



# PROTOTYPING

A Practitioner's Guide

by **TODD ZAKI WARFEL** foreword by Dave Gray



# TABLE OF CONTENTS

---

<b>How to Use This Book</b>	<b>iii</b>
Who Should Read This Book?	iii
What's in This Book?	iii
What Comes with This Book?	iv
<b>Frequently Asked Questions</b>	<b>v</b>
<b>Foreword</b>	<b>xii</b>
<b>Introduction</b>	<b>xiv</b>
<b>CHAPTER 1</b>	
<b>The Value of Prototyping</b>	<b>1</b>
Prototyping Is Generative	3
Prototyping—The Power of Show, Tell, and Experience	3
Prototyping Reduces Misinterpretation	5
Prototyping Saves Time, Effort, and Money	6
Prototyping Reduces Waste	7
Prototyping Provides Real-World Value	9
Summary	12
<b>CHAPTER 2</b>	
<b>The Prototyping Process</b>	<b>13</b>
The Absence of Design Studio	14
What Does the Prototyping Process Look Like?	15
Summary	26
<b>CHAPTER 3</b>	
<b>Five Types of Prototypes</b>	<b>27</b>
Type 1: Shared Communication	28
Type 2: Working Through a Design	33

---

Type 3: Selling Your Idea Internally	33
Type 4: Usability Testing	35
Type 5: Gauging Technical Feasibility and Value	37
Summary	42

## **CHAPTER 4**

### **Eight Guiding Principles** **43**

Principle 1: Understand Your Audience and Intent	44
Principle 2: Plan a Little—Prototype the Rest	45
Principle 3: Set Expectations	46
Principle 4: You Can Sketch	48
Principle 5: It's a Prototype—Not the Mona Lisa	50
Principle 6: If You Can't Make It, Fake It	51
Principle 7: Prototype Only What You Need	53
Principle 8: Reduce Risk—Prototype Early and Often	53
Summary	56

## **CHAPTER 5**

### **Picking the Right Tool** **57**

Influencers	58
What Tools Are People Using?	60
What Kinds of Prototypes Are They Making?	61
Summary	63

## **CHAPTER 6**

### **Paper and Other Analog Methods** **65**

Strengths	66
Weaknesses	67

---

Essential Paper Prototyping Kit	68
Progressive Paper Prototyping	70
Summary	81

## **CHAPTER 7**

### **PowerPoint and Keynote** **83**

Strengths	85
Weaknesses	86
Creating Narrative Prototypes with PowerPoint	86
Creating Interactive Prototypes with PowerPoint	89
AJAX Effects in PowerPoint	93
Summary	95

## **CHAPTER 8**

### **Visio** **97**

Strengths	99
Weaknesses	99
Prototyping with Visio	100
Additional Resources	107
Summary	109

## **CHAPTER 9**

### **Fireworks** **111**

Strengths	113
Weaknesses	114
Prototyping an iPhone Application with Fireworks	115
Additional Resources	125
Summary	127

---

<b>CHAPTER 10</b>	
<b>Axure RP Pro</b>	<b>129</b>
Strengths	131
Weaknesses	132
Building a Video Web Site Prototype with Axure RP	132
Additional Resources	146
Summary	147
<b>CHAPTER 11</b>	
<b>HTML</b>	<b>149</b>
Strengths	151
Weaknesses	152
Prototyping with HTML	152
Creating an HTML Prototype	154
Additional Resources	168
Summary	170
<b>CHAPTER 12</b>	
<b>Testing Your Prototype</b>	<b>171</b>
Common Mistakes	172
Preparing for a Usability Test	176
Design Test Scenarios	177
Test Your Prototype	179
Record Observations and Feedback	180
Analyze and Determine Next Steps	182
A Final Word	184
Summary	185

---

Index	187
Acknowledgments	195
About the Author	197

In 2008, I ran the first survey on prototyping for this book. My goal was to understand what tools UX people are using for prototyping, why they're using those tools, and what they want from those tools. I also wanted to see how my experiences and choices compared to the rest of the UX community.

Roughly 200 people responded from a variety of UX backgrounds, including researchers, designers, developers, product managers, and business analysts.

This chapter discusses the results of that survey and offers some guidance for selecting the right prototyping tool for your environment.

## Influencers

When I started this book, most of my prototyping was done using HTML and CSS. Honestly, I had to fake the majority of my AJAX interactions and transitions. My JavaScript skills were pretty mediocre.

One day, I decided to bite the bullet and get familiar with the Prototype and script.aculo.us JavaScript library. It took a few days of trial and error before I finally got a handle on things. I haven't looked back since.

At the time of writing this chapter, most of my prototyping is still done using HTML, CSS, and Prototype and script.aculo.us. My JavaScript skills have evolved, and I no longer have to fake AJAX.

The available prototyping options continue to grow. By the time I finish this book, my tools of choice could very well change or evolve again. I've been playing with the jQuery JavaScript library. Adobe just released a new version of Fireworks and Flash. Who knows what will be available?

So why choose HTML, Flash, Fireworks, Axure, or something else? According to the survey, these are the top influencers that drive tool choice, in order of importance:

1. Familiarity and availability
2. Time and effort to produce a working prototype
3. Creating usable prototype for testing

4. Price
5. Learning curve
6. Ability to create own GUI widgets
7. Available on my platform
8. Collaborative/remote design capabilities
9. Built-in solutions/patterns for AJAX transitions
10. Built-in GUI widgets
11. Creating usable source code

I can definitely find my own influencers in that list with one exception—knowing the audience and intent. This is my number one concern. But my other concerns, like familiarity, availability, comfort, availability of GUI widgets and AJAX, or ability to create my own are definitely in that list. They just might be in a slightly different order.

You'll see this list again later in the book. I'm going to use it to evaluate a number of different prototyping tools.

You may notice the presence of creating usable source code. You may also notice that it's at the bottom of the list. This one's a bit tricky.

Creating prototypes that are reusable for production isn't something that's commonly done or recommended. There are several reasons for this. Prototyping is a rapid, iterative process. It's a process for exploring ideas, failing often, and learning quickly. If you're focused on creating production-level code, you start to lose some of the value of rapid iteration.

On the other hand, if you can produce reusable source code with very little or no additional effort, then your prototyping tool has the added value of development time.

At the time of this book's publication, there isn't a dedicated prototyping tool for non-coders that can create rapid prototypes with little effort and produce reusable source code. If you want a prototyping tool that produces reusable source code, you'll have to learn to code HTML or use a development framework like Ruby on Rails.

## What Tools Are People Using?

As I expected, the usual suspects like paper, PowerPoint, Flash, and HTML showed up on the most common tools list. Also, not surprising is that participants used more than one tool. For instance, they might use a combination of paper for sketching and then go into Photoshop and finally HTML.

In the past couple of years, a couple of newcomers have hit the scene and continue to gain traction, namely Axure RP and Fireworks. While Fireworks has been around for some time, only recently has it gained built-in prototyping capabilities. Here's a breakdown of the most common tools being used for prototyping in the UX field as of 2008 (see Table 5.1).<sup>1</sup>

**TABLE 5.1 SURVEY RESULTS OF COMMON PROTOTYPING TOOL USAGE**

Paper	77.0%
Visio	59.0%
PowerPoint	43.0%
Dreamweaver	47.0%
Axure	30.0%
OmniGraffle	30.0%
Illustrator	23.0%
Flash	21.0%
Acrobat	19.0%
Fireworks	18.0%
InDesign	12.0%
Photoshop	10.0%
Other HTML editor	4.0%
Keynote	3.0%
Flex	2.0%
Blend	0.2%
iRise	0.1%
Other (Excel, FileMaker)	0.1%

<sup>1</sup> Participants were allowed, but not required to select their most common tools up to a maximum of three—one participant could select Paper, Visio, and HTML. Since participants could select up to three tools, the percentages will exceed 100. [toddwafel.com/archives/first-prototyping-survey-results/](http://toddwafel.com/archives/first-prototyping-survey-results/)

I'm sure you can see a number of familiar tools in that list. Did you notice any surprises?

Probably the biggest surprise to me was the number of people who use Visio for prototyping. I expected to see Visio, but didn't expect nearly 60 percent. Another surprise was the presence of Excel and FileMaker for prototyping. While this made up less than one percent of respondents, I was surprised to see it at all. Finally, the sheer number of prototyping tools being used for prototyping was a bit surprising—over 26 total if you include the items listed under Other.

## What Kinds of Prototypes Are They Making?

Paper continues to be at the top of the list. The continued growth of AJAX libraries like Prototype and jQuery, combined with the increasing number of auto-generating tools like Axure, iRise, and Fireworks, has made creating interactive prototypes easier than ever.

Ruby on Rails (RoR) is another one that has seen an increase in use for prototyping. Rails is a rapid application development framework, which makes it an ideal tool for both prototyping and production. Rails maintains all the advantages of HTML with the added capability to use dynamic data.

Here's a breakdown in Table 5.2 of the most common types of prototypes that people in the UX community are building.

**TABLE 5.2 SURVEY RESULTS OF COMMON TYPES OF PROTOTYPES**

Paper	81.0%
Hand-coded HTML	58.0%
Auto-generated (Axure, iRise, Visio, Fireworks, or similar)	39.0%
Clickable screenshots using HTML	34.0%
Flash, Flex, AIR, or Blend	27.0%
Keynote or PowerPoint	24.0%
Clickable PDFs	21.0%
Production environments (Rails, Java, .Net, PHP, Xcode)	9.0%
3D models (cardboard, foam core)	2.0%

## What Kinds of Prototypes Are They Making?

Most of my prototypes are created using HTML and AJAX. I use an HTML/CSS framework we developed in-house at Messagefirst combined with either the Prototype or jQuery JavaScript library. While my prototypes are all hand-coded, leveraging an existing, well-tested framework and JavaScript library means I can maintain a rapid, iterative approach.

In some cases, I need to use something other than HTML, for example, for mobile or TV. If I need to prototype for mobile, I typically use paper or Flash. If I need to prototype, say, for instance, a DVR interface, I use Flash or Keynote.

## Summary

I've never been one to do something just because others are doing it. And in the case of prototyping that hasn't changed.

Most of the prototypes I create are production level HTML prototypes, but I wouldn't recommend that for most people. I choose that method because it's the method that suits my needs and I'm comfortable with the tools. So, how would I recommend you select a prototyping method or tool? Well, consider the following:

1. **Audience:** Who is going to view or interact with the prototype?
2. **Intent:** Think back to the five types of prototypes. Which of these do you need?
3. **Familiarity and learnability:** Are you familiar with the method or tool, or willing to learn it?
4. **Cost:** Don't just think about the license cost, also consider the cost of downtime if you need to learn it.
5. **Collaboration:** Do you need it? If so, your choices are significantly limited.
6. **Distribution:** How will you share it with others?
7. **Throwaway versus reusable:** If you need reusable source code, then your choices are limited. If you're going to throw it away, which is more likely, then your options are wide open.

You'll find a comparison matrix inside the front cover of this book, which will help you evaluate the methods and tools reviewed in this book.

At this point in the book, you've completed all the overview material for prototyping. From this point forward, I'm going to focus on specific tools, evaluating them based on the criteria listed in this chapter, and I will also cover a few tips and techniques for each of the tools.