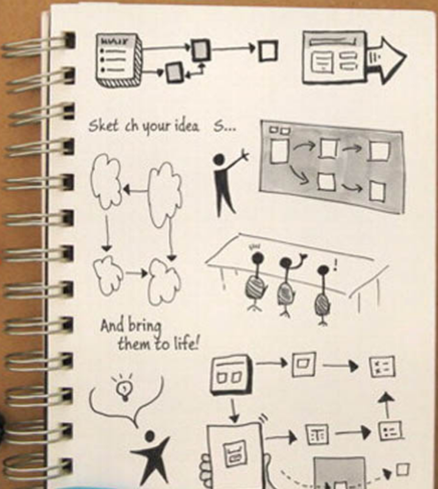
DYNAMIC PROTOTYPING

WITH SKETCHFLOW IN EXPRESSION BLEND



→ BY
CHRIS BERNARD &
SARA SUMMERS

FOrEWOrd BY BILL BUXTOn

QUE

Dynamic Prototyping with SketchFlow in Expression Blend

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

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

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.

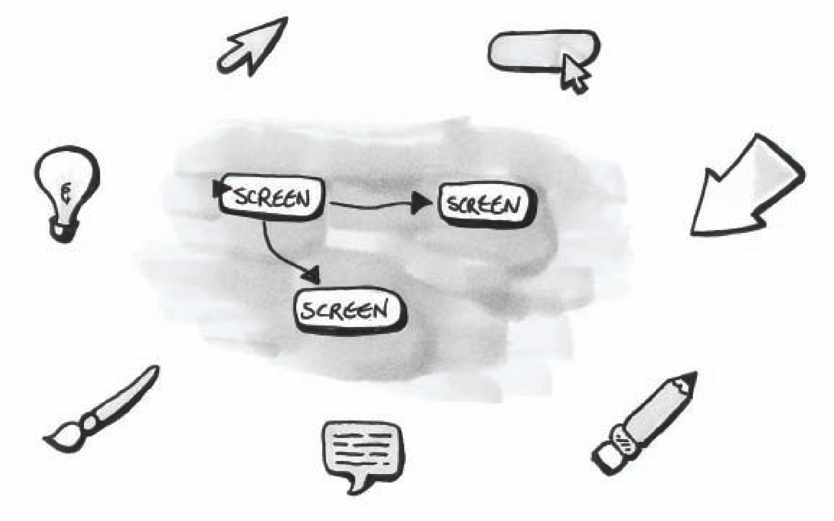
Bulk Sales

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



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.

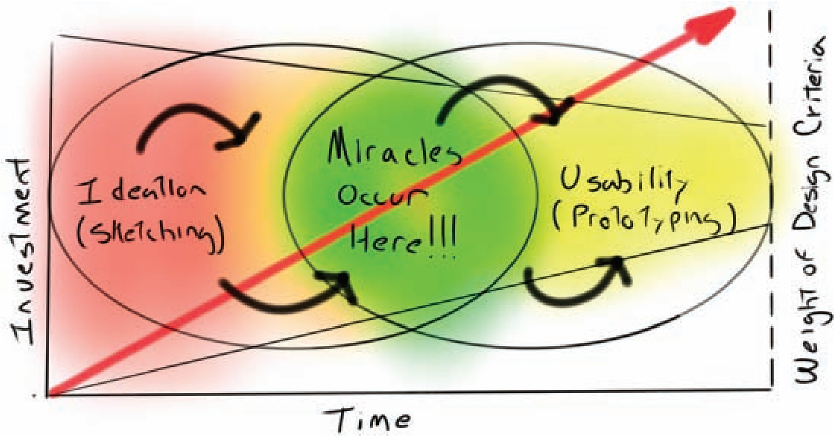


Figure 0.1 *The dynamics of the design funnel shows us the continuum of how sketching and prototypes help us make great experiences. Idea and figure courtesy of Bill Buxton with a few modifications.*

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:

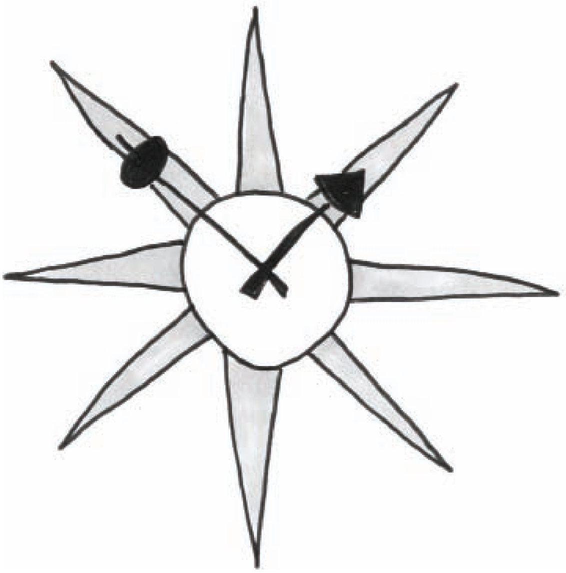
- Validating design research and insights.
- Sharing brainstorming concepts and rough sketches.
- Beginning concept exploration.
- Bringing hand-drawn sketches or digitally created sketches to life in dynamic prototypes.
- Generating consensus and feedback.
- Creating design specification documentation.

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 whit:

- 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.

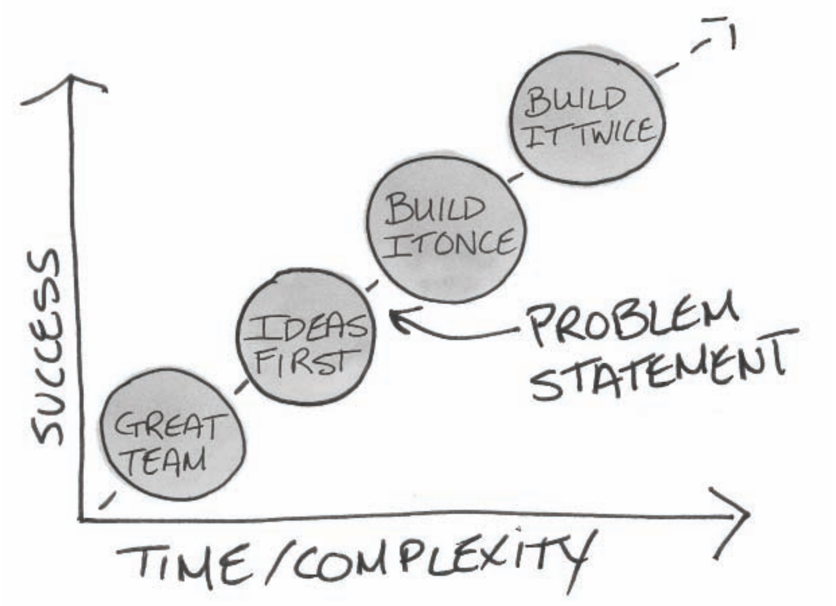


Figure 2.1 The preceding lessons simplified in a bubble chart.

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.

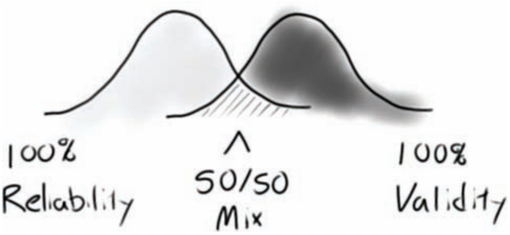


Figure 2.2 A sketch that illustrates Martin's theories.

(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.

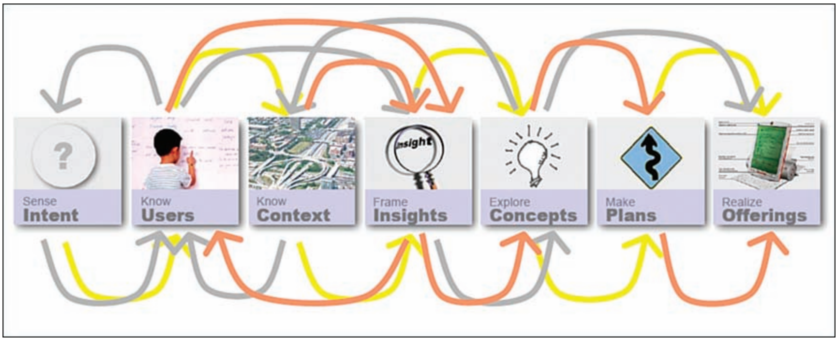


Figure 2.3 An example of the steps and non-linear nature of a design process taught at the Institute of Design by Professor Vijay Kumar. (Image printed with permission from Vijay Kumar.)

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 is starts with a focus on understanding your customers or users first versus starting with a technology or a business.

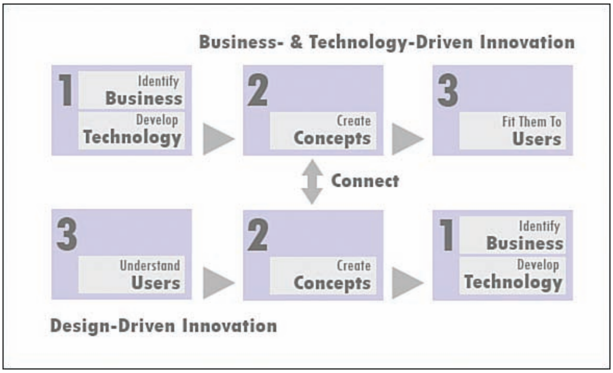
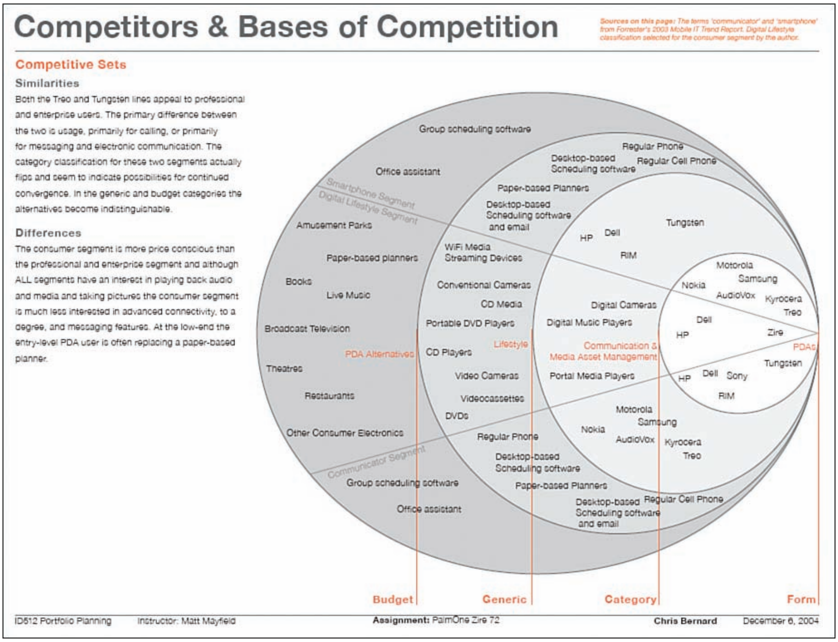


Figure 2.4 The journey that occurs in a design process around design innovation and how it differs from business and technology innovation. (Image printed with permission from Vijay Kumar.)

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.

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.



Competitors & Bases of Competition

Sources on this page: The terms 'communicator' and 'smartphone' from Forrester's 2003 Mobile IT Trend Report. Digital Lifestyle classification selected for the consumer segment by the author.

Competitive Sets

Similarities

Both the Treo and Tungsten lines appeal to professional and enterprise users. The primary difference between the two is usage, primarily for calling, or primarily for messaging and electronic communication. The category classification for these two segments actually flips and seem to indicate possibilities for continued convergence. In the generic and budget categories the alternatives become indistinguishable.

Differences

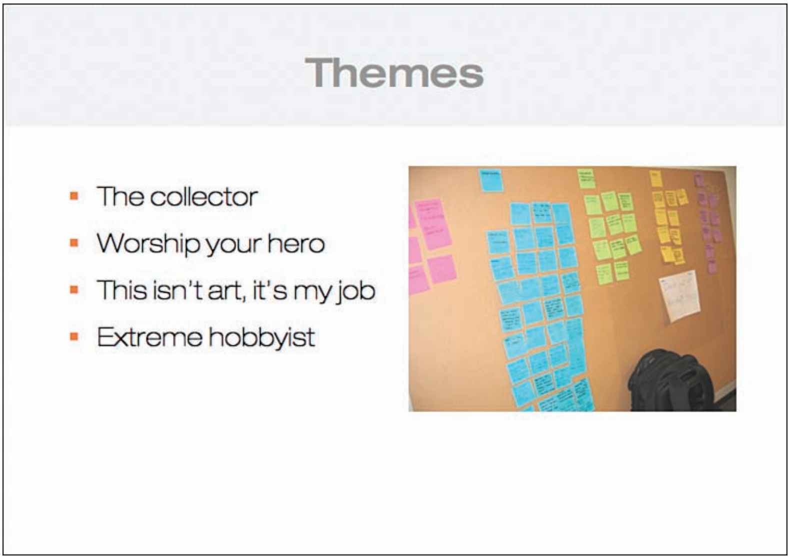
The consumer segment is more price conscious than the professional and enterprise segment and although ALL segments have an interest in playing back audio and media and taking pictures the consumer segment is much less interested in advanced connectivity, to a degree, and messaging features. At the low-end the entry-level PDA user is often replacing a paper-based planner.

ID512 Portfolio Planning Instructor: Matt Mayfield **Assignment:** PalmOne Zire 72 **Chris Bernard** December 6, 2004

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.



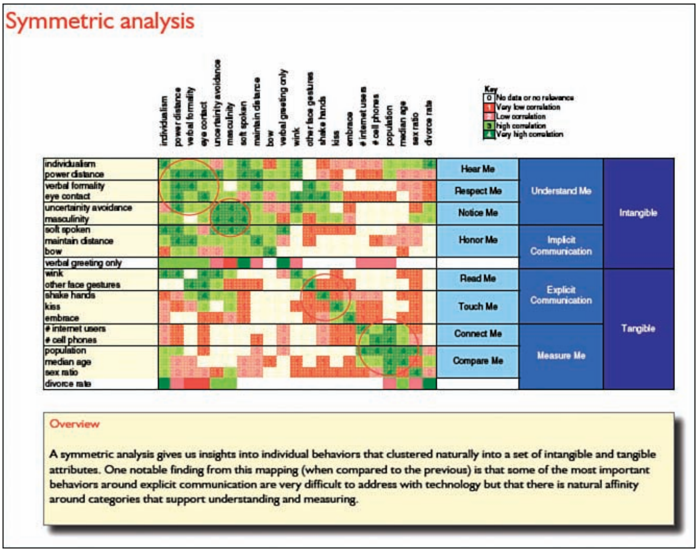
Themes

- The collector
- Worship your hero
- This isn't art, it's my job
- Extreme hobbyist

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.



Symmetric analysis

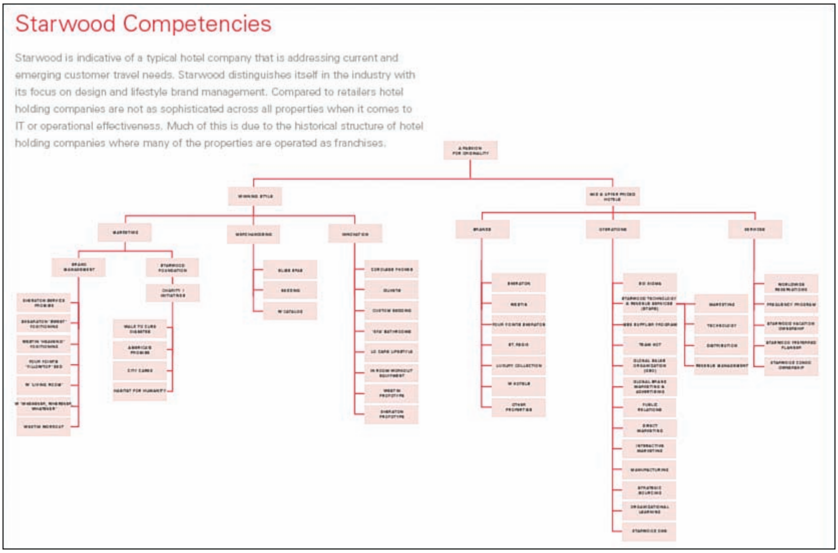
Overview

A symmetric analysis gives us insights into individual behaviors that clustered naturally into a set of intangible and tangible attributes. One notable finding from this mapping (when compared to the previous) is that some of the most important behaviors around explicit communication are very difficult to address with technology but that there is natural affinity around categories that support understanding and measuring.

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.*

SYNTHESIS AND IDEATION

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.



Starwood Competencies

Starwood is indicative of a typical hotel company that is addressing current and emerging customer travel needs. Starwood distinguishes itself in the industry with its focus on design and lifestyle brand management. Compared to retailers hotel holding companies are not as sophisticated across all properties when it comes to IT or operational effectiveness. Much of this is due to the historical structure of hotel holding companies where many of the properties are operated as franchises.

Figure 2.8 *In this example from one of the author's student work, analysis of data enables us to identify competencies for a sample customer. This is an example from a Design Strategy Workshop at the Institute of Design with Professor Vijay Kumar.*

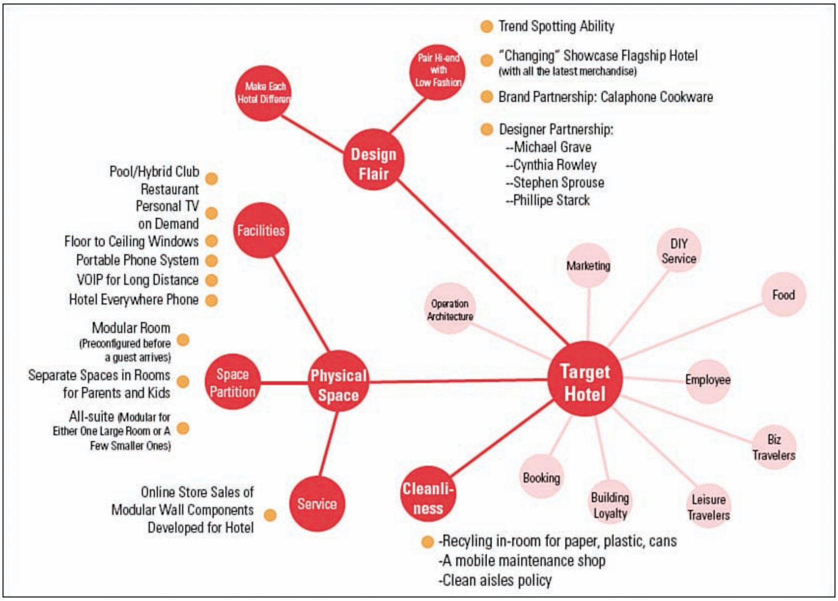


Figure 2.9 Affinity mapping helps designers and business makers uncover new opportunities. This is an example from a Design Strategy Workshop at the Institute of Design with Professor Vijay Kumar.

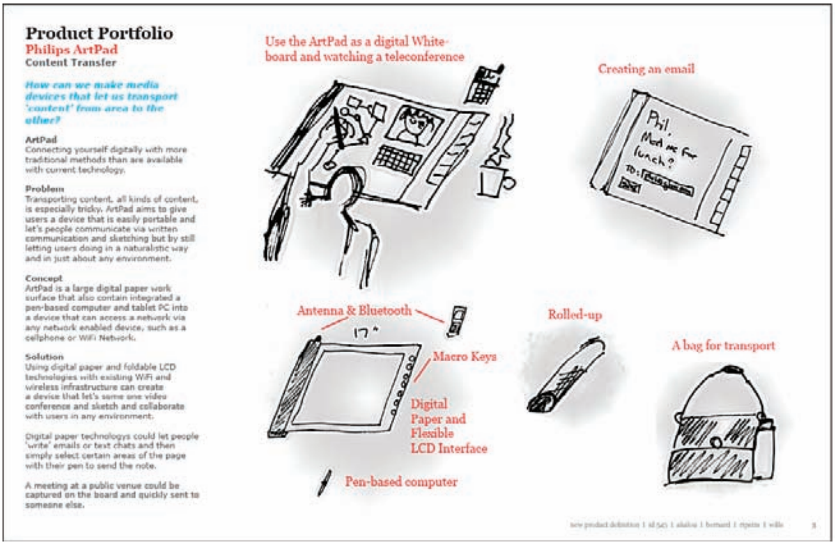
By looking at competitor offerings and matching them with a different company, teams can uncover new business opportunities. Here is an example of student work of one of the authors; it was identifying competencies and needs that would allow for a new business to be created, which would deliver new value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



Product Portfolio

Philips ArtPad

Content Transfer

How can we make media devices that let us transport 'content' from area to the other?

ArtPad
Connecting yourself digitally with more traditional methods than are available with current technology.

Problem
Transporting content, all kinds of content, is especially tricky. ArtPad aims to give users a device that is easily portable and let's people communicate via written communication and sketching but by still letting users doing in a naturalistic way and in just about any environment.

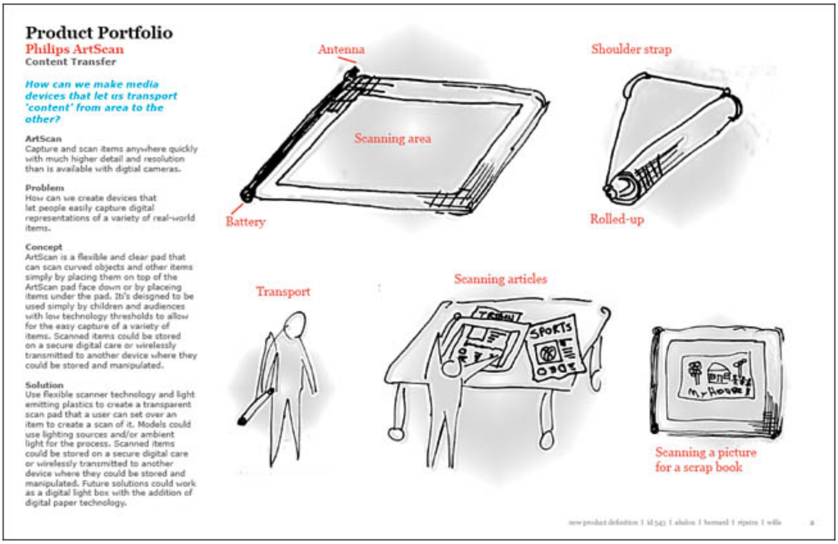
Concept
ArtPad is a large digital paper work surface that also contain integrated a pen-based computer and tablet PC into a device that can access a network via any network enabled device, such as a cellphone or WiFi Network.

Solution
Using digital paper and foldable LCD technologies with existing WiFi and wireless infrastructure can create a device that let's some one video conference and sketch and collaborate with users in any environment.

Digital paper technologys could let people 'write' emails or text chats and then simply select certain areas of the page with their pen to send the note.

A meeting at a public venue could be captured on the board and quickly sent to someone else.

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.



Product Portfolio

Philips ArtScan

Content Transfer

How can we make media devices that let us transport 'content' from area to the other?

ArtScan
Capture and scan items anywhere quickly with much higher detail and resolution than is available with digtial cameras.

Problem
How can we create devices that let people easily capture digital representations of a variety of real-world items.

Concept
ArtScan is a flexible and clear pad that can scan curved objects and other items simply by placing them on top of the ArtScan pad face down or by placeing items under the pad. Iti's deisgned to be used simply by children and audiences with low technology thresholds to allow for the easy capture of a variety of items. Scanned items could be stored on a secure digital care or wirelessly transmitted to another device where they could be stored and manipulated.

Solution
Use flexible scanner technology and light emitting plastics to create a transparent scan pad that a user can set over an item to create a scan of it. Models could use lighting sources and/or ambient light for the process. Scanned items could be stored on a secure digital care or wirelessly transmitted to another device where they could be stored and manipulated. Future solutions could work as a digital light box with the addition of digital paper technology.

Antenna
Scanning area
Battery
Shoulder strap
Rolled-up
Transport
Scanning articles
Scanning a picture for a scrap book
new product definition | id 543 | abalos | bernard | ripstra | wille
2

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