UX ARCHITECTURE FOR DATA COLLABORATION

Stiby Systems Case Presentation

Henrik Korsgaard

June 19, 2023

Outline and focus

- ► UX research process
- ► "Results" and assumptions
- ► Information design for shared views
- ► Interaction design for collaborative features

As a newly hired UX architect, your initial task is to create an outline for the UX work in a project aimed at improving the UX of the **collaboration tooling** in an existing online Excel-like table system.

…assume that you have the necessary budget for it.

UX Research

Discovery: How, when and why do users collaborate?

Define: What are the main user scenarios, information concepts and features?

Prototype: User flows, wireframes, key interfaces and features

Evaluate: User testing and evaluation

Integrate: Plan integration and delivery

Discover > Define > Prototype > Evaluate > Integrate
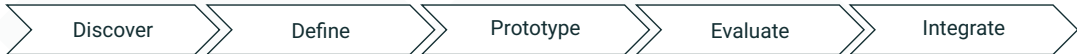
UX Research

Discover

- ► Internal research and analytics
- ► Contextual interviews with users
- ► Observe collaborative task sessions
- ► State-of-art on collaborative applications (CRDT)

Define

- ► Workshops!
- ► Collaborative task objectives
- ► Scenarios, personas and user journeys (user stories)
- ► Information concepts and architecture
- ► UX quality criteria and KPIs

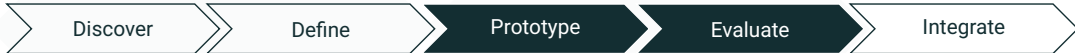| Discover | Define | Prototype | Evaluate | Integrate |

UX Research

Prototype

- ► User and collaborative flow
- ► Information architecture and layout
- ► Data and view operations
- ► Real-time collaboration

Evaluate

- ► Internal review and testing
- ► (informal) user feedback
- ► Think aloud evaluation
- ► Review UX quality criteria and KPIs

| Discover | Define | Prototype | Evaluate | Integrate |

Collaborative data scenarios[1]

Collaborative projects

- ► Peers collaborate on a larger project
- ► Different responsibilities and expertise
- ► Mixed focus with a high degree of coordination
- ► Multiple data views

Real-time collaboration

- ► Peers collaborate on smaller (urgent) tasks
- ► Real-time collaboration with shared focus
- ► Few data views

Training

- ► Expert user provide training or help to one or more trainees, e.g. onboarding
- ► Focused on learning the application and/or data
- ► Tailored data views and exercises

[1] Larsen-Ledet, Ida, and Henrik Korsgaard. Territorial functioning in collaborative writing: fragmented exchanges and common outcomes. Computer Supported Cooperative Work (CSCW) 28 (2019): 391-433.
Larsen-Ledet, Ida, Henrik Korsgaard, and Susanne Bødker. Collaborative writing across multiple artifact ecologies. Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems. 2020.

UX qualities

- ► Sharing with collaborators should be easy and include task assignment and notes

- ► Important to know who did what in a shared data view (awareness, track changes, accountability etc.)

- ► Support sandbox experimentation and analyses before publishing or merging to master

- ► Collaborative features should not overshadow existing task features

Sharing in collaborative applications

- ► Work object is shared by replication (content and formatting)
- ► Communication is transient (chat)
- ► Tools are individual, but similar across users (UI)
- ► Environment is not shared (browser/extensions)
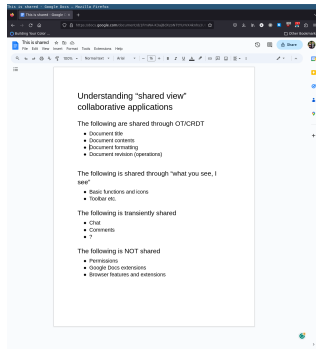
Interaction Design features

- ► The user can save changes as individual **views** (sheets) of data
- ► **The user can share their saved views with other users**
  - → The user can add or remove columns from the **view**
  - → Users can filter and order the table content
- ► **Multiple users must be able to work on the same views simultaneously**
  - → The users of the system may be located on multiple locations

Making 'views' first class objects

- ▶ Data views as the main work object – it's what is shared when collaborating
- ▶ Views can be published in formats fitting the consumer needs
- ▶ A data view encapsulate a data source, users, and the revision history

Creating a new data view

1. Select data source(s) (or import)
2. Preview data
3. Add collaborators
4. Generate view
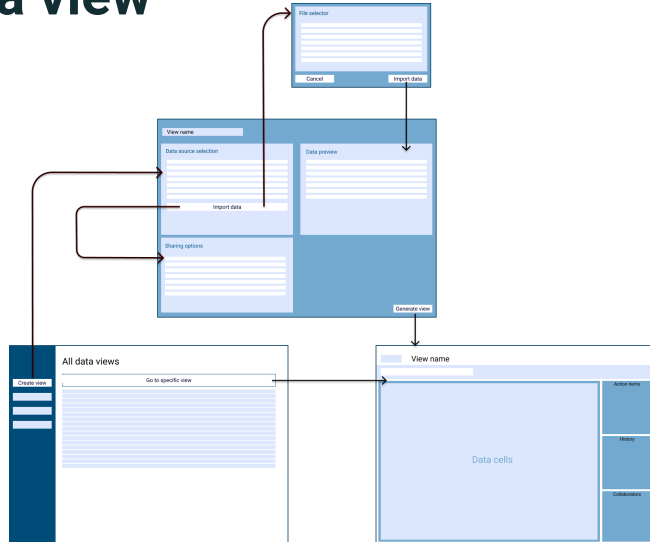
Manipulating shared view

- ► Data operations as the replicated objects:
  - ► Data manipulation
  - ► View manipulation
- ► This ensures locatability from history into the changes
- ► …and accountability (who did what – where are we)
- ► Rolling back changes (undo/hide operation)
- ► A set of operations (a macro) can be exported to other data views
- ► (and reviewed before applied to master data)
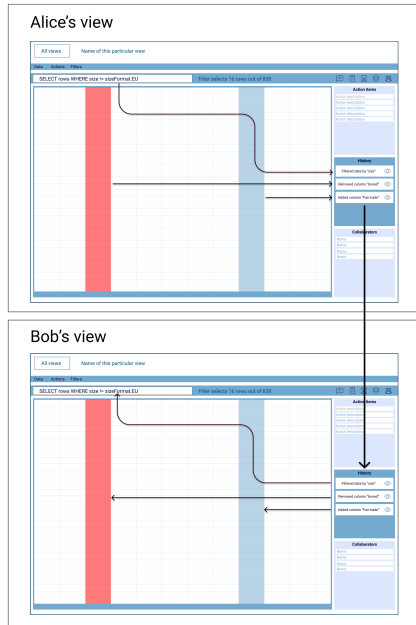
Collaborating with data views: Assign action item
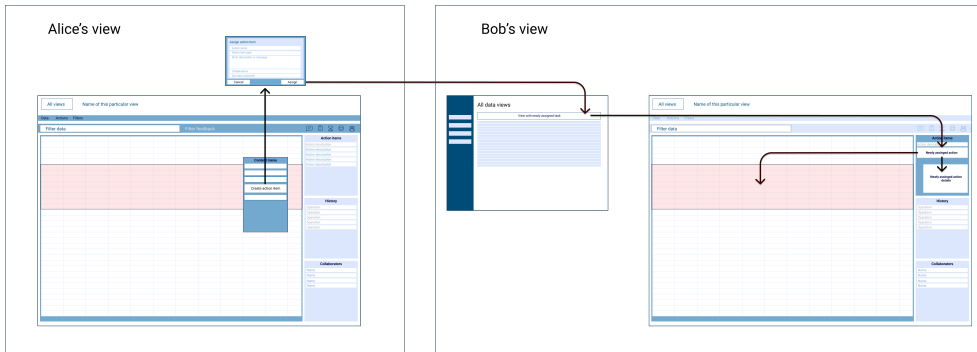
Key IA/IxD challenges

- ► 'Views' might be a difficult concept to grasph for non-data users
- ► Revision history is a difficult concept to get right and powerful for non-programmers
- ► Remote collaboration **require** additional communication channels
- ► What happens if we add automatisation, quality tests etc. that have delay?
- ► Prototyping (CRDT) collaborative tools pose different requirements than single user applications:
  - ► Require more contextual user research and co-design activities
  - ► Hard to study from application analytics (but I have done that)
  - ► Require some technical infrastructure to prototype collaboration (websockets will get you 80% there)