



**BALTIMORE**  
— DRUPALCON 2017 —

# **Less is More – What Modules, Features, or APIs Should We Cut From Core?**

April 26, 2017



# Peter Wolanin

Senior Application Architect  
BioRAFT  
drupal.org: pwolanin



# David Hernandez

Manager of Learning & Contributions  
FFW  
drupal.org: davidhernandez  
@davidnarrabilis





# Why Might Something Be Cut?

- Inherently insecure (PHP)
- Outdated or proprietary standard (drupal & openID)
- No longer a commonly used feature of Drupal sites
- Lack of maintainers or interest
- Excessive tech debt compared to value
- API hard to use or lacking in value
- Faster innovation in contrib
- Better alternatives in contrib

## Cut From 8.x (But not easily)

- PHP module #1203886 (2011 - 2013)
- Trigger module #764558 (2010 - 2012)
- Profile module #301071 (2009 - 2011)
- Poll module #61285 (2006 - 2013)
- Blog module #233301 (2008 - 2011)

# Consequences of keeping a module in

- Stagnated development (priorities)
- Less control for caregivers (no commit access)
- Maintenance burden on core

# What Happens To a Removed Module?

- Goes to contrib
- That means someone has to commit to maintaining
- Usually that means it's actually left to wither
- e.g. blog module does not have a stable 8.x release

# Maintainerless Modules and Subsystems

- Actions module
- Aggregator module
- Asset Library API
- Authentication and Authorization
- Automated Cron / Cron
- Ban module
- Batch API
- Color module
- Contextual module
- PostgreSQL & SQLite DB drivers
- File module
- Filter module
- Help module
- Lock API

- Mail system / API
- Menu UI module
- Node module
- Options module
- Seven theme
- Syslog module
- Telephone module
- Text Field module
- Token module
- Tracker module

\*as of 8.3.0. Plus some like "base system" and "bootstrap" you can't really avoid maintaining.

# Discussion (Hit?) List

- Actions module & actions plugin api
- Aggregator module
- History module
- Statistics module
- Ban module
- Shortcut module
- Tracker module
- Forum module
- Color module
- Book module
- Tour module
- RDF module



# Rough Idea of Maintenance Burden

How many pages of issues since 8.x was started approx 7 years ago  
Include bugs and tasks; critical, major, and normal severity

[https://www.drupal.org/project/issues/search/drupal?status\[0\]=1&status\[1\]=13&status\[2\]=8&status\[3\]=14&status\[4\]=15&status\[5\]=2&status\[6\]=7&priorities\[0\]=400&priorities\[1\]=300&priorities\[2\]=200&categories\[0\]=1&categories\[1\]=2&version\[0\]=9.x&version\[1\]=8.x&version\[2\]=7.x&component\[0\]=book.module&order=created&sort=desc](https://www.drupal.org/project/issues/search/drupal?status[0]=1&status[1]=13&status[2]=8&status[3]=14&status[4]=15&status[5]=2&status[6]=7&priorities[0]=400&priorities[1]=300&priorities[2]=200&categories[0]=1&categories[1]=2&version[0]=9.x&version[1]=8.x&version[2]=7.x&component[0]=book.module&order=created&sort=desc)



# Actions module & actions plugin api

"The Actions module provides tasks"

Enabling the module exposes configurations for Actions plugins, but there is no obvious way in core to trigger them?

In D7 VBO used actions as did Rules. In D8 you can add the "Node operations bulk form" field to allow you to apply actions to the view's results. That's so hidden as to be useless?

Rules module has its own plugin class and does not use core Actions.

**Maintenance Burden: 1**



# Aggregator module

"The Aggregator module is an on-site syndicator and news reader that gathers and displays fresh content from RSS-, RDF-, and Atom-based feeds"

**Maintenance Burden: 3**





# History module

"The History module keeps track of which content a user has read."

Shows that red "**new**" flag in listings.

Maintenance Burden: 0.5





# Statistics module

"The Statistics module shows you how often content is viewed."

This used to only work for uncached or authenticated pages. Now it has a bit of JS that does a POST to an extra php file to record the visit.

However, this means you are doing an extra (light) bootstrap of Drupal even for cached pages!

Only works for nodes, not other pages, might not work with default .htaccess file.

**Maintenance Burden: 1**



# Ban module

"The Ban module allows administrators to ban visits to their site from individual IP addresses."

Uses a request middleware to short-circuit the request from a banned IP. This runs before the page cache.

Still means some database queries and other overhead compared to blocking before the request gets to PHP. Individual IP blocking is not very effective against a non-trivial attacker.

**Maintenance Burden: 1**



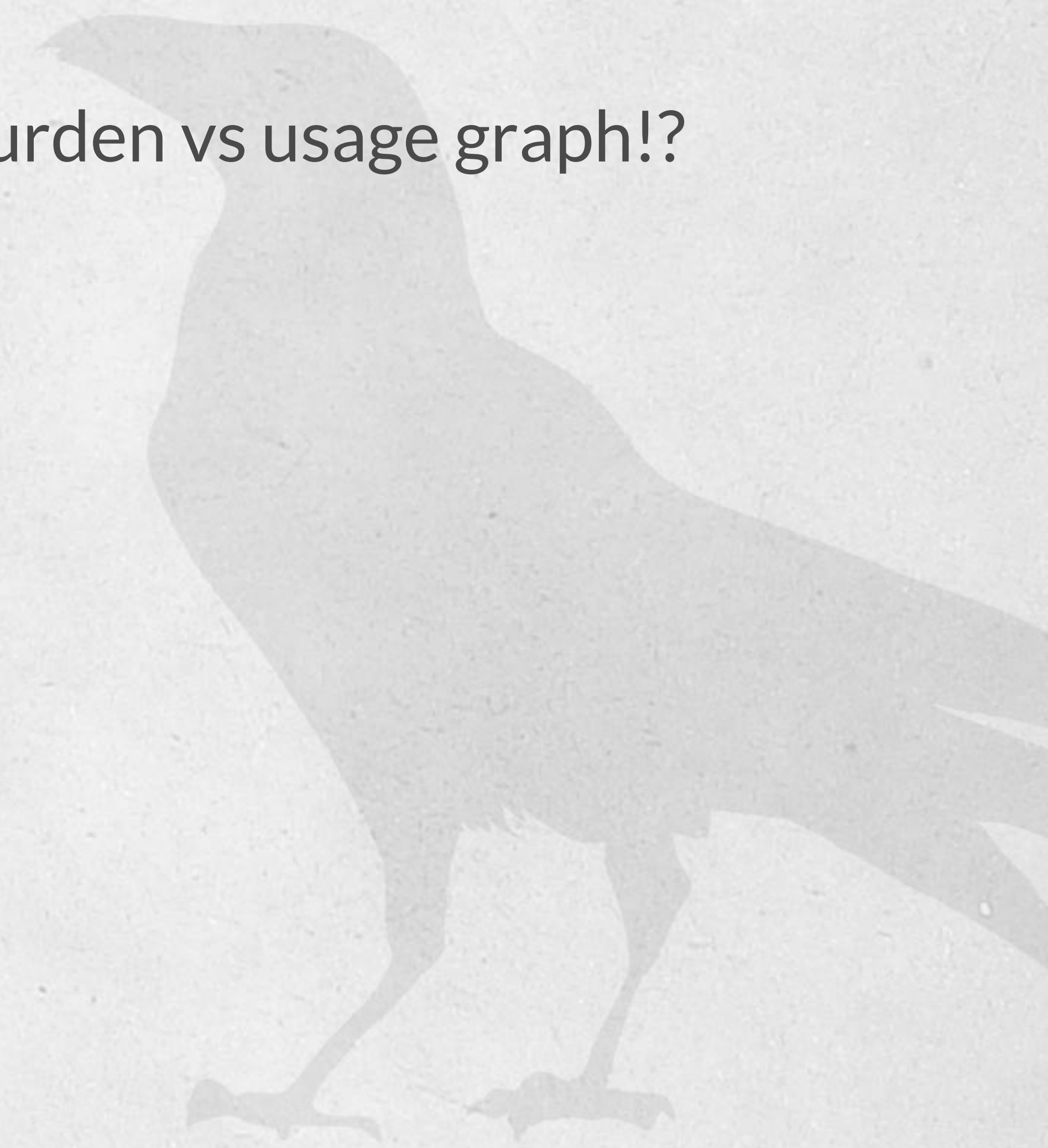
# Shortcut module

"The Shortcut module allows users to create sets of *shortcut* links to commonly-visited pages of the site."

A solution in search of a problem?

Support burden vs usage graph!?

Maintenance Burden: 2





# Tracker module

"The Activity Tracker module displays the most recently added and updated content on your site, and allows you to follow new content created by each user."

**Maintenance Burden: 1**





# Forum module

"The Forum module lets you create threaded discussion forums with functionality similar to other message board systems."

Issue to remove #1898812 open since 2013.

In that issue people wanted to move forum listings to more standard (Views) approaches, but that didn't happen.

**Maintenance Burden: 2**





# Color module

"The Color module allows users with the *Administer site configuration* permission to change the color scheme of compatible themes."

Is this important to the product for first time users?

Maintenance Burden: 1





# Book module

"The Book module is used for creating structured, multi-page content, such as site resource guides, manuals, and wikis."

Is the name alone the reason it's overlooked, or are there other key feature or usability gaps?

**Maintenance Burden: 2**

The core Book module has long been a useful but under-utilized tool for structuring large amounts of content in hierarchies, including for documentation and wiki-like sites.

Replace Book with a core module that is functionally similar (and has an upgrade path) but with a more intuitive name, possibly better re-use of other core code and any key UX improvements?



# Tour module

"The Tour module provides users with guided tours of the site interface."

We only have one written so far - is it useful?

Maintenance Burden: 1





# RDF module

"The RDF module enriches your content with metadata to let other applications better understand its relationships and attributes."

This modifies a lot of markup, so can be problematic for existing themes.

<https://developers.google.com/search/docs/guides/intro-structured-data>  
"JSON-LD is the recommended format."  
"

Though Microdata and RDFa are also listed

**Maintenance Burden: 1**



# Do We Know **ANYTHING** About Usage?

<https://www.drupal.org/node/2867597>

=> Top Drupal 7 and Drupal 8 core sub-modules

Beware - this includes CI, so lower usage range may reflect breadth of test coverage more than usage.



# Can We Have Small Core? + ...

- Drupal is the product - core plus features
- Can we split out the real "core" that is only essential?
- Build core plus all the other modules
- Can that accelerate advancement of the product?

<https://ffwagency.com/digital-strategies-blog/proposal-future-drupal-core-development>



# Join Us for Contribution Sprints

**Friday, April 28, 2017**

**First-Time Sprinter  
Workshop**

9:00am-12:00pm

Room: 307-308

**Mentored Core Sprint**

9:00am-12:00pm

Room:301-303

**General Sprints**

9:00am-6:00pm

Room:309-310

**#drupalsprints**



# WHAT DID YOU THINK?



Locate this session at the  
DrupalCon Baltimore website:  
[events.drupal.org/node/17573](https://events.drupal.org/node/17573)

Take the survey!  
[https://www.surveymonkey.com  
/r/drupalconbaltimore](https://www.surveymonkey.com/r/drupalconbaltimore)

# THANK YOU!