Vi-Linh Nguyen
HCI 430 - Fall 2023

A Look into the Nuance of UX from a Programmer

Technology as a field is a vast field that extends more than what people outside the domain expect it to be often. As someone who comes from an artistic background learning programming, this has been confirmed again and again throughout my time studying computer science. The concentration of human computer interaction, often colloquially referred to as UX, has been the most enlightening part in seeing the intricacies of how people interact with technology. From the developer's side, who tends to focus on the code that builds an application, I've found that the information extracted from user centered tests involving prototypes to have the most nuance in planning out a software product.

The Tiny Fingers article by Mark Rettig talked about the advantages of a paper prototype mainly. Some of the ideas discussed started with how the field of human computer interaction evolved from the emphasis on software and fidelity to the efficiency of low fidelity and how important that can be. The advantages of the low fidelity were explained in the context of the timeline of a product's development so this offered efficiency benefits because it's cheap and easy to change making iteration easier to streamline through.

As mentioned, going with a high-fidelity prototype from the start has the risk of not getting enough valuable information from such a detailed prototype early in the development process (Rettig 1994). The kind of valuable information in question can potentially speak to how well the usability measures to the user. According to Rettig, developers can prioritize usability more during the planning stages of the project. This allows for the “formative evaluation” Rettig explains in detail in the article.

The entire process from ideation to prototyping is consistent with what is taught in DePaul's User Centered Design class. From what has been taught so far in this prototyping class, the development process is the same (because both classes explore the same field), except students will have the opportunity to dive more into the later stages of the development process of UX (where in contrast, the start of the process focused more on the research aspects of UX). Both of the aforementioned articles from Erickson and Rettig explore the standard development and design process in detail. The ideation part of the process does share similarities to how a programmer would plan out how to implement a software, but the high-level concepts differ.

The Notes on Design paper seems to focus more on the design and development process in the scope of how large organizations handle it, so good communication becomes tantamount to the project's chance of success. Although they do bring up the communicative aspect of individual work through the importance of physical artifacts in the design process (Erickson 1995). Furthermore, the influences at what kicks off the communication over the project is also multi-faceted.

There is a bigger focus in the Erickson paper about the different audiences that come up during the design and development cycle of an application or product, however unlike the Tiny Fingers article. The diversity of one single facet of an audience, like the design team as mentioned, is considered more and what kind of knowledge the varying disciplines contribute to the project at hand (Erickson 1995).

Erickson also goes into more detail about design artefacts that Rettig doesn't focus on as much, stories. The story's role in the design and development process explained by both authors play the part in considering how users would approach a certain problem using the product the designers are trying to design and develop, or a “framework” (Erickson 1995). Because of how unpredictable users can be, stories help consider that ambiguity before any solid plan is fully committed to. This would mean that designers are given a fair degree of adaptability to work with during the beginning stages of the design process, what Erickson calls the “exploratory stage” (Erickson 1995). To add on to the use of stories, Erickson goes onto explain the equalizing benefits of them in the team project, similar to the low fidelity prototype that comes up later in the process (Erickson 1995). To summarize, because both authors use the context of a large organization to explain their individual concepts, UX teams in bigger organizations tend to have various disciplines because it's such a collaborative field. Stories help get across concepts that may normally only be clear to certain disciplines if discussed formally.

In the field of UX, especially in large organizations both authors use to contextualize the type of project that's common enough to explain in such articles, teams that design a product and develop it can have people from so many different fields and disciplines. Likewise, the development stages explained in the Notes on Design Paper are emphasized as not “clearly separated” unlike the Tiny Fingers article (Erickson 1995). Take the Refinement stage for example, where the project is at a point where the designers know what the product is supposed to accomplish (Erickson 1995). However, the planning and discussions involved here lends to what is discussed by Rettig when the benefits of the lo-fi prototype in the process get explained more in detail in terms of determining requirements for the project. This is where Rettig describes preparing for testing. Returning to the core of what both mentioned articles share, prototypes, designers must consider that each kind of prototype made will be used for different kinds of information gathering. The testing Rettig describes includes the use of low fidelity prototyping, but the information there is to convey the overall concept of a product (Erickson 1995). Designers at this point tend to not prioritize the interactivity in their prototype just yet. That's saved for when more data is gathered.

From what I have learned in DePaul's User Centered Design class, the low fidelity prototype our team built did in fact go through several iterations just as Rettig's examples had, although the difference there is we used a low fidelity software prototype to get around location differences between the group of designers and the users we tested the prototype with. The reasoning for low fidelity included the tight deadline but the ease at adding changes with each user test was easier to grasp in practice than in theory. The “vision prototype” acted closely to a low fidelity prototype because it's generally quite easy to get lost in adding details when working with a software prototype. The biggest challenge to this phase of our team's product development is the very advantage Erickson described when the development stages were ambiguously separated (Erickson 1995). It may help in flexibility in planning but to a design team that is still figuring out their baring's into a new field, it makes gauging the team progress incredibly difficult. With a tight time, frame of a school project, there isn't always time for lengthy discussions of ideas. Plus, some of our group members were new to the high-level concepts involved in UX, what Erickson calls the “exploratory stage” (Erickson 1995).

In short, technological projects that heavily rely on users have a nuance in their high-level concepts that contrast starkly to the high-level concepts one would find in programming. The user adds a layer of unpredictability that poses both a flexibility and obstacle to how to tackle a problem. And that's if the team of designers can figure out what the problem is at hand. This is why prototyping plays such a vital role in UX, whether that team of designers is multidisciplinary or not. That data gathered thanks to the prototyping and testing lets the team know how to move forward.

Vi-Linh Nguyen
HCI 430 - Fall 2023

Citation

Rettig, Marc. “Prototyping for tiny fingers.” Commun. ACM 37 (1994): 21-27.

Erickson, Thomas. “Notes on design practice: stories and prototypes as catalysts for communication.” (1995).