
Chapter 8. Prototypes

Table of Contents

What is a prototype?	1
The role of prototypes in research	1
Why build a prototype?	1
When to build a prototype	2
Low- and high-fidelity prototypes	2
Prototyping steps	2
Problems with prototyping	2

What is a prototype?

A prototype is a simplified system, designed and built to focus on certain aspects while ignoring others. For example, a prototype might ignore user interface issues entirely, or it may be built solely for ease of development without concern for efficiency, or it may implement only some parts of a system and omit others entirely, etc. A prototype is easier to study and reason about than an abstract model, and quicker to produce than a complete working system. It can also be easier to analyse than a full system because unimportant aspects have been factored out. A prototype is never the major contribution of research, but it is often a helpful tool for achieving some other research goal.

The role of prototypes in research

Prototypes are mainly used in the following ways in IT research:

- To demonstrate that an idea is practical and feasible, by implementing it in some prototypical form
- To enhance an idea/model/system/architecture/approach, and develop it in more detail - by discovering problems, solutions and alternative opportunities in the course of prototype construction
- As a vehicle for conducting an experiment, to test some property of a system or approach — e.g. performance, usability, integration etc.
- To study the use of some technique in a particular application context, one which has characteristics different from other environments where the technique has so far been applied

Why build a prototype?

Constructing prototypes enables us to study, test and improve on ideas and designs. The more prototype systems developed and explored, the more we will learn and the more our research will benefit. The longer a research project proceeds without constructing a prototype, the more time is being invested in an untested concept; and the more choices are being made without verification. Researchers run the risk of developing in more and more detail an idea that is unsound, flawed or inferior.

Building a prototype to test an idea is a type of formative evaluation, so-called because it is being tested in order to help form the solution to the research problem. A prototype facilitates testing decisions already made (are there any problems with what has already been done?) and assists in future decision-making (what is the best alternative for the next step?). Prototypes are particularly useful for checking system performance, but partial systems can also be built to answer specific questions about certain components or aspects of a system.

When to build a prototype

Prototyping is used extensively in engineering. Where there is an accepted theory, model or method of analytical testing this is used initially and prototyping is left until a design is almost complete; otherwise prototypes are built early in the project.

To build a prototype to evaluate an idea, that idea must first be developed in sufficient detail to make implementation possible. You need to identify the components (including hardware, users and external software) and the interactions of interest, what information they require, and how to represent this. Overall system requirements need to be decomposed into tasks and operations, which must then be allocated to individual components. Relationships between components and operations must be decided, for example the temporal ordering of operations and the information passed between components

Low- and high-fidelity prototypes

Low-fidelity prototypes do not look very much like the system envisaged, but are quick and cheap to construct and test, and are useful particularly in the early stages of the research project. A Wizard of Oz prototype is an example of a low-fidelity prototype used in HCI, where an interface design is evaluated by having a (hidden) human respond to inputs in the place of software components which have not been built. User interface prototypes can even involve hand drawings and story boarding in the beginning, to check usability before any software is produced.

Initial prototyping should be fast and simple as possible, to get an idea whether or not the system is on the right track as quickly as possible. Focus on key aspects only, and keep the number of tests and amount of analysis down to a minimum, since only significant benefits and problems are a concern at this stage.

High fidelity prototyping creates systems that are much closer to the final product envisaged. Besides taking longer to develop and having a higher chance of containing hidden bugs, they can cause researchers to be distracted by less important details instead of focusing on the issues that the prototype was intended to explore.

Prototyping steps

1. identify the question(s) to be answered by the prototyping exercise
2. design a prototype specifically for answering those questions
3. design an experiment using that prototype to answer the questions
4. implement the prototype
5. conduct the experiment using the prototype
6. process and analyse the results from testing the prototype
7. use the results to answer the questions and direct future research accordingly

Problems with prototyping

The main problems with prototyping are that it is very time-consuming, and researchers find it difficult to throw away prototype code when drawbacks or problems are identified because of the time they have already invested in that prototype. To avoid these problems it is best to build partial solutions, and gradually extend (or, when necessary, replace) them to test new aspects as they are considered.