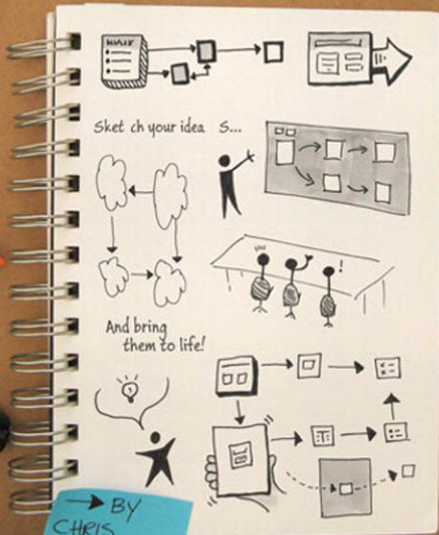


DYNAMIC PROTOTYPING

WITH SKETCHFLOW IN EXPRESSION BLEND



→ BY
CHRIS
BERNARD
SARA
SUMMERS

For eWord BY BILL BUXTON

que

Dynamic Prototyping with SketchFlow in Expression Blend

Copyright © 2010 by Pearson Education, Inc.

All rights reserved. No part of this book shall be reproduced, stored in a retrieval system, or transmitted by any means, electronic, mechanical, photocopying, recording, or otherwise, without written permission from the publisher. No patent liability is assumed with respect to the use of the information contained herein. Although every precaution has been taken in the preparation of this book, the publisher and author assume no responsibility for errors or omissions. Nor is any liability assumed for damages resulting from the use of the information contained herein.

ISBN-13: 978-0-7897-4279-7

ISBN-10: 0-7897-4279-9

Library of Congress Cataloging-in-Publication Data:

Bernard, Chris (Christopher M.)

Dynamic prototyping with SketchFlow in Expression blend : sketch your ideas-- and bring them to life / Chris Bernard and Sara Summers.

p. cm.

ISBN 978-0-7897-4279-7

1. Microsoft Expression blend. 2. Graphical user interfaces (Computer systems)
3. Web sites--Authoring programs. 4. Web sites--Design. I. Summers, Sara. II. Title.

QA76.9.U83B485 2010

005.4'376--dc22

2010001973

Printed in the United States of America

First Printing: March 2010

Trademarks

All terms mentioned in this book that are known to be trademarks or service marks have been appropriately capitalized. Que Publishing cannot attest to the accuracy of this information. Use of a term in this book should not be regarded as affecting the validity of any trademark or service mark.

Warning and Disclaimer

Every effort has been made to make this book as complete and as accurate as possible, but no warranty or fitness is implied. The information provided is on an "as is" basis. The authors and the publisher shall have neither liability nor responsibility to any person or entity with respect to any loss or damages arising from the information contained in this book or programs accompanying it.

Bulk Sales

Que Publishing offers excellent discounts on this book when ordered in quantity for bulk purchases or special sales. For more information, please contact

U.S. Corporate and Government Sales
1-800-382-3419
corpsales@pearsontechgroup.com

For sales outside of the U.S., please contact

International Sales
international@pearson.com

Associate Publisher
Greg Wiegand

Acquisitions Editor
Loretta Yates

Development Editor
Kevin Howard

Technical Editor
Branden Hall

Managing Editor
Kristy Hart

Project Editor
Jovana San Nicolas-Shirley

Copy Editor
Water Crest Publishing

Indexer
Erika Millen

Proofreader
Kathy Ruiz

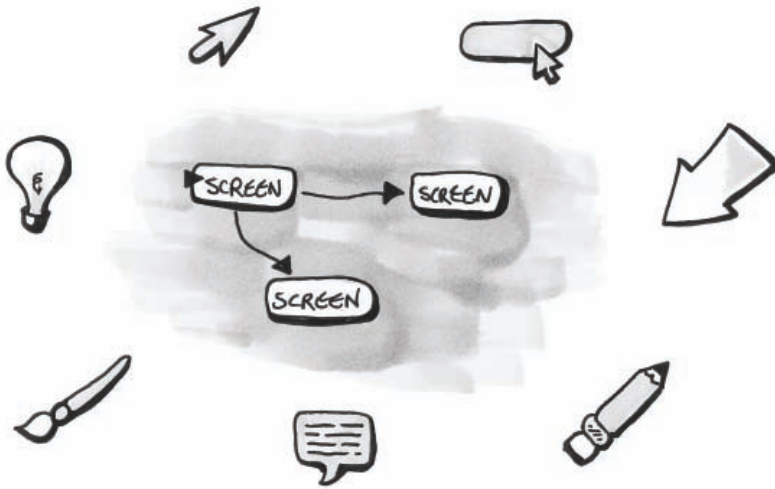
Publishing Coordinator
Cindy Teeters

Interior Designers
Sara Summers
Chris Bernard

Cover Designer
Sara Summers

Cover Coordinator
Ann Jones

Compositor
Gloria Schurick



INTRODUCTION

This section introduces you quickly to the concepts and differences between sketching and prototyping and gives an overview of everything Dynamic Prototyping with Expression Blend covers. It also gives you an explanation of how different sections of the book are arranged.

What's a prototype? This seemingly simple question can have many answers. Throw the word dynamic in there, and things can get even more confusing. If you're contemplating purchasing this book or have already taken the plunge, you might be doing so because you're frustrated with the status quo when it comes to prototyping for interaction—or you may even be uncertain about why sketching and prototyping is something that should become part of your design process.

Bill Buxton has one of our favorite definitions of how we create and design experiences—or rather how we should create and design experiences. It's illustrated by something Bill defined as the dynamics of the design funnel, and he's been gracious enough to let us recreate it for you here.

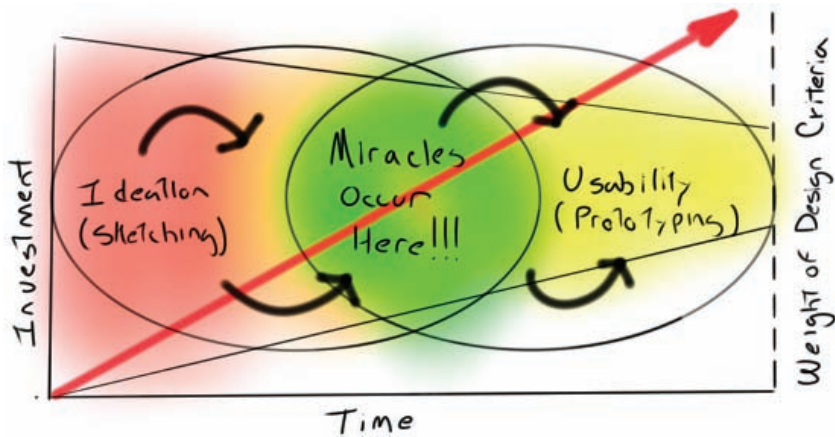


Figure 0.1 The dynamics of the design funnel shows us the continuum of how sketching and prototypes help us make great experiences. Idea and figure courtesy of Bill Buxton with a few modifications.

The design funnel illustrates two disciplines and two concepts. The disciplines are sketching and prototyping, and the concepts are ideation and usability. Sketching is valuable for idea generation because it's fast and enables ideas to be explored quickly and cheaply. More refined designs can be tested at later stages of a project by building prototypes to test the usability of the ideas that we've decided to focus more of our attention on—and this is how we define a prototype in this book—as a tool to help test the usability of an idea with design criteria. The further we get into the world of prototyping, the more invested we become in process, and the more focused our design becomes; this is represented by the funnel in our illustration. As we transition from ideation to usability, the weight of the design criteria we use for our project takes on more importance, and it requires a bigger investment in time and money. This is represented by the arrow in the graphic. Finally, this is not a linear progression; it's iterative, and the circles represent the iterative nature of this process. It also means our criteria and rigor isn't constant in a smart and innovative design process; we want less rigor, not more, in the ideation process; as we move deeper into a project, we want to slowly but certainly ensure that we've got consistent, vigorous, and valid criteria to evaluate our prototypes.

Now if sketching weren't such an important and critical part of prototyping, then perhaps we'd be working with PrototypeFlow instead of SketchFlow in Expression Blend. But the reality is the sketching and prototyping are

different activities that we do at different stages, and great prototypes always start with great sketches. SketchFlow in Expression Blend is designed to let you engage and leverage both activities, and it helps us get to when the miracles occur in good experience design—which we feel is in that intersection of sketching and prototyping.

Many of the existing processes that we use for prototyping were never created for the world of agile design and development or the nuanced world of interaction design. Today, the entire soul of a product or service may hinge on the inherent quality of an interaction, how fast parts of an application load on a device, the gentle bounce of an icon after you've selected it, or the physics or fluidity of an item as you manipulate with your fingers, mouse, stylus, or other input device.

In software design, what we're really describing is the transitions that take place between the different states of an application. In the applications that bring us joy and are simply fun to use, it's the transitions, or the journey, we take between states in an application that brings those experiences to life.

Many creative disciplines have well-established processes and tools that enable us to do our jobs. But for those of us resigned to showing the dynamic nature of what we build in software—especially when it comes to transitions—we are often stuck using the hand-me-downs of our peers. We think SketchFlow in Expression Blend is a killer application for quickly and easily defining, developing, showing, and sharing the transitions that make for a great experience.

By incorporating sketching and prototyping into our design and development process—which we simply call dynamic prototyping—we are able to use sketches and prototypes to determine not just what *to* build but what *not* to build. We are able to iterate in a consequence-free environment and refine and identify the details of interaction design to make products and services that are competitive and differentiated. We get to define and show the states of an application, but we also get to provide equal and perhaps more attention to the transitions. We can do this currently with a variety of tools, but SketchFlow for Expression Blend represents one of the first tools that simplifies how we do this today, while still enabling the designer to play an active role in designing and showcasing dynamic interactions without resorting to code.

WHO SHOULD READ THIS BOOK?

This book will be useful to anyone who wants to get a thorough understanding of how to create dynamic prototypes with SketchFlow in Expression Blend and learn the basics of dynamic prototyping.

SketchFlow functionality was introduced in Expression Blend 3. If you're using a later version of Expression Blend, some of the detailed information provided in this book may no longer be current. However, the fundamentals of understanding SketchFlow and Expression Blend shouldn't change, and we'll provide free updated content at www.dynamic-prototyping.com to cover breaking changes

The target audience for this book is professional designers, tasked with creating interactive and dynamic prototypes without having to rely on a developer or knowing code to bring their solutions to life. Our approach presumes that you want to leverage existing design tools that you currently use and are familiar with, but we also encourage you to focus on simplicity—you'll start the design process with pen, paper, or, perhaps, digital tools that mimic the act of sketching. No matter what your day-to-day role is, or even if you have more than one of the following, you'll find this book a valuable resource.

Visual Designer, Information Designer, Information Architect, or Interaction Designer

This book should strengthen your familiarity and competency with dynamic prototyping and the functionality enabled by SketchFlow in Expression Blend, even with no prior knowledge of the tool. This book also serves as a great foundation for learning more about the full functionality of Expression Blend.

Design Planner or Researcher

You may find that SketchFlow in Expression Blend is a useful tool for getting ideas off of a whiteboard and into the hands of clients for iteration and feedback when dealing with remote or displaced teams.

Motion Graphic Designer or Front-End Technology Specialist

If you frequently use dedicated prototyping tools to collaborate, you may find that SketchFlow in Expression Blend is a powerful new tool in your arsenal—especially if your existing tools require a great deal of scripting and coding to develop dynamic prototypes.

Usability Specialist or Design Strategist

You may find that SketchFlow in Expression Blend is a great tool for testing low-fidelity concepts with potential users of applications and services or as a way to present ideas and concepts to key stakeholders.

Software Architect or Developer

You may find that you are often thrust into playing the role of a designer in your projects and day-to-day activities, you'll find that the techniques put forth here will enhance your ability to do better work. This book shouldn't, however, be regarded as a replacement for having the capability and skill of a professional designer on your project team.

SketchFlow for Expression Blend is easy to learn and can be used to create dynamic prototypes for a variety of mediums, from design to business problems. It's a subset of the very powerful, but more complex, functionality of Expression Blend—a tool designed to create Rich Internet Applications (RIAs) for Microsoft's .NET platform.

This book is focused on the tasks that are essential to effective prototyping using SketchFlow in Expression Blend. The reference section of this book provides a charted course for deeper understanding and continued learning with Expression Blend. Dynamic Prototyping with SketchFlow in Expression Blend should be regarded as a complement to books that focus on Expression Blend in its entirety. This book is designed to give you a deep and comprehensive understanding of not just how to use SketchFlow, but also to help you understand how to bring core concepts around sketching, ideation, and concept exploration to your design process—or how to simply add a design process to your current workflow if you don't have one. Consider Dynamic Prototyping with SketchFlow in Expression Blend to be a hybrid of books that focus on design process agnostic of technology and software instruction manuals that tell you the how and not the why of how things work.

WHAT'S IN THIS BOOK?

Expression Blend is a professional design tool that is really a WYSIWYG (What You See Is What You Get) tool for creating applications that run using Windows Presentation Foundation (applications that run on Windows PCs), commonly known as WPF applications and Silverlight applications (cross-platform applications that can run in a browser or out of a browser on different platforms, including Windows, Macintosh, and Linux platforms). Both WPF and Silverlight are part of the .NET platform. The power of the .NET platform is that it lets developers use a similar set of skills and tools to write applications for PCs, for the Web, and even for devices. Tools like Expression Blend extend this capability to designers who work on products and services designed to take advantage of the .NET platform.

Expression Blend enables you to work with video, text, 2D animation, 3D animation, bitmap images, vector art, audio, advanced typographic and printing functions, data, and code to create these types of applications.

This book will focus on are the features that are required to enable you to be successful in SketchFlow.

This is important because SketchFlow itself is a tool that enables designers to quickly and easily start using Expression Blend and to be productive without a complicated and demanding learning curve. In fact, as designers ourselves, we think SketchFlow is the best way for designers to *start* using Expression Blend. It's important to note, however, that even if you don't want to create applications that take advantage of the .NET platform, SketchFlow in Expression Blend can still be a powerful tool to utilize in your design workflow.

We cover all the features of SketchFlow with which you need to be familiar, including the following:

- Design, sketching, and prototyping **best practices** and **design patterns**.
- Understanding the Expression Blend **interface** and getting your **workspace** set up to work with SketchFlow.
- How to **import** freehand **sketches** into Expression Blend and bring them to life with SketchFlow.

- Creating **navigation flows** that show the **screens** and **interactions** in your application.
- How to refine your application with SketchFlow **styles** that enable you to build real **controls** and functionality into your prototype.
- How to create **reusable assets** that save you time and energy in your projects and professional practice.
- How to use **animation**, **navigation**, **states**, and **behaviors** to add complex interactivity in your applications.
- How to integrate real or simulated **data** sources to showcase complex functionality.
- How to **annotate** your work and package your dynamic prototypes and **distribute** them so stakeholders can provide **feedback**.
- How to create professional design **documentation** automatically with SketchFlow that you can pass on to stakeholders and colleagues in projects.

We've also included a thorough, but optional, reference section you can learn in more detail about some of the key technologies behind Expression Blend, including sections on the following:

- The .NET framework
- Extensible Markup Application Language (XAML)
- The C# programming language
- Windows Presentation Foundation (WPF)
- Silverlight

HOW TO USE THIS BOOK

Every individual has different learning styles. There are four main parts to this book and they can be reviewed in or out of sequence depending on your learning style.

PART I

Part I introduces the fundamentals of the design process that use sketching and prototyping techniques. We also discuss patterns and practices for dynamic prototyping that we've found particularly effective with SketchFlow in Expression Blend. If you need a basic overview of design process, thinking, and practice and how dynamic prototyping enables it before diving into the basics of the tool, start here.

PART II

Part II shows you how to use SketchFlow in Expression Blend to quickly build a prototype. You'll find a basic overview of how to be productive in SketchFlow that can be completed in less than a few hours. If you're the type who doesn't like to read the manual but are new to Expression Blend, start here. We'll introduce all the key concepts of Expression Blend's interface and nomenclature here in great detail so it will be easier for you to navigate about Expression Blend in other sections and chapters.

PART III

Part III takes a deep dive into the functionality and workflow of SketchFlow in Expression Blend. If you're someone who enjoys learning every nook and cranny of a tool before you're comfortable, we'll spill the beans on everything, including the following:

- The SketchFlow interface
- Assets, styles, and components
- SketchFlow animation and states
- SketchFlow behaviors
- Data
- Annotations, feedback, and documentation

PART IV

In Part IV, we focus on applied knowledge. If parts I through III are a bootcamp Part IV is where we drop you off in the forest and you put to work everything you've been taught. If you're the type who learns best in a tutorial-style of environment, this is the best place to start. We take you

though a dynamic prototyping project, starting with a creative brief and all the artifacts we'd typically inherit in a design-focused project; we then create an actual dynamic prototype and even discuss how to begin the detailed design process using Expression Blend.

REFERENCE

In the Reference section, we go deeper into the technology that makes all this happen, including .NET, XAML, C# WPF, and Silverlight. Finally, we provide curated resource lists and recommendations on how to continue learning about Expression Blend, prototyping and design, and WPF and Silverlight.

FAQ

Dynamic prototypes? Sketching and wireframing with my current tools works just fine for me. Why do I need this?

In the world of rich interactive applications, if a picture is worth a thousand words, then an interaction is worth a million. For example, imagine having to create a document that would describe the interface of the iPhone, the Xbox, or TiVo without interactivity, gestures, or motion. Our current static processes, ignore the transitions that must exist in our software and interactive experiences. It's far easier and more effective if we can actually design and show these transitions as part of our design process. We can do that with a variety of tools today, but it can often be labor intensive and time consuming. Should it really take five or six different tools to create a simple prototype? Wouldn't it be nice if we could take our sketches and bring them to life?

The moment design of a product begins, the collection of design documentation becomes outdated and often unmanageable unless documentation is constantly updated with each progression and iteration of the project. Big, complex documents are often difficult for stakeholders to review and understand. Sometimes what you document and think you explain clearly really isn't clear, leading to a miscommunication with the client, which often isn't discovered until much later in the project when changes are costly.

You shouldn't abandon techniques and processes that work for you, but we think you'll find that SketchFlow in Expression Blend offers a faster and

easier way to demonstrate interactivity and create documentation that can be used in creative briefs, project specifications, and design documents.

If you're constantly being asked to do more, but still maintain high quality at an even faster pace, tools like SketchFlow for Expression Blend can be very useful in your design process.

I'm a designer, not a developer. Won't SketchFlow in Expression Blend be too difficult for me to learn?

A good ski instructor can teach someone to ski in a day; this doesn't mean you'll become a master of the carve, skiing down the most difficult runs of the mountain by lunch, but you should at least be able to have fun. SketchFlow for Expression Blend makes Expression Blend fun and accessible to less technical, or even non-technical designers. As a tool it gives you power that typically is only available to those that spend a great deal of time learning how to code.

Most designers can learn to be productive in SketchFlow in a matter of hours. You can gain confidence in the powerful features in SketchFlow for Expression Blend within a few days without bothering with code.

Once you learn how much fun SketchFlow in Expression Blend can be, you may find that you want to delve deeper into the considerable capabilities of Expression Blend. This book is a great way to start. We provide plenty of resources and links should you choose to learn some of the features beyond the capabilities of SketchFlow, but these are optional things that are not required knowledge for using SketchFlow.

However, we strongly encourage any designer to endeavor to understand the elements of what they are working with. If we look at classic artistic and design movements like Bauhaus or even the discipline of architecture, there is often an expectation around understanding the materials that you work with (textiles, photography, concrete, or steel) that is often missing from the interaction design process. You'll find that you're perhaps a better designer if you begin to learn the underlying characteristics of the medium. You don't *need* to do that to use SketchFlow, but we suspect you'll be a better designer and developer as you make those efforts over the course of your career.

My team already has a number of prototyping tools. Why should I learn one more?

If you've spent years learning the ins and outs of prototyping tools and come from a world of Adobe Flash and/or Flex, you should find the learning curve on even advanced functions in Expression Blend to come to you very quickly. In fact, you'll probably discover that you possess about 90% of the skills and capabilities to start learning how to be productive in using Expression Blend if you've been working with Flash, Flex, and ActionScript 3.0.

If you require an application that easily lets you work with sketches in a dynamic fashion, you'll find that SketchFlow does things that other tools currently do not. This includes using traditional animation techniques to create and explore transitions, adding interactivity and linking without code, and creating documentation and sharing feedback—all within one tool.

If you regularly work with teams that create solutions that ultimately target WPF or Silverlight, you'll see that Expression Blend enables a powerful workflow that makes designers first-class citizens in the design process, which allows for round-trip design and iteration that simply isn't available on other platforms. (You can learn how this is possible in the Reference section of this book, which discusses XAML.)

I don't use a PC for my daily work. Can I still use SketchFlow for Expression Blend on a Mac?

Microsoft Expression Tools only run on Windows at the moment. Although this may seem odd, there are some simple reasons for why this is currently the case. Fortunately, there are some easy workarounds for those who find SketchFlow for Expression Blend to be a killer application in their designer toolbox.

Microsoft Expression Tools, most commonly called Expression Studio, were actually designed using the very technology they are designed to help you create (WPF and Silverlight applications). For example, Expression Blend is a WPF application, and so are many parts of Expression Design. By using WPF, Microsoft is able to create new features and versions of the Expression Studio suite far more quickly—in fact, Version 3 of Expression Studio is the third revision of the software in just slightly over two years.

This level of speed comes at a price; it's nearly impossible to move that quickly and support multiple platforms.

In the professional design tools space, this is less of an issue than it was even a few years ago, as most computers can easily run a variety of operating systems natively or through a function called virtualization.

For example, most current users of Apple computers can run Windows on their computers just as effectively as they can run OSX. You can either do this using the Bootcamp feature of Intel-based Apple Computers to run Windows natively on your computer, or you can explore using special virtualization software that enables you to run and use Windows and Windows applications within OSX.

The bottom line? If SketchFlow for Expression Blend makes sense in your design workflow, the barriers to incorporating it are modest, regardless of your primary design operating system.

One feature of SketchFlow in Expression Blend is the ability to preserve assets into production—by doing that, my focus is more on the end product than a prototype. Are you sure this is a tool for prototyping?

One of the most valuable things a designer can do with a prototype is throw it away. The more effort and care put into the prototype, the more likely that a team can become blind to continued exploration or the marriage to bad ideas.

SketchFlow in Expression Blend is designed first and foremost to enable you to leverage the power of sketching and showcase interactivity using animation—perhaps the most effective way to demonstrate complex interactions without writing code—currently a popular method of prototyping among many designers.

But sometimes, our prototypes are not just about concept exploration; sometimes we need to know if something is even possible. This is typically an iterative step in prototyping. In most traditional workflows, we leave the low-fidelity world of sketching and move into higher-fidelity methods because we have to. SketchFlow for Expression Blend lets you seamlessly embrace this evolution and work with real controls, data, behaviors, and (if needed) code to show the most complicated interactive concepts.

SketchFlow can actually make these complex controls look like the same thing you'd sketch on a piece of paper. This is important because it can keep stakeholders focused on what's important—the idea, versus the details. In this stage of the creative process, this is a crucial area of focus.

For projects whose ultimate destiny is WPF or Silverlight applications, SketchFlow projects and the documents they create become valuable resources that can dramatically accelerate production. The analogy that comes to mind is how you might create an illustration for a comic book: Use a pencil to rough in an image, use an inker to build on those initial sketches, and finally use a colorist to add additional detail. Coupled with the copy, a comic book slowly comes to life. SketchFlow for Expression Blend elevates the control that a designer can have in the final result when creating an interactive application.

We won't promise, nor would we recommend, that you take assets and code created for a dynamic prototype wholesale with you into production; it takes away from what the core process of sketching and prototyping should be, iterating quickly and without consequence. But we suspect that those who work with WPF and Silverlight will find that many of the assets and components used in dynamic prototyping can be used to massively accelerate the production process.

SUMMARY

SketchFlow in Expression Blend is a game-changing tool for creating dynamic prototypes for:

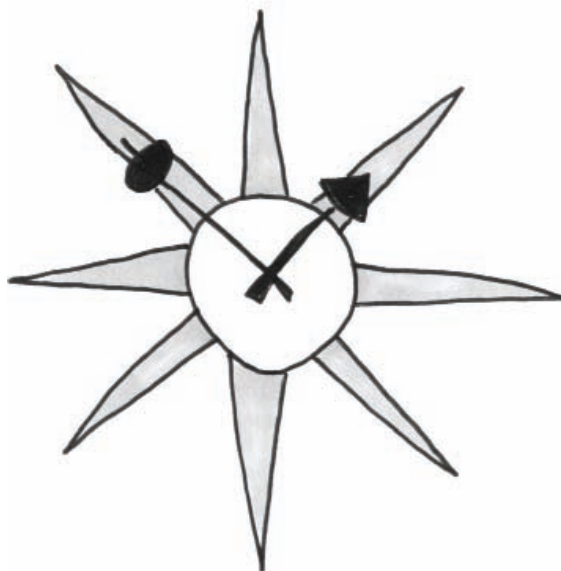
- Validating design research and insights.
- Sharing brainstorming concepts and rough sketches.
- Beginning concept exploration.
- Bringing hand-drawn sketches or digitally created sketches to life in dynamic prototypes.
- Generating consensus and feedback.
- Creating design specification documentation.

Regardless of the final form factor of your output (service, application, product, business idea), SketchFlow in Expression Blend can help you get there—quickly, iteratively, and without code.

SketchFlow in Expression Blend has added benefits for those who are using it to design software or applications that take advantage of WPF or Silverlight, including the following:

- Dozens of controls that can be skinned and linked to simulated data early in the design process with SketchStyles (low-fidelity themed controls that work like a real application control, but look like a sketch) that later can be converted to production assets quickly and easily.
- Great workflow that allows simple importation and manipulation of high-fidelity artwork created in today's most popular design tools.
- Deep integration into professional asset management and application lifecycle management workflows that make the design process a first-class citizen in the development process

So, enough talk. Let's go learn about sketching, prototyping, and SketchFlow!



2

DESIGN PROCESS

Workflow, process, patterns, and methodology are simply a way of life; we typically take advantage of some formalized practice to aid the projects to which we toil and the resulting products we launch. This is especially true of projects or endeavors that involve technology. And, if technology would stay put for a minute, we could reuse the patterns from the last successful project—alas, technology is a constant of change. How do we keep up? Pick the "right" methods? Learn and adapt?

As a rule, software systems do not work well until they have been used and have failed repeatedly, in real applications.

—David L. Parnas, Ph.D.

For several years, the tech industry has become fascinated with failure; stories of failure, learning from failure, and celebrating failure. This is not at all coincidence. As technologists, we are bound to an eternity of failed attempts, in part because of the nature of business and management, the current software used that can't keep up with what we are tasked to do, but more importantly, the complexity of what we are required to create has exponentially increased. Humanized interfaces, gestural touch

technology, and intuitive, predictive, user experiences; this is cause for revolution in every facet of how we think of and execute on ideas.

So now what? How can we expect to change what has historically been a frustrating, pride-swallowing battle? A journey back in time, to how other industries successfully built products in a time of disruptive change, is a great start. The industrial innovation surrounding the mass production of cars, the advent of advanced innovation in architecture practice, and the engineering books that document the techniques used are all ripe for exploration.

It may sound odd to seek-out groundbreaking design processes from a nearly 20-year-old engineering reference but we can learn quite a bit from *The Mechanical Design Process* by David Ullman.¹ Design process for engineers has been explored, tested, and documented in vast detail, because of the high overhead—production cost, expense of manufacturing, and considerable time to take product to market. The following are a few powerful examples of the engineering process that have been adapted to both the design and software engineering process. Most importantly, Ullman discusses why the prototyping process is so critical to our work. To wit:

- Whenever possible, organize the talent around the project.
- Exact measurements or outcomes are not important when prototyping a proof of concept; it's a learning tool.
- Build it twice: once to fail and second to succeed.
- Don't plan for a set number of prototypes. Take what is learned from the first prototype and apply to the second. Rinse, repeat.
- The more complex the function of the product, the longer design prototyping will take.
- Reduce the problem you are trying to solve with the product into one inclusive statement.
- Every phase of product design is iterative; requirements included!
- The human requirements ARE functional specs.

1 *The Mechanical Design Process* by David Ullman

At its most basic level, we can describe the design process as it relates to sketching and prototyping with the following model.

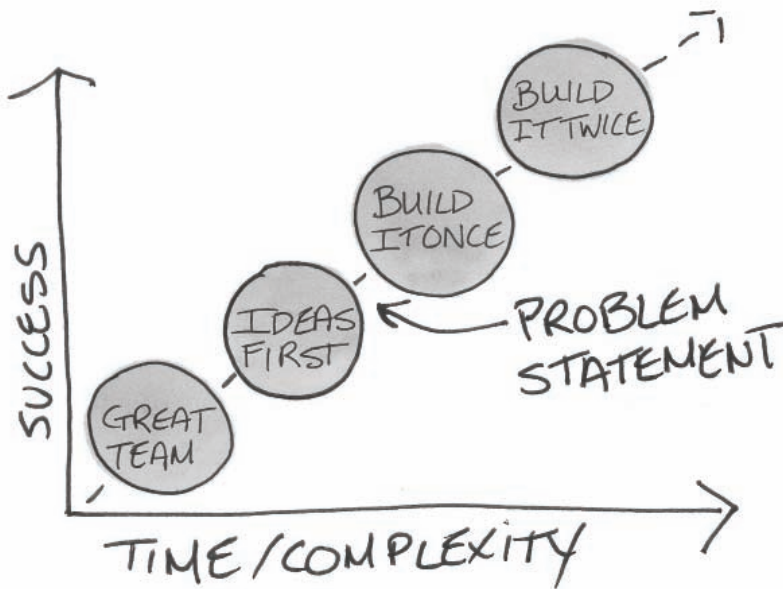


Figure 2.1 The preceding lessons simplified in a bubble chart.

But the act and discipline of design is more nuanced, and it's about far more than aesthetics and surface elements of a given solution or artifacts—in fact, what many folks outside the profession of design consider design to be is only scratching the surface.

Creating the form or appearance of an object or artifact is certainly a part of the design process, and it's often referred to as graphic design, visual design, or communications design. Because it's the most visible and tangible output of design, it perhaps pushes aside some of the other realms of design that are also critical.

Fundamentally, design process is about developing and communicating insights. There are two major influences that we think are important when it comes to the idea of sketching and prototyping.

One is focused on the work of Roger Martin and his thesis around how organizations need to seek validity versus reliability.² Martin's work

² *The Design of Business: Why Design Thinking Is the Next Competitive Advantage* by Roger Martin, p 44.

resonates specifically around our need to balance incremental innovation through refinement of existing knowledge with the need to enable breakthrough innovations—essentially the difference between a version 2.0 product or service versus something nobody has seen before, or thought they needed.

Typically, many organizations focus on developing evidence for future product needs based on past outcomes. They use a limited number of objective variables to remove judgment and bias from decisions to support innovation, along the lines of “Our customers told us they want this” or “This is what everyone else is doing, and they are successful.” Martin characterized this type of *substantiation* as reliability (looking into the past to make an informed judgment about the future). Designers are often called upon to focus on *substantiation* based on future events. This means that they use a broad number of diverse variables. Using processes that integrate judgment and that acknowledge the reality of bias is what design processes are used for. They are needed because most organizations make decisions based on facts, or what Martin called *reliability*. Design operates more in the mode of validity, often asking people to take a leap of faith, and the insights and recommendations can’t always withstand the scrutiny of reliability.

At this point, you may be thinking that it’s because those insights may not matter if they can’t be backed up with reliability, but the reality is that we make decisions based on validity every day: It influences the cars we buy, the people we choose to spend our time with, and the type of phone we buy. Tapping into the gestalt that motivates those decisions is what design thinking and design processes are engineered to facilitate.

Martin characterized the difference in skill sets between designers and business decision makers and their propensities via something he called the predilection gap.³ Organizations that are successful at innovation often have folks that are grounded in theories and practice that value both reliability and validity. Organizations that skew toward reliability often have difficulty innovating or recognizing good ideas. Organizations that skew toward validity often have a hard time getting their good ideas into the marketplace. It’s actually possible to be good at both of these things and not have folks that can serve as translators between these two schools of thoughts as well. The result is often paralysis in an organization.

3 *The Design of Business: Why Design Thinking Is the Next Competitive Advantage* by Roger Martin, p 54.

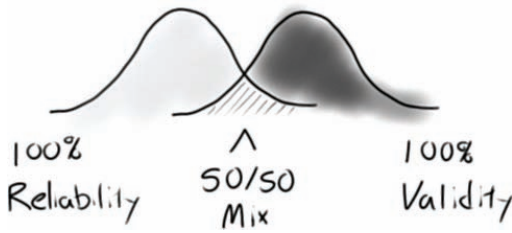


Figure 2.2 A sketch that illustrates Martin's theories.

(Printed with permission from Roger Martin, author of *The Design of Business: Why Design Thinking Is the Next Competitive Advantage*, Boston: Harvard Business Press, 2009.)

So, if we can acknowledge that design can be more than just about aesthetics and form and that validity is an important component of innovation, what's the process that designers use to enable success?

Although there are multiple books on this subject, and some of the names may change, the typical process that designers go through is fairly straightforward. If you ever hired a product design or innovation firm, you would most likely be exposed to a process such as this—or it would at least be the process applied to solving your problem. If you were to explore the techniques being applied in academia around design education and strategy, you would also find that these are the techniques that many designers are being exposed to in graduate and undergraduate education when it comes to learning about “design thinking.” Here we'll use some examples from one of the author's previous work under Professor Vijay Kumar at the Institute of Design at the Illinois Institute of Technology.

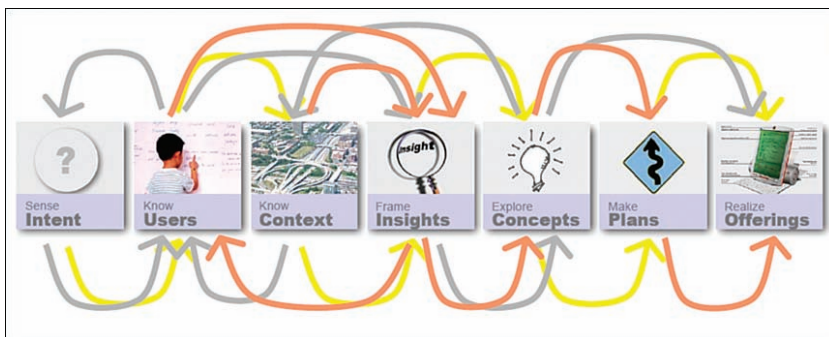


Figure 2.3 An example of the steps and non-linear nature of a design process taught at the Institute of Design by Professor Vijay Kumar. (Image printed with permission from Vijay Kumar.)

A TYPICAL DESIGN PROCESS

Design processes are about a journey of discovery. There are numerous techniques that can be applied to each stage of the design process, and each stage is designed to build on the other. Design-driven innovation is different from business- and technology-driven innovation in that it starts with a focus on understanding your customers or users first versus starting with a technology or a business.

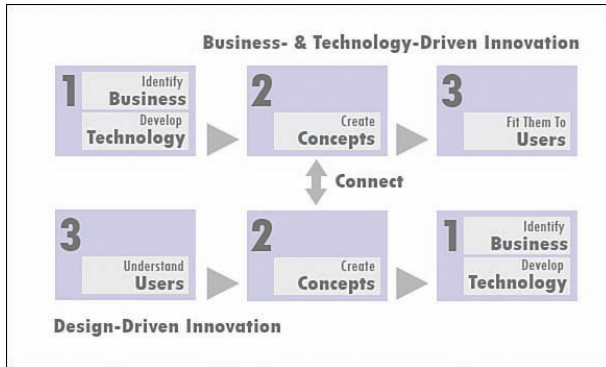


Figure 2.4 The journey that occurs in a design process around design innovation and how it differs from business and technology innovation. (Image printed with permission from Vijay Kumar.)

Most design processes contain the following steps. For large projects, it can take weeks or months to go through all of these phases, but talented design planning teams can also accomplish these steps in a matter of days for focused efforts. Here's an overview of what those processes look like when we follow the techniques and practices that the Intitute of Design employs.

DEFINITION AND INTENT

Design process starts with an idea or, more often, a hypothesis—something you are trying to prove or disprove. This definition state is where designers state their intent, or where the business process is framed and where a research plan is hatched.

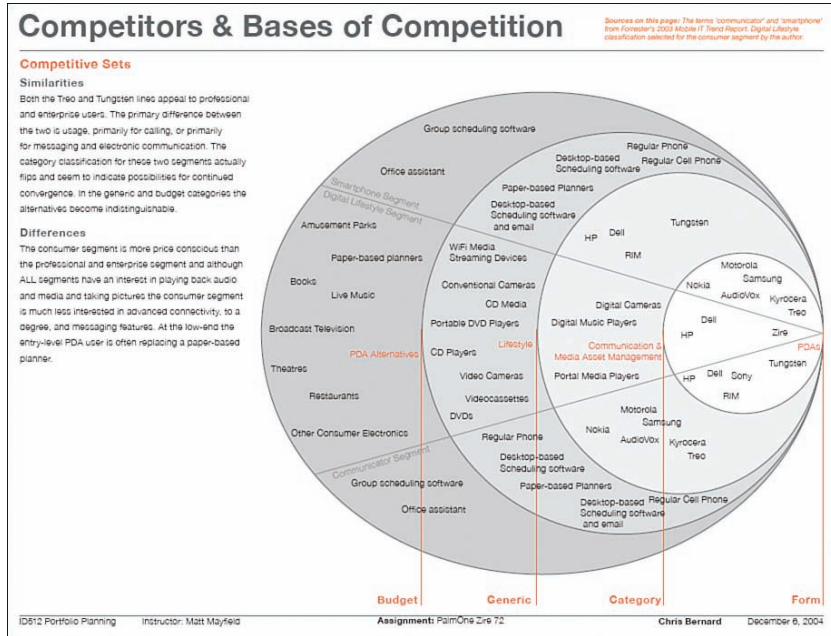


Figure 2.5 Some definition efforts begin with an attempt to capture a snapshot of a landscape that influences a problem space. This is an example of a product mapping completed for a Portfolio Planning Class at the Institute of Design.

RESEARCH

The second stage of a design process is focused on research; it can take many forms, but one key difference from the typical “stakeholder” interviews and secondary research that we might be familiar with is that this research typically takes the form of what is often called contextual or ethnographic research. Direct observation and anthropological techniques are often used.

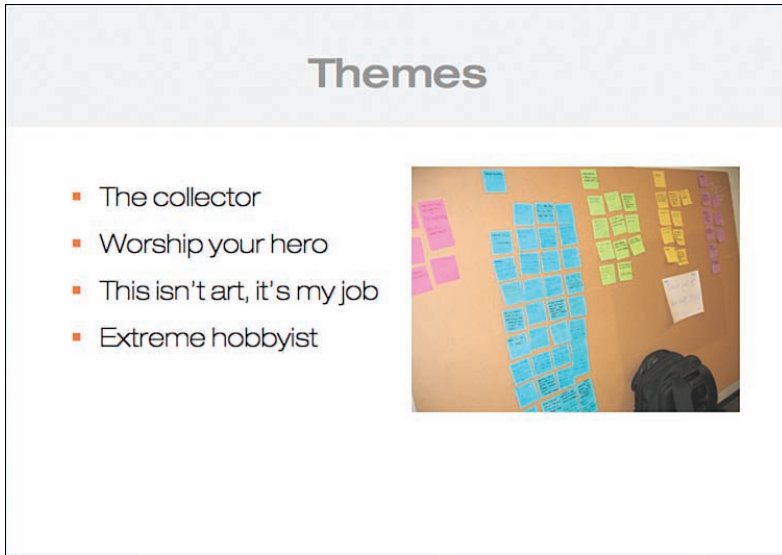


Figure 2.6 Insights or observations from contextual research are often clustered or cataloged at collection or soon after. This is an example from an Understanding Users Class at the Institute of Design.

ANALYSIS

The next stage is really one of analysis: processing and organizing all the data that you've collected. Often one of the biggest challenges for designers is not coming up with ideas, or collecting them, but figuring out which ones are the most important to pursue.

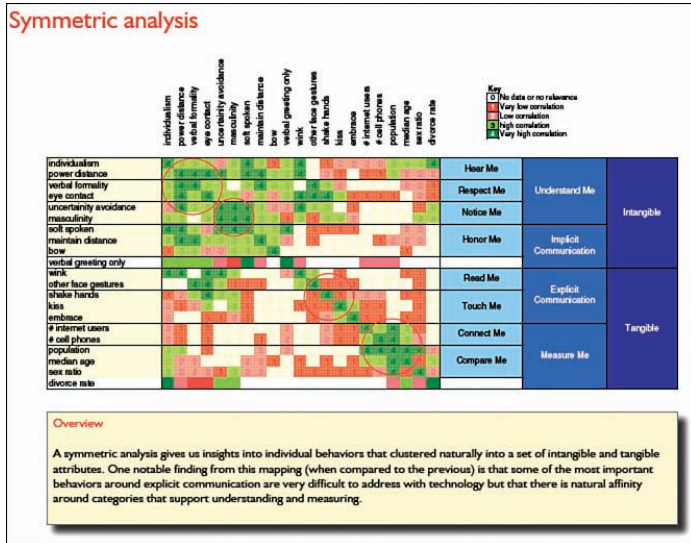
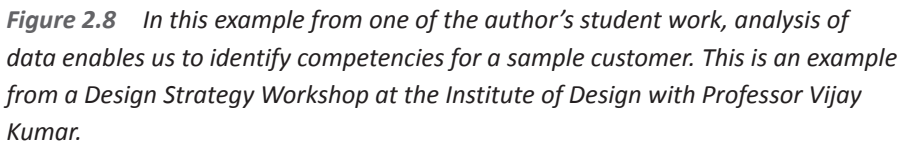


Figure 2.7 Even designers use spreadsheets from time to time; here we see a morphological sort of data and attributes derived from primary research. This is an example from a Design Analysis Class at the Institute of Design with Professor Vijay Kumar.

This is at the point where most people think the design process begins, and that is around the phase of synthesis or ideation. This is the point in the process where we start to develop and flesh out some of the themes and memes that come out of our analysis process. It is also the phase where tools like SketchFlow can start to become useful.



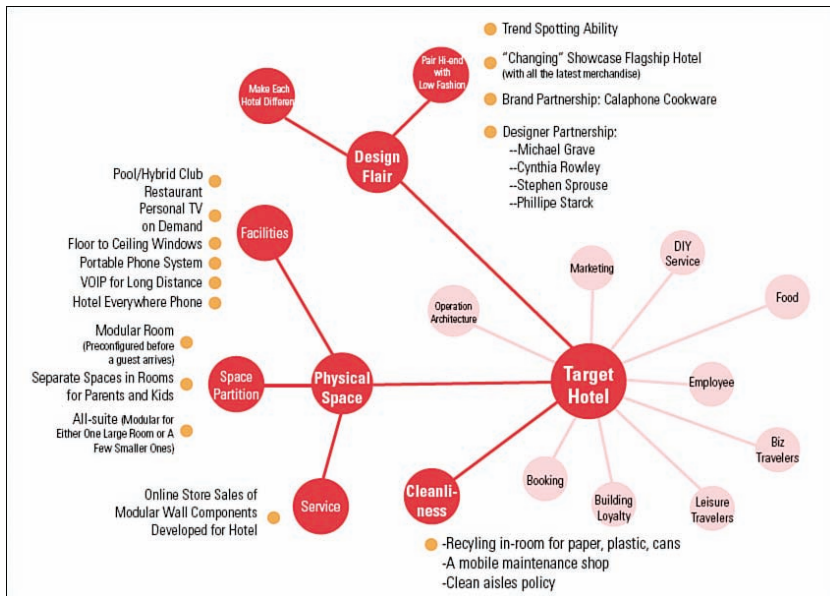


Figure 2.9 Affinity mapping helps designers and business makers uncover new opportunities. This is an example from a Design Strategy Workshop at the Institute of Design with Professor Vijay Kumar.

By looking at competitor offerings and matching them with a different company, teams can uncover new business opportunities. Here is an example of student work of one of the authors; it was identifying competencies and needs that would allow for a new business to be created, which would deliver new value to an existing customer segment.

PLANNING

Have you even been frustrated when you ask someone for an estimate and a plan and their first response is, "It depends." We often roll our eyes at consultants and companies that suggest a six-week project to figure out what the *real* price of a project may be. But the reality is that for complex problems that require breakthrough innovation, if *someone* hasn't gone through a process of research, analysis, synthesis, and ideation, the estimate you receive will be useless or wildly inaccurate. Another way to think about this is that solving hard problems is, well, hard. If it were easy, we wouldn't need to go through all these steps, and we would always make smart choices all the time.

CONCEPTUALIZATION AND PROTOTYPING

Many traditional processes have this phase, and it's often called a macro or high-level design process. It's often the stage where we plan out a slice of a project or develop multiple initial concepts to help us drive toward our final decision. For many people who don't use a fully realized design process, this is often where SketchFlow might first be used.

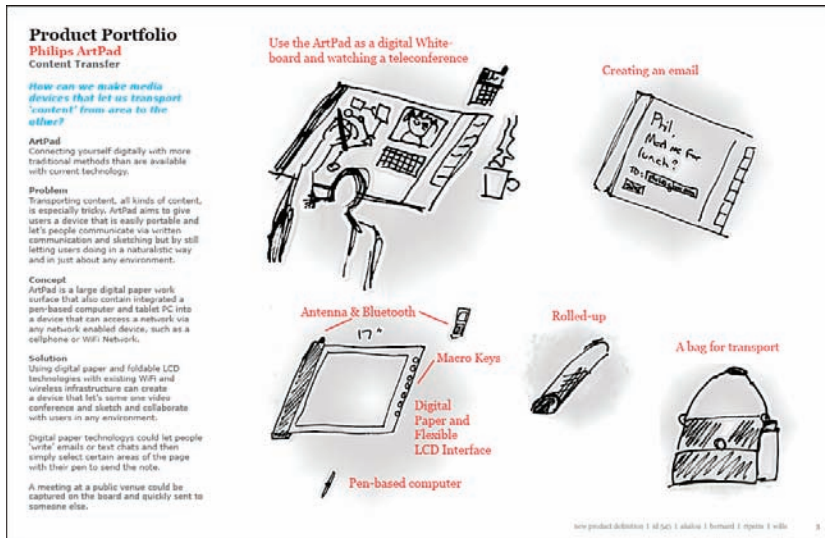


Figure 2.10 In this example from one of the author's student work, we see an encapsulated design process that features intent, a problem statement, conceptualization, and a potential solution. This is an example from a New Product Definition Class at the Institute of Design with Professor Chris Conley.

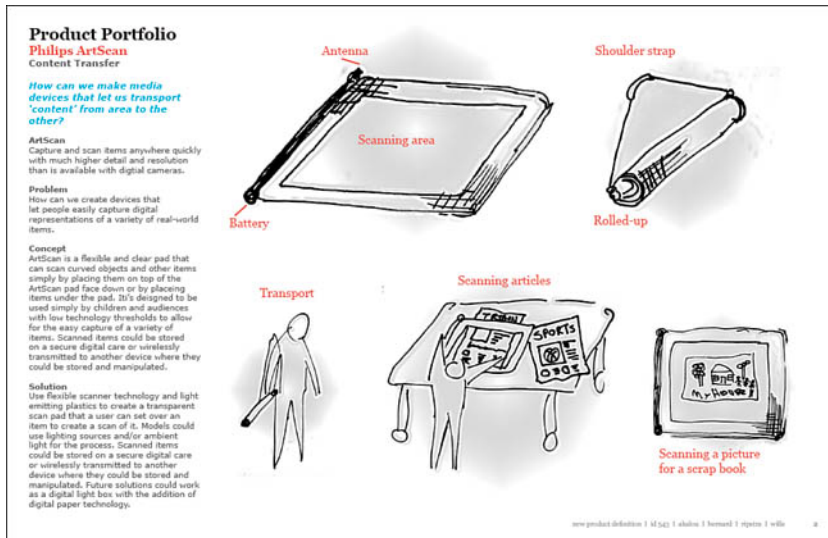


Figure 2.11 Another example of the same encapsulated process. This is an example from a New Product Definition Class at the Institute of Design with Professor Chris Conley.

PRODUCTION AND IMPLEMENTATION

This phase is often known as a micro or detailed design process. In many cases, a designer might not even have been engaged until this stage of the process with a mandate to “Make it pretty” or “Make sure our customers love this”—which may be an impossible task by this point. SketchFlow and technologies like WPF and Silverlight offer a powerful combination when using SketchFlow and a design process because they allow a designer's work and vision to translate seamlessly into the development process.

SENSING AND FEEDBACK

Great products are never really finished until they are abandoned. New releases, features, and innovations follow the lifetime of a product or service. Good design processes acknowledge this and ensure that a mechanism is in place via process or technology to capture, judge, and act on feedback and insights that are collected over the lifetime of product. Just as SketchFlow is a great tool to use at the beginning of a project, its utility can also be realized to support shipping projects and services that need to evolve.

WHAT ABOUT AGILE PROCESSES?

Increasingly, design processes are being combined with agile methods. There are two primary ways to accomplish this. The most common way is to incorporate design processes in each sprint or release of a product. The most valuable way—and some designers might argue the *only* way—is to incorporate design processes in an initial phase zero that takes a long-term and holistic view of project needs before moving into agile processes.

FURTHER READING

The Reference section of this book provides recommendations on other authors' books and work to reference if you'd like to learn more about design processes. Employing design processes into your existing workflow is not a trivial task; in fact, you might not even be convinced that you should at this point. Although it's beyond the scope of this book to be exhaustive in articulating the value of a formal design process, we'll discuss some basic patterns and practices in the next chapter that you can incorporate into your work.

INDEX

A

***About Face: The Essentials of Interaction Design* (Cooper), 473**

accessing

Assets panel, 214

SketchStyles, 219-220

actions, 249-251

ActivateStateAction behavior, 252

Adaptive Path, 479

**Add Existing Item option
(Project panel), 81**

**Add New Folder option (Project
panel), 81**

**Add New Item option (Project
panel), 81**

**Add sample data source icon
(Data panel), 153**

Add State button, 332-334

Add State Group icon, 332

agile processes, 40

analysis (design process), 35

animation

adding to SneakerSpot
project, 311-313

building project, 317

*Easing Function
transitions, 315*

*Properties pane
adjustments, 313*

*SketchFlow Animation
panel, 314*

*SketchFlow animation
progress, 316*

animation models, 226-227

creating, 119-125, 227-235

determining elements to be
animated, 110

easing functions, 232-233

explained, 109-110

frames, 120-122

guidelines, 226-227

importing Photoshop files for
animation, 111-118

keyframe animation and
storyboards, 244-246

SketchFlow Animation panel, 195
states

- adding to buttons, 237-243*
- advantages of, 235-236*
- creating, 137-144, 236*
- definition of, 135-136*
- editing, 138, 145-147, 236*
- explained, 136-137, 235-237*
- previewing, 145*
- repeating behavior for, 245*
- State panel, 138-139, 190*
- transitions between, 148, 242*
- usage tips, 149-150*

Animation category (Properties panel), 204

Annotation icon, 104, 198

annotations

- adding to projects, 169-170, 277-279*
- enabling, 169*
- viewing, 170*

Appearance category (Properties panel), 201-202

Apple operating systems, 11-12

application menus. *See* specific menus

Applications = Code + Markup: A Guide to the Windows Presentation Foundation (Petzold), 471

A Project Guide to UX Design: For User Experience Designers in the Field or in the Making (Unger and Chandler), 475

Archetype, 447

ArcListBoxStyle.xaml file, 362

ArcPanel.dll file, 362

Arrow tool, 103

Artboard

- activating annotations on, 278*
- Annotation icon, 198*
- discussed, 72*
- Effects icon, 196*
- explained, 195-197*
- Grid icon, 197*
- Grid Snapping icon, 197*
- Snapping to Snaplines icon, 197*
- Zoom icon, 196*

Asset view (Tool panel), 87

assets

Assets panel

- accessing, 214*
- Behavior Assets category, 214*
- Behaviors category, 189*
- Categories category, 190*
- Component Screens category, 214*
- Controls category, 189*
- Effects category, 189*
- explained, 188-190*
- Locations category, 190*
- Media category, 189*
- Navigation Screens category, 214*
- Project section, 188*
- SketchFlow category, 188*
- Styles category, 189*

BodyCenter-Sketch, 90
Bodyright-Sketch, 90
Button-Sketch, 90, 166
component screens
 creating, 165-168
 linking to other screens,
 168-169
 naming, 167
discussed, 86-88
explained, 213
PasswordBox-Sketch, 90
Rectangle-Sketch, 90
searching for, 87
styles
 accessing, 219-220
 control-focused SketchStyles,
 217-218
 data-focused SketchStyles, 218
 editing, 222
 explained, 215-216
 layout-focused SketchStyles,
 218
 text-focused SketchStyles,
 216-217
TextBox-Sketch, 90
TitleCenter-Sketch, 90
Assets panel, 88
 accessing, 214
 Behavior Assets category, 214
 Behaviors category, 189
 Categories category, 190
 Component Screens category, 214
 Controls category, 189

Effects category, 189
explained, 188-190
Locations category, 190
Media category, 189
Navigation Screens category, 214
Project section, 188
SketchFlow category, 188
Styles category, 189
attributes (XAML), 402
audio (Silverlight), 460, 467

B

The Back of the Napkin
(Roam), 474

BackEase function, 232

Base category, 120

Behavior Assets category (Assets panel), 214

behaviors

actions, 249-251
ActivateStateAction, 252
adding to projects, 253-258
capabilities of, 251-252
explained, 247-248
NavigateBackAction, 252
NavigateForwardAction, 253
NavigateToScreenAction, 132, 253
PlaySketchFlowAnimation, 253
setting properties of, 254-256
triggers
 explained, 249-251
 TriggerTypes, 254-255

Zoom control (SneakerSpot project), 327-330
Bounce Out and Bounciness properties, 343-344
DragAndReturnAction, 341-343
final refinements, 344-345
HideAction, 340-341
ShowAction, 339

Behaviors category (Assets panel), 189

Bernard, Chris, 478

binding data, 160-163,
260-264, 458

BodyCenter-Sketch asset, 90

Bodyright-Sketch asset, 90

Bois, Pete, 477

Bolt, Nate, 52

books

developer tool books, 470
interactive design books, 473-475
language books, 470-471
theory books, 472-473

Border container, 415-416

Border object, 184

Bounce Out property, 343-344

BounceEase function, 232

Bounciness property, 343-344

brainstorming, 43-45

Brooks, Frederick, 50

Brown, Dan, 475

Brush Transform tool, 181

Brushes category (Properties panel), 201

Build Project command (Project menu), 317

Butler, Jill, 472

Button-Sketch asset, 90, 166

buttons, adding states to, 237-243

Buxton, Bill, 1, 472

C

C#

casting, 431
classes, 423, 434-436
code-behind files
 constructors, 425-426
 namespaces, 424
 page class, 425
data types, 430
event handlers, 426-429, 440-441
events, 423-429, 440-441
explained, 419-420
frameworks, 422
inheritance, 423, 441-442
libraries, 422
methods, 423, 429
 calling, 432
 defining, 432
 explained, 431-432, 436-438
 overloading, 438
 return values, 433-434
objects, 423

- OOP (object-oriented programming), 420-422
- polymorphism, 423
- properties, 423, 429-430, 438-439
- subclasses, 423, 441-442
- variables, 430-431
- calling methods, 432**
- Canvas container, 413**
- Canvas object, 184**
- casting, 431**
- Categories category (Assets panel), 190**
- Chandler, Carolyn, 475**
- Change visual tag feature, 74-76**
- Channel 9, 476**
- child classes. *See* subclasses**
- Chinnathambi, Krupa, 477**
- CircleEase function, 232**
- classes. *See also specific classes***
 - explained, 423, 434-436
 - subclasses, 423, 441-442
- client feedback**
 - incorporating into design process, 319-323
 - SneakerSpot sample project, 319-323
- Close Other Documents command (File menu), 77**
- CLR (Common Language Runtime), 394-395**

- code-behind files, 416-418**
 - constructors, 425-426
 - namespaces, 424
 - page class, 425
- CodePlex, 476**
- collections**
 - ColorSwatch, 349-353
 - data collections
 - adding properties to, 266-268*
 - creating, 154, 266, 270-271*
 - customizing data in, 155-158*
 - editing, 268-269*
 - properties, 267
- color swatch list, customizing, 362-364**
- ColorPaletteDataSource, 349**
- ColorPicker control**
 - ColorPaletteDataSource, 349
 - ColorSwatch collection, 349-353
 - control template, editing, 355-359
 - color swatch list, customizing, 362-364
 - data binding changes to CurrentColor, 360-361
 - goals for, 348
 - visual design, 347-348
- ColorSwatch collection, 349-353**
- ComboBox-Sketch control, 158**
- comic narrative, 46-47**
 - flow, 49
 - frame, 48
 - image, 48

- moment, 47
- word, 49
- commands.** *See specific commands*
- Common Language Runtime (CLR),**
394-395
- Common Properties category**
(Properties panel), 203
- Communicating Design:***
Developing Web Site
Documentation for Design and
Planning (Brown), 475
- communicating dynamically, 49-50**
- complex properties, 267**
- component screens**
 - creating, 165-168, 223-224
 - explained, 222
 - linking to other screens, 168-169
 - naming, 167
 - navigation via, 307-310
 - SneakerSpot sample project,
307-310
- Component Screens category**
(Assets panel), 214
- conceptualization and prototyping**
(design process), 38-39
- configure_page_zoom_**
controller.png file, 327
- configure_page_zoom_**
viewer.png file, 327-329
- configuring workspace, 65-71**
- constructors, 425-426**

- containers, 430**
 - Border, 415-416
 - Canvas, 413
 - Grid, 411
 - ScrollPane, 415
 - StackPanel, 413
- content**
 - adding to navigation flow, 83
 - content protection
(Silverlight), 461
 - creating with SketchStyles, 89-90
 - importing into SketchFlow, 79-82
- control templates, editing,**
355-359
- control-focused SketchStyles,**
217-218
- controls**
 - adding states to, 237-243
 - adding with SketchStyles, 89-90
 - binding data to, 160-163
 - ColorPicker
 - color swatch list, customizing,*
362-364
 - ColorPaletteDataSource, 349*
 - ColorSwatch collection,*
349-353
 - control template, editing,*
355-359
 - xdata binding changes to*
CurrentColor, 360-361
 - goals for, 348*
 - visual design, 347-348*

- ComboBox-Sketch, 158
- control templates, editing, 355-359
- control-focused SketchStyles, 217-218
- creating controls to bind data to, 158-159
- slider controls, binding to text blocks, 260-264
- Zoom, 327-330
 - Bounce Out and Bounciness properties*, 343-344
 - DragAndReturnAction behavior*, 341-343
 - final refinements*, 344-345
 - HideAction behavior*, 340-341
 - ShowAction behavior*, 339

Controls category (Assets panel), 189

Cooper, Alan, 43, 473

Core Presentation Framework (Silverlight), 456

Create a Component icon (SketchFlow Map), 199

Create a Screen icon (SketchFlow Map), 199

Create Data Binding panel, 262, 378

***Creative Whack Pack* (von Oech), 472**

***Critique Magazine*, 20**

CubicEase function, 233

CurrentColor image, binding data to, 360-361

customizing data collections, 155-158

Cynergy, 447

D

data

for ColorPicker (SneakerSpot project)

ColorPaletteDataSource, 349

ColorSwatch collection, 349-353

controls

binding data to, 160-163, 260-264

creating controls to bind data to, 158-159

creating for projects, 265-269

data collections

adding properties to, 266-268

creating, 154, 266, 270-271

customizing data in, 155-158

editing, 268-269

data-focused SketchStyles, 218

Data panel

Detail Mode, 264-265

explained, 205

List Mode, 264

Data panel controls, 151-152

*Add sample data source
icon, 153*

*Define New Sample Data
icon, 153*

Details Mode icon, 152-153

Edit sample values icon, 154

List Mode icon, 152

data sources

ColorPaletteDataSource, 349

defining, 153

naming, 154

data types

casting, 431

explained, 430

explained, 259-260

master-list relationship, creating,
270-275

data collections

adding properties to, 266-268

creating, 154, 266, 270-271

customizing data in, 155-158

editing, 268-269

data-focused SketchStyles, 218

Data panel, 151-152, 264

Add sample data source icon, 153

Define New Sample Data
icon, 153

Detail Mode icon, 152-153,
264-265

Edit sample values icon, 154

explained, 205

List Mode icon, 152, 264

data sources

ColorPaletteDataSource, 349

defining, 153

naming, 154

data types

casting, 431

explained, 430

DeepZoom, 452

**Define New Sample Data
feature, 266**

**Define New Sample Data icon
(Data panel), 153**

defining

data sources, 153

methods, 432

**definition and intent (design
process), 33**

**Delete a Selection icon
(SketchFlow Map), 199**

deleting feedback, 107

derived classes. See subclasses

***The Design of Business
(Martin), 472***

design funnel, 1-2

design process

agile processes, 40

analysis, 35

conceptualization and
prototyping, 38-39

definition and intent, 33

discussed, 27-32

documentation, 296-297

- innovation
 - discussed, 17-20*
 - finding ideas, 21-23*
 - presenting ideas to stakeholders, 23-25*
 - revealing direction, 20-21*
- patterns, 51-52
 - design, test, survey, 53-54*
 - evaluate, amend, release, 55-56*
 - execute, release, iterate, 55*
 - ideate, sketch, experiment, 53*
 - research, interview, observe, 52*
 - tune, life cycle, debrief, 56*
- planning, 37
- production and
 - implementation, 39
- research, 34
- sensing and feedback, 39
- SneakerSpot example. *See SneakerSpot sample project*
- synthesis and ideation, 36, 37
- design, test, survey (design process patterns), 53-54**
- Design Thinking Digest, 478**
- design websites, 478-479**
- designers, advantages of learning SketchFlow, 10**
- Designing for Interaction: Creating Smart Applications and Clever Devices (Saffer), 474***
- Designing for the Digital Age (Goodwin), 474***

- Designing Gestural Interfaces (Saffer), 474***
- Designing Interactions (Moggridge), 472***
- Designing Interfaces (Tidwell), 474***
- desktop applications and .NET framework, 396**
- Detail Mode (Data panel), 264-265**
- Details Mode icon (Data panel), 152-153**
- developer tools**
 - book recommendations, 470
 - for .NET framework, 392-393
 - for WPF (Windows Presentation Foundation), 446
 - websites, 475-477
- Direct Selection tool, 181**
- direction, revealing, 20-21**
- DockPanel object, 184**
- documentation**
 - creating, 171-173, 290-292
 - for design process, 296-297
- Don't Make Me Think! (Krug), 473***
- DragAndReturnAction behavior, 341-342**
- Drucker, Peter, 472**
- dynamic communication, 49-50**
- Dynamic Prototyping, 478**
- E**
 - easing functions, 232-233, 315, 458
 - Edit a Copy command, 367

Edit menu, 207

Edit Resource icon, 240-241

Edit Sample Values dialog, 155

Edit sample values icon (Data panel), 154

Edit Template command, 367

editing

behavior properties, 254-256

control templates, 355-359

data collections, 155-158,
268-269

SketchStyles, 222

states, 138, 145-147, 236

Effects category (Assets panel), 189

Effects icon (Artboard), 196

Einstein, Albert, 23, 391

Elam, Kimberly, 473

ElasticEase function, 233

Electric Bleach, 476

elements, 183-185, 402-403

Border object, 184

Canvas object, 184

DockPanel object, 184

Grid object, 184

ScrollView object, 184

StackPanel object, 184

Uniform Grid object, 185

ViewBox object, 185

WrapPanel object, 184

Ellipse tool, 182

enabling

annotations, 169

navigation in projects, 127-133

EnemyControl class, 436-437

Eraser tool, 103

***The Essential Drucker*
(Drucker), 472**

evaluate, amend, release (design process patterns), 55-56

event handlers, 426-429, 440-441

events, 423, 426-429, 440-441

execute, release, iterate (design process patterns), 55

ExponentialEase function, 233

**Export to Microsoft Word
command (File menu), 172, 291**

exporting feedback, 105

Expression Blend, 463

**Expression Studio Website and
Expression Community Site, 475**

Eyedropper tool, 181

F

feedback

controls, 286-287

deleting from project, 107

Feedback panel, 206, 283

importing into projects, 287-289

incorporating into design process,
319-323

obtaining with SketchFlow Player,
279-282

- providing with SketchFlow Player, 102-108
- SneakerSpot sample project, 319-323, 365-366
- Feedback command (Window menu), 105, 288**
- Feedback panel, 105, 206, 283**
- Feedback Text Entry Box, 102-103**
- File menu commands**
 - Close Other Documents command, 77
 - explained, 206
 - Export to Microsoft Word command, 172, 291
 - Import Photoshop File command, 112
 - Package SketchFlowProject command, 94, 281
- files**
 - ArcListBoxStyle.xaml, 362
 - ArcPanel.dll, 362
 - code-behind files
 - constructors, 425-426*
 - namespaces, 424*
 - page class, 425*
 - code-behind files (XAML), 416-418
 - configure_page_zoom_controller.png, 327
 - configure_page_zoom_viewer.png, 327-329
 - navigation_component.png, 310
 - navigation_component_v2.png, 326

- finding ideas, 21-23**
- Fiore, Quentin, 23**
- Fire method, 438**
- Fit All icon (SketchFlow Map), 199**
- Fit Selection icon (SketchFlow Map), 199**
- flow (comic narrative), 49**
- folders, SneakerSpot3, 326**
- Foundation Expression Blend with Silverlight (Gaudioso), 470***
- Foundation Silverlight 3 Animation (Paries), 470***
- frame (comic narrative), 48**
- frames, 120-122**
- frameworks, 422**
- functions. *See methods***
 - BackEase, 232
 - BounceEase, 232
 - CircleEase, 232
 - CubicEase, 233
 - ElasticEase, 233
 - ExponentialEase, 233
 - PowerEase, 233
 - QuadraticEase, 233
 - QuarticEase, 233
 - QuinticEase, 233
 - SineEase, 233

G

- Gaudioso, Victor, 470**
- Glaser, Milton, 20**
- Goodwin, Kim, 474**

Gradient tool, 181
Green, Jennifer, 471
Grid container, 184, 411
Grid icon (Artboard), 197
Grid Snapping icon (Artboard), 197
Grid Systems (Elam), 473
Grid tool, 183
GUI (graphical user interface),
 moving beyond, x-xiii
Guthrie, Scott, 476

H

Head First C# (Stellman and
 Green), 471
Help menu, 211
HideAction behavior, 340-341
hide state (SneakerSpot project),
 334-335
hi-fi
 applying hi-fi styles, 382-383
 mapping lo-fi to, 375-379
Highlighter tool, leaving feedback
 with, 103
Holden, Kritina, 472
House of Mirrors, 477
Hubel, David, 18

I

ideas
 finding, 21-23
 presenting to stakeholders, 23-25
ideate, sketch, experiment (design
 process patterns), 53

image (comic narrative), 48
Import Adobe Photoshop File
 dialog box, 113
Import Photoshop File command
 (File menu), 112
Import Same Data from XML
 feature, 266

importing

 content into SketchFlow, 79-82
 feedback, 287-289
 Photoshop files into SketchFlow
 for animation, 111-118

Ingebretsen, Robby, 477

inheritance, 423, 441-442

initial sketch concepts
 (SneakerSpot project), 299-301

innovation

 discussed, 17-20
 finding ideas, 21-23
 presenting ideas to stakeholders,
 23-25
 revealing direction, 20-21

Innovative Whack Pack (von
 Oech), 472

Institute of Design, 479

interactive design books, 473-475

interactivity for SneakerSpot
 sample project, 337

 Bounce Out and Bounciness
 properties, 343-344

 DragAndReturnAction behavior,
 341-343

 HideAction behavior, 340-341

 ShowAction behavior, 338-339

J–K

Johnson, Michael B., 50

Kano model, 41-42

keyframe animation, 244-246

Kirupa.com, 477

Krug, Steve, 473

Kumar, Vijay, 31

L

languages

book recommendations, 470-471

for WPF (Windows Presentation Foundation), 446

Layout category (Properties panel), 202

layout containers

Border, 415-416

Canvas, 413

Grid, 411

ScrollPane, 415

StackPanel, 413

layout-focused SketchStyles, 218

LayoutRoot, 183, 408

layouts

creating with XAML (Xtensible Application Markup Language)

Border container, 415-416

Canvas container, 413

Grid container, 411

LayoutRoot, 183, 408

naming objects, 408-410

ScrollPane container, 415

StackPanel container, 413

layout-focused SketchStyles, 218

libraries, 422

Lidwell, William, 472

Lightbulb icon, 103

Line tool, 182

Link to an Existing Item option (Project panel), 81

linking component screens, 168-169

List Mode icon (Data panel), 152, 264

ListBox, 87

localization (Silverlight), 460

Locations category (Assets panel), 190

lo-fi, mapping to hi-fi, 375-379

loyal customer survey (SneakerSpot project), 298

M

MacDonald, Matthew, 471

Macintosh operating systems, 11-12

magnifying glass (SneakerSpot project), 328-330

Bounce Out and Bounciness properties, 343-344

DragAndReturnAction behavior, 341-343

final refinements, 344-345

- HideAction behavior, 340-341
- ShowAction behavior, 339
- Make Into Component Screen**
command (Tools menu), 223
- Making Comics (McCloud), 47**
- Managed Extensibility**
Framework, 460
- Map panel, 99-100, 284**
- mapping lo-fi to hi-fi, 375-379**
- Martin, Roger, 29-30, 472**
- master-list relationship, creating,**
270-275
- McCloud, Scott, 47, 472**
- The Mechanical Design Process***
(Ullman), 28
- Media category (Assets**
panel), 189
- media support for Silverlight,**
465-467
- The Medium Is the Massage***
(MuLuhan and Fiore), 23
- Mental Models (Young, Indi), 473***
- menus. *See specific menus***
- Merholz, Peter, 50**
- methods, 423, 429. *See***
also functions
 - calling, 432
 - constructors, 425-426
 - defining, 432
 - explained, 431-432, 436-438
 - Fire, 438
 - overloading, 438
 - return values, 433-434

- microphone support**
(Silverlight), 460
- Microsoft Design, 478**
- Microsoft Expression Blend***
Unleashed (Williams), 470
- Microsoft Software Development**
Network (MSDN), 476
- Microsoft Visual Studio, 463**
- Miscellaneous category**
(Properties panel), 204
- MIX Online 2.0, 478**
- mobile applications, 399**
- Moggridge, Bill, 472**
- moment (comic narrative), 47**
- MSDN (Microsoft Software**
Development Network), 476
- multicast networking**
(Silverlight), 461
- multitouch support**
(Silverlight), 461
- MuLuhan, Marshall, 23**

N

- naming**
 - component screens, 167
 - data sources, 154
 - objects, 408-410
- Nathan, Adam, 471**
- NavigateBackAction behavior, 252**
- NavigateForwardAction**
behavior, 253
- NavigateToScreenAction, 132**

NavigateToScreenAction
behavior, 253

navigation

- via component screens, 307-310
- enabling in projects, 127-133
- navigation flow, adding
content to, 83
- navigation screens, accessing
through Assets panel, 214
- in SketchFlow Player, 96-100,
282-287

NavigationButtonStyles

- creating, 367-370
- template modifications, 370-373

navigation_component.png
file, 310

navigation_component_v2.png
file, 326

Navigation panel, 283

Navigation Screens category
(Assets panel), 214

Nerd + Art, 477

.NET Compact Framework, 399

.NET framework

- applications
 - desktop applications, 396*
 - mobile applications, 399*
 - Silverlight RIAs (Rich Internet Applications), 397-398*
 - websites/web applications, 397*
- Common Language Runtime (CLR), 394-395

- development tools, 392-393
- explained, 391-392
- .NET Compact Framework, 399
- and Silverlight, 392

Nibbles Tutorials, 477

notifications (Silverlight), 461

O

Object menu, 208

object-oriented programming
(OOP), 420-422

objects

- Border, 184
- Canvas, 184
- DockPanel, 184
- explained, 190-192, 423
- Grid, 184
- naming, 408-410
- objects in Blend, 183
- properties, 423, 429-430, 438-439
- ScrollView, 184
- StackPanel, 184
- Uniform Grid, 185
- ViewBox, 185
- WrapPanel, 184

Objects and Timeline panel,
114-118

- objects, 190-192
- timelines, 193-195, 244-245

Oech, Roger von, 472

offline DRM (Silverlight), 461

OOP (object-oriented programming), 420-422

Out of Browser Mode (Silverlight), 457

overloading methods, 438

P

Package SketchFlowProject command (File menu), 94, 281

page class, 425

Paintbucket tool, 181

panels. *See specific panels*

Pan tool, 85, 181

Paper Prototyping (Snyder), 473

Paries, Jeff, 470

Parnas, David L., 27

PasswordBox-Sketch asset, 90

patterns (design process), 51-52

design, test, survey, 53-54

evaluate, amend, release, 55-56

execute, release, iterate, 55

ideate, sketch, experiment, 53

research, interview, observe, 52

tune, life cycle, debrief, 56

Pen tool, 86, 103, 182

Pencil tool, 86, 182

Perspective 3D

Transformation, 457

Petzold, Charles, 471

Photoshop files, importing into SketchFlow for animation, 111-118

Pixar, 50

pixel effects (Silverlight), 457

planning (design process), 37

PlaySketchFlowAnimation behavior, 253

plug-ins, Silverlight Player, 456

polymorphism, 423

post-mortems

general discussion, 387

project measurement discussion, 386

project process discussion, 385-386

team discussion, 387

PowerEase function, 233

predilection gap, 30

previewing states, 145

printing support (Silverlight), 460

production and implementation (design process), 39

Programming in the Key of C#: A Primer for Aspiring Programmers (Petzold), 471

Project menu

Build Project command, 317

explained, 208-209

Rebuild Project command, 362

Run Project command, 94

SketchFlow Project Settings command, 75

Project Rosetta, 477

Project section (Assets panel), 188

Projects panel, 81, 186-188

properties, 423, 429-430

adding to data collections,
266-268

behavior properties, setting,
254-256

collection properties, 267

complex properties, 267

explained, 438-439

simple properties, 267

Properties panel

Animation category, 204

Appearance category, 201-202

Brushes category, 201

Common Properties category, 203

explained, 200-201

Layout category, 202

Miscellaneous category, 204

Transform category, 203

Trigger category, 203

***Pro Silverlight 3 in C#*
(MacDonald), 471**

prototyping

dynamic prototypes, 9-10

WPF (Windows Presentation
Foundation) applications, 453

***Prototyping* (Warfel), 474**

Q

QuadraticEase function, 233

QuarticEase function, 233

QuinticEase function, 233

R

**Rebuild Project command (Project
menu), 362**

recommended reading

developer tool books, 470

interactive design books, 473-475

language books, 470-471

theory books, 472-473

Rectangle-Sketch asset, 90

Rectangle tool, 182

Redo icon (SketchFlow Map), 198

**refinement and master sketches
(SneakerSpot project), 302-306**

reliability versus validity, 29-30

RepeatBehavior, 245

repeating animations, 245

research (design process), 34

**research, interview, observe
(design process patterns), 52**

resources

data resources. *See* data

recommended books

developer tool books, 470

*interactive design books,
473-475*

- language books*, 470-471
- theory books*, 472-473
- websites, 475-479
- Resources panel**, 205
- Results panel**, 199-200
- Results window**, 94
- return values (methods)**, 433-434
- revealing direction**, 20-21
- reviewing client feedback**, 319-323
- RIAs (Rich Internet Applications)**, 397-398, 455
- Roam, Dan**, 474
- root UI updates (SneakerSpot project)**, 380-381
- Run Project command (Project menu)**, 94
- runtime**, 394, 456

S

- Saffer, Dan**, 474
- Save as New Workspace command (Window menu)**, 71
- Schorman, Christian**, 476
- Schrage, Michael**, 23
- Scott Guthie Blog**, 476
- ScrollPane container**, 415
- ScrollViewer object**, 184
- searching for assets**, 87
- Selection tool**, 85, 181
- sensing and feedback (design process)**, 39
- Serious Play* (Schrage)**, 23

- Set as Start feature**, 76
- setting up. *See* configuring workspace**
- sharing projects**, 93-95
- Show Component Connections Dimmed icon (SketchFlow Map)**, 199
- Show Feedback options icon**, 104
- Show Navigation Connections Dimmed icon (SketchFlow Map)**, 199
- show state (SneakerSpot project)**, 332-333
- ShowAction behavior**, 338-339
- Sierra, Kathy**, 19
- Silverlight**, 64, 392, 457-461.
 - See also* XAML
 - architecture, 456
 - compared to WPF (Windows Presentation Foundation), 448-452
 - Core Presentation Framework, 456
 - explained, 455-456
 - media support and streaming, 465-467
 - .NET Framework, 456
 - Player, 456
 - RIAs (Rich Internet Applications), 397-398
 - tools and workflow, 462-463
 - website, 47
- simple properties**, 267

SineEase function, 233

**SizePicker combo box
(SneakerSpot project), 383**

SketchFlow, advantages of, 10-13

**SketchFlow category (Assets
panel), 188**

SketchFlow Map

Create a Component icon, 199

Create a Screen icon, 199

creating, 74-77

Delete a Selection icon, 199

explained, 73-74, 198-199

Fit All icon, 199

Fit Selection icon, 199

Redo icon, 198

Show Component Connections
Dimmed icon, 199

Show Navigation Connections
Dimmed icon, 199

Undo icon, 198

Zoom control, 198

SketchFlow Player, 322

navigating, 96-100, 282-287

obtaining feedback with, 279-282

providing feedback with, 102-108

**SketchFlow Project Settings
command (Project menu), 75**

sketching

comic narrative, 46-47

flow, 49

frame, 48

image, 48

moment, 47

word, 49

presenting ideas with
sketches, 23-25

SneakerSpot project

initial sketch concepts, 299-301

*refinement and master
sketches, 302-306*

team sketching, 43-45

visual storytelling, 45-46

**Sketching User Experiences
(Buxton), 472**

SketchStyles

accessing, 219-220

adding controls with, 89-90

control-focused SketchStyles,
217-218

data-focused SketchStyles, 218

editing, 222

explained, 215-216

layout-focused SketchStyles, 218

text-focused SketchStyles,
216-217

**slider controls, binding to text
blocks, 260-264**

**Snapping to Snaplines icon
(Artboard), 197**

SneakerSpot sample project

animation, adding, 311-313

Easing Function
transitions, 315

Properties panel
adjustments, 313

SketchFlow Animation
panel, 314

SketchFlow animation
progress, 316

applying hi-fi styles, 382-383

building, 317

client feedback, 319-323

ColorPicker control

ColorPaletteDataSource, 349

ColorSwatch collection,
349-353

control template, editing,
355-359

color swatch list, customizing,
362-364

data binding changes to
CurrentColor, 360-361

goals for, 348

visual design, 347-348

control templates, editing,
355-359

feedback, 365-366

initial sketch concepts, 299-301

interactivity, 337

Bounce Out and Bounciness
properties, 343-344

DragAndReturnAction behavior,
341-343

HideAction behavior, 340-341

ShowAction behavior, 338-339

loyal customer survey, 298

mapping lo-fi to hi-fi, 375-379

navigation via component
screens, 307-310

NavigationButtonStyles

creating, 367-370

template modifications,
370-373

project brief, 297

refinement and master sketches,
302-306

root UI updates, 380-381

SizePicker combo box, 383

SneakerSpot3 folder, 326

states, 331-332

hide state, 334-335

show state, 332-333

visual states, 379-380

Zoom control, 327-330

Bounce Out and Bounciness
properties, 343-344

DragAndReturnAction behavior,
341-343

final refinements, 344-345

HideAction behavior, 340-341

ShowAction behavior, 339

SneakerSpot3 folder, 326

Snyder, Carolyn, 473

Split view, 405-406

stack order, 115

StackPanel container, 184, 413
stakeholders, presenting ideas to, 23-25

State panel, 138-139, 190
states

- adding to buttons, 237-243
- advantages of, 235-236
- creating, 137-144, 236
- definition of, 135-136
- editing, 138, 145-147, 236
- explained, 136-137, 235-237
- previewing, 145
- for SneakerSpot sample project, 331-332

- hide state, 334-335*

- show state, 332-333*

- State panel, 138-139, 190
- transitions between, 148, 242
- usage tips, 149-150
- visual states, 379-380

Stellman, Andrew, 471

storyboards, 244-246

storytelling, 45-46

streaming (Sliverlight), 465-467

styles

- accessing, 219-220
- control-focused SketchStyles, 217-218
- data-focused SketchStyles, 218
- editing, 222
- explained, 215-216
- hi-fi styles, 382-383

- layout-focused SketchStyles, 218

- NavigationButtonStyles

- creating, 367-370*

- template modifications, 370-373*

- text-focused SketchStyles, 216-217

Styles category (Assets panel), 189

subclasses, 423, 441-442

substantiation, 30

Summers, Sara, 478

SuperEnemyControl class, 442

surveys, loyal customer survey, 298

synthesis and ideation (design process), 36-37

system requirements, 64

T

teams

- questions for port-mortems, 387
- team sketching, 43-45

templates

- control templates, editing, 355-359

- explained, 221-222

- for NavigationButtonStyles, 370-373

text

- binding slider controls to, 260-264

- text-focused SketchStyles, 216-217

text-focused SketchStyles, 216-217

Text tool, 185

TextBox-Sketch asset, 90

theory books, 472-473

Thirteen23, 447, 465

Tidwell, Jennifer, 474

timelines, 193-195, 244-245

TitleCenter-Sketch asset, 90

Tools menu

explained, 210

Make Into Component Screen
command, 223

Tools panel, 84-87, 180

Brush Transform tool, 181

Direct Selection tool, 181

elements, 183-185

Border object, 184

Canvas object, 184

DockPanel object, 184

Grid object, 184

ScrollView object, 184

StackPanel object, 184

Uniform Grid object, 185

ViewBox object, 185

WrapPanel object, 184

Ellipse tool, 182

Eyedropper tool, 181

Gradient tool, 181

Grid tool, 183

Line tool, 182

objects in Blend, 183-185

Paintbucket tool, 181

Pan tool, 181

Pen tool, 182

Pencil tool, 182

Rectangle tool, 182

Selection tool, 181

Text tool, 185

Zoom tool, 181

**Transform category (Properties
panel), 203**

Transition Style, 76

transitions

adding to SneakerSpot
project, 315

between states, 148, 242

**Trigger category (Properties
panel), 203**

triggers

explained, 249-251

TriggerTypes, 254-255

Tulathimutte, Tony, 52

**tune, life cycle, debrief (design
process patterns), 56**

U

Ullman, David, 28

***Understanding Comics*
(McCloud), 472**

Undo icon (SketchFlow Map), 198

Unger, Russ, 475

Uniform Grid object, 185

Universal Principles of Design
(Lidwell, Holden, and
Butler), 472

user experience, Kano model,
41-42

User Guide for Expression
Blend, 86, 182

UX Array, 478

V

validity versus reliability, 29-30

variables, 430-431

View menu, 207

ViewBox object, 185

viewing

annotations, 170

XAML (Xtensible Application
Markup Language), 404-405
Split view, 405-406
XAML view, 405

views. *See specific views*

Visual State Manager, 451

visual states, adding, 379-380

visual storytelling, 45-46

Visual Studio, 463

W

Warfel, Todd Zaki, 474

web applications and .NET
framework, 397

Web Form Design
(Wroblewski), 474

webcam support (Silverlight), 460

website resources, 475-479

Weisel, Torsten, 18

Williams, Brennon, 470

WIMP (windows, icon, menu,
pointing) interface, x-xiii

Window menu, 70, 211

Feedback command, 105, 288

Save as New WorkSpace
command, 71

Windows Communication
Foundation RIA Services
(Silverlight), 460

Windows Forms, 396, 445

Windows Presentation
Foundation. *See* WPF

Windows Presentation Foundation
***Unleashed* (Nathan), 471**

WinForms, 396, 445

word (comic narrative), 49

Word documents, creating from
SketchFlow projects, 291-292

workspace

Artboard

activating annotations on, 278

Annotation icon, 198

discussed, 72

Effects icon, 196

explained, 195-197

Grid icon, 197

Grid Snapping icon, 197

- Snapping to Snaplines icon, 197*
- Zoom icon, 196*
- explained, 178-180
- setting up, 65-71
- SketchFlow Map
 - creating, 74-77*
 - discussed, 73-74*
- WPF (Windows Presentation Foundation), 396**
 - capabilities, 447-448
 - compared to Silverlight, 448-452
 - explained, 445-446
 - languages, 446
 - prototyping, 453
 - tools, 446
- WrapPanel object, 184**
- Wroblewski, Luke, 474**
- X – Y – Z**
- XAML (Xtensible Application Markup Language)**
 - capabilities, 403-404
 - code-behind files, 416-418
 - constructors, 425-426*
 - namespaces, 424*
 - page class, 425*
 - creating layouts with
 - Border container, 415-416*
 - Canvas container, 413*
 - Grid container, 411*
 - LayoutRoot, 408*
 - naming objects, 408-410*
 - ScrollPane container, 415*
 - StackPanel container, 413*
 - elements, 402-403
 - explained, 401-402, 463-465
 - viewing, 404-405
 - Split view, 405-406*
 - XAML view, 405*
- Young, Indi, 473**
- Zoom control, 198, 327-330**
 - Bounce Out and Bounciness properties, 343-344
 - DragAndReturnAction behavior, 341-343
 - final refinements, 344-345
 - HideAction behavior, 340-341
 - ShowAction behavior, 339
- Zoom icon (Artboard), 196**
- Zoom tool, 86, 181**