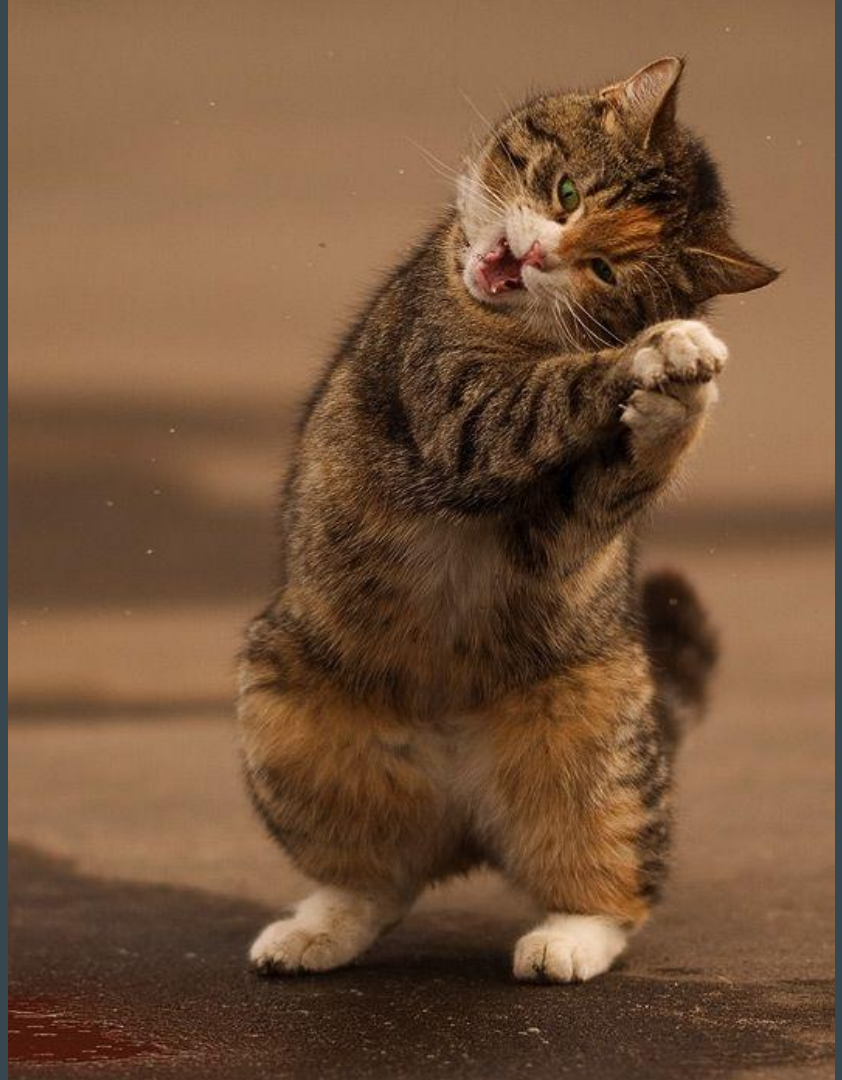
NEW ORLEANS
DRUPALCON 2016

The potential in Drupal 8.x and how to realize it

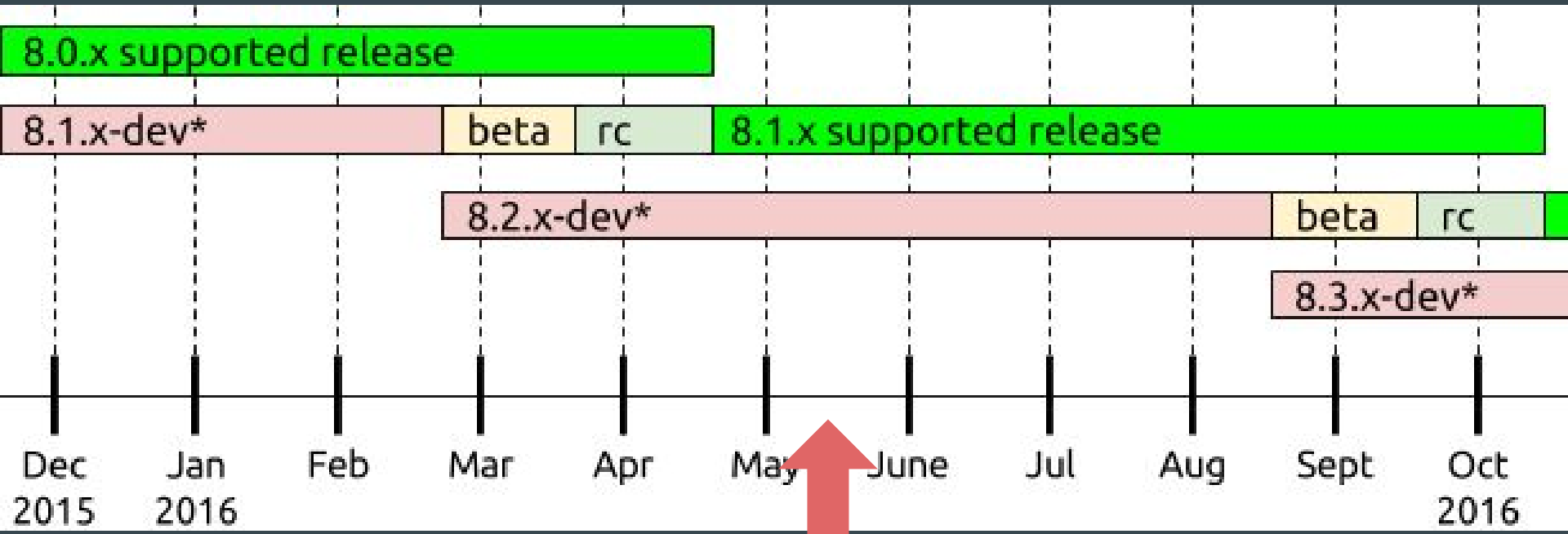
Angela Byron, Gábor Hojtsy

1. Drupal 8:
The dawn of new possibilities
...



Making big changes in 8.x: It's possible!

Intro to semantic versioning



Predictable

Improvements every 6 months

Incentive to contribute

Backwards-compatible

2. What to improve?

...

"Top-down" goals (from committers)

Migrate UI	Frontend testing	Blocks and layouts
Configuration Management	Media	Workflow

"Bottom-up" goals (from community)

BigPipe	Twig	Admin style guide
Contact for basic web forms	Picture support	Improved entities

www.drupal.org/core/roadmap

Drupal development roadmap

Last updated April 29, 2016.

Starting with Drupal 8, the core release cycle has "minor" versions every 6 months with backwards-compatible features and API improvements.

This page is intended to be a living document for both official priorities from the core committer team for each minor release, as well as approved community initiatives from core developers at large.

Issue queue for feature requests and bugfixes

This document only includes high-level plans for initiatives that span many issues and need community signoffs. No prior approval is needed for bug fixes or small feature requests; those are found in the Drupal 8 issue queue.

How are initiatives chosen and approved?

For Drupal core initiatives that will require more than a few issues to complete, contributors first create an initiative proposal issue. (See the initiatives under review below for plans that are still under discussion, or start your own!) The community discusses each proposal until it receives the signoffs needed according to the Drupal core governance. Then, the community begins work on the initiative and it will be highlighted below under either the priorities for the upcoming minor (which are decided by core maintainers with community input) or the ongoing initiatives (for both community proposals and ongoing work from previous priorities.

Initiatives for Drupal 8 minor releases must be backwards-compatible. See the allowed changes in minor versions and the backwards compatibility policy.

Key #

- 8.2 Focus area for 8.2.x
- 8.1 Focus area for 8.1.x
- Initiatives with this icon were proposed by the community.
- Plan under review.
- Under development (not yet fully in core)
- In core as an experimental module.
- Stable in core

Page status

Incomplete

Edit this page

About this page

Drupal version
Drupal 8.x

Audience
Contributors, Site administrators, Site builders

Level
Beginner, Intermediate

Getting Involved Guide

- Where is the Drupal Community?
- Ways to get involved
- Why get involved?
- Talk with the community
- Contributor tasks
- Create a project (module or theme) on Drupal.org
- Contribute to documentation
- Contribute to Drupal.org
- Use the issue queue
- Community Initiatives
- Testing code
- Contribute to development
  - Core vs. contributed projects
  - Patches
  - Drupal core
    - Drupal core maintainers
    - Criteria for evaluating proposed

3. But… how?

…

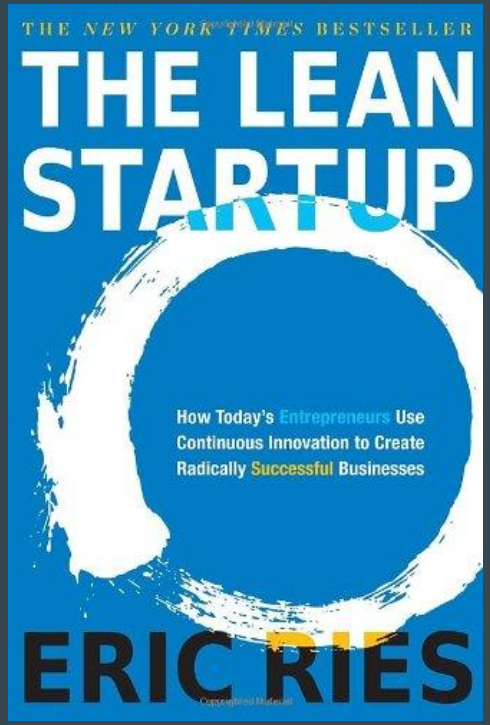
Pain points from Drupal 7 and Drupal 8.0.x

- Bikeshedding, especially of user-facing changes
- Work hard on something, may still get rejected
- Directional feedback vs. standards nitpicks
- Don't validate ideas until *after* shipping; now too late to fix
- Giant core patch vs. sandbox vs. contrib vs. core

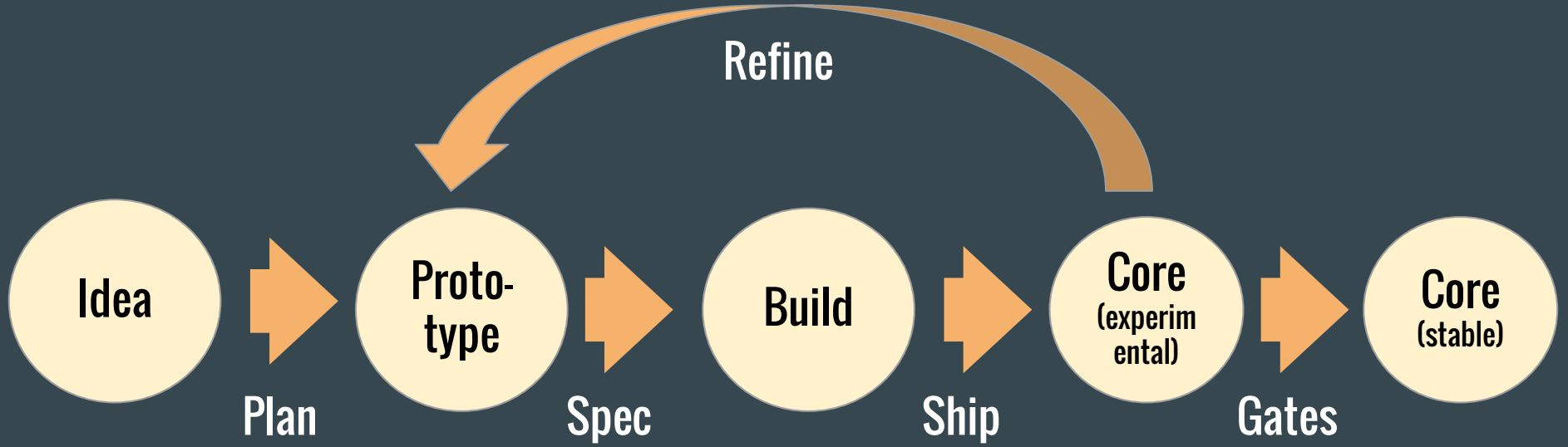
Ideas for improvement

1. Iterate quickly and cheaply on ideas
2. Clear sign-off points to avoid wasting time
3. Involve the right stakeholders at the right time
4. Gain visibility for proposals from committers
5. Reduce barriers to entry into core for new ideas
6. Clear visibility of priorities for the community

How *other* people improve products



Possible implementation for Drupal core

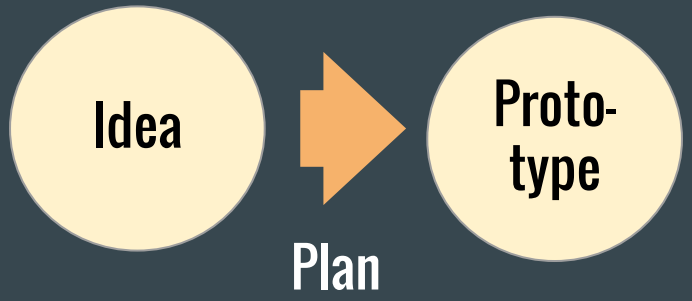


Note: This is *just* a proposal

...

...about how to make proposals. ;)

Your feedback needed!



1. "Idea" is just a few sentences (lean UX-style)
2. Get sign-off / rejection right away (product management)
3. To get to next phase, formulate a "Plan"

Plan template (beta)

Creating a plan issue for an initiative proposal

Last updated May 6, 2016.

```
<h3>Background</h3>

High-level information that a contributor should know to understand the proposed
initiative.

<h3>Goal</h3>

A high-level statement of the goal for your proposal (and why).

<h3>Process and where to find it<h3>

Does the initiative have a team site/group/etc.? Where will the work be done? Can
it be added to core as a new experimental module
https://www.drupal.org/core/experimental, or does it need to make changes across
existing subsystems and therefore need a feature branch? Is there a
sandbox/existing contrib module/other repo?

<h3>Proposal roadmap</h3>

A rough roadmap of the specific work planned. (Note that your roadmap will likely
evolve as you work on the initiative, so don't try to make this section perfect!
The goal is to get you and other stakeholders to think about this from the start,
```

Page status

Incomplete

Log in to edit this page

About this page

Drupal version
Drupal 8.x

Level
Intermediate

Getting Involved Guide

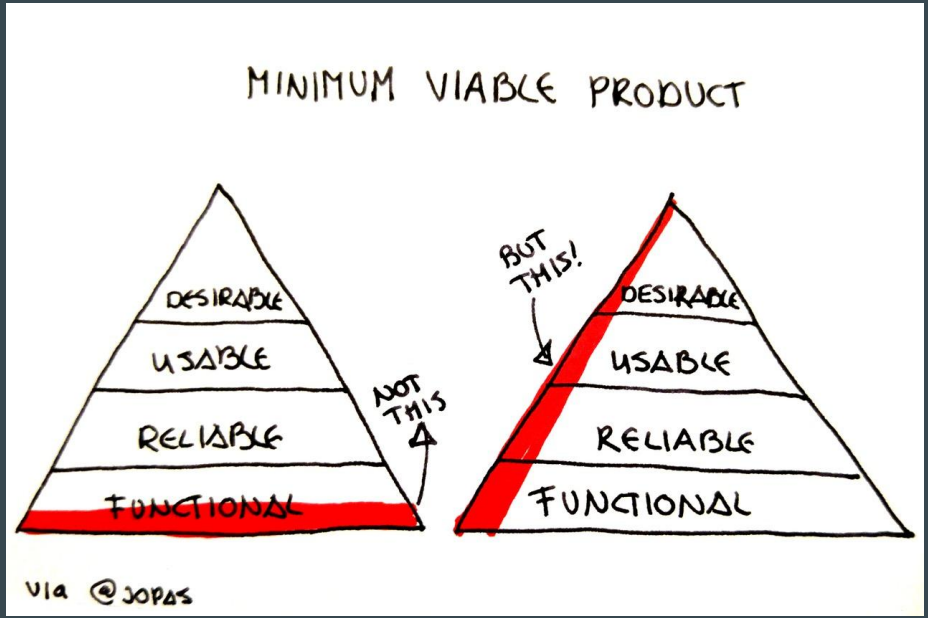
- Where is the Drupal Community?
- Ways to get involved
- Why get involved?
- Talk with the community
- Contributor tasks

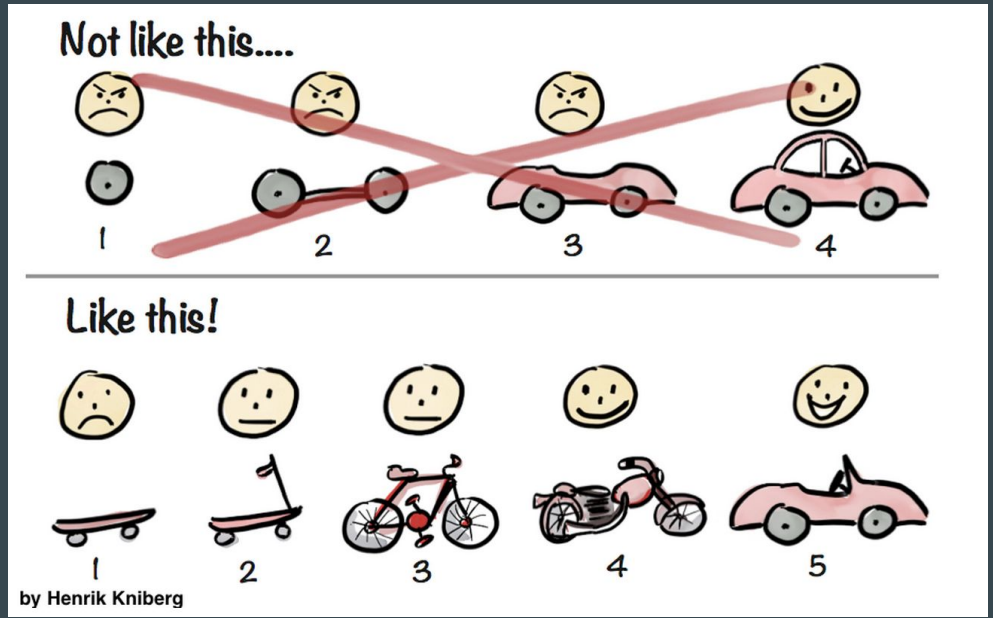
in a backwards-compatible way for an upcoming minor version (not necessarily the next one), and avoid disrupting existing sites, modules, themes, etc. For example,

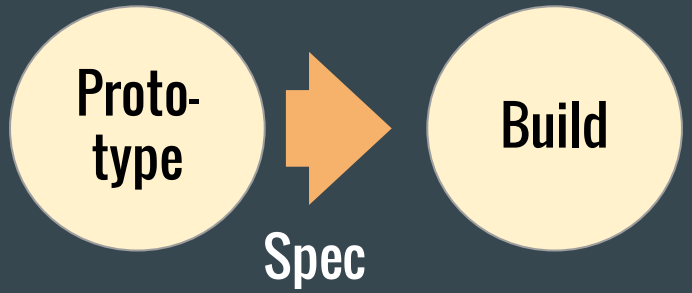
- Use the issue queue
- Community Initiatives

https://www.drupal.org/core/initiative-proposal-template

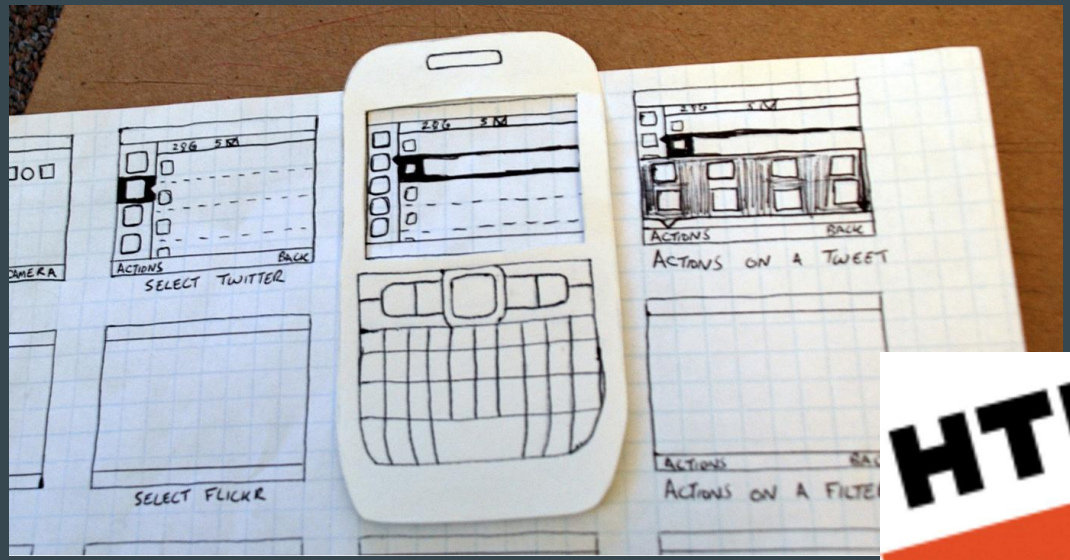
For compelling 8.x minor releases...







1. Prototype iteratively, as cheaply as possible
2. Validate prototype with real users
3. Once validation occurs, the prototype becomes a spec
4. Now, No. More. Bikeshedding. ;)


ACTIONS
BACK
SELECT TWITTER
SELECT FLICKR
ACTIONS ON A TWEET
ACTIONS ON A FILTE

HTML
CSS

Who to talk to? At least some of these folks.

Initiative Coordinators

- Multilingual
- Web Services
- Media
- Content Workflow
- ...

Subsystem Maintainers

- Field system
- Queue system
- Block module
- Shortcut module
- ...

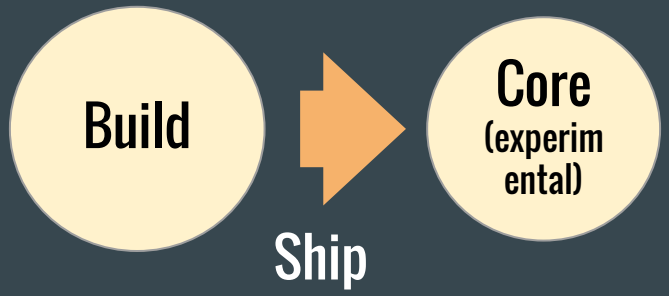
Topic Maintainers

- Testing
- Performance
- Documentation
- Usability
- Accessibility

Committers

- Product Managers
- Framework Managers
- Release Managers

MAINTAINERS.txt && d.o/project/governance



1. Now, spec becomes core patch
2. However, most "core gates" (except MVP testing! :)) are bypassed
3. Initially goes in as "Experimental" module
4. Bikeshedding opens again after shipping. ;)

Experimental modules

Pros

Already in core

Can be less stable

Familiar core process

Easy for end users

Iterate quickly

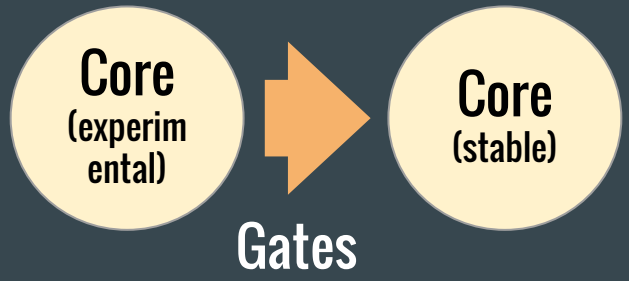
Cons

Cannot commit directly

Needs reviewers

System-wide changes not possible

Risk of lingering technical debt



1. Once iterated on a few times, move to "proper" core module.
2. This requires all sign-offs, core gates, etc.
3. Radical refinements no longer possible without a *new* experimental module.
4. Enjoy!

Summary

1. Get sign-off/rejection *before* doing tons of work
2. Validate direction with real-world data vs. bikeshedding
3. Make it cheaper/easier/faster to improve core all around
4. Jump through the right hoops at the right time
5. ...
6. Profit! :P

4. What do you think?

…

Discuss!

Are the pain points addressed?

Balance of bureaucracy vs. unpleasant surprise?

How do we get ideas on the roadmap?

What about the implementation details?

NEW ORLEANS
DRUPALCON 2016

Join us for Sprints

Friday, May 13 at the Convention Center

First-Time Sprinter Workshop - 9am-12pm in Room 271-273

Mentored Core Sprint - 9am-6pm in Room 275-277

General Sprints - 9am-6pm in Room 278-282

So How Was It? - Tell Us What You Think

Evaluate this session - https://events.drupal.org/node/9866

Thanks!