

# Rebuild Your Drupal Engine

*Drupal-to-Drupal Migration  
with Migrate and Migrate D2D*

Kristen Pol /  
kristen@hook42.com



# Me

## *Kristen Pol*

- Drupal Architect
- 10 years of Drupal!
- [kristen@hook42.com](mailto:kristen@hook42.com)
- [drupal.org/user/8389](https://drupal.org/user/8389)
- [@kristen\\_pol](https://twitter.com/kristen_pol)

## *Hook 42*

- [hook42.com](https://hook42.com)
- [@hook42inc](https://twitter.com/hook42inc)



# Migration

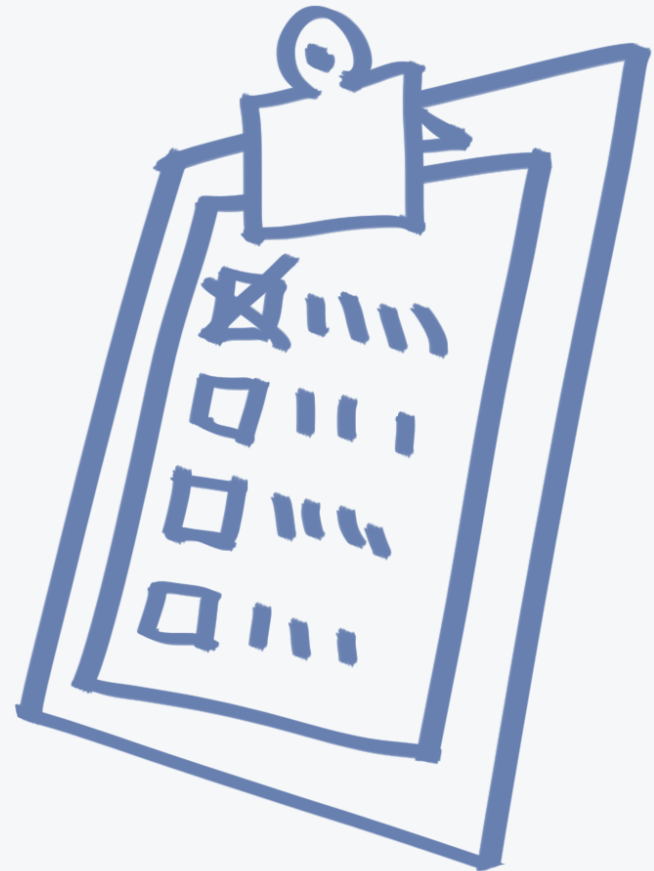
## *Planning*



# Planning

## *Why a Migration?*

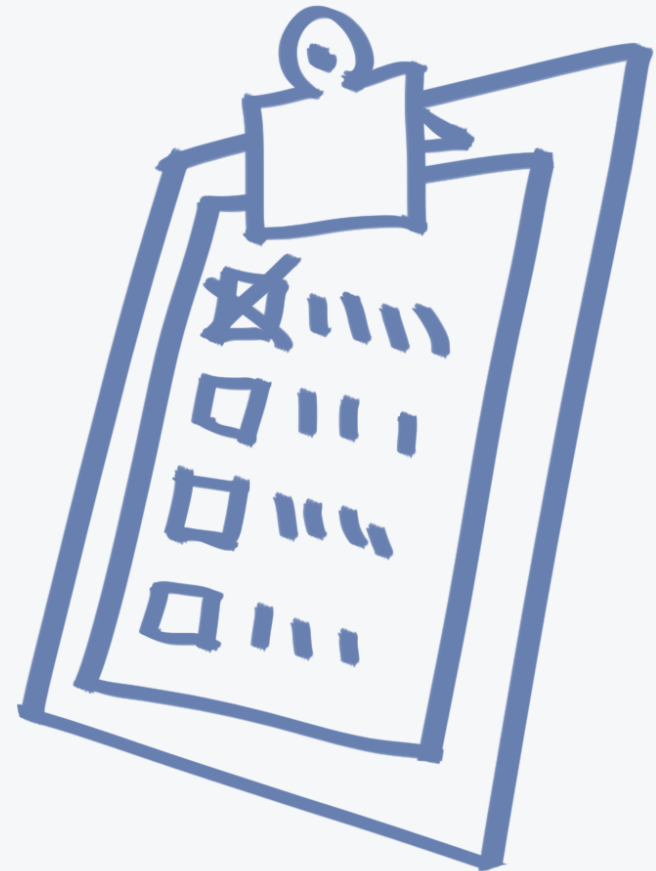
- Core no longer supported
- Contrib no longer supported
- Hacked code
- New Drupal features
- Big redesign / refactor
- All of the above



# Planning

## *Migration Options*

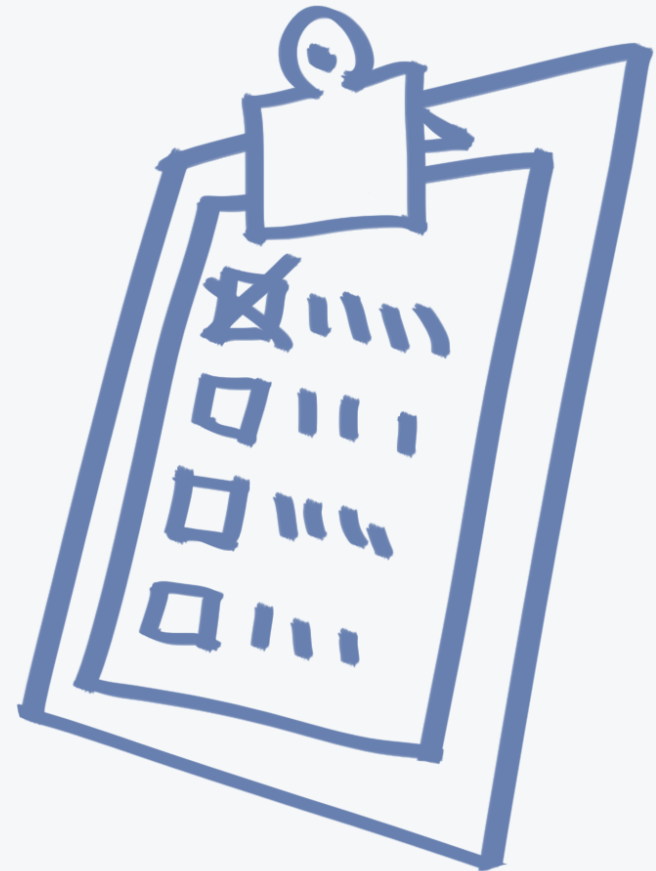
- Upgrade\*
- Manual
- Custom Code
- Feeds modules
- Migrate modules
- Combination of above



# Planning

## *Migrate vs. Others*

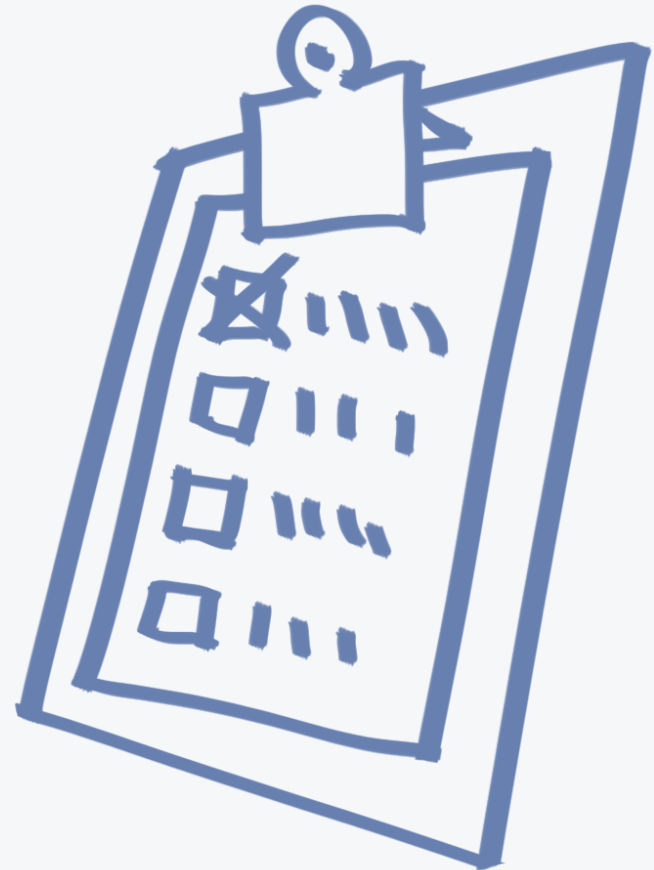
- Advantages over upgrade
  - Hacks
  - Redesign / refactor
- Advantages over manual
  - Large amounts of content
  - Content conversion
- Advantages over custom
  - Speed of execution
  - Maintainability



# Planning

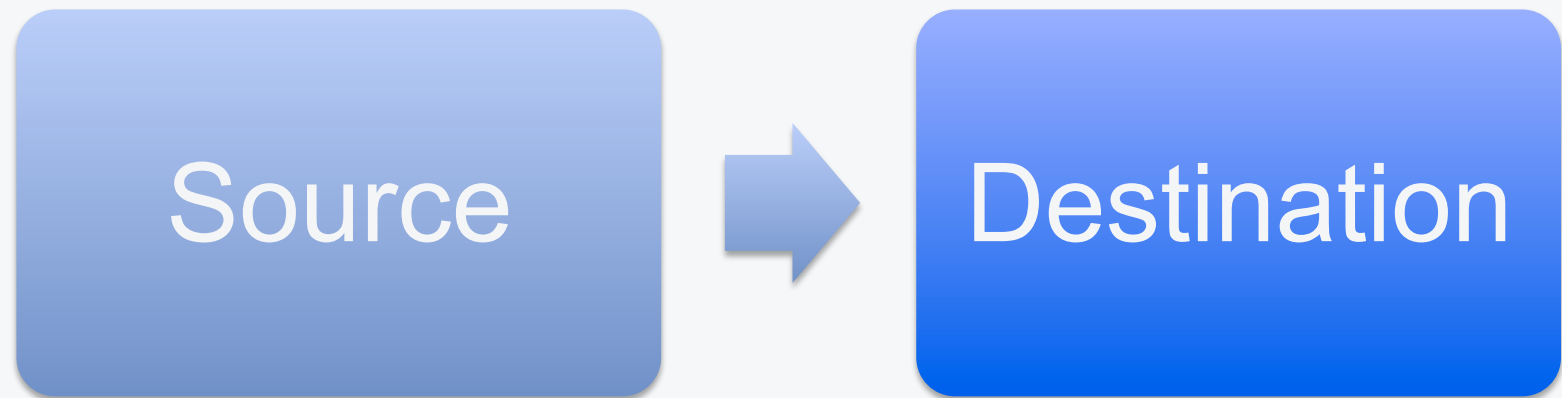
## *Migrate vs. Others*

- Advantages over Feeds
  - Roll backs
  - Dependencies
  - Performance
  - Custom logic



# Planning

## *Migration Mapping*

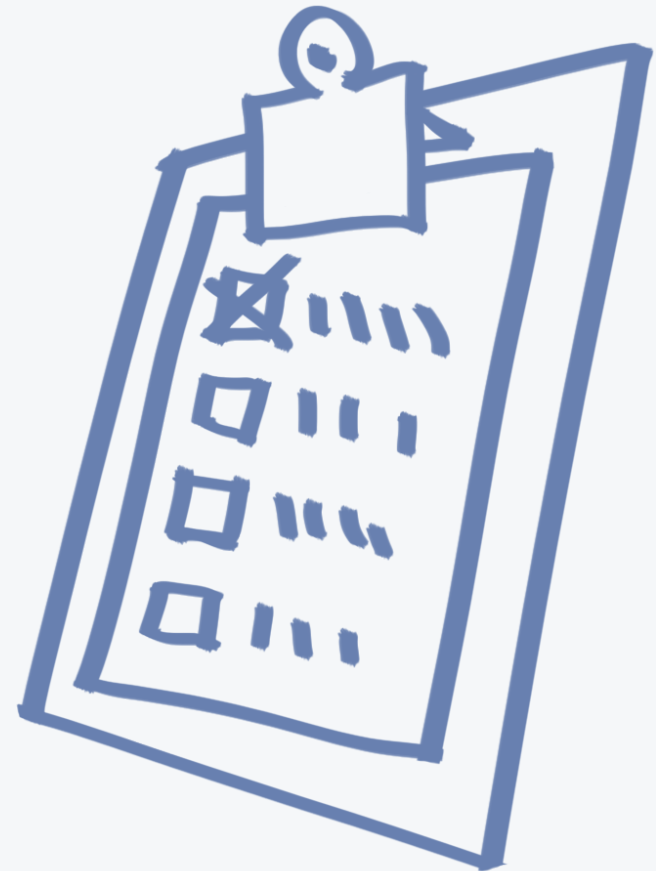




# Planning

## *Migration Mapping*

- Track in spreadsheet
- Source to destination
- Entity type
- Number of entities
- Field human name
- Field machine name
- Field type
- Field config notes
- Field migration notes



# Planning

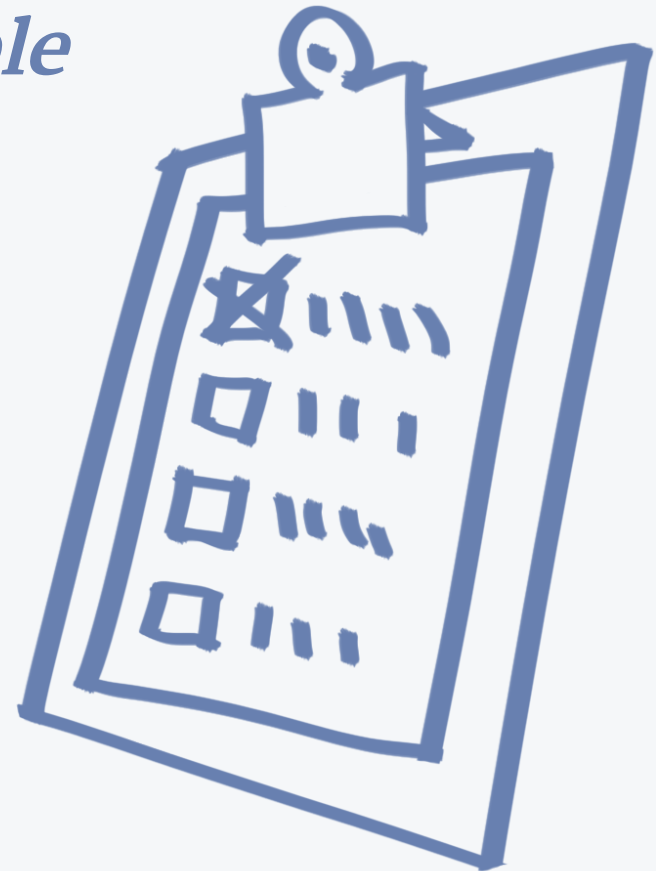
## *Simple Mapping Example*

<b>Press</b>	Press releases and news articles							
	<b>Entity</b>	<b>Machine Name</b>						
<b>D5</b>	content type	press						
<b>D7</b>	content type	press						
<b>D5 Name</b>	<b>D5 Machine Name</b>	<b>D5 Type</b>	<b>D5 Config</b>	<b>D7 Name</b>	<b>D7 Machine Name</b>	<b>D7 Type</b>	<b>D7 Config</b>	<b>Notes</b>
Title	title	core	core	Title	title	core	core	Press from [r
Date	field_press_date	Date	popup calen	Date	field_date	Date	use d5 conf	use this as p
Publication	field_press_source	Text	1 row / requ	Publication	field_publication	Text field	required / p	use this as p
Logo	field_press_logo	Image	max res = C	Logo	field_logo	Image	same as d5	media file se
Article Link	field_press_link	Link	optional title	Article Link	field_external_link	Link	use d5 config	
n/a	n/a	n/a	n/a	D5 Nid	field_d5_nid	Integer		special migr
Weight	field_weight	Integer	values = -10	(use Weight r	(no field)	(no field)	(no field) / c	special migr
Website	field_press_website	Text	main websi	n/a	n/a	n/a	n/a	do not map

# Planning

## *Simple Mapping Example*

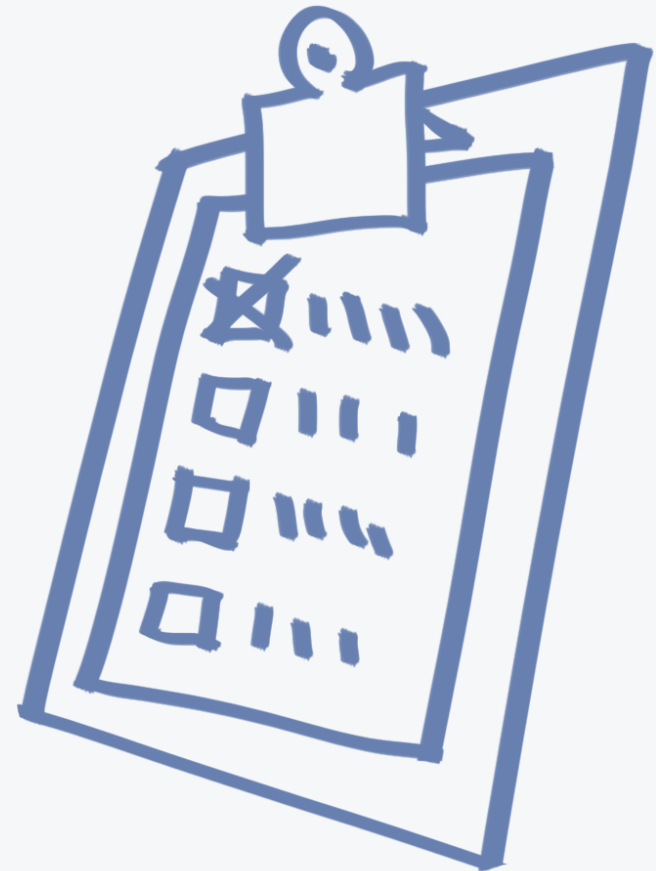
- Content type to content type
- 1 unmapped old fields
- 0 unmapped new fields
- 7 fields mapped
- Consider manual migration if small number of fields and low 100s of nodes or less



# Planning

## *Real Life - Simple*

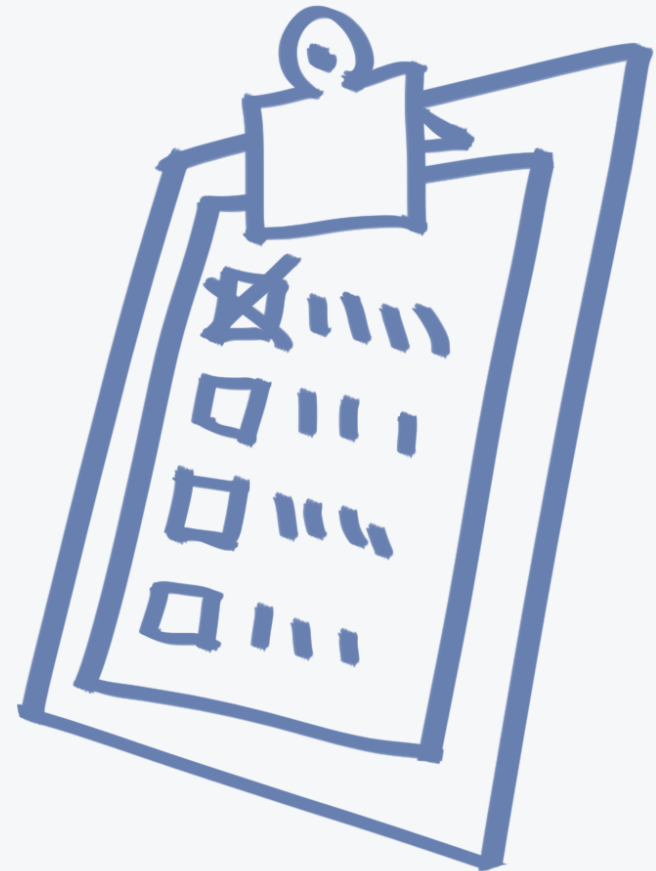
- Content type to content type
- 0 unmapped old fields
- 7 unmapped new fields
- 4 mapped fields
- 137 lines of code\*
- 30 nodes
- Initial migration took minutes
- Could do manual migration



# Planning

## *Real Life - Complex*

- Two content types merged into one content type
- 1 unmapped old field
- 19 unmapped new fields
- 23 mapped fields
- 732 lines of code\*
- 83741 nodes
- Initial migration took days
- Final, incremental migration took minutes



# Migration

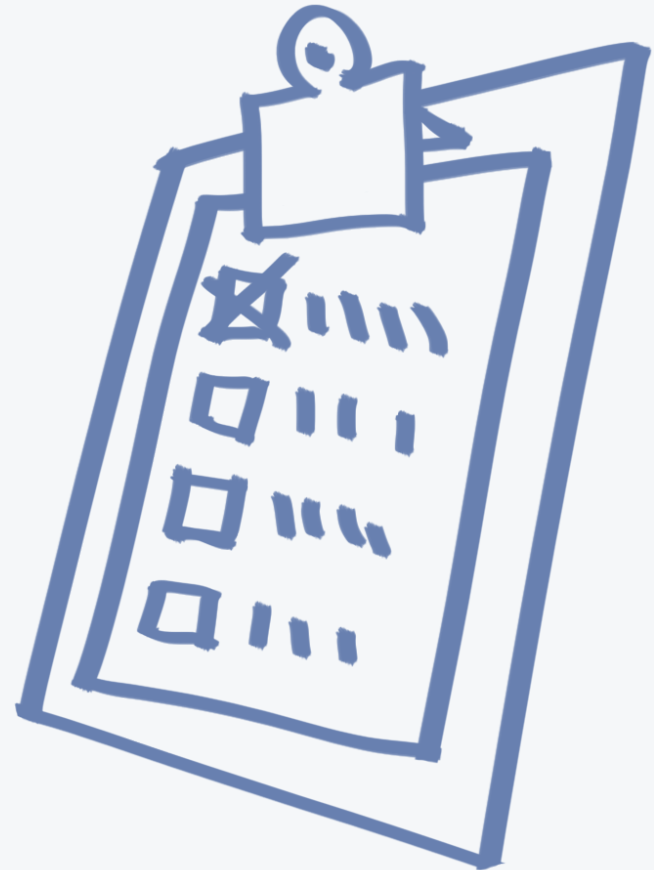
## *Considerations*



# Considerations

## *Development*

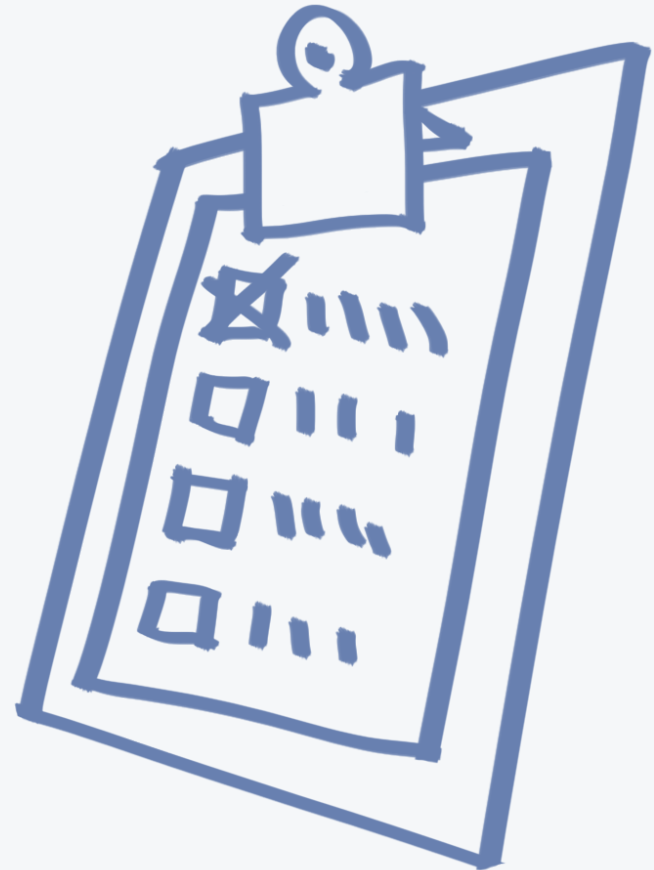
- Test D2D Wizard for simple mapping
- Creating migration code in parallel with development is tricky
- Watch out for field constraints like max size
- Post migration cleanup is sometimes necessary



# Considerations

## *Data*

- Really understand the old data and how it is used (example: text field for URL)
- Beware of bad data that needs to be cleaned up first or not migrated
- Meta data (alt / title text) is stored in different places in different versions

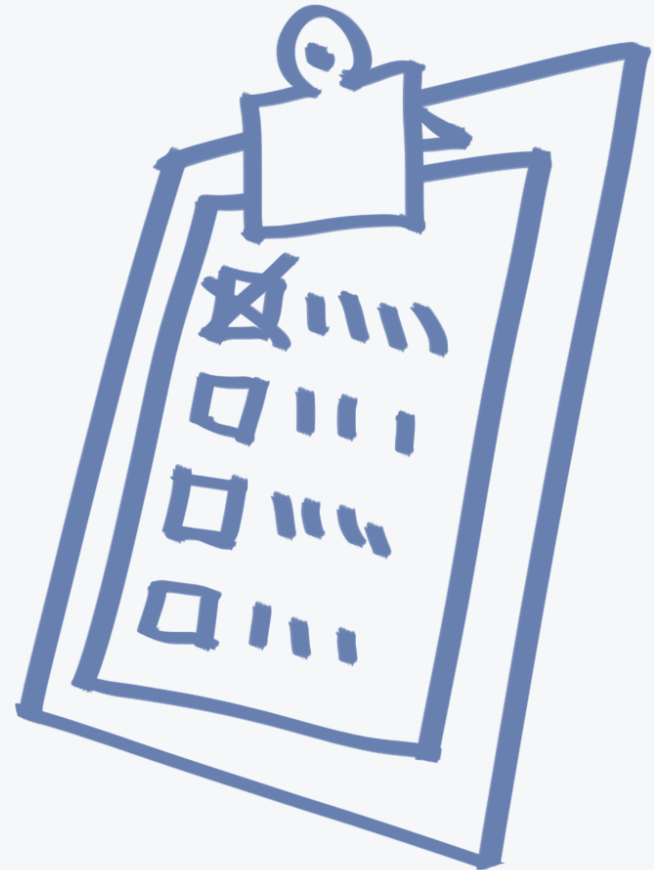




# Considerations

## *File*

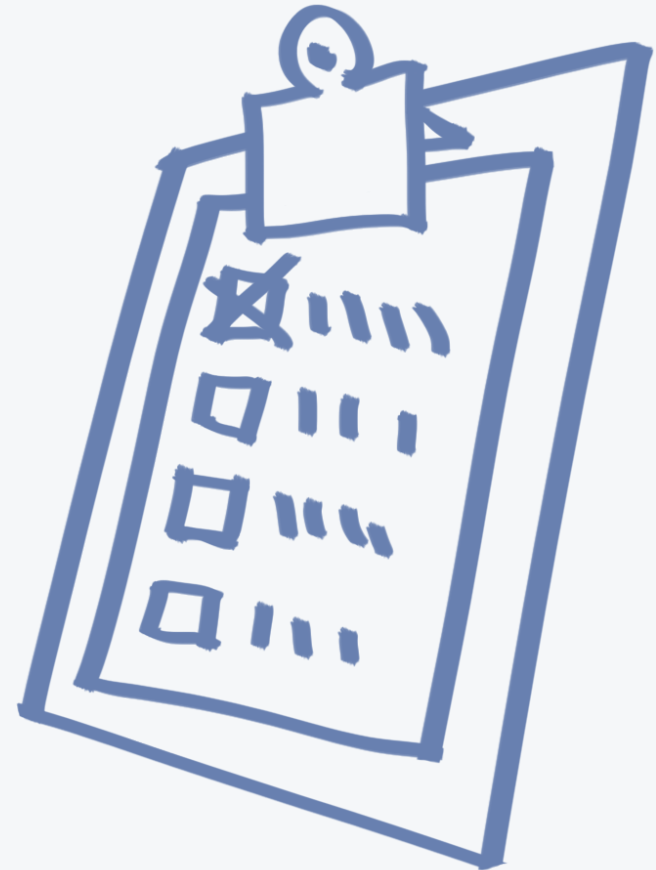
- Capitalization in filenames
- Check for max file size, duplication, special characters (cleanup prior if possible)
- There might be corrupt (0 bytes) files that cause errors
- Can take a LOT of time to copy (e.g. several days)



# Considerations

## *Performance*

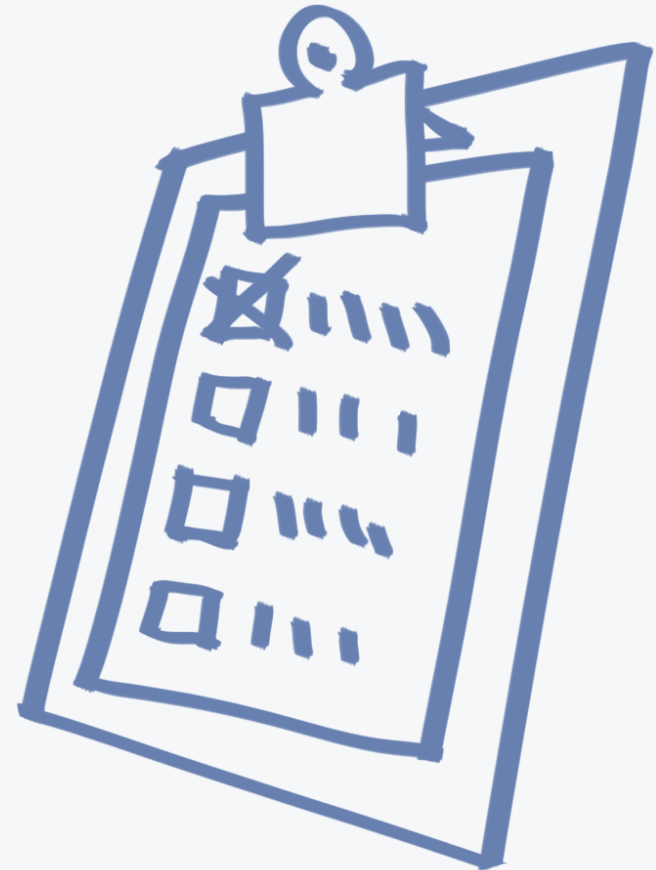
- Performance can be an issue if processing lots of content... time it
- Have both databases on same server if possible
- Deliver files to install directly on new infra if possible (for initial migration)



# Considerations

## QA

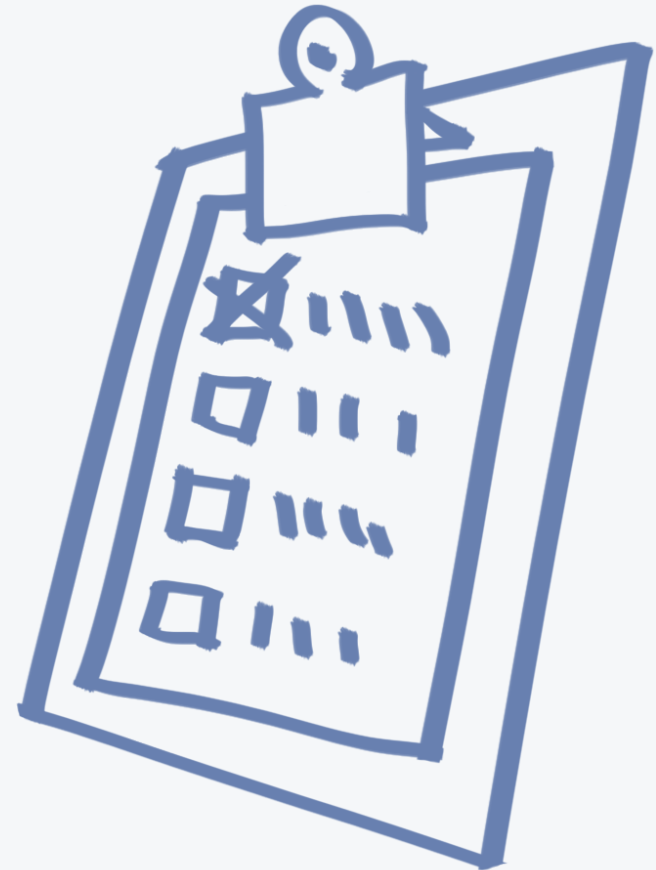
- Architect must peer review migration code
- Architect \*and\* client must QA sample migrated data before official migration starts
- Spot check each migration batch prior to starting next
- Be OCD with QA :)



# Considerations

## *Workflow*

- Babysitting migrations takes time and focus
- Rolling back once official migration starts is painful
- Run through several incremental migrations with checklist to mimic go-live
- Run incremental migration ~day before go-live to keep delta small



# Migration

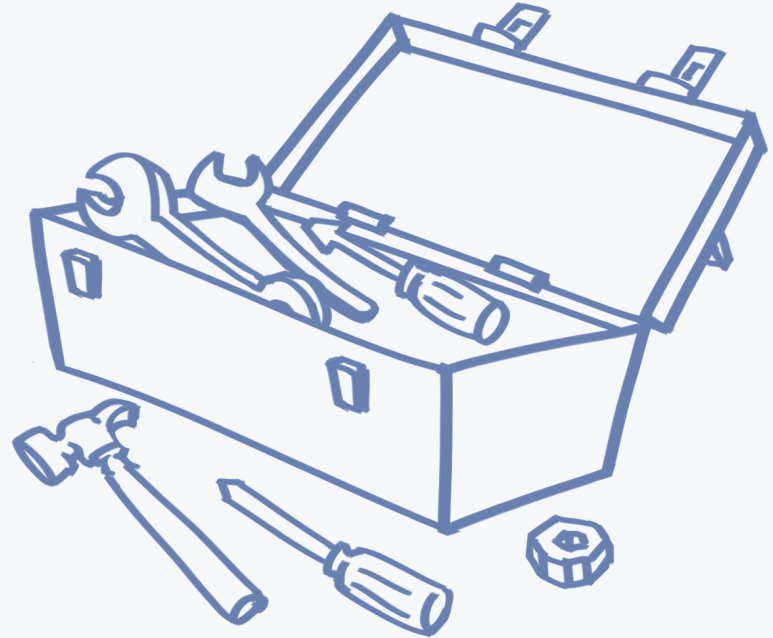
## *Modules*



# Modules

## *Migrate*

- Admin UI
- Object-oriented framework with base classes
- Drush integration
- Example code included:  
migrate\_example and  
migrate\_example\_baseball



# Modules

## *Migrate UI*

Home » Administration » Content » Migrate

### MIGRATE DASHBOARD

CONTENT

COMMENTS

FILES

MIGRATE

Dashboard

Configuration

The current status of each migration group defined in this system. Click on a group name for details on its configuration.

<input type="checkbox"/>	GROUP	STATUS	SOURCE SYSTEM
<input type="checkbox"/>	<a href="#">baseball</a>	Ready to import	
<input type="checkbox"/>	<a href="#">default</a>	Ready to import	
<input type="checkbox"/>	<a href="#">example_group</a>	Ready to import	

#### OPERATIONS


Import

Execute

# Modules

## *Migrate UI*

Home » Administration » Content » Migrate

**BASEBALL** 

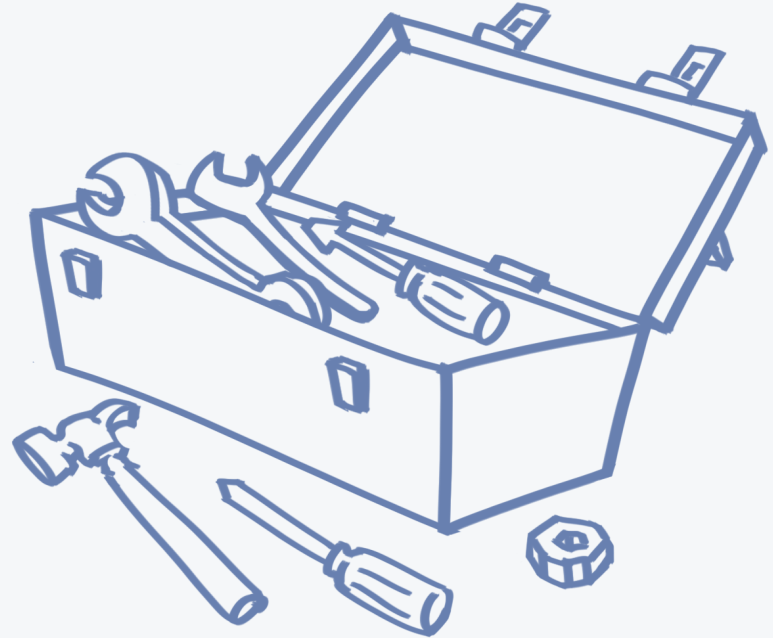
<input type="checkbox"/>	TASK	STATUS	ITEMS	IMPORTED	UNPROCESSED	MESSAGES	THROUGHPUT	LAST IMPORTED
<input type="checkbox"/>	GL2000	Idle	2429	2429	0	0	1352/min	2014-04-17 22:51:29
<input type="checkbox"/>	GL2001	Idle	2429	2429	0	0	1355/min	2014-04-17 22:53:21
<input type="checkbox"/>	GL2002	Idle	2426	791	1635	0	1344/min	2014-04-17 22:54:44
<input type="checkbox"/>	GL2003	Idle	2430	0	2430	0	Unknown	
<input type="checkbox"/>	GL2004	Idle	2428	0	2428	0	Unknown	
<input type="checkbox"/>	GL2005	Idle	2431	0	2431	0	Unknown	
<input type="checkbox"/>	GL2006	Idle	2429	0	2429	0	Unknown	
<input type="checkbox"/>	GL2007	Idle	2431	0	2431	0	Unknown	
<input type="checkbox"/>	GL2008	Idle	2428	0	2428	0	Unknown	
<input type="checkbox"/>	GL2009	Idle	2430	0	2430	0	Unknown	



# Modules

## *Migrate D2D*

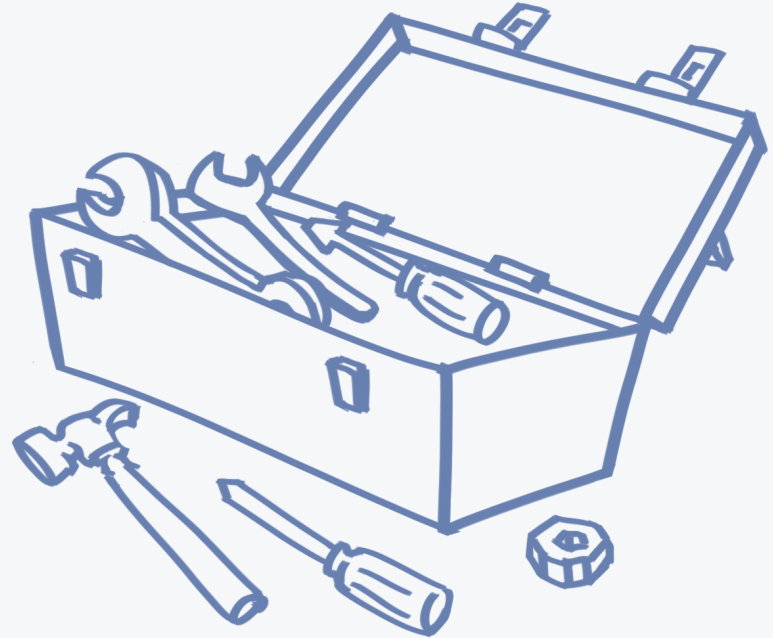
- Extends Migrate to make migrating from D5, D6 or D7 to D7 easier
- Supports all core and standard CCK fields
- Mapping Wizard for simple mapping
- Example code included:  
`migrate_d2d_example`



# Modules

## *Migrate Extras*

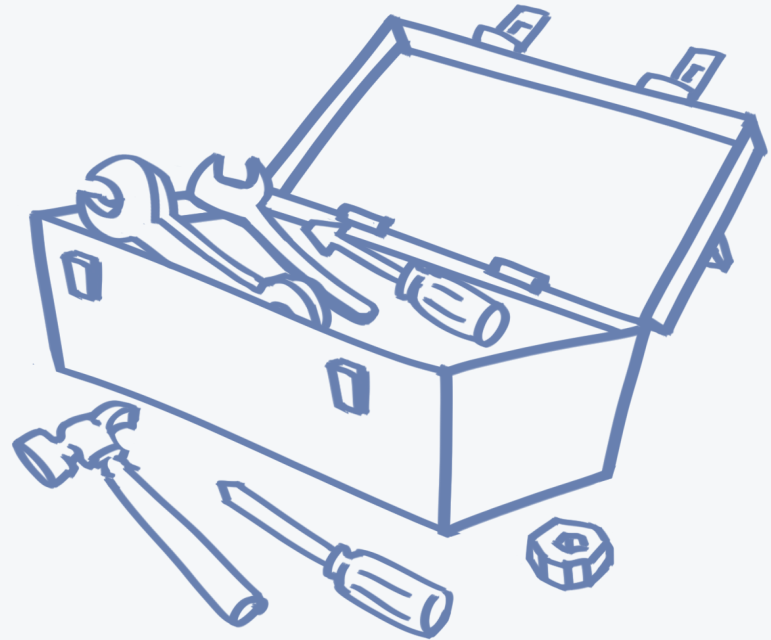
- Extends Migrate for some contrib modules, e.g.
  - Address
  - Flag
  - Media
  - Pathauto
  - Private Messages
  - Profile2
  - Rules
  - Voting API
  - Webform



# Modules

## *Migrate in Drupal 8*

- D8 will not have an in-place upgrade process like previous versions
- Migrate and Migrate D2D code has been modified for D8 core
- Git: [drupalcode.org/project/drupal.git/tree/HEAD:/core/modules/migrate\\_drupal](https://drupalcode.org/project/drupal.git/tree/HEAD:/core/modules/migrate_drupal)



# Migration

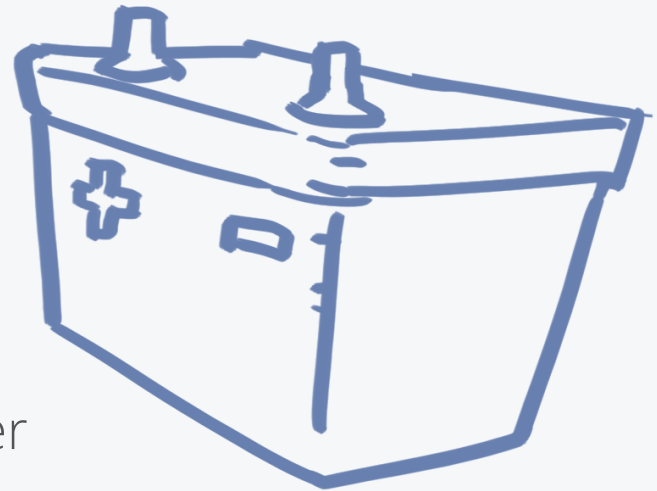
## *Dependencies*



# Dependencies

*Order is Important!* 😊

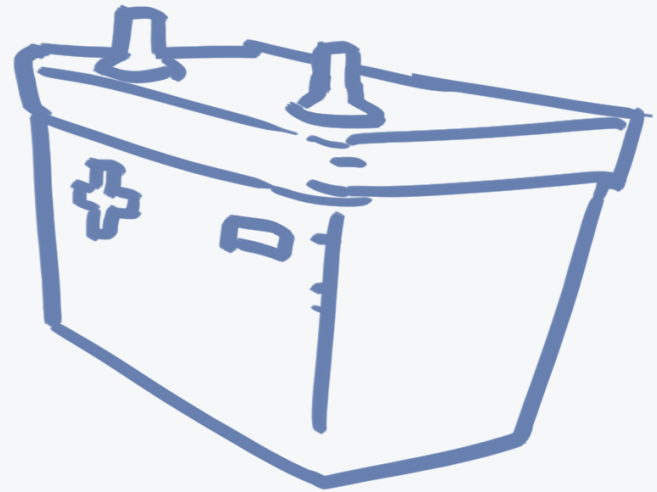
- Build / debug them in order
- Use stubs if necessary
  - Example: self-referencing
  - Docs: [drupal.org/node/1013506](https://drupal.org/node/1013506)
- Run migrations in dependency order
- Check for issues / errors in dependent migrations before proceeding



# Dependencies

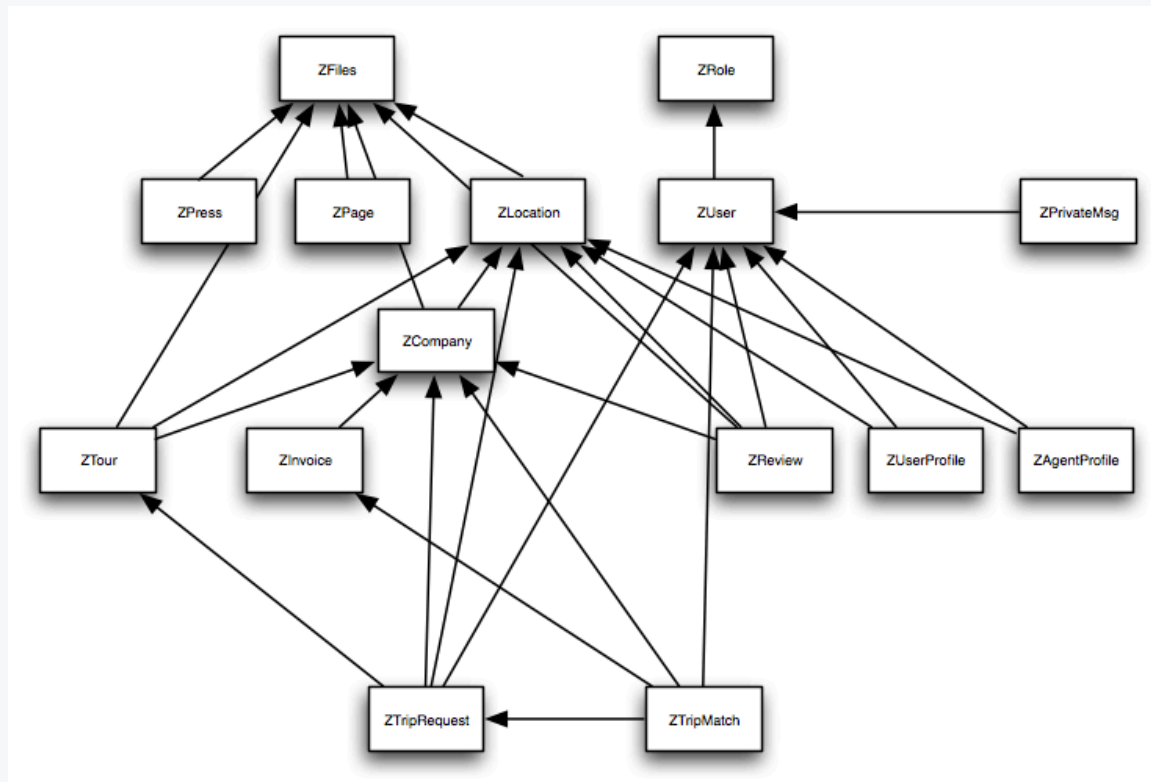
## *Simple Example*

- Roles
- Users
  - Depends on Roles
- Files
  - Depends on Users
- Profile
  - Depends on Users and Files
- Blog
  - Depends on Users and Files
- Team Member
  - Depends on Users and Blog



# Dependencies

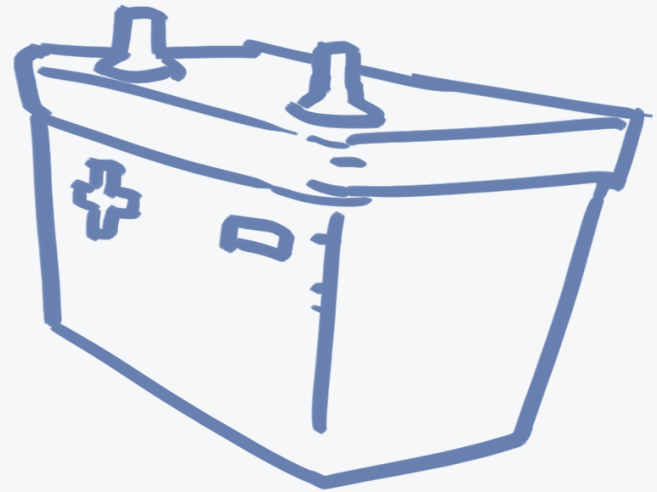
## *Real Life*



# Dependencies

## *Files*

- Transfer physical files prior to running migration
- Rsync or ftp script
- Can run into issues depending on hosting (e.g. timeouts)
- Watch out for errors in file transfers!
- Don't proceed with rest of migration if any files are missing





# Migration

## *Code*



# Code

## *Migration Class*

- Description
- Source
- Destination
- Map fields
- (optional) transform data as needed
- Register class in .info file



# Code

## *Import Execution*

- `__construct($arguments)`
- `preImport()`
- for each row:
  - retrieve source data
  - `prepareRow($row)`
  - create destination entity
  - `prepare($entity, $row)`
  - save destination entity
  - save map data
  - `complete($entity, $row)`
- `postImport()`



# Code

## \_\_construct(\$arguments)

- Constructor for specific migration (e.g. page, article, blog, user)
- Define dependencies, highwater mark, source, destination
- Add field mappings (mapped and unmapped)
- Docs: [drupal.org/node/1006984](https://drupal.org/node/1006984)



# Code

## *construct examples*

- Integer

```
$this->addFieldMapping('field_d5_nid','nid');
```

- Image meta data

```
$this->addFieldMapping('field_logo:alt',  
    'field_travelcompany_logo:alt');  
$this->addFieldMapping('field_logo:title',  
    'field_travelcompany_logo:title');
```

# Code

## *construct examples*

- Unmapped Source Field

```
$this->addFieldMapping(NULL,'field_meta')  
->issueGroup(t('DNM'));
```

- Unmapped Destination Field

```
$this->addFieldMapping('field_archive')  
->issueGroup(t('DNM'));
```

# Code

## *preImport()*

- Per migration
- Runs before migration starts
- Runs once for entire migration
- Docs: [drupal.org/node/1343112](https://drupal.org/node/1343112)



# Code

## *preImport examples*

- Disable rules from firing
- Disable email from sending
- Special logging
- Example: [drupal.org/node/1327812](https://drupal.org/node/1327812)





# Code

## *`prepareRow($row)`*

- Runs during migration per row
- Lets you modify or skip data
- Look at other row data to figure out how to modify data
- Docs: [drupal.org/node/1132582](https://drupal.org/node/1132582)



# Code

## *prepareRow examples*

- Use value in a source select list to set a checkbox value in destination
- Set workflow state
- Set weight
- Alter title text
- Save multiple value field into single
- Set fields from consolidated content type values



# Code

*`prepare($entity, $row)`*

- Runs during migration
- Called after all fields have been processed
- Use when entity changes are required
- Last chance to modify entity
- Handle fields that don't have handlers
- Docs: [drupal.org/node/1132582](https://drupal.org/node/1132582)



# Code

## *prepare examples*

- Cleanup node alias
- Map address field values
- Workbench moderation state manipulation
- Weight mapping
- Role manipulation
- Pre-fill missing translations with placeholder text



# Code

*complete(\$entity, \$row)*

- Counterpart to prepare()
- Runs right after destination entity is saved
- Updates based on new entity id
- Updating data related to saved entity
- Update relationships between data not mapped with migrate\_d2d
- Don't change/save this entity



# Code

## *complete examples*

- Special logging
- Organic Groups manager replaced User Role
- Build workflow history from source
- Flagging entity using source field data
- Workbench moderation states
- Tagging other entities based on saved entity



# Code

## *postImport()*

- Per migration
- Runs after migration ends
- Runs once for entire migration
- Docs: [drupal.org/node/1343112](https://drupal.org/node/1343112)



# Code

## *postImport examples*

- Re-enable rules
- Re-enable email
- Special logging
- Example: [drupal.org/node/1327812](https://drupal.org/node/1327812)





# Migration

## *Drush*



# Drush

*drush make-life-easier*

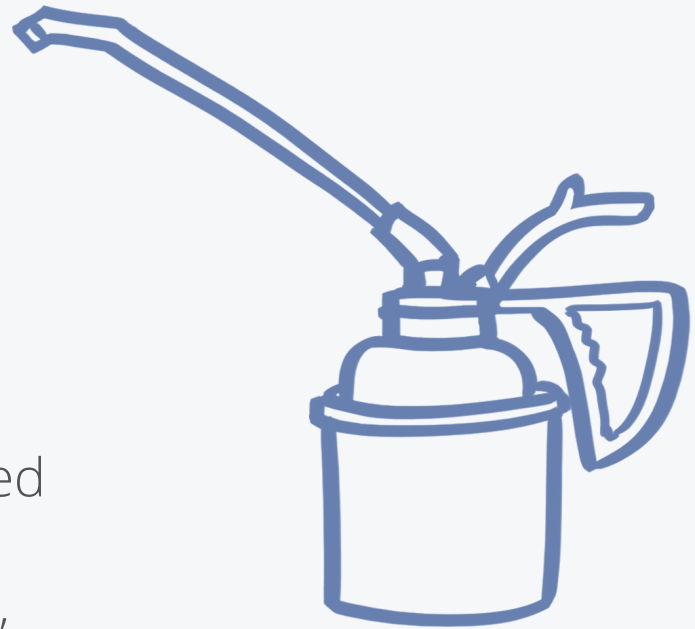
- Command line
- Don't do a migration without it
- `drush --filter=migrate`
- Docs: [drupal.org/node/1561820](https://drupal.org/node/1561820)



# Drush

## *drush migrate-status*

- drush ms [name]
- Name
- Total
- Number imported and unprocessed
- Status (idle, importing, rolling back, stopping, disabled)
- Last imported



# Drush

## *drush migrate-status*

```
[mac:kristen:migrate]$ dr ms
```

Group:	baseball	Total	Imported	Unprocessed	Status	Last imported
GL2000		2429	0	2429	Idle	
GL2001		2429	0	2429	Idle	
GL2002		2426	0	2426	Idle	
GL2003		2430	0	2430	Idle	
GL2004		2428	0	2428	Idle	
GL2005		2431	0	2431	Idle	
GL2006		2429	0	2429	Idle	
GL2007		2431	0	2431	Idle	
GL2008		2428	0	2428	Idle	
GL2009		2430	0	2430	Idle	

```
[mac:kristen:migrate]$
```

# Drush

## *drush migrate-import*

- `drush mi [args] [name]`
- `--user=1`
- `--feedback="10 items"`
- `--limit="100 items"`
- `--idlist=145,231`
  - source ids
- `--update`
  - import old and new
- `--needs-update`
  - `needs_update=1` in db and source db on different server



# Drush

## *drush migrate-import*

```
mysql> select * from migrate_map_zcompany limit 10;
```

sourceid1	destid1	needs_update	rollback_action	last_imported	hash
0	1583913	1	0	0	NULL
711	1583751	0	0	0	
831	1583557	1	0	0	
835	1583758	0	0	0	
836	1583502	1	0	0	
837	1583518	1	0	0	
838	1583763	0	0	0	
839	1583575	1	0	0	
840	1583524	1	0	0	
841	1583584	1	0	0	

```
10 rows in set (0.00 sec)
```

# Drush

## *More drush*

- drush migrate-rollback
- drush migrate-stop
- drush migrate-reset-status
- And more... [drupal.org/node/1561820](https://drupal.org/node/1561820)



# Migration

## *Takeaways!*

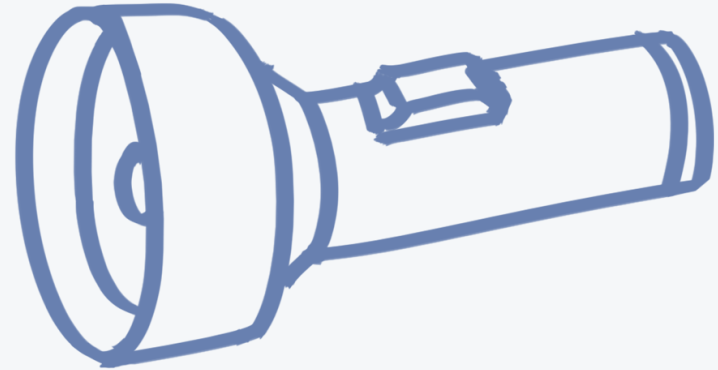




# Takeaways

## *Planning*

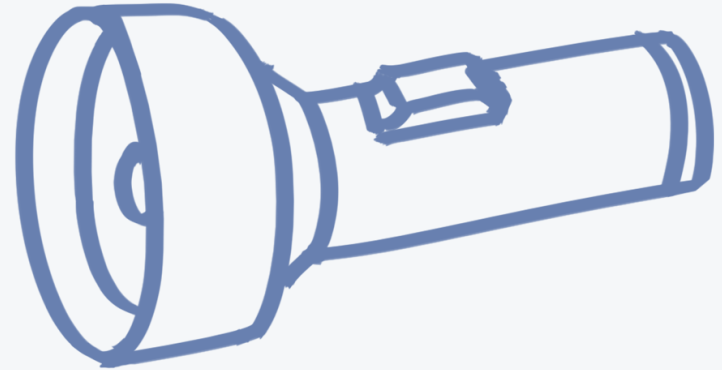
- Careful architecture, planning, and mapping
- Document before implementing
- Think three times about adding anything
- Think four times before naming anything



# Takeaways

## *Development*

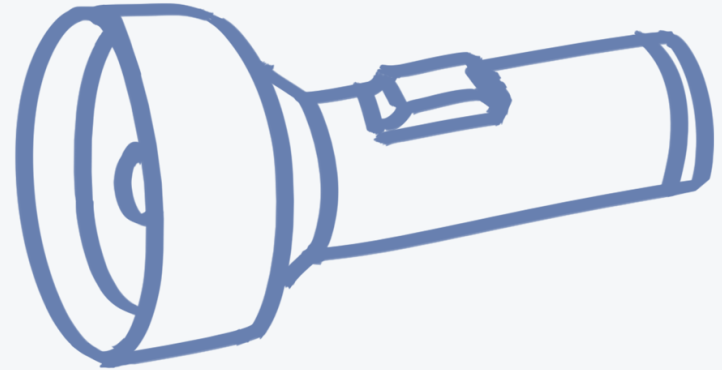
- TEST! TEST! TEST!
- TEST! TEST! TEST!
- TEST! TEST! TEST!



# Takeaways

## *Deployment*

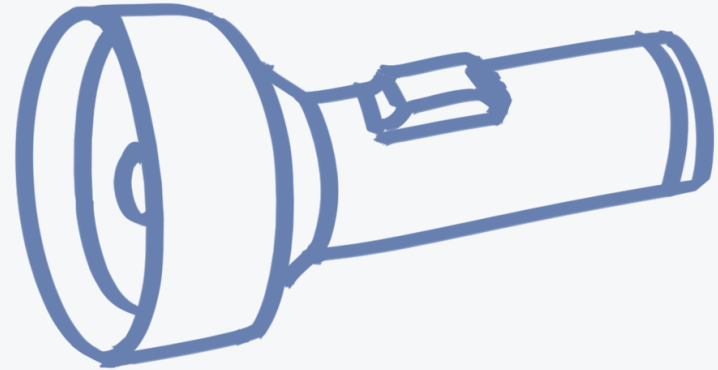
- Have checklist
- Rehearse
- Assume you'll have to rollback
- Rest up before go-live



# Takeaways

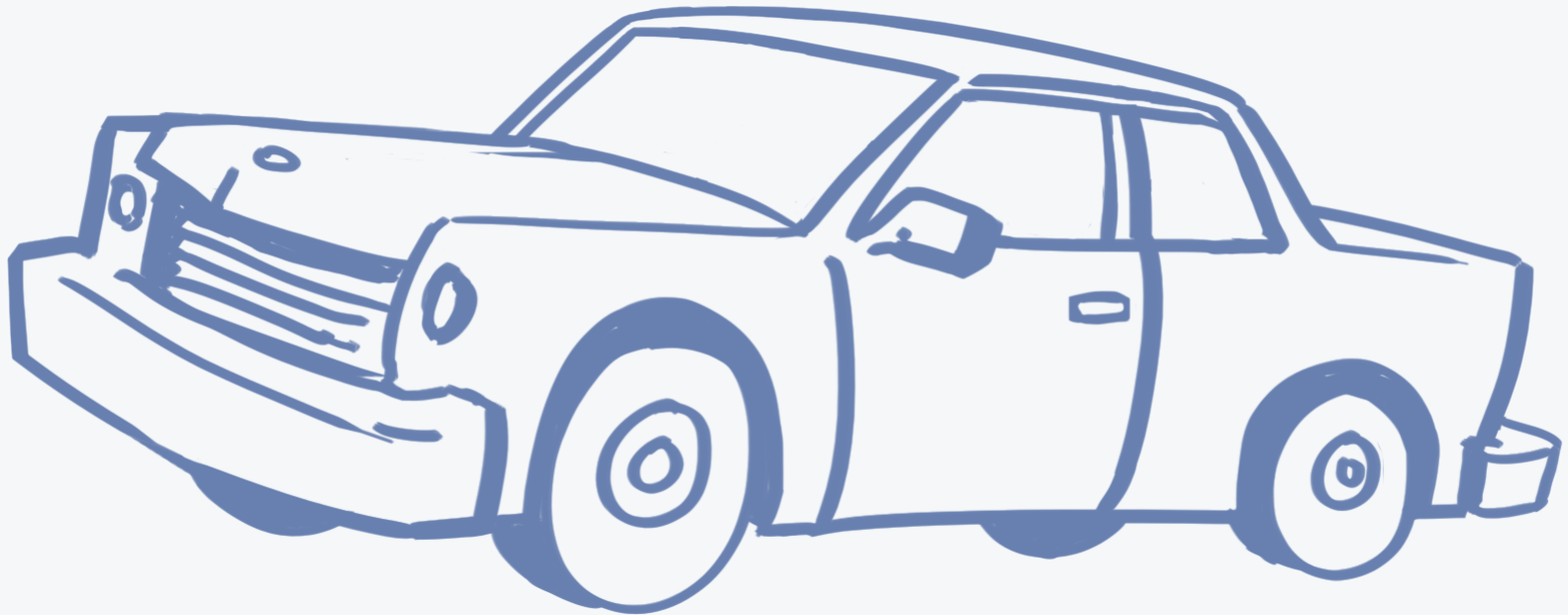
## *Post Go Live*

- Be ready for something that got missed
- Might need to create scripts to fill in missing data or manipulate existing data
- Do not delete your migrate tables or remove access do old database for awhile



# Migration

*Vroom! Vroom!*



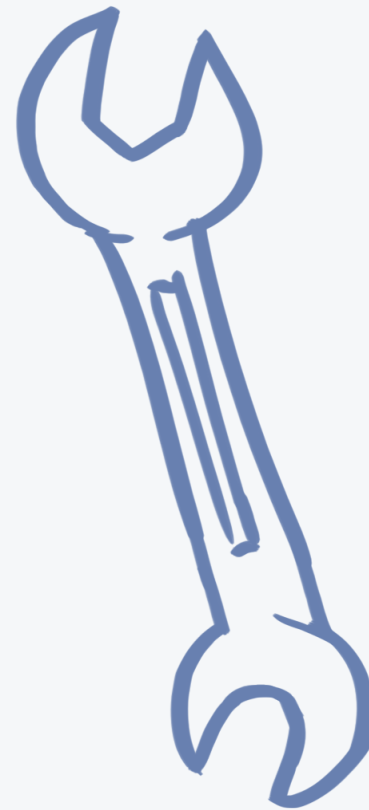
# Questions?



# Resources

## *Learn more*

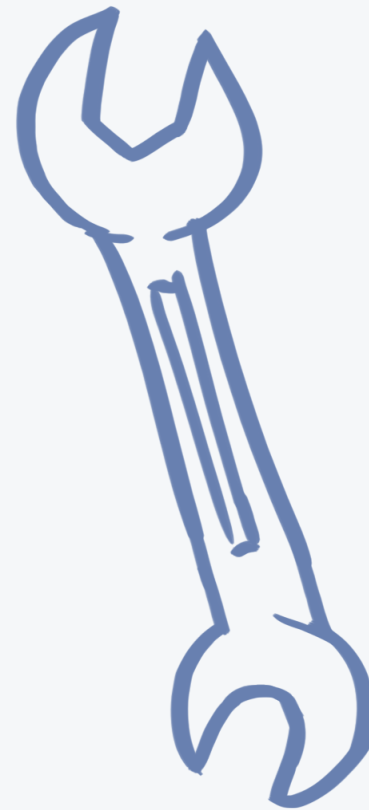
- Drupal.org migrate:  
[drupal.org/node/415260](https://drupal.org/node/415260)
- Drupal.org migrate\_d2d:  
[drupal.org/node/1813498](https://drupal.org/node/1813498)
- Ryan Weal:  
[verbosity.ca/working-drupals-migrate-module](https://verbosity.ca/working-drupals-migrate-module)
- Drupalize.me:  
[drupalize.me/series/importing-data-migrate-and-drupal-723](https://drupalize.me/series/importing-data-migrate-and-drupal-723)



# Resources

## *Learn more*

- John Bickar, Stanford:  
[techcommons.stanford.edu/  
topics/drupal/drupal-6-  
drupal-7-migration-worksheet](http://techcommons.stanford.edu/topics/drupal/drupal-6-drupal-7-migration-worksheet)
- Andrew Morton:  
[denver2012.drupal.org/  
program/sessions/getting-it-  
drupal-migrate](http://denver2012.drupal.org/program/sessions/getting-it-drupal-migrate)
- Ashok Modi:  
[slideshare.net/nyccamp/move-  
into-drupal-using-the-migrate-  
module](http://slideshare.net/nyccamp/move-into-drupal-using-the-migrate-module)





# D8 Resources

*Learn more*

- Core Initiative:  
[groups.drupal.org/imp](https://groups.drupal.org/imp)
- Ryan Weal:  
[verbosity.ca/migrating-multilingual-data-drupal-8](https://verbosity.ca/migrating-multilingual-data-drupal-8)
- Melissa Anderson:  
[dspeak.com/fldc14](https://dspeak.com/fldc14)

