Planning a real-life Drupal 8 migration

•Kafei Ineractive Inc.

•Ryan Weal (@ryan_weal)

•Novella Chiechi (@italiatina)

Do we really have to move?

- Frankly, you don’t have to migrate. But you probably should (copy and paste can take a *long* time).
- Here are your two options for migration:

  1) Keep the same architecture, or

  2) create a new structure.
- You’re likely going to go down this sophisticated route (migration) bc your needs are sophisticated.
- As Drupal 8 is currently in beta, it is easier to

What is currently supported:

- D6 -> D8 migrations,
- CSV -> D8 migrations,
- Incomplete: D7->D8,
- Incomplete: XML->D8.

Before you migrate

- Have a content stakeholder to review the migrations.
- This person should have at least a rudimentary knowledge of Drupal's working parts:
  - Content types
  - Nodes
  - Taxonomy

Enumerate the migrations

- Enumerate the migrations - how many content types, entitiy types, etc.
- Is the whole site being migrated, or just part of it?
- Create a spreadsheet mapping out old to new types/fields/etc. Get your client to review and approve the plan.

Have a bug tracker or spreadsheet

- Have a bug tracker or spreadsheet for content stakeholder to review post-migration
- Spreadsheets are ideal if the client doesn't want to learn yet another system.
- Grant access to your bug tracker if the client is very savvy and/or can be trained to comply with your processes.
- General spreadsheet tips:
- Organize your tabs by the NEW content types.
- Have an “overview” tab, where you keep a quick list of what is going into what.

The “Moving House” Analogy: Six Action Items for Your Next Migration

1) Mappings - how to do, how to review. Get approved.
   - Client wants to see what’s coming and what’s not. Tech wise this means spreadsheet.
2) Migrate all the simple things, content types and standard CCK fields, users, files, menus, easy things.
   - There are migrations that will bring across your content types as is.
3) Migrate complex things, write custom

Where will things go in the new house?

- Create the new content types/new architecture. Here you can decide how best to get your legacy data into customized structures.
  - Funnel multiple content types into one (one migration for each source)
  - Break out one legacy content type into multiple new content types (this can be especially helpful if you are improving admin interface or a portal website.

Moving van - get the right vehicle, and keep it tidy

- Consider your workflows carefully. All or part of your migration project may benefit from non-standard treatment (ie, not just using migrate). Keep an open mind when you are planning your migration.
  - Drupal Migrate
  - CSV, XML (soon!) imports
  - Copy & Paste

Unpack and put away

- Create a list of things for testing. Standard testing stuff also includes
  - tags in body text
  - Stubs
- When you ask people to test, make sure to include a representative node, so that people know what they're looking at/for. Otherwise, they won't do it. They really won't.
- Create a bug/issue master list spreadsheet (or

Housewarming! (And Goobye)

- Housewarming!
  - Take your time when showing clients how to navigate the new site.
  - Ask them to collaborate on help text - for more advanced/trusted users, have them modify the help text.
  - “Give” the Test site to the client to make changes on.

Tools for developers

- Drupal’s core migrate and migrate_drupal modules
- Contrib migrate_ui (all-in-one), migrate_plus (CSV)

Process Advice

- An important part of migrations are keeping things orderly, documenting your process, using clear and consistent naming conventions.
- This also means that you’ll need to keep conscious of legacy data (whether or not it comes with you, it should be purposeful), and try to structure the new site to be a good archive.

  – For example, are there content types that have been doing too many things? Are there things in the WYSIWYG that should be