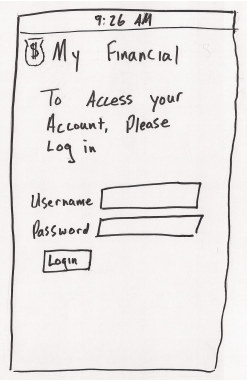
The Role of Prototyping

Before we can cover the role of prototyping, we need to explain what prototyping is. The process of prototyping creates a model of a system, and in most types of prototyping visual and interactive models. Part of the effort in creating these models is looking for the best way to implement features to help a person accomplish their tasks. These models are created in an effort to obtain early feedback on design decisions about the user interface and interactivity of the system while the cost is low. Usually the prototype has a limited set of functionality or features. This is done to allow for quicker and easier creation, feedback and refinement of design ideas, as well as reducing the expense and effort required to create the prototypes.



Part of the idea of prototyping is to come up with a series of design alternatives. With each of these alternatives, a prototype demonstrates a different possible "look and feel" for how the system could be put together. These prototypes then give users a concrete example that they can look to for what is possible. This allows the design team to obtain feedback on designs to iteratively create a better design.

There are many types of prototypes, each of which have their own goals, strengths, and weaknesses. These types vary mostly in two ways, their scope, that is how much of the system they represent, and their realism, that is the amount to which they represent the final system.

Scope

Vertical Prototypes

With a vertical prototype, the amount of effort required to create a prototype is achieved by limiting the number of features that will be explored. The included features however will be created in depth, and will likely link to real data. As vertical prototypes are near fully functional, they can be tested under realistic conditions with real tasks.

Horizontal Prototypes

Unlike vertical prototypes, horizontal prototypes do not include full functionality, but would include a user interface for the entire system. The large benefit to horizontal prototypes is that they can demonstrate the look and feel of all the features, and as such, it is possible to see how different features fit together. Unlike with a vertical prototype, since the features are not fully functional, tests cannot be performed with the same realism.

Scenarios

By reducing both the number of features and the amount of functionality, it is possible to reduce the prototype of the system to a number of scenarios. As the prototype has been reduced in both regards, they are easier to create require less effort and cost. These allow the prototype to be tested under a scripted interaction, and as such do not provide the same amount of realism as do the vertical or horizontal prototypes.

Realism

Low-Fidelity

The major characteristic of low-fidelity prototypes is that they provide a limited functionality and limited amount of interaction. Low-fidelity prototypes are useful in coming up with various alternative designs in a fast and cheap manner. Low-fidelity prototypes are mainly performed to show the concept, screen layout, or alternatives in design. These are also useful in the requirements elicitation phase to help gain feedback on potential designs. These can be thought of as trying to get the right design. Another feature of low fidelity prototypes is that they demonstrate what the system is supposed to do, but not the full details of the interactivity. As such, low-fidelity prototypes are usually presented in a very specific way to tell a story of how the system with function. These include fairly simple methods such as sketches and storyboards.

High-Fidelity

High-fidelity prototypes are designed to showcase the interactivity of the system. In many ways, the user treats the prototype just like they would the real system. Unlike the low-fidelity prototyping methods, high-fidelity prototypes have a high degree of realism. They take more effort to create then do the low-fidelity prototypes, but they are far more accurate in their depiction of the interface. These can be thought of as trying to get the design right. High-fidelity prototypes are usually almost complete versions of the product.

When and where Prototyping is normally used

Prototyping can be performed throughout the development process, however it makes the most sense early on to midway through the process, as this is when the most benefit can be taken. That being said, the types of prototypes differ at differing stages of the process.

Low fidelity prototypes can be used early on in the development process to help with eliciting requirements form the end user. This because it gives the user possible alternatives that they can use when they are trying to describe what they want. It also helps the design team by providing them with a means to make sure they are on the same page with the customers.

High fidelity prototypes are often created further along in the development process, when a general design has been decided on. They can also be simplified versions of the final system. These are useful for evaluation purposes, as they give concrete examples that can be tested by users. This allows the ability to ensure that more difficult concepts are handled in the correct way. At the same time, they can and should be performed early enough that changes can actually take place.

Benefits and Drawbacks of Prototyping

Prototyping as a whole

The big benefit of Prototyping is that compared to the development effort, it is fast and cheap. Prototyping also helps elicits more and better feedback early on in the development process. This can help identify some of the design problems in the system before the coding for the system has even begun. With these cheap and fast prototypes, it is usually possible to find the big usability issues or brick walls as they are commonly referred, with little effort, and usually no code. As

such, problems can be addressed when they are less costly, and little or no development effort has been taken. Another benefit is that prototyping encourages creativity in the possible solutions that it may produce.

One drawback of prototyping is that some view it as a duplication of effort. Depending on the appearance of the prototype, customers may get the impression that the product is nearly finished, when much of the production code is not yet complete. Finally, not all ideas that can be prototyped are necessarily feasible.

Low-Fidelity

The biggest benefit to low-fidelity prototyping is the low cost and effort required to create these prototypes. They are also useful when eliciting requirements from a customer, as they make the customer aware of what alternatives are possible.

The biggest issue with low-fidelity prototypes is that many people argue that more issues can be determined with high-fidelity prototypes. Also, it is not necessarily possible to implement all ideas that can be prototyped with low-fidelity prototypes, and sometimes these efforts can even overlook many design decisions.

High-Fidelity

The biggest benefit for high-fidelity prototypes is that as they are working systems, they are fully interactive, and provide the functionality required for testing. From these prototypes, it is also much easier for users to understand how the final system will operate. As they provide many major features, it is possible to use them as a marketing tool.

High-fidelity prototypes require far more effort and resources to create. It is also possible that with a higher degree of finish, that the customer could believe that the product is or is nearly complete. Unlike low-fidelity prototypes, it is not realistic to demonstrate many alternatives.

How to create a prototype

Prototypes are based on a set of realistic tasks that the end user of the system will need to perform. To create a prototype you will need to perform a number of steps. This example is suitable for a scenario or a vertical prototype. For different types of prototypes, consult the other readings.

1. Determine the Goal of the task; this will help you determine when you are finished. Should describe what the user wants to accomplish, but not how to do it.
2. Determine what input the user will need. The Inputs are those information or resources that the user will need to accomplish the task.
3. Determine what assumptions (if any) there are about the state of the system at the beginning of the task.
4. Break the task into a series of steps that the user will need to perform to complete the task.
5. Create screens for the steps as needed (usually 1 screen per step)

Materials needed

Pencil(s) and Paper

Example

We are creating a new iPhone application for a company called My Financial. This allows the users to do normal banking tasks, such as checking balance, recent transactions, transfers, etc. using their iPhone.

For this example, we are interested in creating a prototype for viewing recent transactions task.

My Bank has a new iPhone application, which allows me to check my bank account. I want to see if my rent check has gone through yet.

Goal

To determine if my rent check has be deposited yet

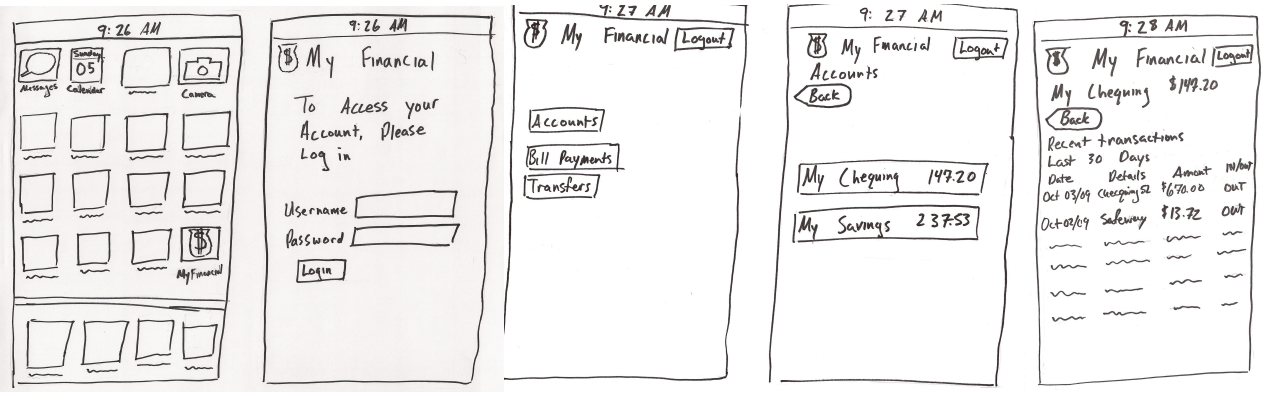
Inputs

- A Cheque Number (for the rent cheque)
- A user account (so the bank knows who they are)

Steps

1. Go to the home screen on your phone
2. Open the bank application
3. Authenticate
4. Go to Accounts
5. Open Chequing to view transactions

Screens



Exercise

You are building a new phone that can take pictures of barcodes, and present the user with a "buy this book" page.

For this exercise, we are interested in the task of buying a book.

Bibliography

Required

Nielsen, J. (1993) Usability Engineering, Chapter 4.8 Prototyping. Academic Press.

- This is a good overview of different prototyping techniques, mainly the vertical, horizontal, and scenario prototypes. I would recommend that

Other Useful Papers (If you are interested in Prototyping, these may be useful)

Rudd, J., Stern, K. and Isensee, S. (1996) **Low vs. high fidelity prototyping debate.** Interactions 3(1), p76-85, ACM Press.

- This paper is a good look at the debate over low fidelity and high fidelity prototypes, where each is useful, and the drawbacks of each.

Snyder, C. (2003) Paper Prototyping, Morgan Kauffman Publishers.

- Paper prototyping is one method of prototyping that is fast and cheap. In some ways it can provide feedback on early design ideas when no code has been written.

Greenberg, S. **Working through task-centered system design**. in Diaper, D. and Stanton, N. (Eds) The Handbook of Task Analysis for Human-Computer Interaction. Lawrence Erlbaum Associates.

- Describes a task-centered walkthrough, which is a cheap and easy way to evaluate a prototype