# Features Strategy

structured development and deployment

Robert Foley - Technical Architect

Acquia®
THINK AHEAD

# Introduction – About Me

Robert Foley
Technical Architect at Acquia since 2013

Using Drupal since 2007

Working in the industry since 1997 and have worked in C# .NET, Java J2EE, and roughly 30 some odd client and server side languages, libraries and frameworks.

# Introduction – What about you?

How many know what features are?

How many have built features?

How many of you have created custom modules?

How many of you have had problems working as a team using features?

# Overview

- → Brief overview of features
- → Structuring Features
- → Feature “types”
- → Workflow and Feature Foundations
- → Deployment, Rollback, change management

# What are Features?

Acquia®
THINK AHEAD

# In the Past – Hand Crafted Goodness

Teams wrote custom modules with one-off functions to define structure, save settings, and content structures.

Each project (or even module) was unique and no two were alike.

Worse, teams would manually “setup” module configuration on each site instance (dev, stage, prod)

# Features?

https://www.drupal.org/documentation/modules/features

Structured data

- Configuration
- Settings
- Content (less so)?

Enables:

- **Standard for Integration**
- **Encapsulated into code**

It can contain:

- Global settings
- Module settings
- Entity Definitions
- Field Definitions
- Display structure
- Behaviors (rules, etc)
- Module specific elements/configuration

# Structured Features

Acquia®
THINK AHEAD

# Considerations

- → Working in teams (two or more) and or dispersed (off site/remote) teams
- → Necessitates consistency for initial releases and future releases over time

# Considerations Continued

- → Features should be structured to support modularity
- → Features “sets” should be designed to be additive.
- → “Try” to reduce interdependencies between features.

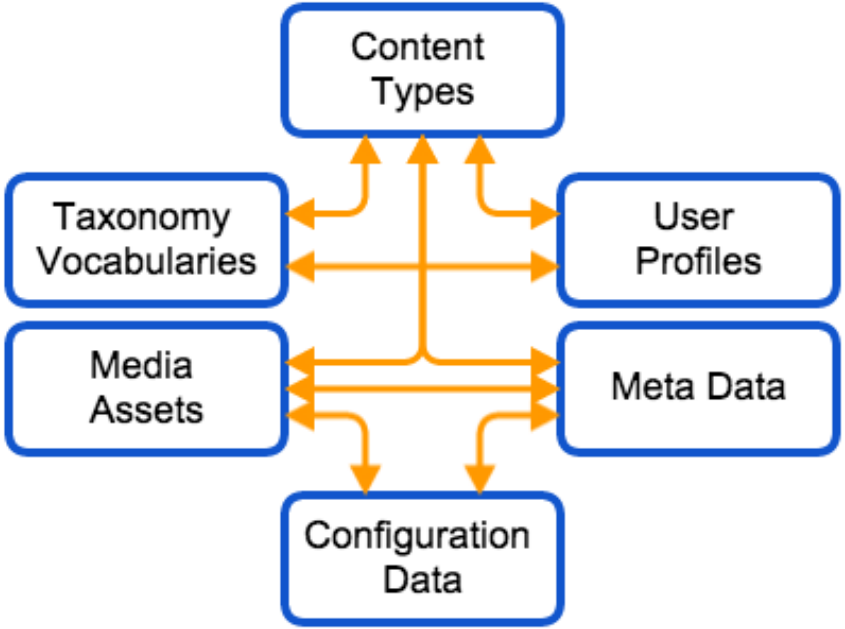
# Cookie Monster Cookie Co.

Mr. Cookie Monster needs help.

- List favorite cookie recipes
- Group by ingredients, color, fruit, vegetable, and astronomical sign.
- Needs ability for chiefs to write reviews.
- Needs ability for head chief to post recipes.
- Maybe let visitors rate cookies.

# Where do we start?

# Where do we start?



# Basic Model

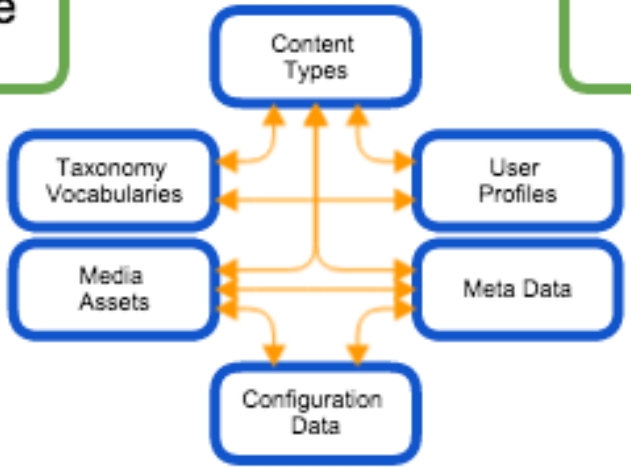
Color

Recipe

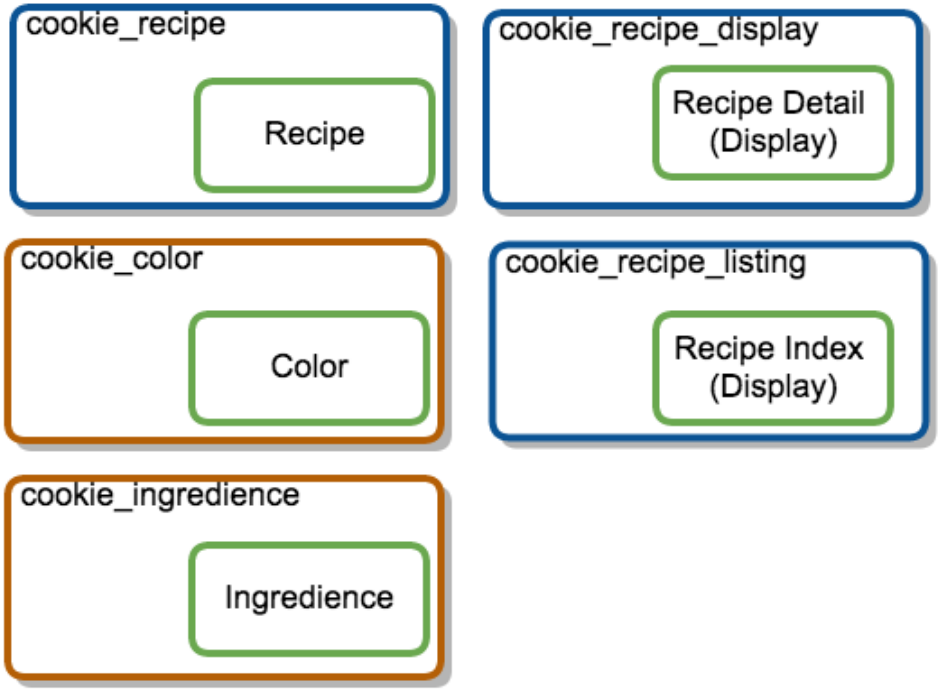
Recipe Detail (Display)

Ingredience

Recipe Index (Display)



# Basic Model


cookie_recipe
Recipe
cookie_recipe_display
Recipe Detail (Display)
cookie_color
Color
cookie_recipe_listing
Recipe Index (Display)
cookie_ingredience
Ingredience

# Grouping Features

- → Working in a team of more than two requires separating effort into small tasks.
- → Separating data modeling from display work helps specialists continue to contribute without conflict.
- → Separation promotes modular refactoring and or extension over time.

# Naming Conventions

https://www.drupal.org/node/299070

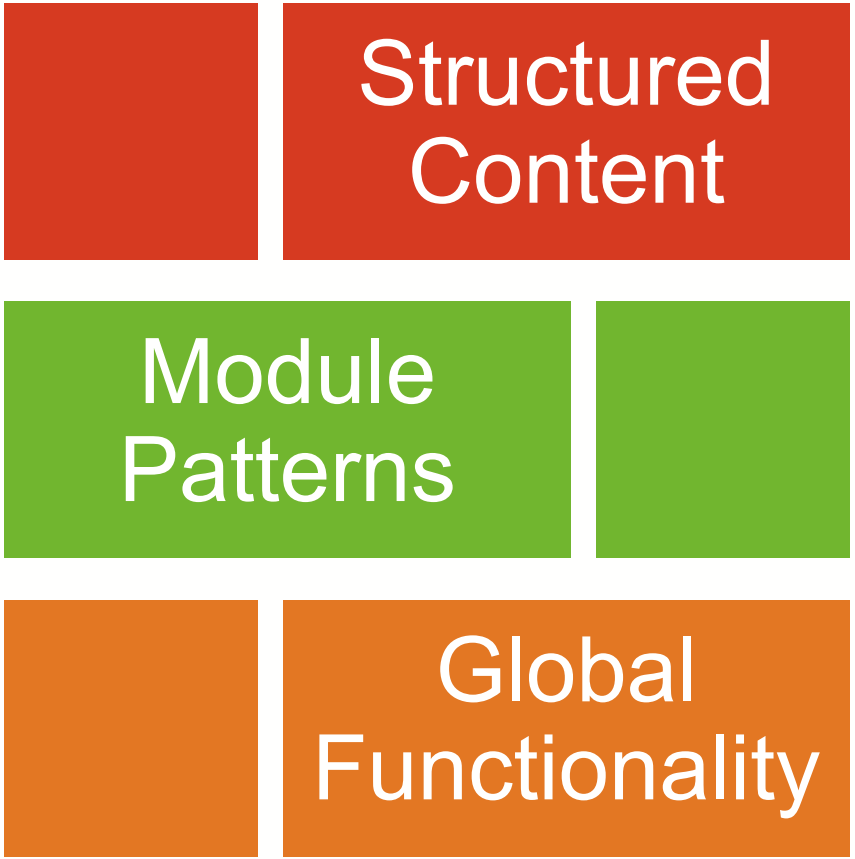
- → Features are modules.
- → The machine (file) name used defines a Name Space of the functions encapsulated within the feature.
- → Your team may expand the feature with custom functions and integration with custom or contributed modules or 3rd party libraries.

# Types of Features?

Acquia®
THINK AHEAD

# Feature Types

- Features provide consistency
- Structured data, configuration, and settings for core Drupal objects
- Contributed modules have been extended to support “exportability”
- Ctools provides a foundation for exposure to an even larger set of modules.


Structured Content
Module Patterns
Global Functionality

# Feature Types

Features can package multiple sets of configuration, settings, structure, and even content elements.

Design of a feature is what you make it.

Global Functionality

Module Patterns

Structured Content

# Global Functionality Features

Foundational Features define global sets of configuration, settings, or sets of modules required for the business goals of the site.

Permissions Matrix
Solr Search config
Global Image Cache
Media config
Media display
Wysiwyg Profiles

# Module Pattern Features

Patterns (recipe) Features define a dependency of modules and encapsulate configuration and settings that are not tied to a specific entity or single module.

- Wysiwyg profile and file management
- Search configuration and Solr profile
- Panels Layouts and sub-module configuration

# Structured Content Features

Traditionally used as examples for feature usage. Defining a new content type and exporting it.

Content Type Fields
Content Type Displays
Views
Panel layouts
Blocks
Panes
Custom Forms
Administration indexes

cookie_recipe

cookie_recipe_display

cookie_recipe_listing

cookie_color

cookie_ingredience

cookie_input_profiles

cookie_wysiwyg_profiles

cookie_search_base

cookie_search_solr

cookie_permissions

cookie_default_media_files

cookie_default_media_files_display

cookie_default_image_styles

cookie_default_path_aliases

**cookie_master_control**

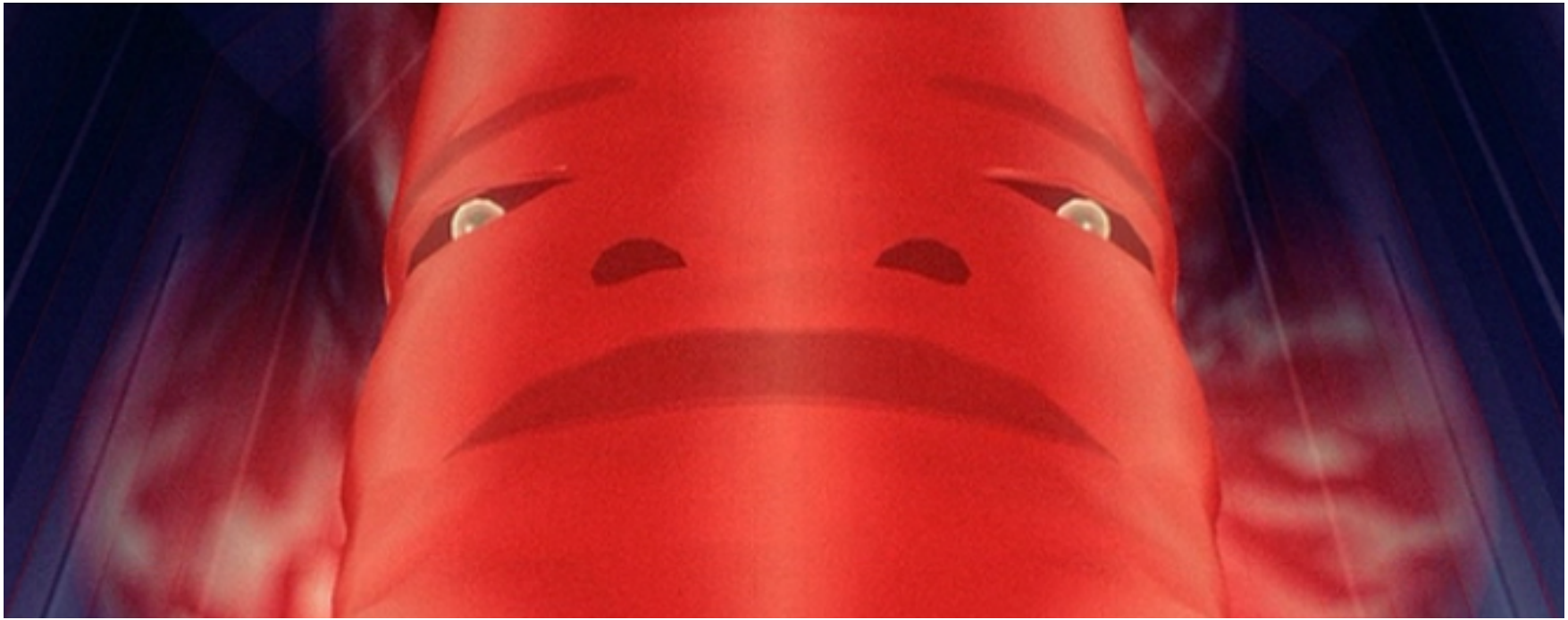
# Workflow and Foundations

Acquia®
THINK AHEAD

# Infrastructure and Standards

- → Revision Control System (Git, CVS, Subversion, etc)
- → Bug/Task Tracking System (Jira, Bugzilla, etc)
- → Checklists and Peer review Processes
- → Release Methodology

# “Master Control” Module/Install Profile



# Master Control

- Defines Default modules to enable on install
- Provides sequenced releases
- Enables modules/features
- Reverts Features
- Sets/Resets Variables
- Manage content structure

Usually comprised of:

- Empty module file
- Install file
- info file

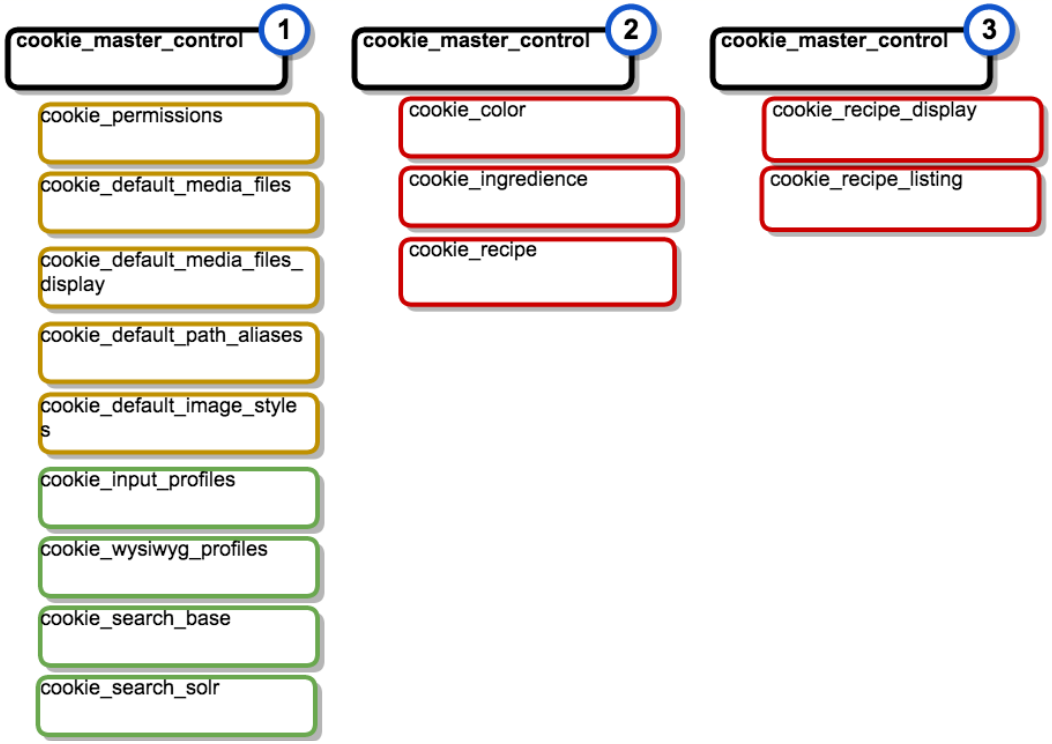
# Updates?

https://www.drupal.org/node/876250

## cookie_master_control.install

```
/**
** ticket-573 - Enable cookie events.
*/
cookie_master_control_update_7001() {
     if (!module_exists('cookie_events')) {
         module_enable('cookie_events');
     }
}
```

# Central Release Structure



# Questions and Answers?

# Thank You

Acquia®
THINK AHEAD

# Notes

**Features Packages**
Features
https://www.drupal.org/project/features

Features Extra
https://www.drupal.org/project/features_extra

StrongArm
https://www.drupal.org/project/strongarm

Ctools
https://www.drupal.org/project/ctools

UUID
https://www.drupal.org/project/uuid

UUID Features
https://www.drupal.org/project/uuid_features

Automated updating Datasets
https://www.drupal.org/project/deploy

Larger datasets with dependencies
https://www.drupal.org/project/migrate