

2011-07-07
DUB - Drupal Usergroup Berlin
„Drupal Deployment“

features+

„A Presentation about monsters, features and the plus“



Florian Latzel

- @fl3a
- <http://netzaffe.de>
- <http://is-loesungen.de>

whoami: f - l - 3 - what?!

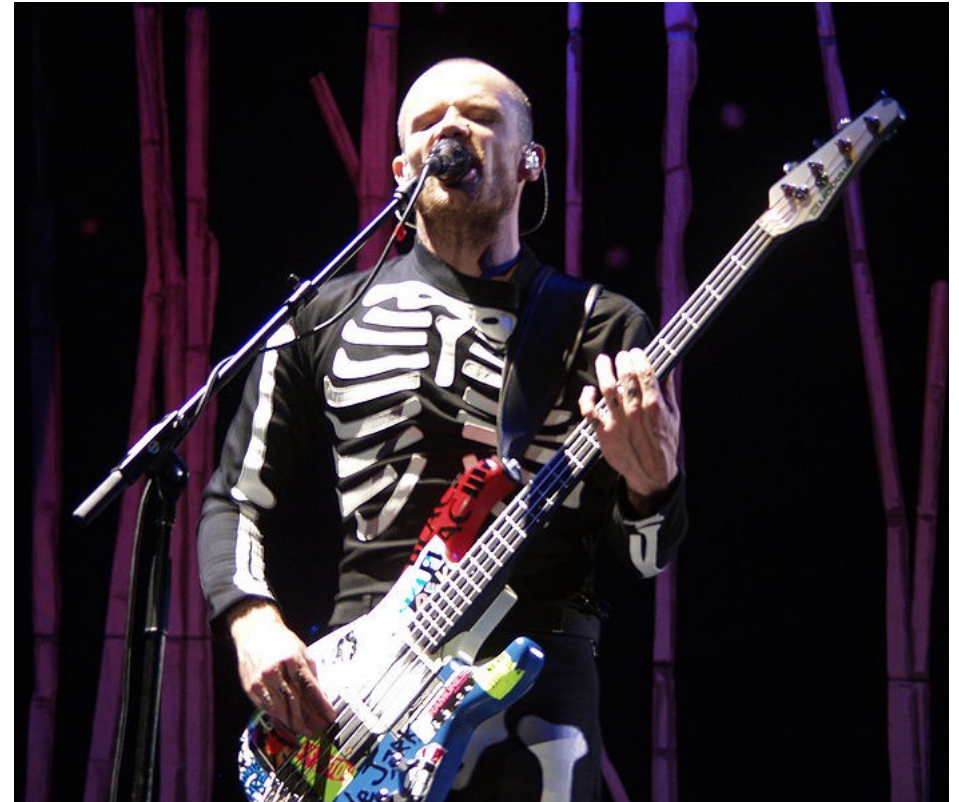
- @fl3a → flEa

whoami: f - l - 3 - what?!

- @fl3a → flea [fli:]

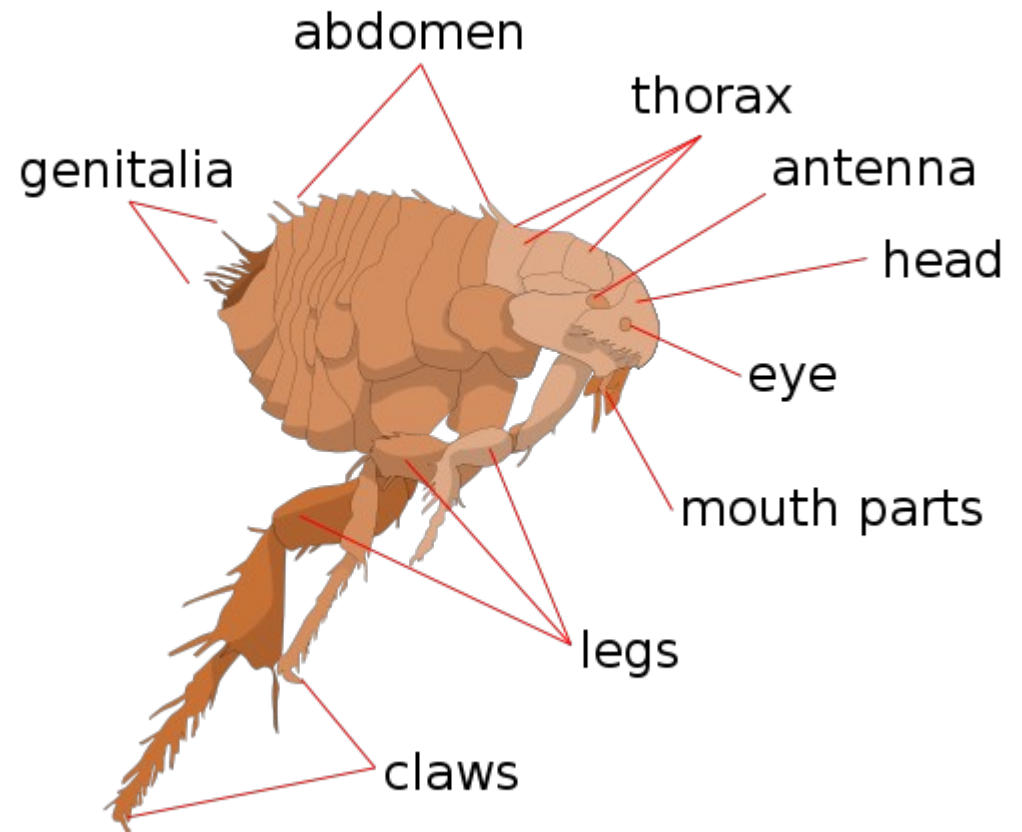
whoami: f - 1 - 3 - what?!

- @fl3a → flea [fli:]



whoami: f – l – 3 – what?!

- @fl3a → flea [fli:]



whoami: fl3a → flEa → flea → floh...

- @fl3a → floh

cat features+/toc.txt

- Questions
- Conclusion
- Features
- Drupal API
- Drush
- Installation profiles
- The Plus

Questions: What is stored where?

Filesystem:

- Code
- Code Settings

Database:

- Drupal Settings
- Modules Settings
- Content Types
- Content
- Views
- Rules
- ...

Questions: What is exportable* right now?

Filesystem:

- Code
- Code Settings

*(and importable)

Database:

- ~~Drupal Settings~~
- ~~Modules Settings~~
- Content Types
- ~~Content~~
- Views
- Rules
- ...

Questions: How to cooperate with a team*?

- Working together on live?
- Working with ftp?
- Sharing sqldumps?
- Overwriting others' work?

*(Perhaps a distributed team)

Questions: How to handle the staging issue?

Everything works on dev. Now what?

- Install modules again on stage
- Export Views, Rules, CT's manually from dev^{*}
- Import them manually into stage

* Also possible via code for Rules[1] and Views[2]

Questions: How to handle the staging issue?

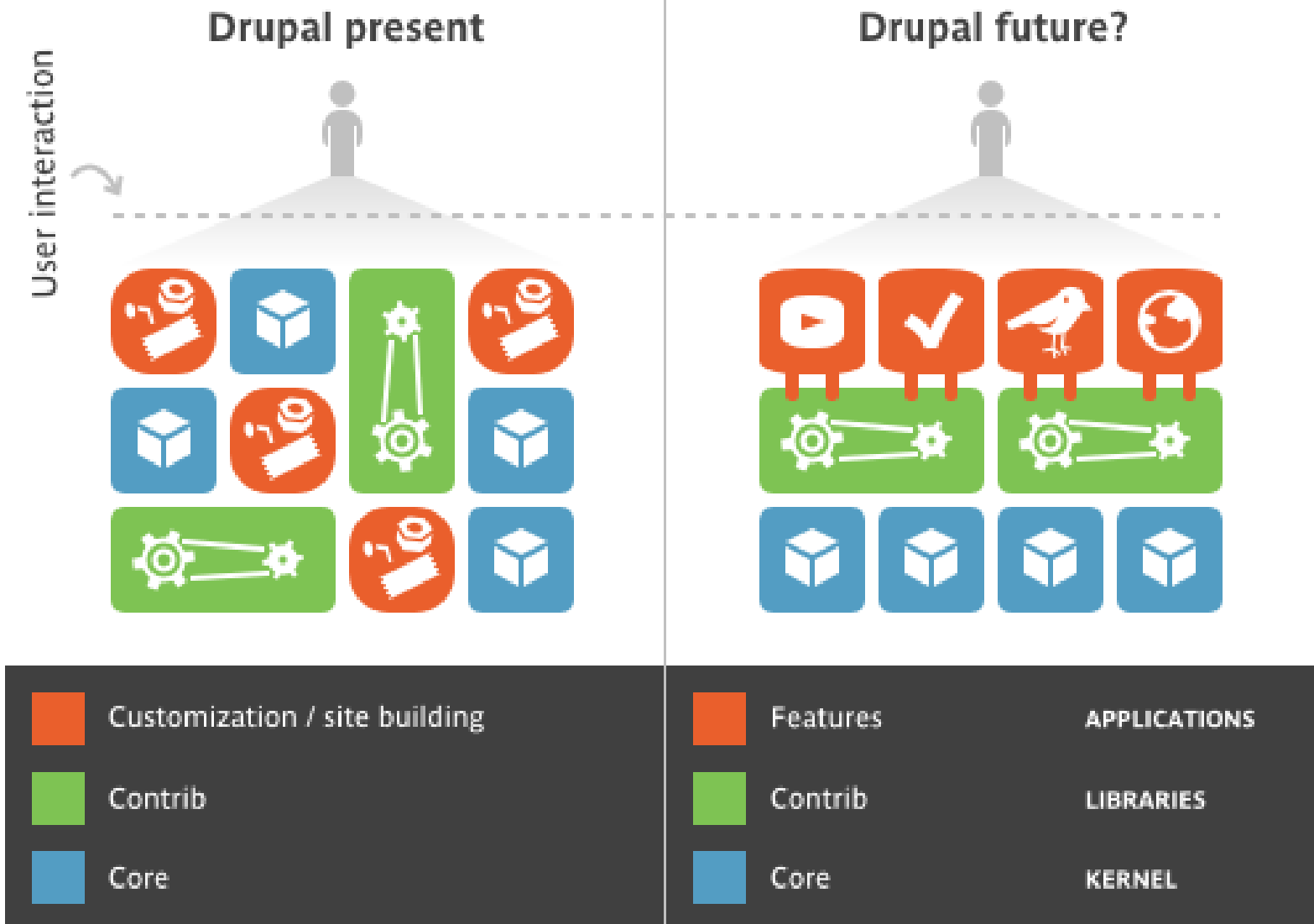
Reproduce „unexportable“ functionality manually:

- Module Settings / Configuration
- Roles and its Permissions
- And so on ...

Conclusion

- Not a good strategy for a team...
- No granularity...
- Mostly cumbersome work...
- Hard to use repeating patterns...
- Prone to errors...
- No revision history...
- No VCS / SCM...

features



features: The Features Module

Features

<http://drupal.org/project/features>



Requires ctools

<http://drupal.org/project/ctools>

- Uses the ctools export API
- Export functionality and configuration into code
- Code means Drupal Module

features: What can it handle?

Drupal Core:

- CCK Fields
 - CCK Fieldgroups
 - Content Types
 - Dependencies (required modules)
 - Menus and Menu links*
 - Permissions
 - Roles
 - Taxonomy (Vocabularies)*
- * Currently in development.

features: What can it handle?

Contributed Modules:

- Context
- CKEditor Profiles
- Feeds
- Flags (from 6.x-2.x)
- ImageCache
- Panels (Page Manager)
- Rules
- Views

features: What is missing?

- Configuration
- Custom Blocks
- Input Formats
- Content

features: Rule #35?

Rule #35: "There's a Drupal module for that." [3]

features: Rule #35 – Configuration

Strongarm

<http://drupal.org/project/strongarm>

- Settings are usually stored in `{variables}`
- Strongarm makes variables exportable via features
- Uses the ctools export API



features: Rule #35 – Boxes

Boxes

<http://drupal.org/project/boxes>

- A reimplementaion of custom blocks (boxes)
- Uses the ctools export API



features: Rule #35 – Content



features: Rule #35 – Content

Nodes, Node revisions, User, Taxonomy

http://drupal.org/project/uuid_features

- Requires uuid
→ Universally Unique Identifier
<http://drupal.org/project/uuid>
- e.g. `node/259` gets the following uuid:
`f3159efc-3e1c-102e-882c-fd9c148d8804`

Features: Rule #35 – import formats

Input Formats

http://drupal.org/project/input_formats

- Allows you to export / import Input Formats manually like CT's and Views
- Features Module Integration
- Requires Exportables
<http://drupal.org/project/exportables>
- Requires Strongarm and Ctools, too

features: myfeature

A Profile CT with...

- content_profile
- cck_private_fields
- flag_friend, flag_friend_access
- content_taxonomy
- And a lot of fields...

features: myfeature

myfeature/

- myfeature.features.content.inc
- myfeature.features.fieldgroup.inc
- myfeature.features.inc
- myfeature.info
- myfeature.module
- myfeature.strongarm.inc

Features: myfeature.info

```
core = "6.x"
dependencies[] = "cck_private_fields"
dependencies[] = "cck_private_fields_flag_friend"
dependencies[] = "content_profile"
...
description = "Myfeature description"
features[ctools][] = "strongarm:strongarm:1"
...
features[fieldgroup][] = "profile-group_personal"
...
features[node][] = "profile"
...
features[user_permission][] = "edit own profile content"
features[user_permission][] = "edit privacy options for field field_relationships"
features[user_permission][] = "view private data for field field_relationships"
...
features[user_role][] = "Community member"
features[variable][] = "comment_anonymous_profile"
...
name = "myfeature"
package = "Features"
```

features: myfeature.module

```
<?php
```

```
include_once('myfeature.features.inc');
```

Features: myfeature.features.inc (1)

```
<?php

/**
 * Implementation of hook_ctools_plugin_api().
 */
function myfeature_ctools_plugin_api() {
  list($module, $api) = func_get_args();
  if ($module == "strongarm" && $api == "strongarm") {
    return array("version" => 1);
  }
}
```

Features: myfeature.features.inc (2)

```
/**
 * Implementation of hook_node_info().
 */
function myfeature_node_info() {
  $items = array(
    'profile' => array(
      'name' => t('Profile'),
      'module' => 'features',
      'description' => t('A user profile built as content.'),
      'has_title' => '1',
      'title_label' => t('Title'),
      'has_body' => '0',
      'body_label' => '',
      'min_word_count' => '0',
      'help' => '',
    ),
  );
  return $items;
}
```

Features: myfeature.features.inc

```
<?php

/**
 * Implementation of hook_ctools_plugin_api().
 */
function myfeature_ctools_plugin_api() {
  list($module, $api) = func_get_args();
  if ($module == "strongarm" && $api == "strongarm") {
    return array("version" => 1);
  }
}
```


features: myfeature.features.content.inc

```
<?php

/**
 * Implementation of hook_content_default_fields().
 */
function myfeature_content_default_fields() {
  $fields = array();
  ...
  // Exported field: field_relationship_status
  $fields['profile-field_relationship_status'] = array(
    'field_name' => 'relationship_status',
    'type_name' => 'profile',
    'display_settings' => array(
      'weight' => '37',
      'parent' => 'group_common',
      'label' => array(
        'format' => 'above',
      ),
      ...
      'allowed_values' => 'solo^M
In a relationship^M
complicated',
      ...
      'label' => 'relationship status',
    );
}
```

features: Specifications

Kit [4]

- KIT Feature Specification (kitf 1.0-draft)
- KIT Theme Specification (kitt 1.0-draft)

features: Known Issues

Lost in translation [5]...

- e.g. a field is named `'Beziehungsstatus'`
- e.g. in your `myfeature.content.inc` you will get: `t('Beziehungsstatus')` that will be used as translation source
- but it should be `t('relationship status')`
- Still a open issue in Features Module [6]
- Solution: Features Export Language [7]

Drupal API: If a feature is not enough...

You know, Drupal has an API...

Create Upgrade-Path's with Drupal's `update.php`

- Create `myfeature.install` file
- Implement `hook_install()`
http://api.drupal.org/api/drupal/developer--hooks--install.php/function/hook_install/6
- Implement `hook_update_N()`
http://api.drupal.org/api/drupal/developer--hooks--install.php/function/hook_update_N/6

Perhaps you might use `db_query()`

or implement `hook_you_name_it()`

drush



Features Module provides some drush commands:

- drush features
- drush features-export [feature name] [component list]
- drush features-update [feature name]
- drush features-revert [feature name]
- drush features-diff [feature name]

drush: Useful features related extensions

- Features Clone
http://drupal.org/project/features_clone
- Features Plumber
http://drupal.org/project/features_plumber
- Features Fetch
http://drupal.org/project/features_fetch

Installation profiles

You might know Installation Profiles from...

- DrupalCenter
- OpenAtrium
- Acquia Drupal
- Commons
- Open Publish
- ...

Installation Profiles: myprofile

```
drupal-6.x/  
├── COPYRIGHT.txt  
├── cron.php  
├── includes  
├── index.php  
├── install.php  
├── misc  
├── modules  
├── profiles  
│   ├── default  
│   │   └── default.profile  
│   ├── myprofile  
│   │   ├── myprofile.profile  
│   │   └── translations  
│   │       └── de.po  
├── robots.txt  
├── scripts  
├── sites  
├── themes  
├── update.php  
└── xmlrpc.php
```

Installation Profile: myprofile.profile

```
/**
 * Implementation of hook_profile_modules()
 *
 * @return
 *   array of modules to be required and activated
 */
function myprofile_profile_modules() {
  return array(

    // feature-stuff
    'myfeature',
    ...
  );
}
```

The Plus

- Perhaps your own Drupal Distribution?
- Feature Server
<http://drupal.org/project/fserver>
- Drush Make
http://drupal.org/project/drush_make
- Drush make UI
http://drupal.org/project/drush_make_ui
- drush site-install
(in drush core \geq 4.0)
- CI – Continuous Integration
(with e.g. Hudson / Jenkins)

```
cat ~/feature+/appendix/references.txt
```

- [1] <http://drupal.org/node/298634>
- [2] http://api.lullabot.com/hook_views_default_views
- [3] Robert Douglass at DrupalCamp Stockholm
<http://twitter.com/#!/todddross/statuses/1970578988>
- [4] <http://drupal.org/project/kit>
- [5] <http://blog.erdfisch.de/2011/05/features-international>
- [6] <http://drupal.org/node/874760>
- [7] <http://drupal.org/sandbox/xamanu/1161994>

cat ~/feature+/appendix/image-credits.txt

- Flea (RHCP)
http://en.wikipedia.org/wiki/File:Flea_in_skeleton_t-shirt.jpg
- Flea (anatomy)
http://en.wikipedia.org/wiki/File:Scheme_flea_anatomy-en.svg
- Feature Monster
http://drupal.org/files/images/features_monster.png
- The future of drupal features
<http://www.flickr.com/photos/developmentseed/3575550279/>
- Strongarm Monster
<http://drupal.org/files/images/strongarm.png>
- Boxes Monster
<http://drupal.org/node/820924>
- „Node Collision“
<http://www.slideshare.net/eaton/drupal-deployment-presentation>
- Adrian <3 drush
<http://www.flickr.com/photos/developmentseed/3614279203/>

Feature related images by developmentSEED, monsters by Saman Bemel Benrud