

# Didact overview planning

13 NOV 2020

1

## parallel use cases

- Notebook w/ file org
- Repeating previous related work
  - soft linking to prior entries
  - facilitates narrative in Data org

causal tree / provenance/  
pedigree graph  
of Data

- formalizing patterns of work into workflow w/ explicit names in 10 structures, docs, automation etc
- using workflow to outsource exec to employee / 3rd party, (scaling), while auto-collecting results

- publishing/sharing workflows
- finishing is remixing workflows

Ultimately, a library of workflows + types  
+ narrative that allows discovery of  
solutions based on desired shape of I/O  
not knowing names of tools or  
terms of art

- Data model supports revisioning, forking workflow

- Tech  
foundations
- changing "program"
  - and/or
  - changing data
  - offline first

## Primary Sections

- 1-2 Pg Exec Summary
- Storyboards + paragraphs showing use cases
- Big google doc describing architecture & goals in design
- Market + biz Models
- Talk / Video presentation

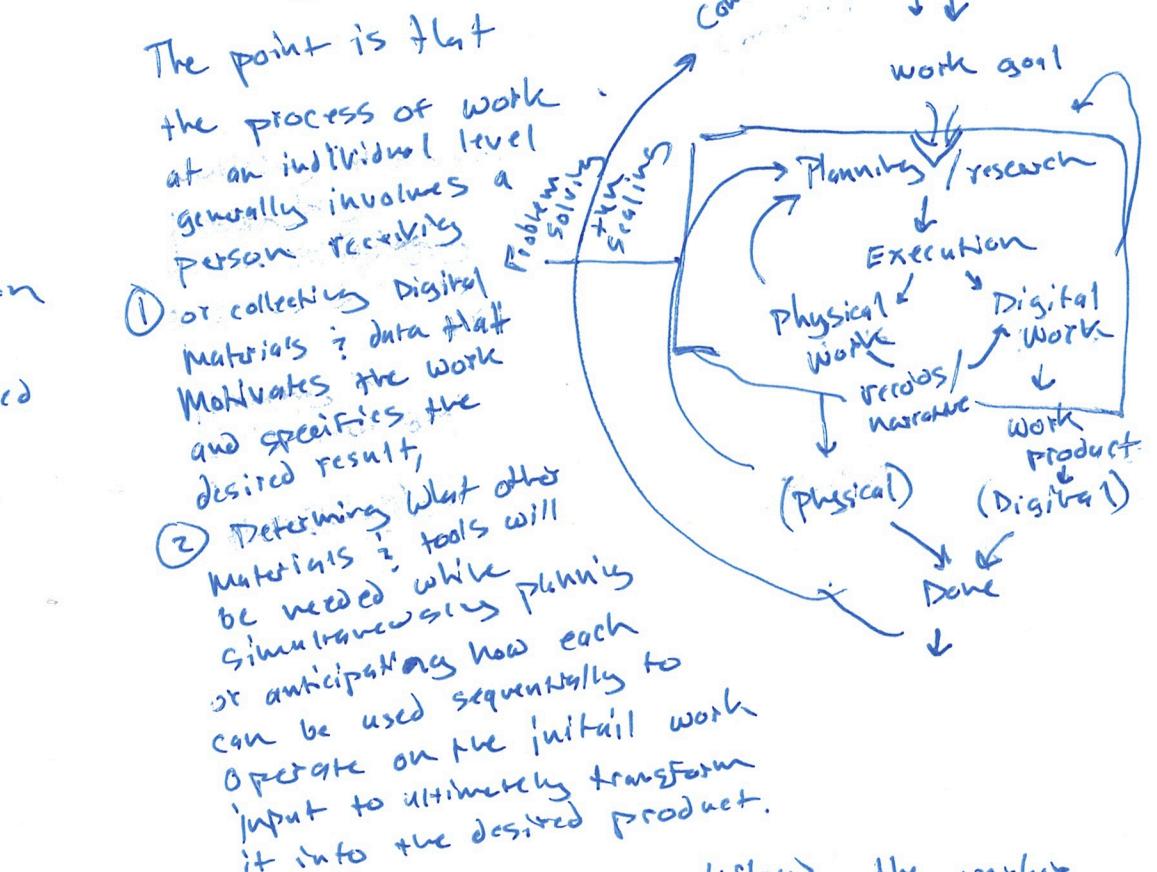
① No uniform container for organizing work notes, procedures, and digital results of completing the work at personal scale. It's ad-hoc. Ours too, except when cost of spending manpower on standardization efforts / Benefits of reliability, quality, logistics, make it worth it.

<sup>the way</sup> We manage digital knowledge; work today is like how work knowledge was managed in the early middle ages, when the highest form of info tech was The Book - a concentration & externalization of knowledge, each produced ad-hoc, by hand... until the printing press provided the technology to standardize & scale the production & consumption of Books.

<sup>Toolbox</sup> Propat Today there are thousands of useful knowledge, team, database, workflow, document etc management tools, each useful, as the old bespoke Books were, some powerful & esoteric, others deeply simple - offering pragmatic affordances that solve a single point of friction in how digital work is done well.

Many Many Many Wonderful tools, & systems that create value independent of one another. Some subset could be condensed into a basic set of inventions that when integrated provide far more

Value than the sum of its parts. 13 NOV 2020  
② Here is a schematic of how value is usually created by an individual worker in the modern knowledge economy!



③ Examples cap/cam/fab is likely to produce digital artifacts, incidentally or directly; by definition if they are working within inventory/orders a purely digital toolchain, or because they use digital tools to record, organize, inspect/validation saves, versions scans, configurations plan, schedule, document, & manage their physical work, if 1 step removed, or b/c those physical tools are themselves digital machines.

To put it simply, more often than not work is conducted with digital tools, on digital materials, to produce more digital stuff. When the work involves physical stuff, digital artefacts are nonetheless consumed and produced along the way - digital proofs going to printer, biz docs, receipts, reports, logs, purchase orders, photographs, forms, the list is endless... infinite varieties of forms (structure) but all of the same material ... information... Digital Information.

- good old fashioned files,
  - Structured information  
(Data, DB, tables)
  - = Words, narratives, reports

## Archetypal Examples

# Q Bookstore inventory

The act of keeping records  
is first ingredient to  
documenting a repeatable  
process, although that may not  
be initial purpose nor is  
the record-keeping necessarily  
repeatable or consistent itself.

\* By a human in a  
surprisingly before  
hands on way

(11) (cont.) Note fairly's  
record keeping is first step  
to repeatability but repeatability  
is not what motivates the  
record-keeping, but it  
could be if the record-  
keeping tools were designed  
to facilitate it, to suggest  
it, nudge towards it by virtue  
of their affordances?  
tools that structure the  
info? Knowledge so that next  
time it's a little easier to  
~~improve upon it~~ by reuse  
Have ^ it

It's all just structure,  
that's the nature of digital  
encoding... but I'm trying  
to get at conceptual categories  
at different levels of abstraction?  
grammaticality

*concept* Create common structure  
table for organizing/editing  
each example

Describe reusable work  
process - how would you  
do it conceptually

1.3 How to implement it w/  
google Docs / Dropbox / Evernote

1.4 How it would look in didatt  
↳ files look like files  
are also + linked deep in  
notes

13 Nov 2020

(4)

2.0 got process figured out -  
now we want the employees to do  
it.

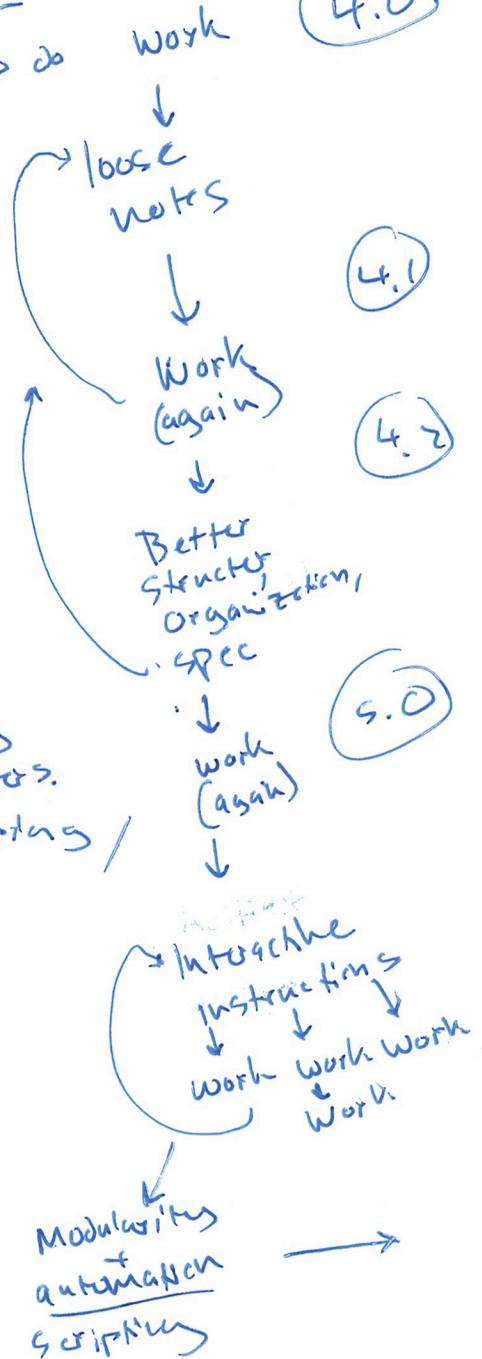
2.1 How does that work w/  
gdocs / Druplx / Evolve

2.2 How bout Didact?  
Implicitly defining 3 scaffolding  
goal requirements is free

3.0 lot of manual Data processing  
- waste of human life. Computers.  
How does some script prototyping /  
playtesting work...

3.1 in existing tools

3.2 Didact



4.0 Creating summary reports,  
Discover Discrepancy work,  
Review records to resolve

4.1

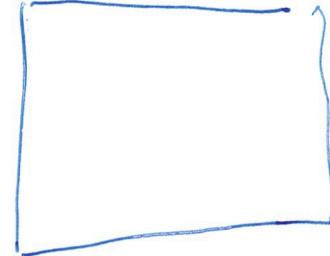
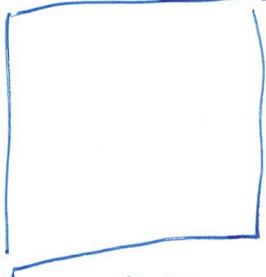
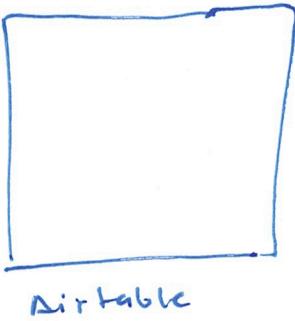
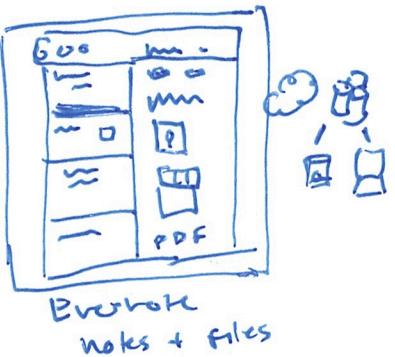
4.2

Workflow encodes data provenance,  
easy to instantly get causal  
records

sharing  
Perhaps PhD discovers library  
api script from 3.0 is rewrites  
it

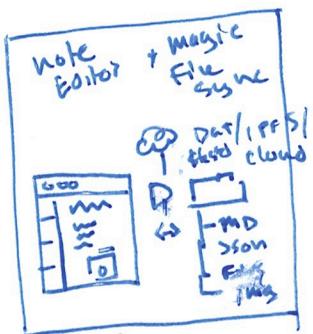
or

go deeper on affordances & design  
for "planning - prototyping mode"

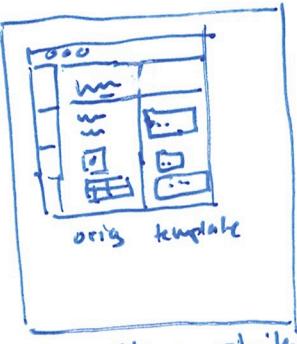


"work"	
Goal	Notes
Plan	notes?
inputs	"files"
Exec	tools
Result	"files"

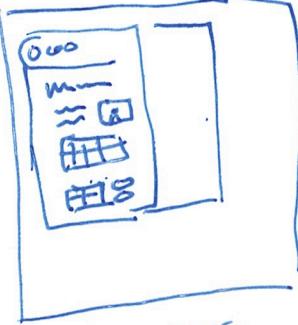
Most work processes  
create succession of  
files + data



Better  
Notetaking  
File Management  
than EN or DB alone



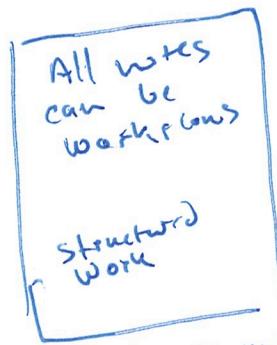
Repeating similar  
work is easier



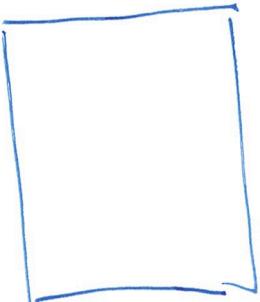
where Does  
Data live?  
Many places  
1. in APP/DB  
Runtime env  
2. Folders  
Plain files

NB supports data  
joins into composites  
while keeping components  
as atomic files in  
Folders

- like Airtable



Workflow is made  
by connecting blocks (steps)  
(unique feature)



### Need:

#### Architecture Diagram

- correspondence b/w NB Features + Folder structure
- NB note, workflow
- Data store
- workflow logic
- partitions
- terms

#### UI closers

- structured Data/File storage
- workflow creation
- workflow reuse
- Excel/func aggregation

# Reviewed project Notes - key ideas

16 Nov 2020

## NB Projects:

- Notes w/ linked files
- facilitate schema/def creation to org / extract files/data into collections or DB
- syncs to flat folder on disk
- Everything is revisioned ↳ can be rewound/ff/forked/merged
- Projects organize notes ↳ workflows
- Notes can have programmable behavior (ops/code) that operate on Data
- code editor ↳ playground env for experimenting
- if note's data deps change, op is re-run
- triggered by file change on FS or via App
- Data model built w/ event sourcing (append only log) + causal tree CRDT model of notebook workflow
- dweb protocol: hyperspace, IPFS, teardrop?

- files on disk are effectively Materialized views of slices of Event log
- ⑥
- NB knows how to parse file change on disk into correct events
- ultimate source of Truth for all content is Merkle-DAG Akin to IPFS unix-fs Data Model
- But also, this can be compacted serialized into snapshot ↳ as single JSON file + FS or optionally multiple connected files

Project.json

```

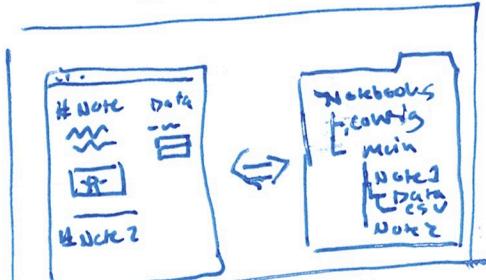
  |- Block.json
  |- Block.md
  |- log.json
  |- Defs.json
  |- collections.json
  |- views.json
  |- Flow.json
  |- deps.json
  |- Blobs (attachments)
  }
```

But all of these files are really "sust" projections from slice of NB Event log (maybe Head, Maybe tail)

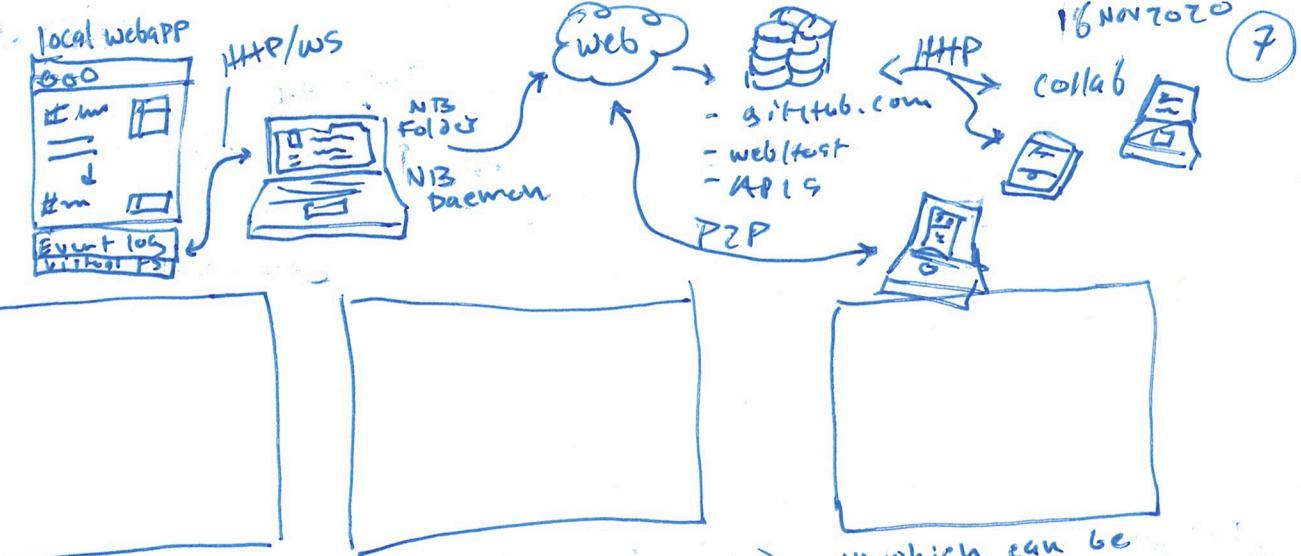
↳ so project folder structure on disk can be automatically converted to diff FS trees

NB API provides API over Event log that automatically indexes data entities into collections / DB & indexes → can trigger construction of hosted DB as provider or view cache

OFFLINE first  
revisioned, fastalive  
distributed (P2P)  
extensible real-time  
Data structure +  
Application (SPA)



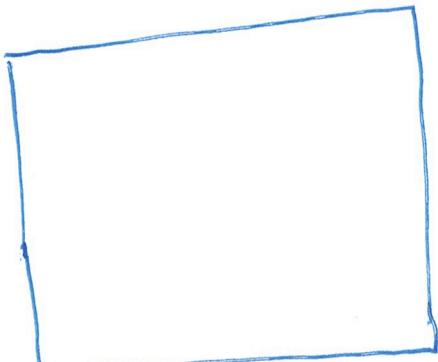
Notebooks are stored in a simple format on disk to maximize the flexibility & control of your data



FILES ? Data can be easily referenced ? reused in multiple notes while staying organized on disk

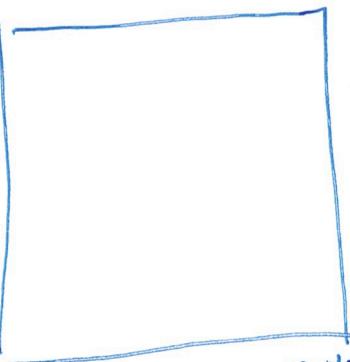
Or extracted, transformed, and combined to create composite data bases in the notebook ...

... which can be used to automatically generate new files or update a remote API.

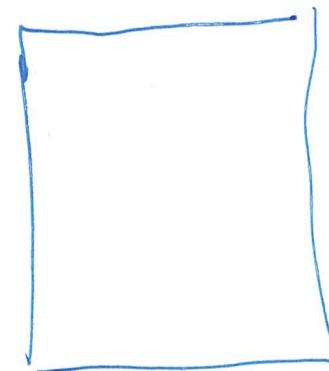


This combination of roles + data collection + composition + output is called a workflow.

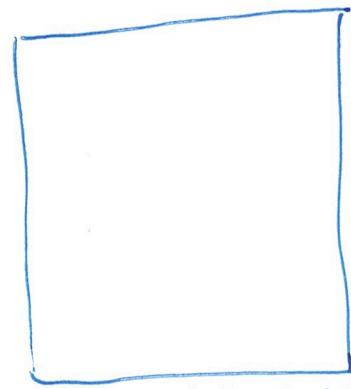
Workflows are optional and are just NB notes that have been linked together to accomplish these steps.



You can use a workflow to speed up and standardize tasks that you may repeat...



Or you can send it to an employee or colleague and have them do the work. It will automatically be synced to the notebook, organized (and optionally validated) just the way you specified.



The Notebook engine analyzes your workflows & data models - and those published online - to make intelligent suggestions and autocompletions that help you reuse existing data structures and workflow steps.

When you are creating notes to document your work, you can use the Data editor to define the scope and structure of your goal at any time. This goal-oriented planning can help you add the notebook both simultaneously backwards (from the end) and from the beginning (could be worded better), as you prototype data operators & workflows,

Notes can contain embedded operations, code, API calls, and even entire workflows, all of which can be used to process your data.

But many tasks can only be done by a person in the real world. In these cases, steps in a workflow can include any kind of file or reference instructions, as well as definitions of what data (checklist? image? CSV?) is expected to validate the task succeeded (or didn't).

In this way you can take notes on the fly, and link them together into a reusable workflow that is purely human-driven if you find yourself doing the same task again. When it's worth it, programmatic steps can be converted into automated operations & programs for the notebook to do for you.

- real-time collaboration, P2P
- as a hosted web app on the internet
- or even as a single static HTML file you can email. It will self-edit when used and can be emailed back, or synchronized over P2P protocol.

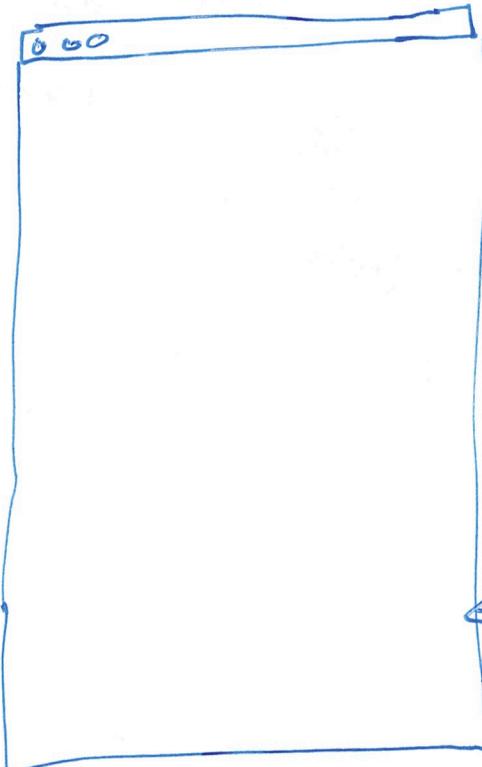
\* almost - it saves a fork of itself w/ updated data.

Because NB is built w/ distributed web technology, a workflow or even whole notebook can be shared.

NB can infer schemas for your data from their names, content and operations applied to them. You can add metadata & validation logic to a schema's definition if you like, too - but all this extra clutter is always optional.

# Architecture Draft Diagram

local  
Desktop / Browser App



CRAP  
↓

local filesystem

↳ Notebooks ↳  
archive/  
main /  
notebook.json

notes/  
2020-01-01-title-uuid/  
2020-01-01-title-uuid.json  
2020-01-10-title-uuid-MD

files/  
profile.pic.jpg  
CSV.json  
outputs/ENTS/  
EXCS/

collections

plugins/  
core  
NPM/  
definitions/  
workflows/

access-control.json  
↳ Shared views  
snapshots

Handles (binding)  
Definitions  
Flow (steps/ops)  
Parent state  
DEPS

YAML

```
wild:
DEPS:
"scans": spec
ins:
name: scans
type: file/CSV
outs:
ISBNS:
- ISBN10
- date
---
```

# Markdown Content

(!@/ref/short)

UI:

Edit note DOC (md)

Creates CT Events

- Serialized into Note?

= No, kept in sep files

- but minimal def + UUID links

Serialized into Note

- sort or back-links

- facilitate file-based

collab

- or as separate snapshot files

Filesystem "UI" ignores

updates to "protected" fields

in files

Editing content ok

UI

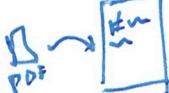
Data Model

remote

Disk

- note UUID
- content (MD)
- ...

ADD file



PDF

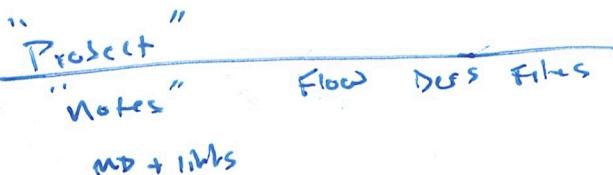
Data Mappor

file UI

Each file is actually a  
SLICE OF CT Events  
Projected at particular time

walkup  
Mon Jan 11

11 am



- Empty Notebook
- add file
- Data Mapper / ETL
  - type def
  - collection API
  - preferences
- composite Data

- Project UI
- goal driven  
then iterative
- workflow creation
- workflow sharing
  - hosted SPA
  - Data collection