EXPLAIN CASE

- DO
    - Show website
        - Map, shop view, location view
        - Migrate UI: locations already migrated
- EXPLAIN
    - Background
        - open data, where we get the info from
    - Migrate UI
- RESULT
    - Understanding background

PHASE 1 - INSPECT

- DO
    - Show source file
        - Columns
    - Show target: website
        - content type
        - taxonomy
- EXPLAIN
    - Investigate target
    - Investigate source
    - Clean up source
- RESULT
    - We have a clue what should go where and where to start

FASE 2 - GENERATE SKELETON

- DO
    - Code module
    - Code Migrate class
        - info
        - source
        - source fields
        - destination
        - mapping table

    - Demonstrate UI
        - Check detail => source, destination fields
- EXPLAIN
    - Parts of a migration
        - Subclass
        - Info
        - Source + fields
        - destination
        - Mappingtable
- RESULT
    - Show up in UI

FASE 3 - TRAIL RUN
- DO
    - Code a simple mapping (title)
    - Run via UI
    - Show shops overview page
    - Rollback via UI
- EXPLAIN
    - How to map
    - How to run a migration
        - Run, rollback...
- RESULT
    - A working run!

PHASE 4 - CLEAN UP x
- DO
    - Change title mapping to simple mapping
    - Mapping default fields
    - Set items do not map
    - Demonstrate migrate UI: the mappings and donotmap
- EXPLAIN
    - Simplemapping
    - Defaults

    - Dont map items => better use of ui
- RESULT
    - More items mapped
    - Clean ui

PHASE 5 - ITERATE OVER EACH FIELD - BODY + TAXONOMY
- DO
    - Code taxonomy & body field
    - Run import via command line
    - Demonstrate vocabulary
    - Demonstrate body: go to detail of an item
    - Rollback via command line
- EXPLAIN
    - FieldsMapping can have arguments for options
    - Different fields have different mappings
    - Taxonomy is a special field => will generate tax for us
    - We can import/rollback via drush
- RESULT
    - Step closer to end result
    - Data is imported in a drupal way (taxonomy)

PHASE 5 - ITERATE OVER EACH FIELD - NODE REF FIELD

- DO
    - Code field mapping: node reference
    - Run import from command line
    - Demonstrate detail page => see location info below
    - Rollback from command line
- EXPLAIN
    - We can handle relationships between
    - Since content is migrated, source table is used to join the items (mapping old identifier with new one)
- RESULT
    - Relationships between content appear

PHASE 5 - ITERATE OVER EACH FIELD - IMAGES

- DO
  - Code image mapping
  - Run from command line
  - Check image folder => they are downloaded
  - Check detail: image on page
- EXPLAIN
  - Image is specific field
  - Migrate takes care over downloading and stuff...
  - Different options
- RESULT
  - Images in site as drupal images
    - In image table

PHASE 5 - ITERATE OVER EACH FIELD - MULTIPLE LINKS

- DO
  - Code multiple links mapping
  - Run from command line
  - Open (edit) up detail page: see multiple links
- EXPLAIN
  - Migrate handles multiple fields
  - How to map fields for which no fieldhandler exist
  - Since we map them directly, some validation is skipped (WARNING)
- RESULT
  - Automatically made multiple
  - Closer to end result
  - We have real links with links module

C

PHASE 5 - ITERATE OVER EACH FIELD - GEO REFERENCE

- DO
  - Code geofield
  - Run from command line

    - Watch the homepage map populate
- EXPLAIN
    - We can do mad data alterations during import
    - Prepare
- RESULT
    - Finished
    - Data in website the drupal way

SUMMARY OF PHASES

- DO
    - Back to presentation
- EXPLAIN
    - Phases
    - What we did
- RESULT

CONCEPTS

a migration