

Team dependency onboarding.

The user doesn't fully see how they can use this. There is a lot of information and it is overwhelmed. I should consider adding a few sentences or quotes or so that helps the user understand how they can use this.

Save button.

Users would like a save button somewhere. So they can come back later to improve it.

Divide Team Dependencies into Departments and Products.

Users would like to build the organization from the top down. Starting with the user first and then moving to name/frame a product, and then moving to the teams. Ultimately this would show value stream and dependencies within and between. And possibly cost per product.

"Feature" business case.

Users would like to do the ROI Projection/Find breakeven point (that I want to rename to Build Business Case) but for a specific feature. We could possibly divide the "build business case" divider into two parts. Platform (which is what is there today), and then "Feature". For example, they might specify how much time it takes to build the feature, and what the impact is of that (time saved, for how many, what does that time cost) and thus they can build a feature business case.

Suggest whether Outsourcing or Platform automation is better

When a user selects outsourcing, there's a chance that automation is a better deal long term. So it would be nice with some form of button which the user can click on "Compare this with a platform". And if they click it, they can see the platform budget part. Essentially "If you can solve this problem for X money, it is better to build a platform than to outsource it".

Commenting.

Users would like to comment a business case, or budget, or dependency visualization. For example, someone enters data. And then shares it with their coworker. That coworker can comment sliders or fields. Similar to how commenting works in google docs. I am aware this will require refactoring and db-storage.

Change-log.

Users would like to be able to save their analysis of their current state, and their future state. They also want to be able to add milestones showing what improvements they have implemented. For example, after a few months, a user makes a new entry of what their organization looks like from a dependency point of view. And the application makes it possible to see the users progress towards their end goal, like a milestone.

Current state -> Target

User adds entry

Current state becomes Starting State -> New entry becomes Current state -> Target state remains

And the graph should update to show this.

Zooming in and Out.

Users would like to define their organization and to be able to zoom in and out. For example, a user creates their org under team dependency, specifies the team, and can add information for each team (like the Build business case for each team)