Software Design & architecture

S39

Interaction Flow

As someone with an engineering background, I understand that the technical perspective is often different from the understanding of non-technical people. For example, in making enterprise application software (EAS). When developers create software according to the client's needs (non-technical), the client is still confused when using it. Even in practice, the user has agreed with the flow of the system that will create or sometimes the user himself makes the system to become a blueprint of the product.

Of course, user confusion can be overcome by having a training or user guide book. But it will become another problem when the system created in the form of software does not meet the needs of the user and causes changes to the addition of features.

This problem is not only experienced by users and EAS developers. UI designers also experienced the same thing when they had to hand off their work to developers. Although this problem can be solved with the help of a variety of prototype applications that can translate animations made by UI designers, time remains a challenge.

Flowchart, IFML, and prototype

- Flowcharts, as describe diagram system, are sufficient to be a reference for developers and users. But the size and complexity of a system can cause ambiguity, especially for users. Notably, there is no complete explanation of one function created. Imagine when the user requests a mechanic chair that can be folded. Then the engineer creates its focus on the size, capacity, material, and how to arrange it to fit with the request. Indeed, a chair will form, but the chair may be uncomfortable, not following the size of the user's room, and the nightmare, the user does not know how to fold or open the chair because they do not see the location of the button.

- Interaction Flow Modeling Language (IFML), which functions as expressing the content, user interaction, and control behaviour of the front-end, is beneficial for communicating user needs. In IFML, the functions of the product system well explained. The weakness of this IFML in the form of diagrams. The diagram is too technical and not easy to understand for the layman.

- Prototyping can be used as an answer to harmonize thoughts between developers and users. The prototype has functions and content that resembles the original product so that users with diverse backgrounds will easily understand it. But prototype with more detail emphasizes the minimum capacity of the product (MVP) rather than the entire system that should be made. Thus, prototypes cannot represent the software system, especially if the prototype made with a high fidelity interface that will take more time.

Prototyping can be used as an answer to harmonize thoughts between developers and users.

Combining the three main roles to get the key role of “interaction”

Interaction is the key to getting a complete picture of the system desired by the user. How to achieve it, users and engineers must be able to communicate about their needs and limitations, and this can be accomplished if both actors understand the communication used.

users and engineers must be able to communicate about their needs and limitations, and this can be accomplished if both actors understand the communication used

The idea is to combine the main concept from flowcharts, IFML, and prototypes. Content visualization is made with a low fidelity prototype to cut time. Low fidelity of this prototype can be made through wireframe with lines or simple shapes as notice that the form contains content or buttons. A shape and frame connected through arrows. Like the line of the flowchart, the direction of the arrow following the function of the shape. If there any decision o the system, a diamond symbol can be added. All functions are made with the same pattern until the features mapped and produce wireflows. Each flow change from one task to another explained through separate text outside the frame. This idea is similar to the pseudocode function in programming. Pseudocode is an informal way of programming description that does not require strict programming language syntax. In other words, explain the diagram and flow with a text outside the frame.

Micro-interactions as The Touch-up

Details needed so that the interaction feels more intense and clear. In this section, micro-interactions will help user interaction in understanding the functions provided. For example, the image of a hand gesture pressing or sliding a button that explains that the user can interact with the shapes made on the frame. While micro-interactions commonly used in experience design, with a customized formula, micro-interactions can also be used in interaction design and play an active role for users and developers to understand the functions of the system better.

However, micro-interaction not recommended to excessively used, and its use must adjust to the type of product/software. Excessive use of micro-interaction can result in confusion among users when using the software. While on the developers' side, the details provided by micro-interaction will result in code addition and creates runtime takes much longer (even the possibility is quite less)