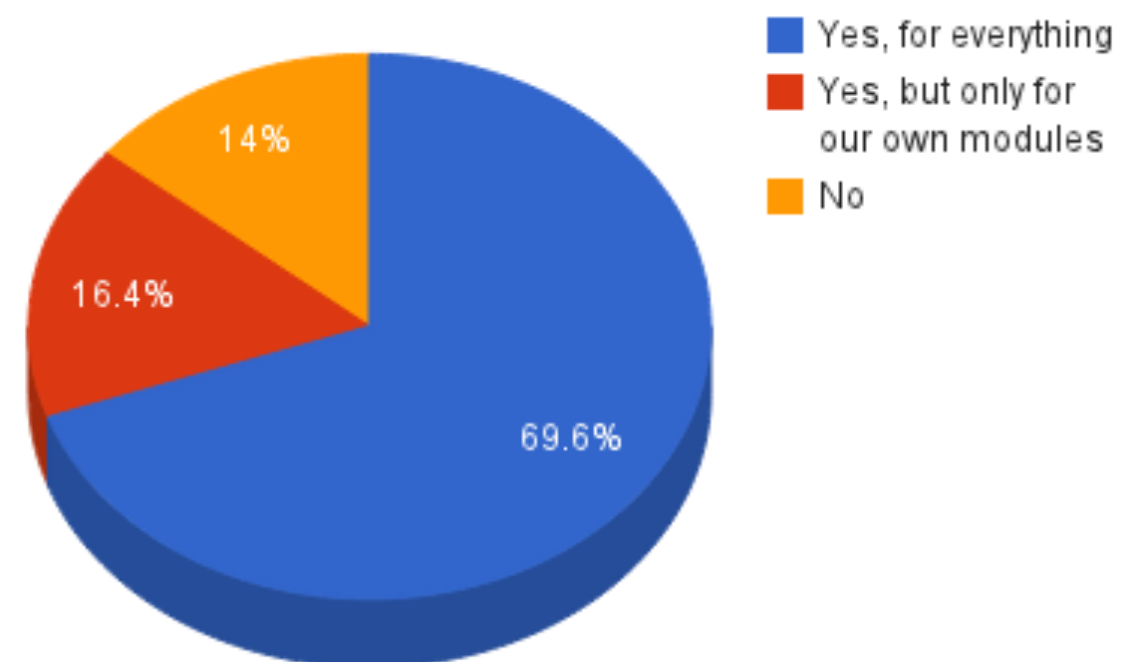
@CESD

Working on live production sites

2014 Do you use any form of version control ?



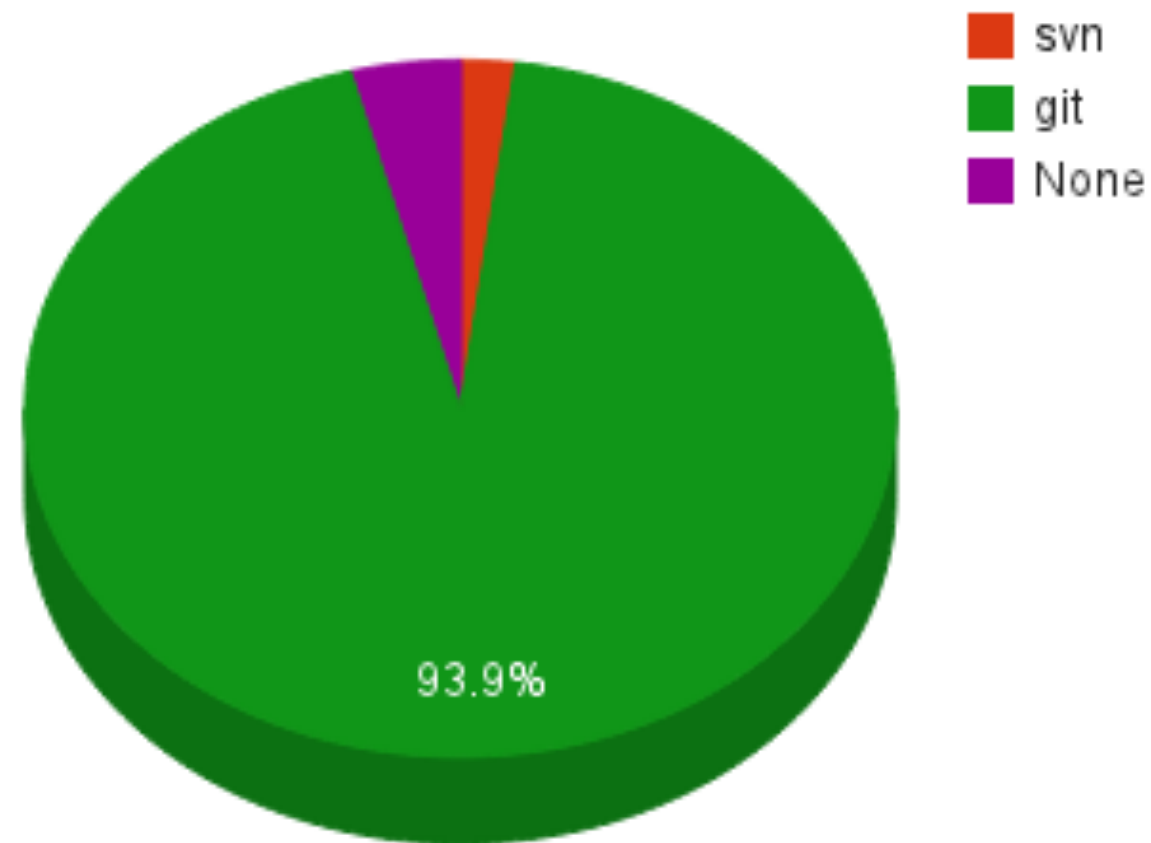
2012 Do you use any form of version control ?



@CESD

No version control for Drupal

2014 What version control system do you mainly use?



Now: Shared

- Last-writer problem
- Limited experimentation
- Unpredictable state
- Version control avoidance
- Performance
- All-or-nothing reversion

Alternative: Local

- Heterogeneous
- “Works on my machine” problem
- Everyone becomes a sysadmin

Solution: Virtualization

- Standardizes environment
- Simplifies configuration
- Enforces version control
- Environment = temporary / Code = permanent

How

- Personal virtual machines (Vagrant)
 - 1+ VM / project
- Compartmentalized features
- Deploy & collab. via code

Vagrant

- Windows/Linux/Mac
- Track Vagrantfiles in Git
- VDD (Vagrant Drupal Dev.)
 - Drush is your friend
- Customization via Chef

Code as Config

- Drupal development is ONLY code
- No more database dumps/loads
- Must use CINC or Features
- Tickets cannot be closed without git ref to code

Deployment

- Features and Components tracked in Git
- Push to development group repo
- Pulled by OIT admins for deployment