### Drupal In Gov

Lessons learned from years of doing Drupal in the public sector



### About us

#### Joshua Li

- Drupal Developer
- Drupal.org ID: RLi
- Email: josh.li@datacom.com.au
- Experience: 4 years

### Chris Cook

- Drupal Specialist (Technical Lead)
- Drupal.org ID: Feng-Shui
- Email: chris.cook@datacom.com.au
- Experience: 6 years



### Topics

- Requirements development
- Project politics
- On site or off site?
- Let developers develop
- Knowledge transfer
- Test, test, test
- Infrastructure
- Plan for organisational change
- Module shopping & performance
- Documentation



#### Requirements development Would you build a house without detailed plans?



### Where do we start?

### Selecting a CMS

- High level requirements developed
- Include infrastructure, security and high-level functionality, eg:
  - Integrates with Active Directory
  - Can enforce role based permissions
  - Can publish XML feeds
- These are used to select a CMS
- We choose Drupal because it's awesome
- Requirements done!



### So what's the problem?

#### In a word: Detail.

- High level requirements do not contain enough detail to actually build a site
- "Can publish XML feeds" how many? what type of content? sorted by? URLs? access control?
- Hundreds of small decisions like these are required to build a Drupal site

### What can you do?

Let your consultant fill in the blanks or make it up as you go

- Very common solution
- Slow, unplanned and uncontrolled progress
  - Experienced Drupal consultants won't do a bad job, but, you' re putting them in control of the deliverable
- If the deliverables are "wrong" it's not good for morale, deadlines or the budget

Develop detailed, low-level requirements that eliminate all ambiguity and ensure the high-level requirements will be met.





DATACOM

### Sounds like a lot of work...

- It is! This is the hardest, but most important phase of your project
- Building in Drupal is the easy bit
  - Even the most complex requirements are achievable when planned well
- You have more chance to:
  - Deliver a site that meets your user's needs
  - Deliver your project on time and on budget
  - Come out of an IT project at least as sane as you went in
- I have never seen a Drupal project with detailed requirements (that were followed) delivered late or over budget



#### **Project politics** Help your developers understand the project's politics.



### Stakeholders' view







### Inform your consultant

## Talk to your consultant about the attitude of stakeholders

- This will allow your consultant to:
  - Tailor any information with sensitivity to audience
  - Ensure they don't inadvertently commit themselves or you to specific functionality





#### On site or off site? Where do you want your consultants located?



### On site

#### You may feel more in control, but:

- Productivity can drop
- Consultants can find themselves
  - In conversation with stakeholders
  - Writing documentation for executive level meetings
  - Answering "Googlable" questions from colleagues

#### Make sure you:

- Determine the tasks for consultants
- Confirm the schedule for each task
- APS-ify consultants for documenting



### Off site

#### Make sure they are working

- Participate in scrum (you should being doing this anyway)
- Check the code repository
- However:
  - Developers may not make any obvious (to you) progress on issues
  - Developers need to explain the situation



#### Let developers develop Would you let your bricklayer pick your curtains?



### **Developers develop**





#### *Knowledge transfer This is not The Matrix, knowledge transfer takes time.*



### Do you need it?

Who is going to manage the website in future?

• Ask their opinions

Prioritise it

• It can be forgotten





### What to transfer?

- Infrastructure management
- Site building
- Module development
- Regression testing
- Troubleshooting WSoD (White Screen of Death)
- Source control and release management

Friday afternoon "master class"



#### Test, test, test How confident are you that this release won't cause errors?



### Errors

#### **Regression errors**

- A change 'here' broke something totally unrelated over there
  - We've seen an update to the **UUID** module prevent the **Date** module providing **Tokens** to **FileField Paths** module. Who was the culprit? The **Entity API** module. Complex? Yep!
- Very hard to track down

#### **Development errors**

• How do you know that new functionality works?





### Test

### Simple test

- Developers build tests as they build site functionality
- Ensure the site is functioning correctly
- Drupal tests itself automatically!

#### Time consuming

- It takes developers time to write tests
- Won't be efficient if quick solution is needed

### But!

- No more testing by hand, saves hours (of testing and troubleshooting)
- Reproducible



#### Infrastructure VHS or Betamax?





### Do you have the infrastructure?

Make sure your consultants talk to your infrastructure team

- Your consultants may use different infrastructure
- Make sure your consultants know the infrastructure you are going to use
- Does your infrastructure team know what they are doing?
- Drupal often requires changes to default settings
- Let your infrastructure team decide based on their expertise



#### Plan for organisational change What happens when you get 'MoG'ed, again?







#### Module shopping & performance How many modules is too many modules?



# How many modules do you have?

#### There is no limit...

- Meet your user requirements first
- Then your performance requirements

So why do we care how many?





### Performance

The more modules we have, the slower is the site?

- Yes but, be clear about performance requirements
- Performance requirements
  - Do you need to be able to serve 30,000 hits "a day" or 30,000 hits between 9am and 5:00pm?
  - Does the system need to be available 24 hours a day, or just between business hours?
- Cache everything
- Coding standards



#### Documentation Why did we do it that way?



### Developers don't like it

So, help them understand why it's important to you!

- If they understand, they will be more willing to help
- We often find that documentation sits in a shelf and is never used hence the reluctance

#### Make sure your developers document:

- In the standard for your Department
- From the beginning of the project

#### Does it make sense to the intended audience?



### What to document?

- What requirements each module supports
- Module dependencies
  - Really important to help ensure that modules are uninstalled if they are no longer needed
- Patches
- Comment in code





### Questions





### Thank you





