1 What is a Prototype?

1.1 General Definition

The American Heritage Dictionary gives the following definitions for a prototype (TAHDotEL04):

1. An original type, form, or instance serving as a basis or standard for later stages.

2. An original, full-scale, and usually working model of a new product or new version of an existing product.

3. An early, typical example.

As can be seen from these definitions prototypes can take many different forms from very simple early examples to full-scale working models of a product. The goals for developing a prototype are usually the same: To provide an early or continuous ability to observe something about the nature of a product (i.e. evaluate ideas about a product or weight alternatives for a product) (HH93). In interface development prototyping is often used to receive feedback from users for the refinement of the final product in a usability evaluation of the prototype. The term rapid prototyping refers to prototyping methods that produce prototypes fast enough to leave a substantial amount of time for actual changes of the product. Rapid prototyping also typically leaves enough time for several prototyping
iterations in the design life cycle during which the prototype can be refined based upon earlier evaluation steps (HH93).

1.2 Prototype Categorization

There are several different ways to categorize prototypes. Typically, categories emphasize one or more points along prototype dimensions as those presented in Figure 1 (VSK96):

- **Features**: How many features of the final product does the prototype include?
- **Functionality**: How much of the functionality of the final product is included in the prototype (for each feature)?
- **Interaction**: How similar is the interaction with the prototype to the interaction with the final product?
- **Design**: How similar is the prototype in design to the final product (color, graphic design, . . .)?

![Figure 1: Important aspects for categorizing prototypes.](image)

A prototype that compromises at least one of these dimensions is often called a **low-fidelity prototype** (VSK96). Low-fidelity prototypes often have limited functionality, features and interaction. They are built mostly to depict concepts, design alternatives, or screen layouts. Typical examples of low-fidelity prototypes include storyboards, drawings, paper mockups, etc. **High-fidelity prototypes**, in contrast, are typically fully interactive, represent the product’s core functionality and are often built with prototyping systems (e.g. Smalltalk, Visual Basic). They are used mostly for exploration and tests of the look and feel of the final product (RSI96). There is no clear separation between low-fidelity and high-fidelity prototypes, several different techniques can also be classified as medium-fidelity prototypes.
A third way of categorizing prototypes is based on how a target system is represented in the prototype. This categorization was introduced by Nielsen (Nie87). **Horizontal prototypes** (cf. Figure 2) include the full set of features that will be integrated in the final product but with limited functionality. This prototype is often involved in a simulation of the interface. A **vertical prototype** covers only a small set of features but those with full functionality. This prototype is used to evaluate parts of the system in depth, often under real circumstances with real tasks. **Scenarios** are a mixture of vertical and horizontal prototypes. In a scenario a user is led through the system along one or just a few paths. It only covers as much functionality as is currently needed.

![Figure 2: Categorization of prototypes according to the representation of the target system in the prototype by (Nie87).](image)

### 2 Advantages and Pitfalls of Prototyping

#### 2.1 Why Prototype?

The development of prototypes is useful for a number of different reasons (HH93):

- Early validation of applications with users, clients.
- Users can take an active part in the development of a product.
- Users are encouraged to share needs and wishes for the final product.
- They produce more visible results earlier (good for managers to show-off!).
- Improved collaboration & communication among developers, analysts, users.
- Encourages reflection about the product.
- Finds answers to questions about the design.
- Many prototypes are very easy to build.
- **Reduced risk of project failure!!**

### 3 Pitfalls of Prototyping

There are a number of pitfalls to prototyping. They are called pitfalls rather than disadvantages because they might be avoided with careful planning (HH93):

- Attempt to use prototyping techniques before securing cooperation from all parties involved in the procedure.
- Established management procedures might not involve prototyping.
- Reduction in programming discipline.
- Pressure to later use the prototype as the real-thing (from client or management).
- Overpromising or misleading with the prototype (prototyping something that cannot be included with the available resources).
- Trap of overdesign (too much time is spent on the prototype).
- Depending how the prototype was designed it might be hard to extend.

### 4 Prototyping in the Design Life Cycle

Prototyping can take place at different stages in the development cycle depending on the goals of the prototype. For different goals different prototyping techniques should be used at different stages. High-fidelity prototypes are full scale prototypes of the whole system. They do not (significantly) compromise any of the points as presented in Figure 1. These prototypes are used late in the design life cycle to test details of the design and the usability of the system. Figure 3 gives an overview of how different prototyping categories can be included in the design life cycle.
The following sections introduce a few low-fidelity prototypes that are used earlier in the design life cycle to test the overall high-level conceptual ideas about the product.

Low fidelity prototypes typically use a medium unlike the final medium. They are quick, cheap, and easily changed. These prototypes identify interaction styles, user needs and characteristics or usability goals and usually produce representations far from final code or GUI. Examples include:

1. *Interface sketch or drawing:* Drawings typically convey the look of the interface. Drawing interface ideas helps the design team to develop a shared representation. Drawings can be created fast and with hardly any costs involved. They are effective in communicating ideas in a group with people of many different backgrounds (designers, programmers, managers, etc.) Drawings should be made as soon as the tasks are identified that the interface should support (CV90).
2. **Storyboard**: Storyboards are a series of sketches showing how a user might progress through a task with a devise, similar to a comic strip. They are often used with scenarios. Storyboards include more detail than static drawings. A storyboard should include all feedback that can be received by the user, i.e. visible, tactile, or audio clues following or demanding interaction from the user. Verbal descriptions should typically be put below the graphics that show the interaction involved in the storyline. Storyboards should be developed after a task description has been developed (CV90).

3. **Flipbook**: Flipbooks are an ordered stack of separate images that when flipped create the illusion of movement. They are used to convey a dynamic in the interface. Flipbooks should be made before other types of animations are made that are more precise in their graphical layout like powerpoint slide shows or flash mockups. First, a detailed task description has to be made from which each sequence for the flipbook can be extracted. The system reactions and input options have to be known before creating the flipbook. Important or critical points in the interaction can be marked off in the flipbook with sticky notes. Flipbooks are usually used to resolve interaction and sequence detail (CV90) (reference image also taken from this source).
4. **Physical mockup (cardboard, wood, etc.):** Many physical prototypes are built on paper. In these prototypes a number of system interactions and reactions are created physically for a later simulation (for example, menus drop down, dialogs pop up, etc.) (Ret94). Physical mockups should be made out early in the design life cycle to test interaction metaphors. These prototypes are particularly useful for an evaluation as will be discussed in the following.

An important part of prototyping involves testing the prototypes on potential users. It has been shown that usability evaluation of prototypes, even with those of low-fidelity, can be very effective ((VSK96), (RSI96)). Evaluation of prototypes can, therefore, be conducted throughout the design lifecycle (cf. Figure 3). Two techniques will be introduced that were
specifically developed for an evaluation. In the class exercise we will try out a method for usability evaluation of a prototype as suggested by (Ret94).

- **Wizard-Of-Oz:** This technique involves a human ‘wizard’. The wizard simulates reactions from the system and, therefore, the system’s ‘intelligence’ (MGM93). The wizard is typically hidden from the user. The user only interacts with a mockup interface. Typically, the Wizard-Of-Oz method is used to evaluate a system or user behavior with a system that cannot be built with current resources. This form of prototyping and evaluation should be performed early in the design cycle to study user expectations and requirements.

- **Pictive:** Pictive stands for **P**lastic **I**nterface for **C**ollaborative **T**echnology **I**nitatives through **V**ideo **E**xploration) (Mul91). It is a paper mock-up (rapid prototyping) technique. It involves recording a design session in which different design team members (designers, programmers, potential users, etc.) participate. The team members change the mockup on-the-fly. No special knowledge is required to change a paper mockup so that each design team member has the same opportunity to contribute to the project. The video records make record-keeping easier and serve as a replacement for a design document. Videos can later be used to evaluate the different perspectives of different groups of participants that might have not been noticed during the design session.
5 The Class Exercise

In the year 2100 our cars still need gas (for some reason) and we still have enough natural resources (we live in Alberta). One major thing has changed about gas-stations, however. Cars now fill up by themselves, there is no need to get out of the car anymore. A new type of interface is needed through which the car driver can specify how much gas the car should get. The input device for this information is a cd-sized round touch-screen display that is placed inside the car. Our task is to deliver and test a prototype for the interface.

We will follow these steps in the development of the prototype (as suggested by (Ret94)):

1. Split in two groups. Each group gets one whiteboard for rapid prototyping.

2. Each group gets 5 minutes to sketch an interface on the whiteboard for the new car-self-fillup mechanism. The main interface should be sketched on the whiteboard while any changing content (based on user interaction) should be put on sticky notes or another exchangeable medium. The interface has to support one simple task: input of the amount of gas to fill the car.

3. For a later evaluation each group needs to assign team members to the following four positions:

   a) **Greeter:** Welcomes the user and tries to put them at ease. Might ask the user to fill out an experience profile (we don’t have time for that here).

   b) **Facilitator:** Takes the lead once the test is set up. Is the only person allowed to speak freely during the test. The facilitator gives the user instructions, encourages him/her to express thoughts or concerns, makes sure the evaluation gets done in time. Saul has shown us how to do this before. Try to remember how he did it.

   c) **Computer:** One team member acts as the computer. He/she knows the interface and reactions of the system to user interaction. The 'computer' arranges the prototype to simulate a response from the system. He/she does not speak.

   d) **Observer:** The remainder of the team. Should take notes of observations. Observer do not talk!

4. One person from the other team will act as a subject. Try to run through an evaluation of the prototype in each group and identify possible weaknesses of your prototype.
6 Annotated Bibliography

Jakob Nielsen: Usability Engineering, Chapter 4.8, (Nie93)

According to the author the idea behind prototyping is saving time and cost in the production and refinement of a product. He introduces two general dimensions of prototyping involving vertical and horizontal prototypes. He then presents several methods for producing prototypes faster, including: accepting poor code, wizard-of-oz methods, fake data, etc. The discussion of advantages and disadvantages of prototyping methods is kept very short. A brief description of scenarios as the ultimate minimalist prototype concludes the chapter.

Marc Rettig: Prototyping for Tiny Fingers, (Ret94)

The author is mostly concerned with low-fidelity prototyping. High-fidelity prototyping is touched shortly and some problems are discussed. The remainder of the article explains how to build a low-fidelity prototype, how to prepare for a test, and how to conduct and evaluate a test using such a prototype. According to the author the main advantage of low-fidelity prototyping is the maximization of the number of times a product can be refined.

Rudd et al.: Low vs. High-Fidelity Prototyping Debate, (RSI96)

As the title suggests this article summarizes advantages and disadvantages of low- and high-fidelity prototyping. The authors introduce guidelines on when, where, or for what purpose to use both prototyping methods. According to the authors, low-fidelity prototyping should be used before program requirements have been established whereas high-fidelity prototyping should be performed in conjunction with the development of a written specification.

Virzi et al.: Usability Identification Using Both Low- and High-Fidelity Prototypes, (VSK96)

This paper introduces two studies that conclude that low-fidelity prototypes can be effective throughout the product development cycle not just at the beginning. It stands in contrast to earlier studies by Nielsen((Nie90)). Effectiveness was measured by the number of usability problems uncovered in a user interface.
Gayle Curtis and Laurie Vertelney: Storyboards and Sketch Prototypes for Rapid Interface Visualization (CV90)

This conference tutorial booklet gives a great and simple overview of prototyping in interface design. The authors discuss a number of techniques and give short examples for each.

Carol Snyder: Paper Prototyping (Sny04)

There is a book by Carol Snyder with the same title as the Website I visited. The website provides great resources of additional material accompanying the book. This material includes downloads of pdfs with checklists of when to use paper prototyping, an extensive article list, and a short introduction to paper prototyping.

Chris Farnum: What an IA Should Know About Prototypes for User Testing (Far02)

A short but interestingly written introduction to low-, medium-, and high-fidelity prototypes from the point of view of an information architect. Also gives a good introduction to other attributes of prototypes like interactivity (static vs. automated prototypes).

Deborah Hix, H.Rex Hartson: Deveoping User Interfaces (HH93)

One of the most cited and best introductions to prototyping is given in Chapter 9 of the book. The authors are concerned with rapid prototyping, give some classifications, discuss advantages and pitfalls, and give many examples.
References


