

ASSIGNMENT 3:

DEFINING DEVELOPMENT SLICE

THE OBJECTIVE

The goal of this assignment is to learn how to split features to smaller increments.

THE ASSIGNMENT

The assignment continues from the Assignment 2, where you defined a feature based on a customer need, using job-to-be-done -framework. Part of the assignment was to validate the need and plan the validation of the suggested feature.

At some point you can't validate the feature further without building a part of it. However, building the whole feature might be too expensive and risky, and therefore, finding the smallest useful slice of functionality is a good practice to structure the development.

Take the feature you specified in Assignment 2 and define the first slice(s) by considering (at least) the questions below. The result should be a description of the slice of the feature, preferably with UI mockups to help you explain the difference between the full feature and the first slice.

THE TASK IN STEPS

- 1. Evaluate your proposed feature (from A2) and refine it, if needed.** Does your suggestion for the new feature in Assignment 2 define a functionality? If yes, use that as the basis of this assignment. If not, sharpen the definition. If you want to use some other feature for which you have sufficiently well-defined groundwork done, you can use that.
In any case, define the feature in your reporting.
- 2. Form a light sketch of an architecture for the feature¹.** What new functionalities and elements the feature requires, and what changes and additions this implies to the product²?
Consider this on the level that matches your know-how in terms of software development. If you are not familiar with the field, feel free to use "black box" -architecture³, that is, just describe the what new functionalities the implementation of the feature needs from the system.

¹ Another wording for the same thing: What new elements the current solution would need if the feature is implemented? Form a blueprint/wireframe/sketch that illustrates how the existing system (i.e. the current solution, e.g. software) can be augmented and with what components.

² "Architecture" as a term comes from software development, but it is essentially a specific case of a more universal idea. All technological solutions can be (presumably) thought as some sort of a system, regardless of whether they are software, hardware, or non-tangible social routines. "Architecture" of such a system usually describes the elements that the systems consist of and their relationships. Here the point is to show what new elements are needed to the system.

³ The level of description should be chosen according to the accuracy of conclusions and understandability of the argument. If an electric scooter needs better ways for stopping, "add a new break to rear wheel" is probably sufficient level. If the scooter should also use the breaking energy to load the battery, then "add a regenerative breaking-system to rear wheel and change the controller for battery charger to one that can use the breaking energy", would more likely lead the audience to understand how this change would affect the overall scooter-system.

3. **Analyse the development path for the feature.** Consider the following questions and assess what are the slices (the smallest useful increments) that you could and should implement when developing the feature:

- What should you implement to eliminate the biggest unknowns?
- What can be left out from a minimal version?
- Is there a "20% implementation" that gets you 80% of the value?
- If you could ignore all error scenarios, what would be the simplest happy path?
- Can you focus only on one persona?

4. **Report the results.** The result should be a description of the first slice(s) of the feature, preferably with UI mockups and associated development path that can help you explain the difference between the full feature and the first slice.

5. **Return as a single pdf-file:** Report the results focusing on clarity and brevity. When formatting the reporting, consider a format that can function for a PM when leading the discussion of the product team on what to prioritize in development.

Use a minimal number of pages (1 would be splendid). Detail the sources and potential notes on the last page.