CHAPTER 15
Risk reduction through prototyping

Objectives

- This chapter describes how prototyping provides value to the project and different kinds of prototypes you might create for different purposes.
- It also offers guidance on how to use them during requirements development, as well as ways to make prototyping an effective part of your software engineering process

Contents

1. Prototyping: What and why
2. Mock-ups and proofs of concept
3. Throwaway and evolutionary prototypes
4. Paper and electronic prototypes
5. Working with prototypes
6. Prototype evaluation
7. Risks of prototyping
8. Prototyping success factors

Prototyping: What and why

- Purpose:
  - Clarify, complete, and validate requirements
  - Explore design alternatives
  - Create a subset that will grow into the ultimate product
- Three classes of prototype attributes
  - Scope
  - Future use
  - Form

Mock-ups and proofs of concept

- Definition: people say “software prototype,” they are usually thinking about a mock-up of a possible user interface. A mock-up is also called a horizontal prototype.
- A proof of concept, also known as a vertical prototype, implements a slice of application functionality from the user interface through all the technical services layers
- Purpose:
  - demonstrate the functional options the user will have available, the look and feel of the user interface (colors, layout, graphics, controls), and the navigation structure
  - explore some specific behaviors of the intended system, with the goal of refining the requirements
  - represent the developer’s concept of how a specific use case might be implemented
  - …

Throwaway prototypes

- Purpose: build a throwaway prototype to answer questions, resolve uncertainties, and improve requirements quality (Davis 1993)
- A wireframe is a particular approach to throwaway prototyping commonly used for custom user interface design and website design.
- You can use wireframes to reach a better understanding of three aspects of a website:
  - The conceptual requirements
  - The information architecture or navigation design
  - The high-resolution, detailed design of the pages

Evolutionary prototypes

- Purpose: provides a solid architectural foundation for building the product incrementally as the requirements become clear over time (McConnell 1996).
- Agile development provides an example of evolutionary prototyping. Agile teams construct the product through a series of iterations, using feedback on the early iterations to adjust the direction of future development cycles.
- Evolutionary prototyping is well suited for web development projects

Evolutionary prototypes

TABLE 15-1 Typical applications of software prototypes

	Throwaway	Evolutionary
Mock-up	■ Clarify and refine user and functional requirements. ■ Identify missing functionality. ■ Explore user interface approaches.	■ Implement core user requirements. ■ Implement additional user requirements based on priority. ■ Implement and refine websites. ■ Adapt system to rapidly changing business needs.
Proof of concept	■ Demonstrate technical feasibility. ■ Evaluate performance. ■ Acquire knowledge to improve estimates for construction.	■ Implement and grow core multi-tier functionality and communication layers. ■ Implement and optimize core algorithms. ■ Test and tune performance.

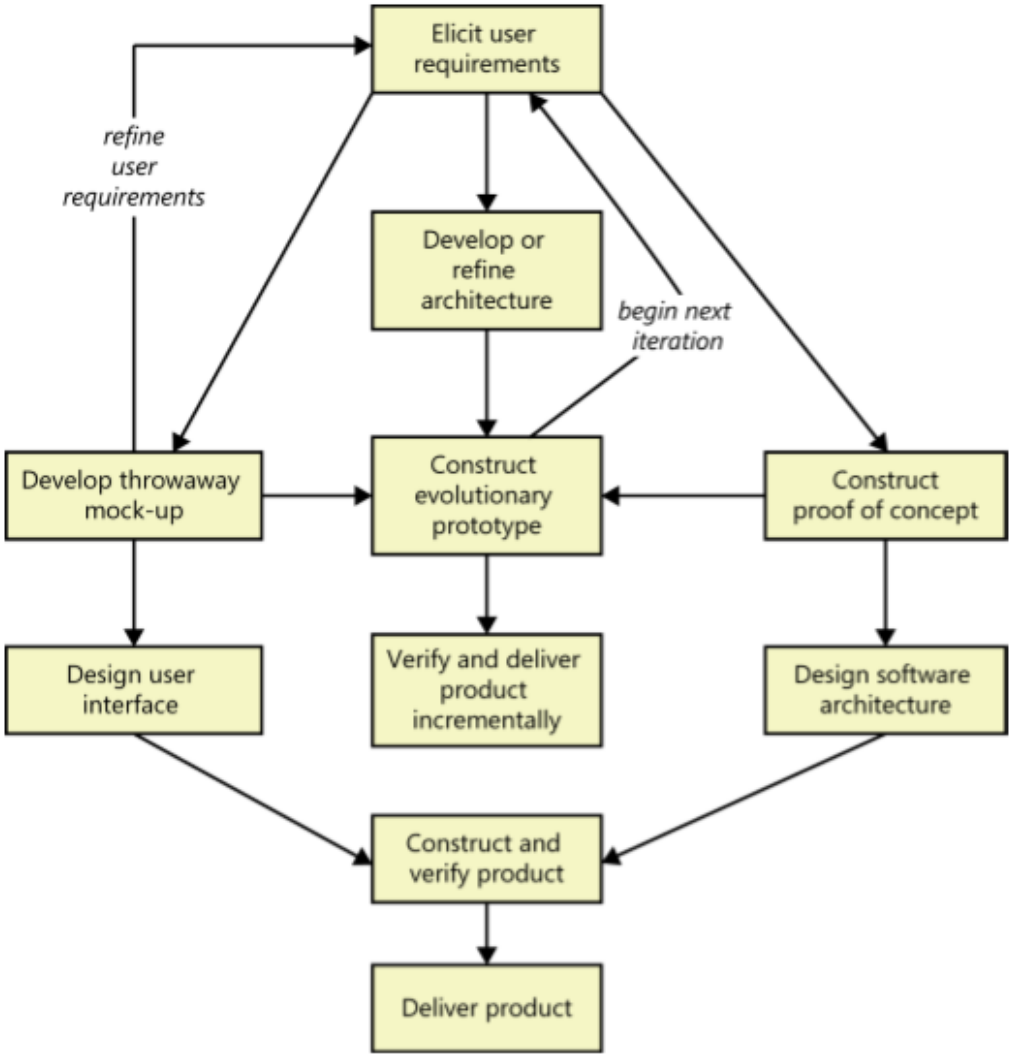
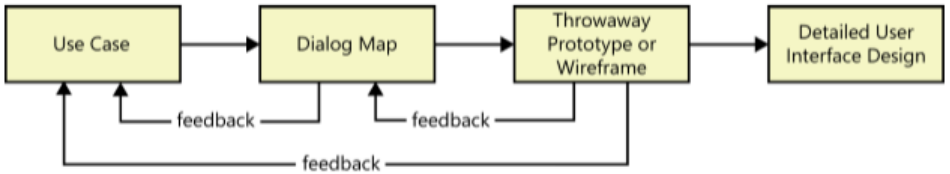


FIGURE 15-1 Several possible ways to incorporate prototyping into the software development process.

Paper and electronic prototypes

- Paper prototype
  - Pros
  - Cons
- Electronic prototype

Working with prototypes



n use cases to use

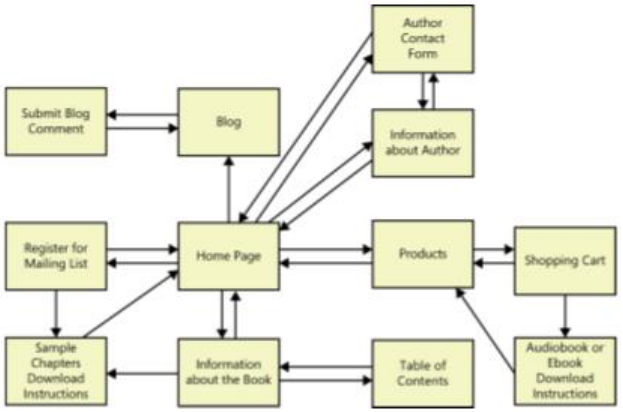


FIGURE 15-3 Partial dialog map for PearlsFromSand.com.

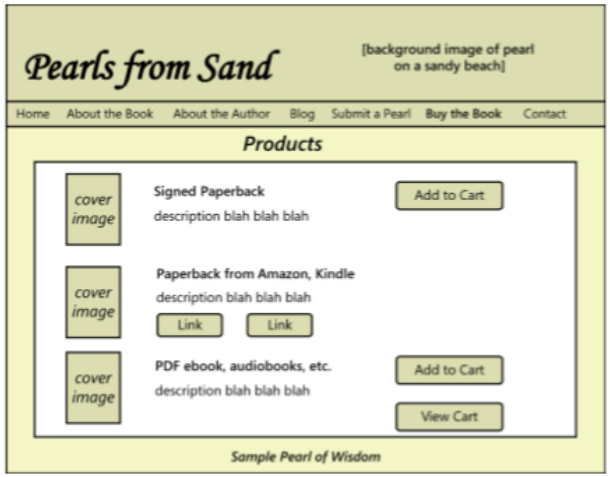


FIGURE 15-4 Sample wireframe of one page for PearlsFromSand.com.

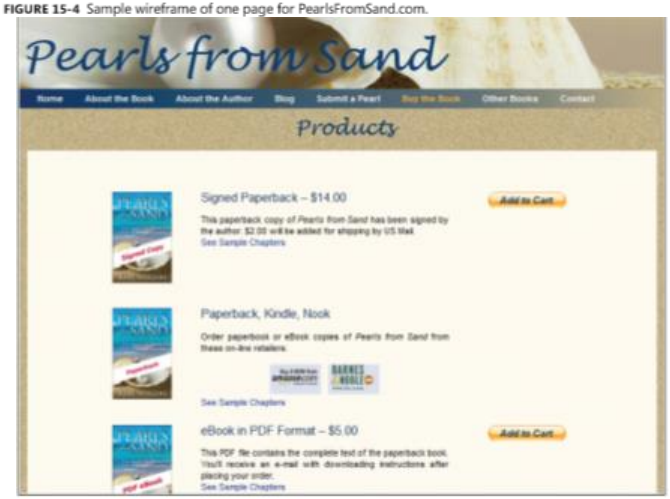


FIGURE 15-5 A final implemented page from PearlsFromSand.com.

TABLE 15-2 Some use cases for PearlsFromSand.com

User class	Use case
Visitor	Get Information about the Book Get Information about the Author Read Sample Chapters Read the Blog Contact the Author
Customer	Order a Product Download an Electronic Product Request Assistance with a Problem
Administrator	Manage the Product List Issue a Refund to a Customer Manage the Email List

Prototype evaluation

You might ask the following questions:

- Does the prototype implement the functionality in the way you expected?
- What functionality is missing from the prototype?
- Can you think of any possible error conditions that the prototype doesn't address?
- Are any unnecessary functions present?
- How logical and complete does the navigation seem to you?
- Are there ways to simplify any of the tasks that require too many interaction steps?
- Were you ever unsure of what to do next?

Risks of prototyping

- Pressure to release the prototype
- Distraction by details
- Unrealistic performance expectations
- Investing excessive effort in prototypes

Prototyping success factors

- Include prototyping tasks in your project plan.
- State the purpose of each prototype before you build it, and explain what will happen with the outcome: either discard (or archive) the prototype, retaining the knowledge it provided, or build upon it to grow it into the ultimate solution.
- Plan to develop multiple prototypes.
- Create throwaway prototypes as quickly and cheaply as possible.
- Don’t include input data validations, defensive coding techniques, error-handling code, or extensive code documentation in a throwaway prototype.
- Don’t prototype requirements that you already understand, except to explore design alternatives.
- Use plausible data in prototype screen displays and reports.
- Don’t expect a prototype to replace written requirement