1.3 Product prototyping

Product prototyping is the process of developing an early version of a product to test your ideas and to convince yourself and company funders that your product has real market potential. You use a product prototype to check that what you want to do is feasible and to demonstrate your software to potential customers and funders. Prototypes may also help you understand how to organize and structure the final version of your product.

You may be able to write an inspiring product vision, but your potential users can only really relate to your product when they see a working version of your software. They can point out what they like and don't like about it and make suggestions for new features. Venture capitalists, whom you may approach for funding, usually insist on seeing a product prototype before they commit to supporting a startup company. The prototype plays a critical role in convincing investors that your product has commercial potential.

A prototype may also help identify fundamental software components or services and test technology. You may find that the technology you planned to use is inadequate and that you have to revise your ideas on how to implement the software. For example, you may discover that the design you chose for the prototype cannot handle the expected load on the system, so you have to redesign the overall product architecture.

Building a prototype should be the first thing you do when developing a software product. Your goal should be to have a working version of your software that can be used to demonstrate its key features. A short development cycle is critical; you should aim to have a demonstrable system up and running in four to six weeks. Of course, you have to cut corners to do this, so you may choose to ignore issues such as reliability and performance and work with a rudimentary user interface.

Sometimes prototyping is a two-stage process:

1. *Feasibility demonstration* You create an executable system that demonstrates the new ideas in your product. The goals at this stage are to see whether your ideas actually work and to show funders and company management that your product features are better than those of competitors.
2. *Customer demonstration* You take an existing prototype created to demonstrate feasibility and extend it with your ideas for specific customer features and how these can be realized. Before you develop a customer prototype, you need to do some user studies and have a clear idea of your potential users and scenarios of use. I explain how to develop user personas and usage scenarios in Chapter 3.

You should always use technology that you know and understand to develop a prototype so that you don't have to spend time learning a new language or framework. You don't need to design a robust software architecture. You may leave out security features and checking code to ensure software reliability. However, I recommend that, for prototypes, you should always use automated testing and code management. These are covered in Chapters 9 and 10.

If you are developing software without an external customer, such as software for a research group, it may be that a prototype system is all you need. You can develop and refine the prototype as your understanding of the problem develops. However, as soon as you have external users of your software, you should always think of your prototype as a "throw-away" system. The inevitable compromises and shortcuts you make to speed up development result in prototypes that become increasingly difficult to change and evolve to include new features. Adding security and reliability may be practically impossible.