ELAG 2016, Copenhagen

Data migration: feedback from an Open Source company
Who I am

➔ Paul Poulain, French (Marseille)
➔ Open source addict since 1997
➔ Involved in Koha since early 2002
➔ Former Koha Release Manager [v 2.0, 2.2, 3.8, 3.10]
➔ Member of Coral Steering Committee
➔ Founder of BibLibre, Open Source software for libraries

Paul.poulain@biblibre.com
Twitter: paul_poulain
(Not on facebook)
+33 6 14 38 05 56
Who I am

BibLibre want to help libraries deploying Open Source

- Migration, tuning, training,…
- Hosting (SaaS of all the software we support)
- Koha, Omeka, Coral, Bokeh, Piwik

BibLibre facts

- 1M€ yearly income
- 17 people
- 150+ customers, in 8 countries, 600+ libraries, 200+ data migration
Note:

➔ You’re librarians, I’m a support provider, I’ll highlight

➔ Things you should put on your tenders
➔ How the work should be splitted between you and us
So you want to

But ... you must exit with...

... your data
What can be migrated

- Bibliographic records (of course)
- Items (of course)
- Authorities
- Patrons
- Issues (current/history)
- Holds (current/history)
- Vendors & orders (current/history)
- Serials, printed (well, not really in fact) and electronic resources
Data migration

Steps

- Extracting
- Specifying
- Transforming
- Integrating
- Testing
“Easy” step

Pre-requisite

➔ Which format
  ➔ Iso2709
  ➔ XML ( ! schema)
  ➔ CSV ( ! vocabulary)

➔ Which encoding ?
  ➔ Unicode

➔ What if there is a mix

➔ Who extract your data from your old software?
Specifying

What must be specified

➔ Format description
  ➔ including codes and their meaning
➔ Old software <> New software equivalences
➔ Links between files, where applicable
➔ Transformations to do
➔ Expected results
➔ Test plan
Specifying

RFP

➔ In your RFP, write as many things as possible about your expectations & requirements

➔ Examples:

➔ Great: “the library want to migrate the bibliographic records and the items. 10% of the record are using a wrong encoding. The subject fields will have to be retrieved from the LoC and appended.

➔ Acceptable: “the library want to migrate the bib. records and items. Some transformations will have to be done during the transition to the new system”

➔ Wrong: “everything must be migrated”
Specifying

Who specify

- Best: together
  - You know your data better
  - Your vendor knows the new system better
- Good: you
- Not good: your vendor
Transforming

Sometimes not needed, but rarely

Transformations required by the new ILS

- Very frequent for items (for example: locations, call number, holds/issue status)
- Who does it?
  - Good: your vendor
  - Not good: you
Transforming

Transformations required by your data

➔ Cleaning old & irrelevant data
➔ Merging and de-duplicating
➔ Who does it?

➔ If you expect your vendor to do it, write it in your RFP!

Transformation to enrich your data

➔ Merge
➔ De-duplication
➔ (Linked data)
Transforming

Tools

- Marcedit
- MARC::Transform to Transform Marc records
- Spreadsheet
- Openrefine
- MARC::Loader to create MARC Records
- Catmandu

Repeatability of the process.
Integrating

Made by your vendor (except if you’re going Open Source ;) )

If you do it yourself
   ➔ Use official APIs (if you do it yourself)
   ➔ Who endorses errors?
**Testing**

Underestimated in most cases

Who write the test plan

- Better: together
- Good: you
- Not good: Your vendor

Who execute the test plan

- Better: Your vendor, then you
- Good: You
- Not good: Your vendor

Migration logs are very important
Building the test plan

- Volumes expected
- Use case
- Comparison between old & new system

Caveats

- You must “think new software”, but you still “think previous software”
- You can’t test everything, select!
Reporting problem

➔ Give identifiers
➔ Give URL
➔ Give screenshots
➔ Example

➔ “if I search, it does not work” is not a usable report!
➔ “If I search for blabla, I expect to get 100 results, I get only 20” is a usable report
➔ If I search for blabla, I expect to get 100 results, I get only 20, the following are missing: x, y, z” is a good report
HINT:
➔ Keep old identifier somewhere in your new system, if may be useful (and remember Murphy’s law)
➔ It’s a logical process, check there are no hole
  ➔ “if this contains “B” do that, if this contains “A” do that”. What if not A or B ? (murphy’s law again !!!)
➔ Move field 600 in 601 if 600$a start by ’A’. What if there is already a 601 ?
Iterating

Why iterating?

➔ Nobody’s perfect
➔ Cost of mistake discovery
  ➔ 1 at spec. step
  ➔ 10 at test step
  ➔ 100 immediately after going live
  ➔ 1000 far after going live
➔ Make the process as automated as possible
  ➔ The bug is usually between the chair and the keyboard

For a small library, 2-3, for a large 3-5
Final migration

Go live
References

Main websites:

➔ MARC::Transform: http://search.cpan.org/perldoc?MARC%3A%3ATransform
➔ MARC::Loader: http://search.cpan.org/perldoc?MARC%3A%3ALoader
➔ Openrefine: http://openrefine.org/
➔ Marcedit: http://marcedit.reeset.net/
➔ Catmandu: http://librecat.org/
THANK YOU!