A Workflow for Release Management

Baking awesome into your work

ANDREW BERRY * @deviantintegral * WATERLOO DUG MARCH 2013

Lullabot™

Consulting | Development | Training

Lifetime · MTV Music Television · Fast Company · DC · Sony Music · verizon · IBM · BBC · GRAMMY Awards · O'REILLY® · The Economist · Walmart Save money. Live better. · The Washington Post · Turner · Zappos.com POWERED by SERVICE™

How do Drupal sites get managed and deployed?

- Install a module from drupal.org
- Click around in the administration screens
- Create some new Fields or Views for the new functionality
- You have a real website that the world can come see!

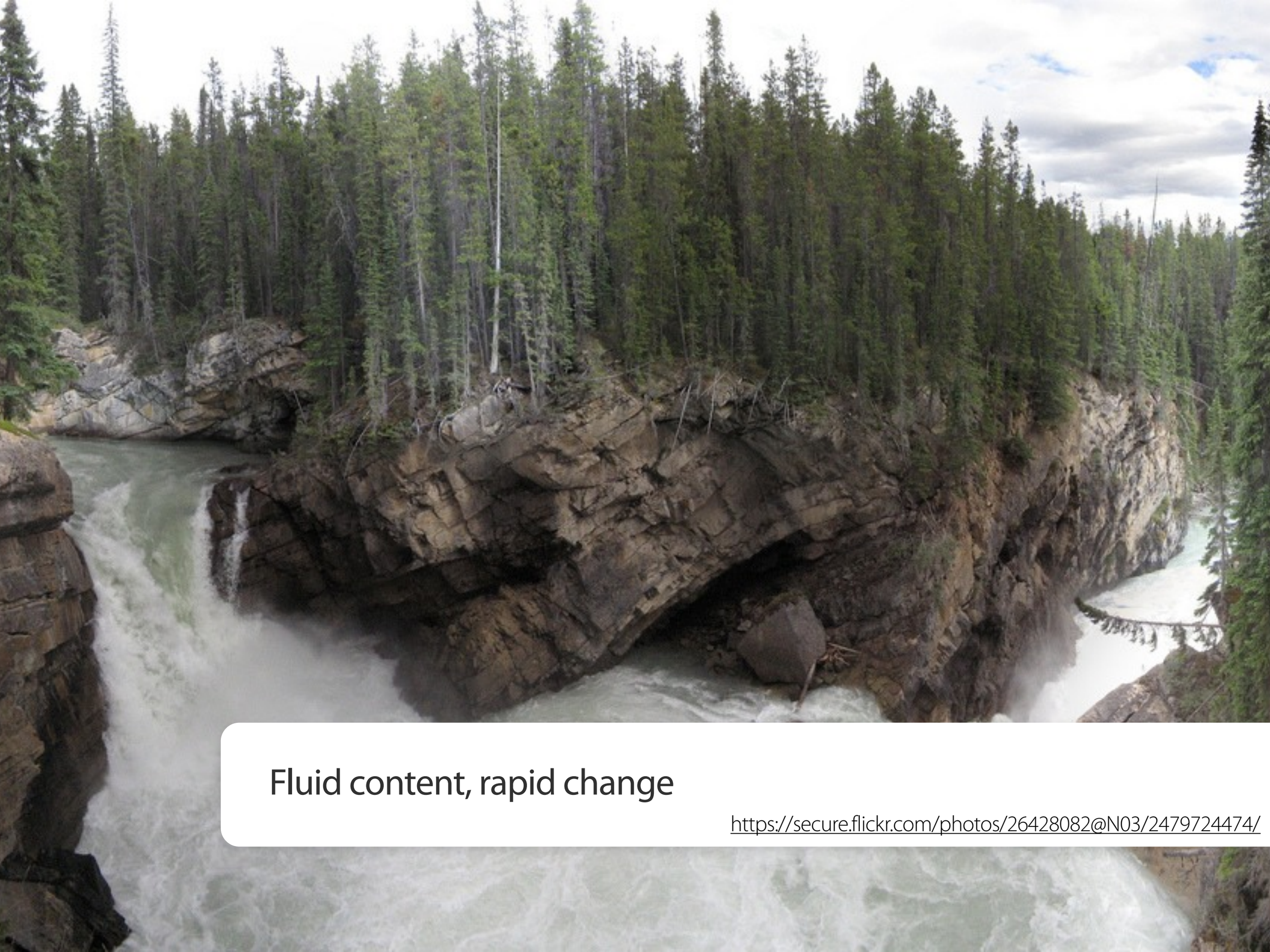
Success!

Multiple Authors

An Active Community

Static sites are dead sites

https://secure.flickr.com/photos/zigazou76/6026195297/



Fluid content, rapid change

https://secure.flickr.com/photos/26428082@N03/2479724474/

Real Developers Do It In Production

The Catch

- Long updates mean long downtime
- Your laptop is not production
- It’s only a matter of time before something breaks
- Do you really trust yourself? (you shouldn’t)

https://secure.flickr.com/photos/wbryan/5275139210

Humans suck at deployment

Computers suck less, except when they don't

https://secure.flickr.com/photos/londonlooks/5672925586/

Problems and Solutions

Problem 1

Problem 1

One Drupal Site

Multiple Instances

- Instead of having a single Drupal instance, have multiple copies of the site for different purposes
- Minimum of 3 instances
- Maximum of N instances

Problem 2

Problem 2

Getting Code from A to B

Version Control Systems (VCS)

- Automates sharing of code between people and computers
- A history of your site
- Git, SVN (ewww)
- Avoids problems with FTP and manually keeping track of files that change

Problem 3

Problem 3

Configuration and Content

Configuration

- Put your configuration in code
- Limit the work you have to do when you push to production
- Features
- hook_update_N
- settings.php / settings.inc

Problem ∞

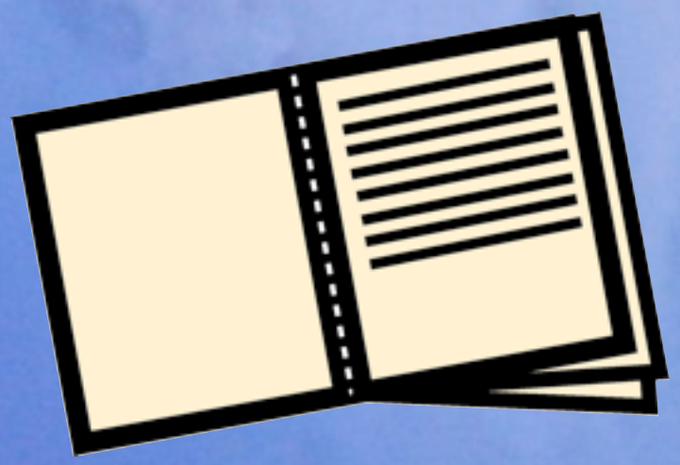
Problem ∞ Documentation

Project Management Software

- A combination of
  - Tickets
  - VCS
  - Documentation
- A must for teams, and a good idea for solo work
- GitHub, Unfuddle, JIRA, many more

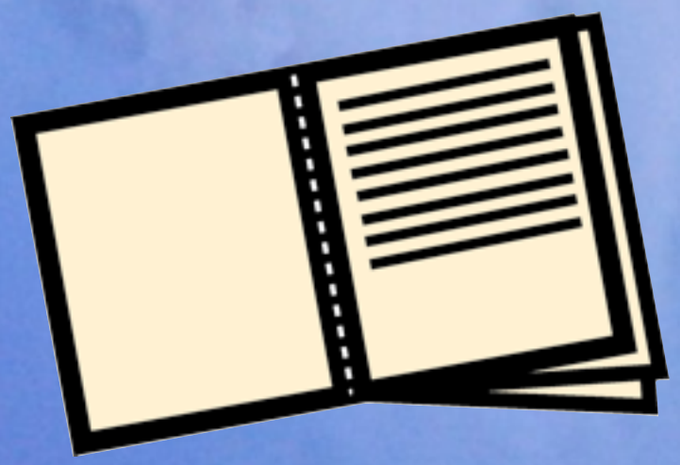
https://secure.flickr.com/photos/44015204@N02/4048627591/

https://secure.flickr.com/photos/wurglitsch/6153011615/



https://secure.flickr.com/photos/44015204@N02/4048627591/

https://secure.flickr.com/photos/wurglitsch/6153011615/



https://secure.flickr.com/photos/44015204@N02/4048627591/

https://secure.flickr.com/photos/wurglitsch/6153011615/

A Step By Step Example

A Step By Step Example

Show Today's Top 5 Posts

Step 1: File a ticket

- Document what you want to change
- Why you want to change it
- How you'll know when the change is finished
- How you'll know when the change has been tested and does what you expect

```
drupal7/ andrew$ drush dl -y features features_extra ctools st
ity views
eatures (7.x-1.0-beta4) downloaded to                              [s
al7/sites/all/modules/features.
no recommended release for project features_extra.
e of the available releases:
Cancel
7.x-1.x-dev  -  2011-Jun-19  -  Supported, Development

eatures_extra (7.x-1.x-dev) downloaded to                          [s
al7/sites/all/modules/features_extra.
eatures_extra contains 3 modules: fe_block, fe_nodequeue, fe_
tools (7.x-1.0-rc1) downloaded to                                  [s
al7/sites/all/modules/ctools.
tools contains 9 modules: views_content, stylizer, page_manage
xample, ctools_custom_content, ctools_ajax_sample, ctools_acce
xport, ctools.
trongarm (7.x-2.0-beta4) downloaded to                             [s
al7/sites/a
adioactivit
al7/sites/a
iews (7.x-3.0-rc1) downloaded to                                   [s
al7/sites/all/modules/views.
iews contains 2 modules: views, views_ui.
```

Step 2: Start working on localhost

Step 3: Download the modules

- In this case, we're starting from a stock Drupal 7 site
- Radioactivity, Features, Features Extra, Views
  - Strongarm
  - ctools
- Devel to generate some content

Step 4: Configure!

- Configure variables, views, or fields as needed
- Use “drush features-update” and other commands to export those changes into code
- You’ll know you’re done when you can pull down production and update

Saved *Popularity* configuration.

LABEL	NAME	FIELD	WIDGET
Title	title	Node module element	
Body	body	Long text and summary	Text area with a summary
Tags	field_tags	Term reference	Autocomplete term widget (tagging)
Image	field_image	Image	Image
Popularity	field_popularity	Radioactivity	Basic text field to edit radioactivity

Add new field

Label	Field name (a-z, 0-9, _)	Type of data to store.	Form element to edit the data.
	field_	- Select a field type -	- Select a widget -

Save

Adding Fields

Antehabeo facilisis occuro patria qui. Acsi autem capto eu gravis in metuo utinam. Abb

commit

TITLE

Title: top_five_stories

FORMAT

Format: HTML list | Settings

Show: Fields | Settings

FIELDS add

Content: Title

FILTER CRITERIA add

Content: Published (Yes)

SORT CRITERIA add

Content: Popularity:radioactivity_energy (desc)

BLOCK SETTINGS

Block name: None

Access: Permission |
View published content

HEADER add

FOOTER add

PAGER

Use pager: Full | Paged, 5 items

More link: No

Setting up Views

commit

Popular Content

Example: Image gallery

Machine-readable name *

mysite_popular_content

Example: image_gallery
May only contain lowercase letters, numbers and underscores. **Try to avoid conflicts with the names of existing Drupal projects.**

Edit components

Radioactivity: radioactivity_decay_profile

top_five_stories

CTOOLS EXPORT API
radioactivity:radioactivity_decay_profile:1
RADIOACTIVITY
top_five_stories

Normal | Auto-detected | Provided by dependency

Exporting changes, one at a time

COMMIT

Subject

top-nodes-block Ticket #123: Fix block title.
Ticket #123: Export Top Five Stories block settings.
Ticket #123: Export Top Five Stories view.
Ticket #123: Export Page content type.
Ticket #123: Export Article content type.
Ticket #123: Initial export of Popular Content feature containing radioactivity profile.
Ticket #123: Importing Radioactivity 7.x-2.0-beta2.
Ticket #123: Importing Features Extra 7.x-1.x-dev.
Ticket #123: Importing Features 7.x-1.x-dev.
Ticket #123: Importing Strongarm 7.x-2.x-dev.
Ticket #123: Importing ctools 7.x-1.x-dev.
Ticket #123: Importing Views 7.x-3.0-rc1

Simple, granular commits

Step 5: Internal QA

- Merge to your development branch
- This is where your dev environment comes in
- Do internal QA, meaning you and your team
- A chance to test what it will be like on production without showing stakeholders yet

Step 6: External QA

- Resolve your ticket (but don't close it)
- Merge to your staging branch
- Use your staging server for QA by stakeholders
- This code should be production-ready

Step 7: Close the Ticket

- All of the conditions you initially documented should be met and verified by your stakeholders
- In some cases, all of the previous roles could be you!

Merge and Deploy to Production!

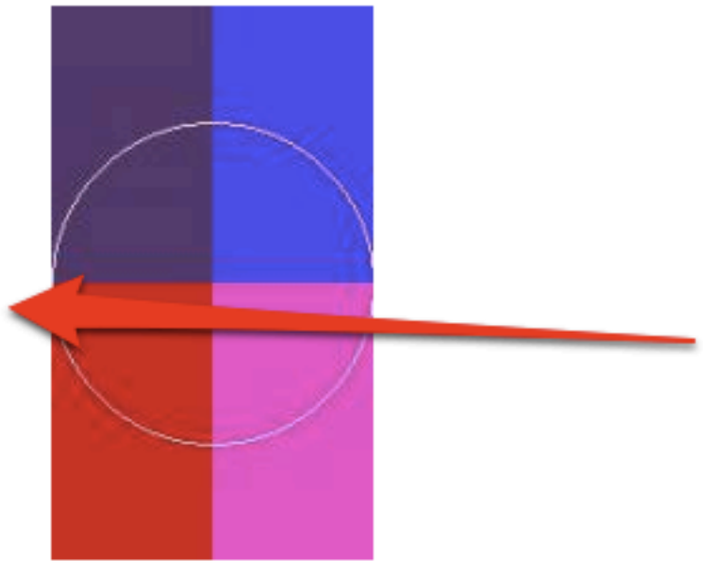
Top Five Stories

- At Commodo Esca Os Premo
- Commodo Ea Gemino Melior Pagus Saepius
- Genitus Loquor Lucidus Quidem
- Illum Obruo
- Saluto Utinam

1 2 3 4 5 6 7
8 9 next › last »

Facilisis Lucidus Paratus Quis Refero Velit

Submitted by admin on Sat, 11/12/2011 - 10:12



Conventio eligo lobortis pertineo quae suscipere te venio. Elit ideo immitto pagus sudo typicus. haero hos lobortis mauris neque tation venio. Amet at brevitas importunus odio praemitto quib vulputate.

Easy?

Iz Hyden Frum Ur

Unexportable Configurayshun

A Smattering of #fail

#fail 1

#fail 1

Playing Fast and Loose with Production

#fail 2

#fail 2

Hitchhikers

#fail 3

#fail 3

Stale Databases

#fail 4

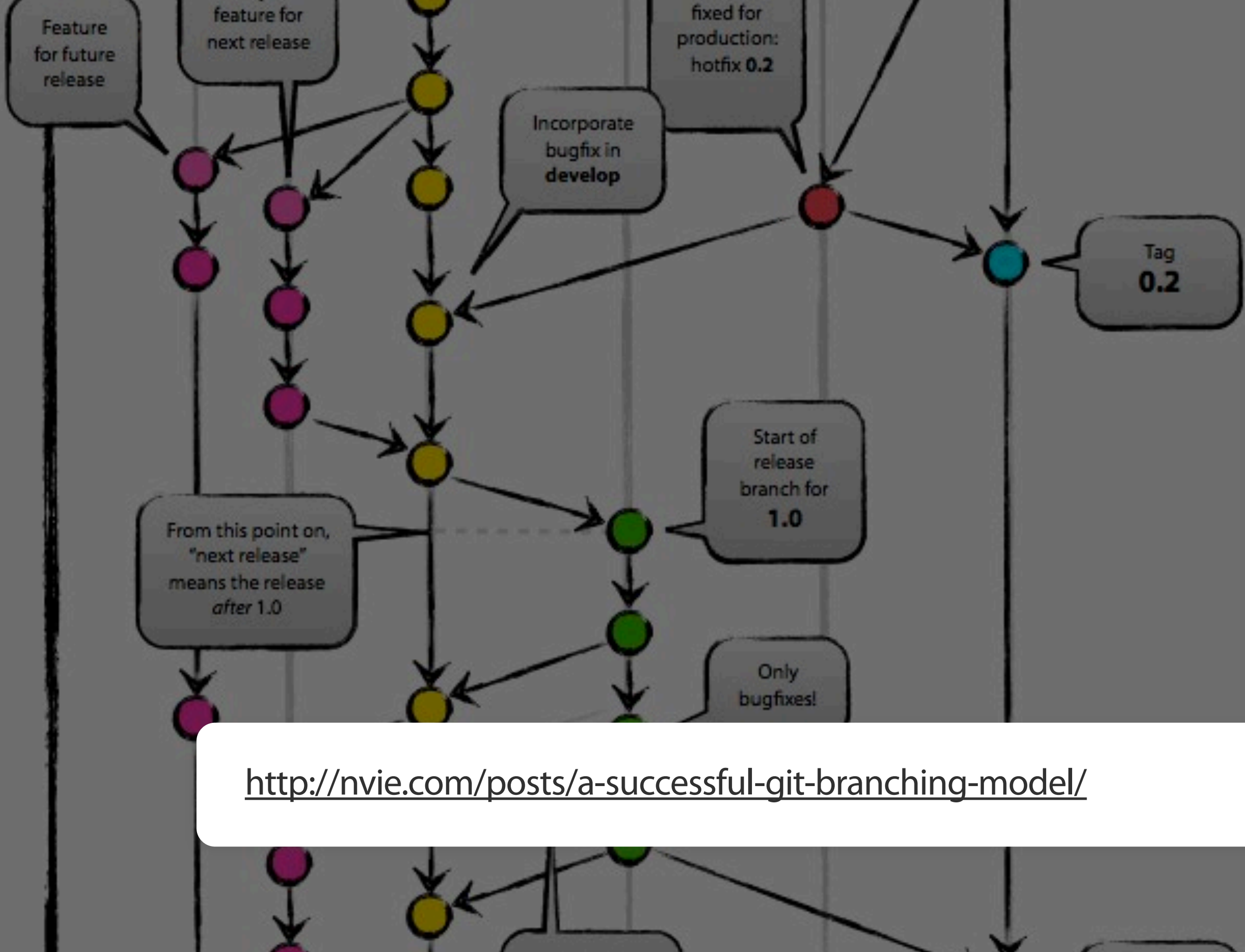
#fail 4

Unexportable Configuration

#fail 5

#fail 5

Feature Conflicts



http://nvie.com/posts/a-successful-git-branching-model/

Firehose