

# Drupal Site Tuneup

***Vroom! Vroom!***

Kristen Pol /  
kristen@hook42.com



# Me

## *Kristen Pol!*

- Drupal Architect
- 10 years of Drupal!
- kristen@hook42.com
- drupal.org/user/8389
- @kristen\_pol

## *Hook 42!*

- hook42.com
- @hook42inc



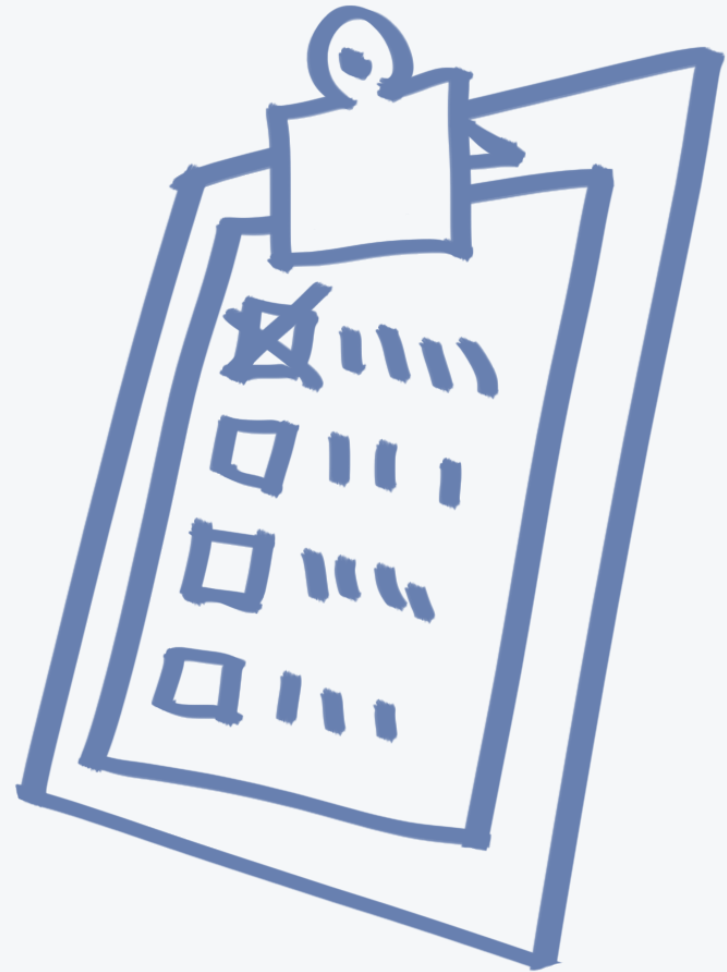
# Tuneup *Strategy!*



# Strategy

## *Why Tuneup?!*

- Better performance
- Easier site maintenance
- Easier onboarding
- Better credibility
- Better user experience
- Better editor experience



# Strategy

## *Make A List!*

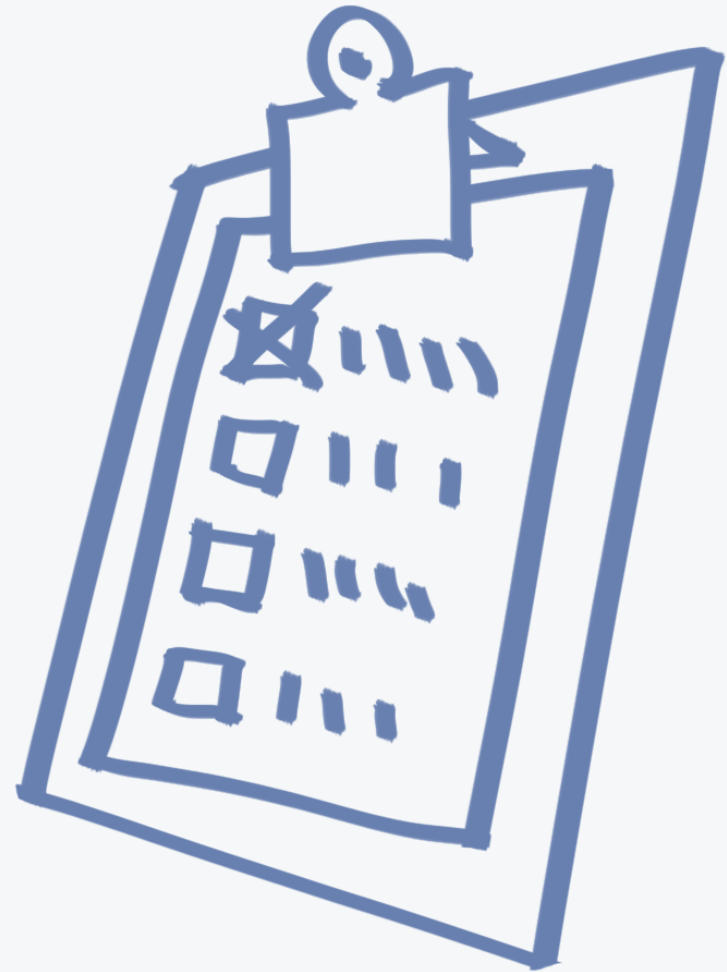
- Users
- Content
- Content types
- Code
- Modules
- Views
- Data
- Files



# Strategy

## *Make A Plan!*

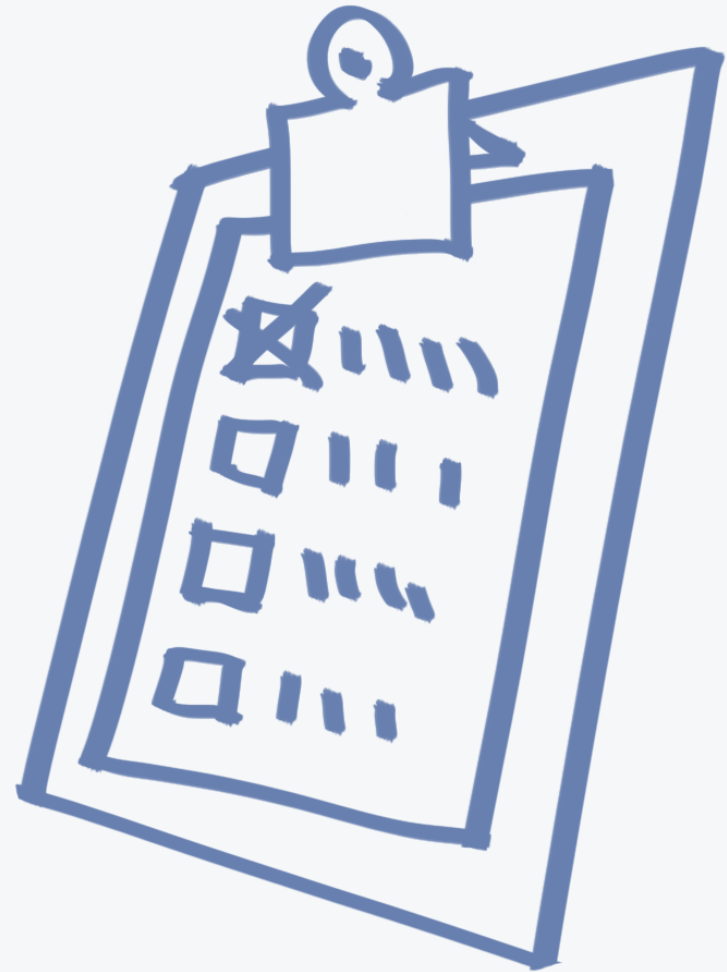
- Prioritize
- Slow and steady
- Balance with ongoing enhancements
- Small changes
- Document



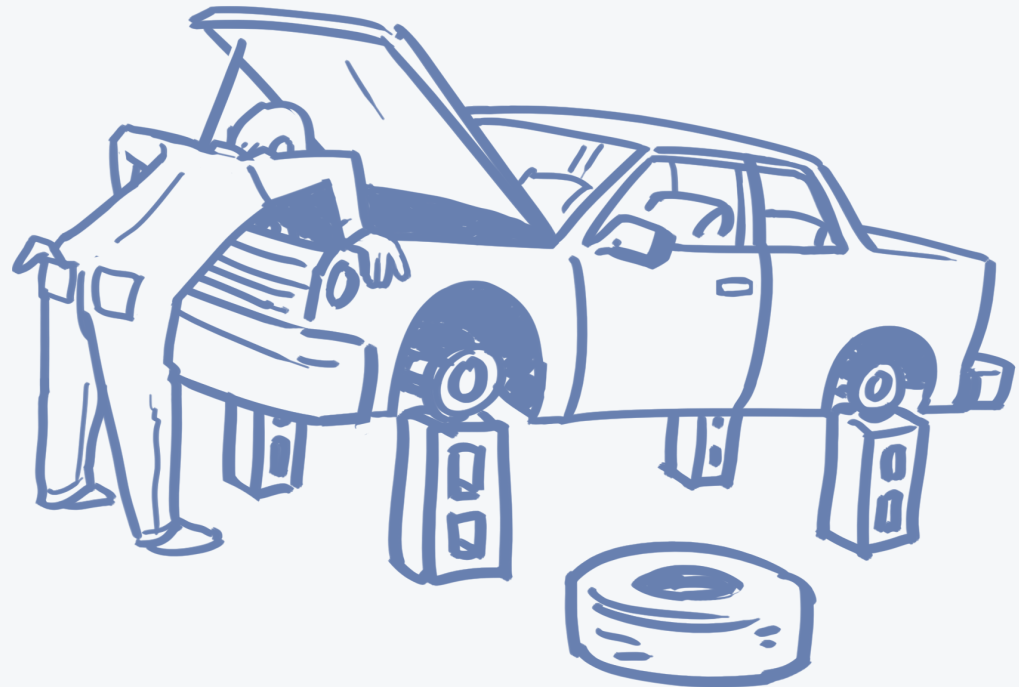
# Strategy

## *Make It Happen!*

- Track with bug / task tracker
- Beware of Features
- Lots of testing
- Peer review
- Local => Dev => Test => Prod
- BE CAREFUL... YOU ARE DELETING THINGS!



# Tuneup *Users!*





# Users

## *Why Spam?!*

- Spambots
- Paid spammers
- Competitors
- Looky-loos
- Testing



# Users

## *Identify Spam!*

- “Bad” countries
- Last access time
- Bounced emails
- Wrong role
- Other users
- Test email domains
- Manual checking



# Users

## *Remove Spam!*

- Modules
  - Administration Views
  - Advanced User
  - LoginToboggan
  - Inactive User
- Custom solution
  - Rules
  - Cron + user\_delete (and node\_delete)



# Users

## *Reduce Spam!*

- Prevent site access
  - IP blocking :/
  - Cloud Flare
  - Bad Behavior
- Prevent form submissions
  - Botcha, Honeypot, Mollom\*, Spamicide\*, Spambot\*, etc.
- Hide user/register page
  - Send link directly
- Control roles
  - Role delay
  - Manual role promotion
  - Coded role promotion



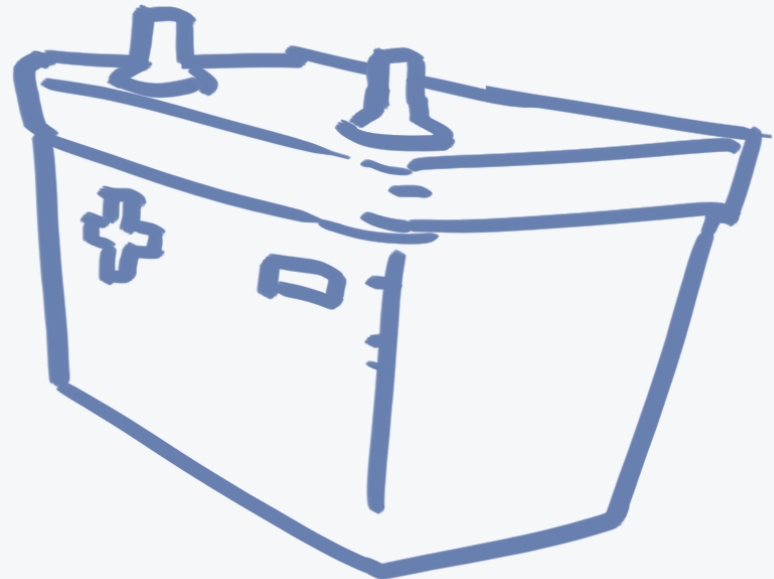
# Tuneup *Content!*



# Content

## *Spam!*

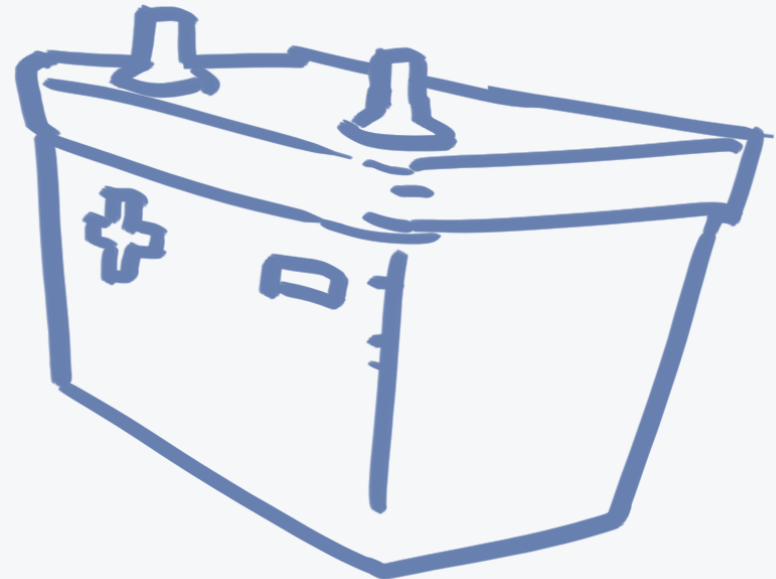
- Nodes, comments, entityform data, etc.
- Similar to users but need a way to mark as spam



# Content

## *Why Duplicates?!*

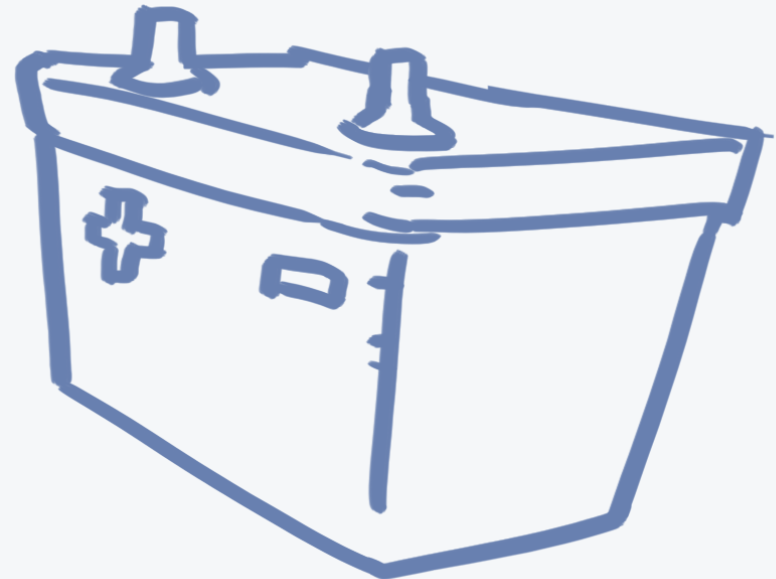
- Spammers
- Multiple submit clicks
- Copy/paste again
- Node clone



# Content

## *Identify Duplicates!*

- Google Webmaster Tools
  - Search Appearance => Optimization => HTML Improvements => Duplicate title tags
- Uniqueness module
- Query database
- Mark duplicates
  - Taxonomy
  - Flag
  - Title

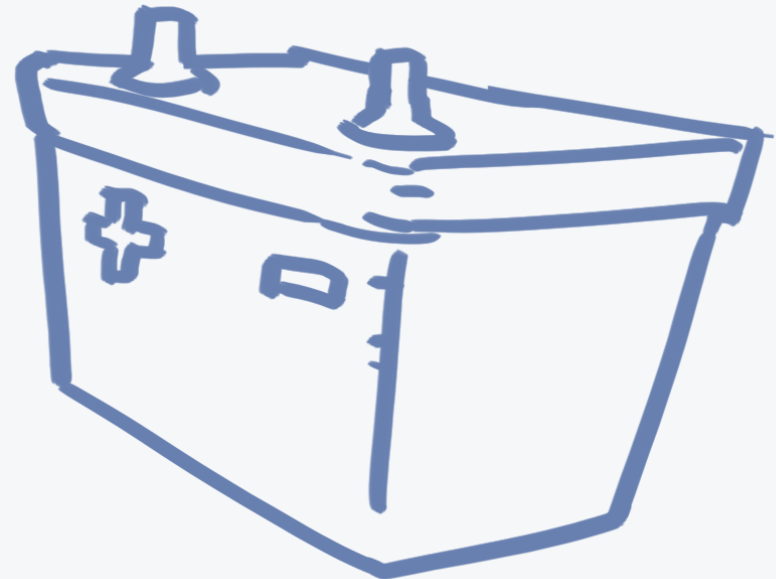




# Content

## *Remove Duplicates!*

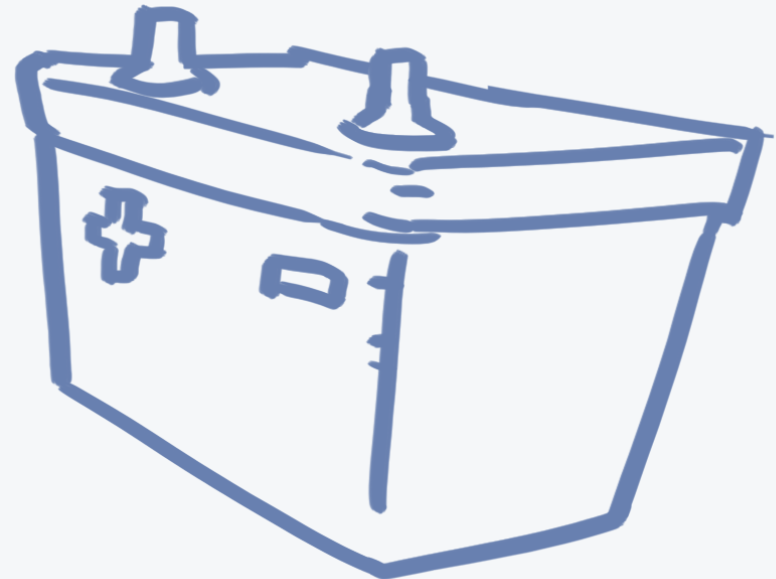
- Manual
  - Custom view with Views Bulk Operations
- Automated
  - Custom cron code (node\_delete)
  - Custom rules



# Content

## *Reduce Duplicates!*

- Restrict access
- Prevent multiple button clicks (jquery)
- Uniqueness module
- Unique Field module
- Workbench Moderation module



# Tuneup

## *Content Types!*



# Content Types

## *Why Cruft?!*

- Requirements changed
- Replaced functionality with community module
- Content type no longer used
- Field no longer used
- Field needs replacement
- Temporary fields
- Identified fields to delete for better performance



# Content Types

## *Identify Cruft!*

- Fields page (admin/reports/fields)
- Talk to client and team
- Research alternative modules/methods



# Content Types

## *Remove Content Type!*

- Use grep
- Remove all nodes (or change node type, if desired, via Node Convert)
- Remove all references in code
- Remove all references from Views, Rules, Context, etc.
- Delete from Features modules
- Remove via UI or `hook_update_N` (see `node_type_delete_confirm_submit`)



# Content Types

## *Remove Field!*

- Use grep
- Remove all references in code
- Remove all references from Views, Rules, Context, etc.
- Delete from Features modules (field base and field instance)
- Remove via UI or via `hook_update_N` (`field_delete_field`) or via `drush field-delete`



# Content Types

## *Reduce Cruft!*

- Careful content architecture
- Document content types in spreadsheets first
- Reuse content types and fields when it makes sense





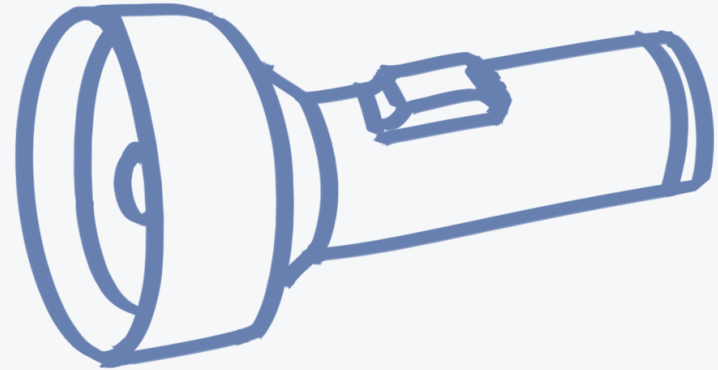
# Tuneup *Views!*



# Views

## *Why Cruft?!*

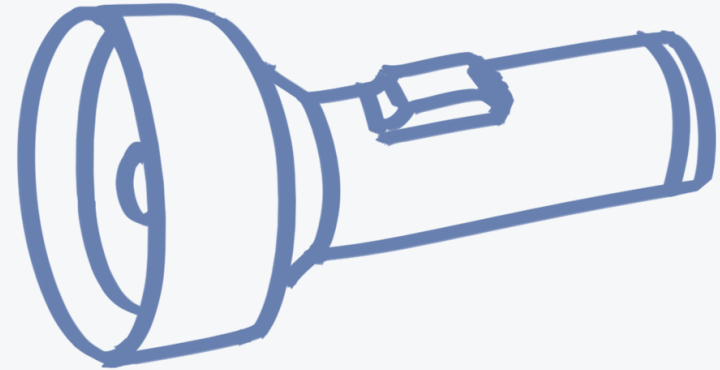
- Requirements changed
- View no longer used
- Replaced functionality with community module
- Identified views to delete or replace for better performance



# Views

## *Identify Cruft!*

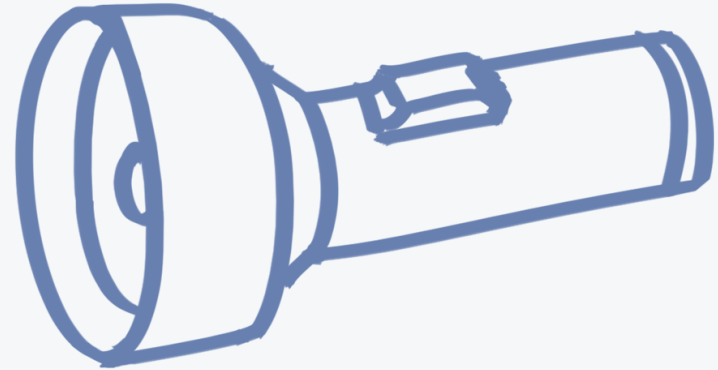
- Views Usage Audit module
  - Use temporarily
- Performance checks
- Manual audit
- Talk to client and team



# Views

## *Remove Cruft!*

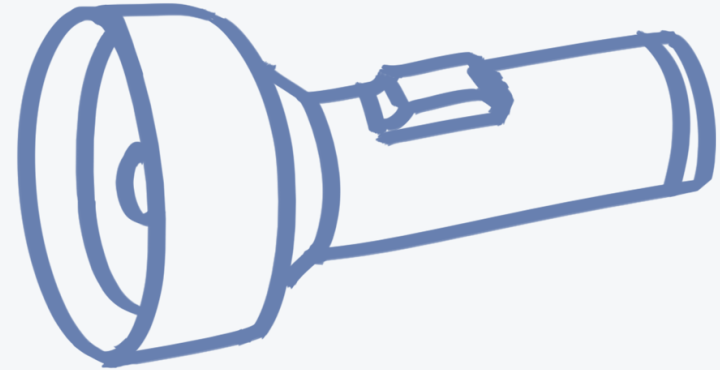
- Use grep
- Remove all references in code
- Remove all references from Context, Panels, other Views, etc.
- Delete from Features modules
- Remove via `hook_update_N` (`views_delete_view`) or via UI



# Views

## *Reduce Cruft!*

- Careful views architecture
- Add good machine names
- Use view modes
- Reuse views when it makes sense



# Tuneup

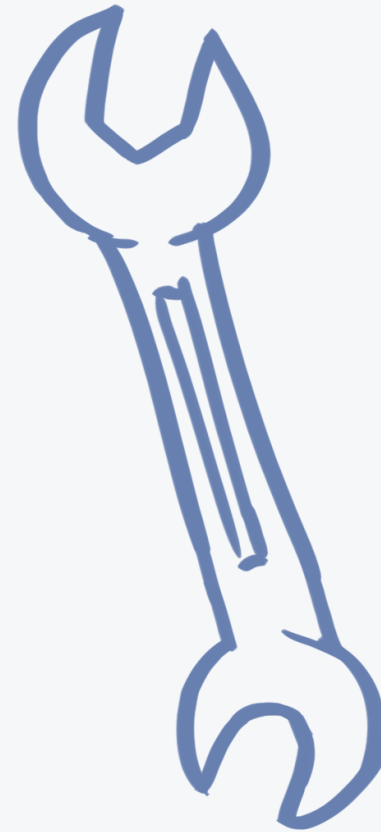
*Code / Modules!*



# Code / Modules

## *Why Cruft?!*

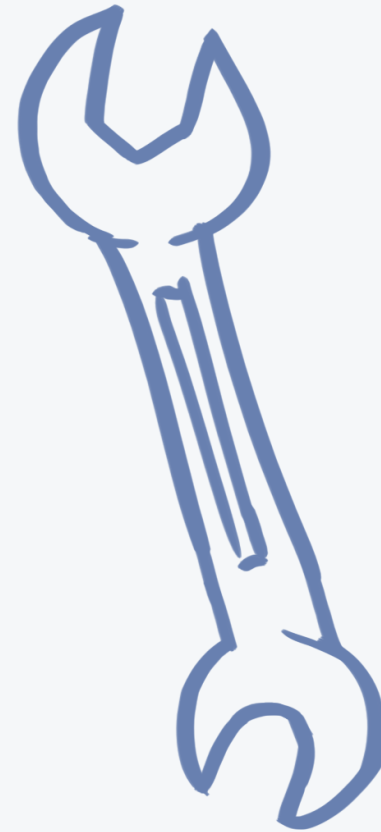
- Modules not uninstalled properly
- Requirements changed
- Code or module no longer used
- Replaced functionality with another module
- Identified code to replace for better performance
- Patch has been applied to module
- Code has been hacked



# Code / Modules

## *Identify Cruft!*

- Disabled modules list  
(admin/modules/uninstall)
- Site Audit module
- Auditor module
- Hacked module
- Review patches file and issue queues
- Performance checks
- Manual audit

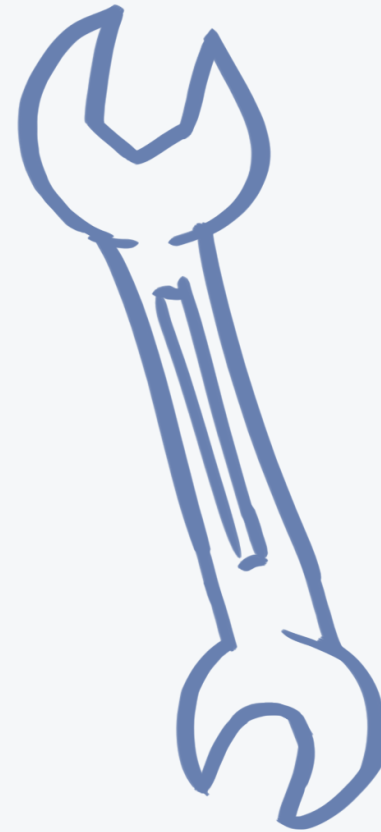




# Code / Modules

## *Remove Module!*

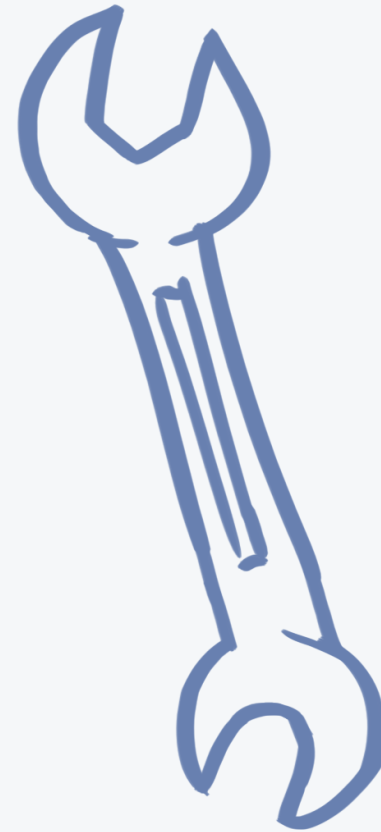
- Remove all references in code
- Delete from Features modules
- Backup prod db
- Uninstall on prod, test, dev, and local (or on prod and then copy db to test, dev, and local)
  - hook\_update\_N (hook\_uninstall\_modules) or UI or drush pm-uninstall
- Remove from git
- Push to prod on next release



# Code / Modules

## *Remove Patches!*

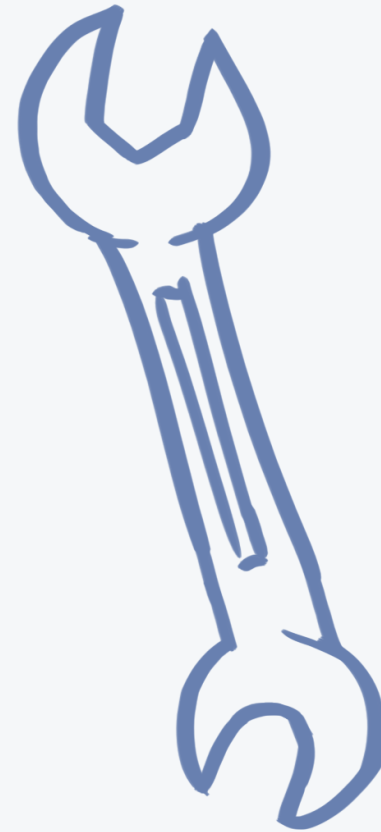
- Make sure patch has been applied to latest release
- Check release notes
- Update module code locally
- Confirm patch is there
- Run `update.php` or `drush updatedb` locally
- Test!
- Remove patch from patches directory and README file
- Normal dev cycle



# Code / Modules

## *Reduce Cruft!*

- Careful code architecture
- Reuse code when it makes sense
- Check module usage and issue queues
- Think carefully about when to use community modules vs custom code
- Keep all custom and community patches in directory with README
- Uninstall modules before deleting code



# Tuneup

*Data / Files!*



# Data / Files

## *Why Cruft?!*

- Modules deleted before uninstalled
- Modules not deleting variables upon uninstall
- Files uploaded again
- Files changed on content and old files remain on file system
- Someone deleting files
- File migration incomplete or had errors



# Data / Files

## *Identify Cruft!*

- Variable Cleanup module
- Audit Files module (issue queue)
- Manually check database tables



# Data / Files

## *Remove Crufty Variables!*

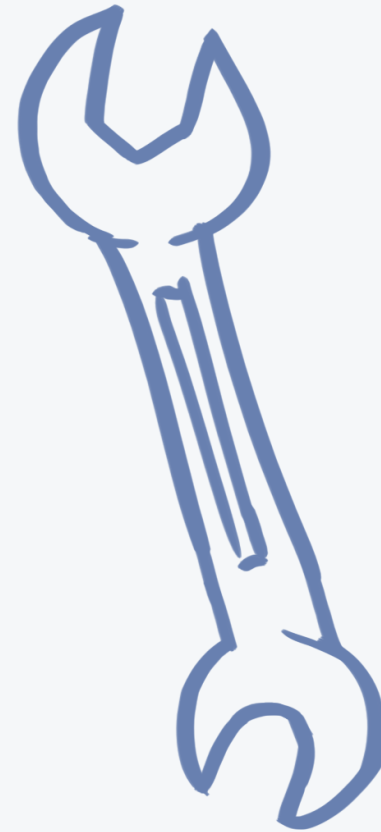
- BE VERY VERY CAREFUL!
- Variable Cleanup module
- Test locally first
- Backup prod db before using
- Only delete variable if you know what it is and are positive it can be deleted
- TEST TEST TEST!



# Data / Files

## *Reduce Cruft!*

- Uninstall modules before deleting code
- Reuse files with Media module





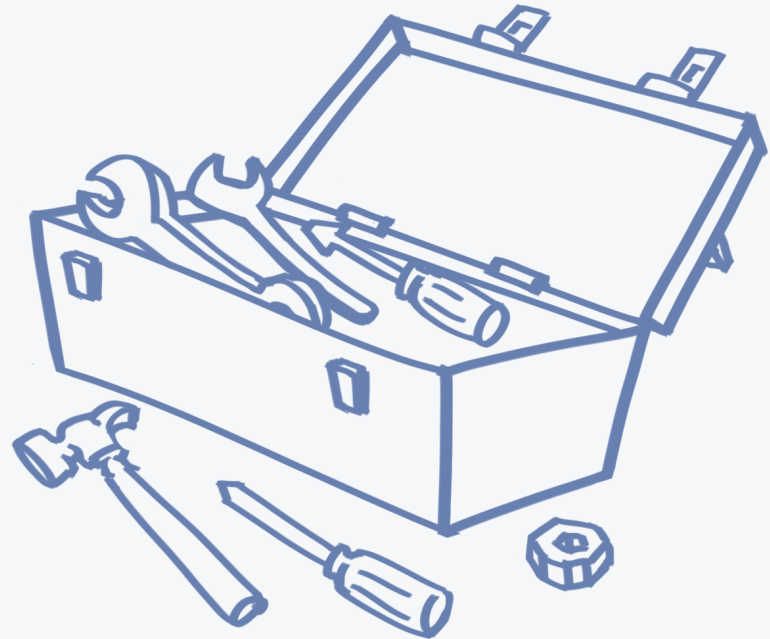
# Tuneup *Takeaways!*



# Takeaways

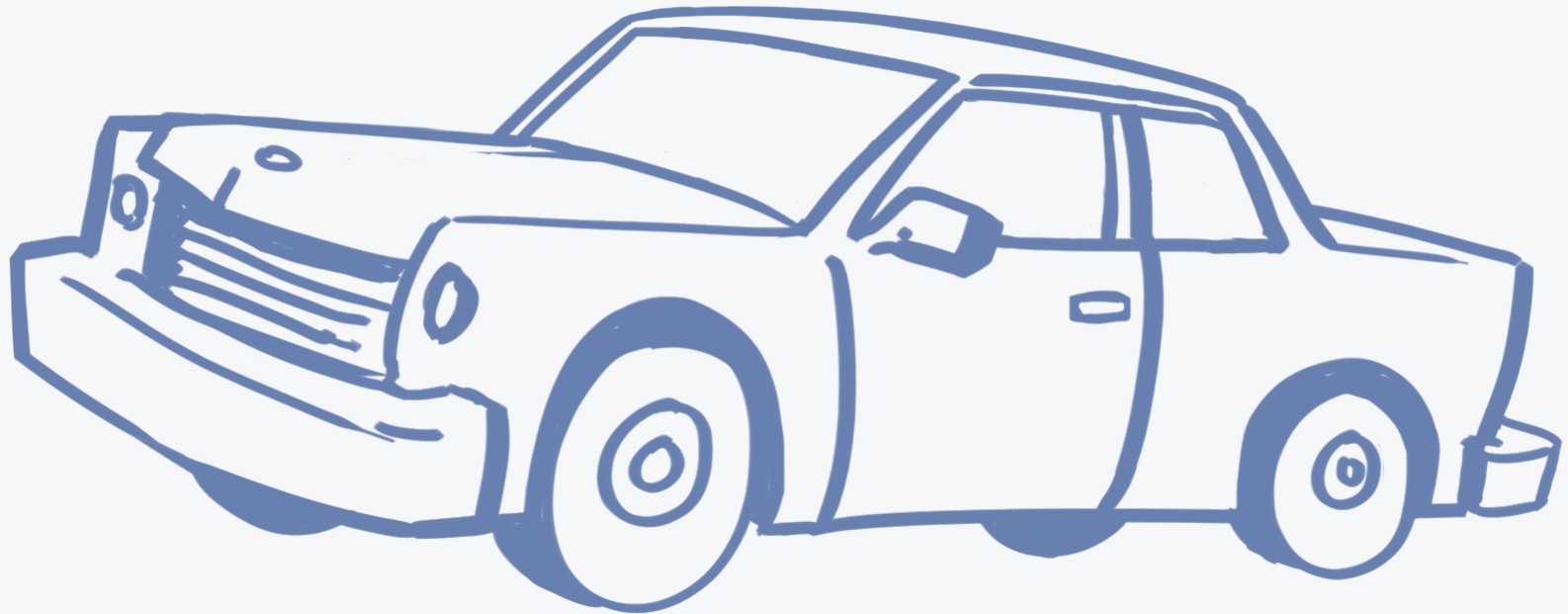
## *Reduce Cruft!*

- Careful architecture and planning
- Document before implementing
- Reuse when it makes sense
- Keep up-to-date on latest community modules and Drupal trends
- Think three times about adding anything
- Think four times before naming anything



# Tuneup

*Vroom Vroom!*



# Questions?

