A PROJECT GUIDE TO UX DESIGN

FOR USER EXPERIENCE DESIGNERS
IN THE FIELD OR IN THE MAKING

SECOND EDITION

RUSS UNGER AND CAROLYN CHANDLER

Contents

INTRODUCTION xiii

CHAPTER 1: The Tao of UXD 1
What Is User Experience Design? 3
The Broad Definition 3
Don't Forget the Tangible 4
Our Focus 5
About UX Designers 6
Where UX Designers Live 7
Let's Get Started! 8

CHAPTER 2: The Project Ecosystem 9
Identify the Type of Site 10
Brand Presence 11
Marketing Campaign 14
Content Source 16
Task-Based Applications 18
E-Commerce Sites 19
E-Learning Applications 20
Social Networking Applications 21
Mobile Sites and Applications 21
Choose Your Hats 30
Information Architect 30
Interaction Designer 31
User Researcher 32
Other Roles You May Play or May Need 35
Building a Network of User Advocacy 41
Understand the Company Culture 42
History 43
Hierarchy 45
Logistics 46
Pulling It Together 47

CHAPTER 3: Proposals for Consultants and Freelancers 49
Proposals 50
Creating the Proposal 51
Title Page 52
Revision History 53
Project Overview 54
Project Approach 55
Scope of Work 57
Assumptions 57
Deliverables 58
Ownership and Rights 59
Additional Costs and Fees 60
Project Pricing 61
Payment Schedule 62
Acknowledgment and Sign-Off 63
Statements of Work 65

CHAPTER 4: Project Objectives and Approach 67
Solidify Project Objectives 68
How Can a UX Designer Help? 71
Understand the Project Approach 73
Waterfall Approach 74
Agile Approaches 75
Modified Approaches 80
How Does the Approach Affect Me? 81

CHAPTER 5: Business Requirements 83
Understand the Current State 86
Heuristic Analysis 86
Gather Ideas from Stakeholders 90
Outline Responsibilities 91
Gather the Right Stakeholders 92
Create a Plan for the Meetings 94
Sales: Requirements-Gathering Meeting 94
Run the Meetings Effectively 96
Coalescing Requirements 98

CHAPTER 6: User Research 101
Basic Steps of User Research 102
Define Your User Groups 102
Create a List of Attributes 103
Prioritize and Define. 105
Choosing Research Techniques 107
How Many Research Activities Can I Include? 110
User Interviews 111
Contextual Inquiry 114
Surveys. 118
Focus Groups 121
Card Sorting 124
Usability Testing 127
After the Research. 128

CHAPTER 7: Personas 129
What Are Personas? 130
Why Create Personas? 130
Finding Information for Personas 131
Creating Personas. 131
Minimum Content Requirements 134
Optional Content 137
Advanced Personas 139
Guerrilla Personas: The Empathy Map 141
Final Thoughts on Personas 143

CHAPTER 8: Content Strategy 145
Why Do You Need Content Strategy? 146
When Do You Need Content Strategy? 147
Who Does Content Strategy? 149
How Long Does Content Strategy Last? 149
This Sounds Familiar... 150
Tools of the Trade 152
The Artifacts 152
What is the One Artifact You Need? 156
Additional Resources 156
Things to Look Out For 160

CHAPTER 9: Transition: From Defining to Designing 163
Ideate and Visualize Features 165
The Basic Process of Storyboarding 166
Facilitate the Prioritization Process. 169
Maintain a Good Tension 173
The Development Advocate 175
Managing Conflict During Prioritization 177
Plan Your Activities and Documentation 181

CHAPTER 10: Design Principles 185
Visual Design 186
Unity and Variety. 187
Hierarchy and Dominance 189
Economy of Elements 191
Proportion and Balance. 193
Interaction 196
Associations and Affordance 197
Economy of Motion. 200
Response. 202
Psychology 205
The Effect of Attractive Design. 205
Flow & Game Design. 207
Social Proof. 212
Creating Your Own Guiding Principles 215

CHAPTER 11: Site Maps and Task Flows. 219
Tools of the Trade 221
Basic Elements of Site Maps and Task Flows. 222
Page 222
Pagestack 222
Decision Point. 223
Connectors and Arrows 223
Conditions. 224
Common Mistakes 224
Sloppy Connections. 225
Misaligned and Unevenly Spaced Objects 225
Poorly Placed Text 226

Lack of Page Numbering 226
The Simple Site Map 228
Advanced Site Maps 228
Breaking the Site Map Mold 230
Task Flows 231
Taking Task Flows to the Next Level 234
Swimlanes 234

CHAPTER 12: Wireframes and Annotations 237
What Are Annotations? 239
Who Uses Wireframes? 239
Creating Wireframes 240
Tools of the Trade 241
Start Simply:
Design a Basic Wireframe 244
Getting Started 245
The Wireframes and Annotations 246
Creating Wireframes: A Sample Process 249
What is This Sketching You Mention? 250
Into the Digital: Wireframes 251
Into the Digital: Visual Design 253
Hey, What About This Responsive Design Stuff I Hear About? 254
Wireframes Vs. Prototypes 256
Which Design Is Right? 256
A Final Note on Presenting Wireframes 257

CHAPTER 13: Prototyping 259
How Much Prototype Do I Need? 260
Paper Prototyping 261
Digital Prototyping 262
Wireframe vs. Realistic Prototypes 263
HTML vs. WYSIWYG Editors 264
Additional Tools for Prototyping 274
Working with a Developer 275
Prototype Examples 276
What Happens After Prototyping? 278

CHAPTER 14: Design Testing with Users 279
Exploring Visual Design Mock-Ups. 283
Choosing a Design Testing Approach 284
Qualitative Research vs. Quantitative Research 285
In-Person Research vs. Remote Research 286
Remote Research Considerations 287
Moderated Techniques vs. Automated Techniques. 288
Usability Testing. 292
Planning the Research. 295
Recruiting and Logistics 299
Writing Discussion Guides. 304
Facilitating 306
Analyzing and Presenting Results. 308
Creating Recommendations. 310

CHAPTER 15: Transition: From Design to Development and Beyond 311
Almost Done... 312
Visual Design, Development, and Quality Assurance 312
Design Testing with Users (Again) 315
10, 9, 8, 7, 6, 5, 4, 3, 2, 1 ... Launch! 315
Personal Advantage 316
Support 316
Network Opinion 317
Postlaunch Activities. 317
Postlaunch Analytics 318
Postlaunch Design Testing with Users (Again, Again) 319

CHAPTER 16: A Brief Guide to Meetings 321
The Agenda 323
Meeting Rules 325
After the Meeting 328
Dealing with Nonconformers 330
A Final Note on Meetings. 331
All Done, Right? 332
Just Like Starting Over... 332

INDEX 321

4 Project Objectives and Approach

Know Which Star to Navigate By

One of the keys to a good project is to start the team out with clear project objectives and a well-understood approach. Ideally, the project leadership will have this defined for you—but how do you know if they don't?

This chapter talks about forming project objectives and offers some questions that will help you solidify those goals. We'll also discuss some common project approaches (or *methodologies*) and how they may influence the way you work.

Carolyn Chandler

You're in the project kickoff, with the full team for the first time. The project manager hands out some materials and gives you an overview of the project. By the end of the meeting, ideally, you should have the following information:

- Why is the project important to the company?
- How will stakeholders determine if the project was a success?
- What approach or methodology will the project follow?
- What are the major dates or *milestones* for key points, such as getting approval from business stakeholders?

All of these questions concern the expectations that stakeholders have for the project: *what* the project will accomplish and *how* they will be involved in it. The first two questions pertain to the project's objectives and the last two to the project's approach.

A *project objective* is a statement of a measurable goal for the project. Let's talk about objectives in more detail.

Solidify Project Objectives

Objectives are an important focusing lens that you'll use throughout the project. They should spring from the client company's overall business strategy, so the project objectives should be in line with the strategic initiatives within the company. For example, if there is a strategic initiative to appeal to a new group of prospective customers (called a *market*), the site or application you're creating may be an effort to provide that market with online access to products and services relevant to them. The objective for that project would then be focused on reaching and engaging that market.

A clear objective resonates throughout a project. It helps you:

- Ask the right questions as you gather ideas from business stakeholders
- Plan research with users and focus your analysis of the results
- Detail the ideas gathered from stakeholders and users and convert them into a consolidated list of project requirements
- Prioritize those project requirements based on their value to the company

- Create effective interaction designs
- Manage requests for changes to the design once development begins
- Focus efforts during deployment activities (such as training and communications to users about the new site or application before and during its launch)
- Determine whether you've met the needs of the client company, once the project is launched

When you start a new project, you probably have project objectives from the project's sponsor (the business stakeholder who has direct responsibility for the success of the project), as well as a set of project-related requests coming from business stakeholders and from customers, but they all may be a bit fuzzy (**Figure 4.1**). Your goal is to clarify these into solid statements that you can use as a yardstick for the project's success.

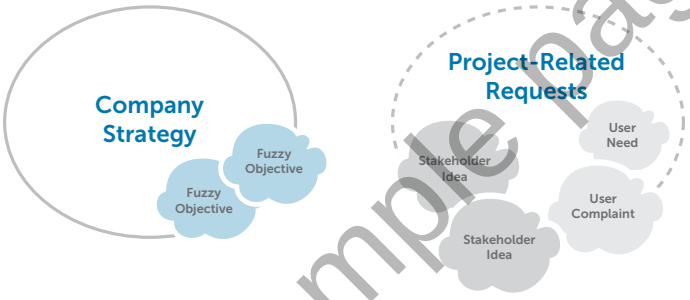

Figure 4.1 *Fuzzy objectives, ideas, and needs*

A solid objective has the following characteristics:

- **Easy to understand.** Avoid insider terminology
- **Distinct.** Avoid vague statements; instead, use wording that seems like it will be useful when you're prioritizing requirements
- **Measurable.** Make concrete statements that you can set an independent measurement against to determine your success

As you define a fuzzy objective, making it clear and measurable, it becomes a solid objective that you can base decisions on (**Figure 4.2** on the next page).

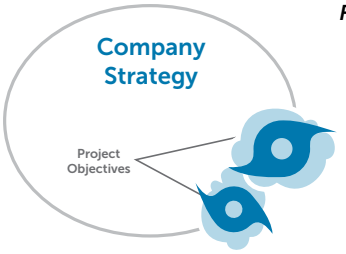

Figure 4.2 *Objectives being solidified*

You'll hear many statements that could be considered objectives. Analyzing fuzzy ones such as those below will help you solidify your objectives and communicate more effectively within the project team.

Business Advocate

"Our objective is to become the market leader in industry x."

This is an objective for the entire company, but is too broad for a specific project. Multiple initiatives at the company need to come together to make this happen; any one site or application may *help* with this but will be very unlikely to be able to handle the entire burden—unless the entire company is about this one site or application and it ends up being wildly successful.

Business Advocate

"Our objective is to generate excitement among our customer base."

This one is better, because a site or application could have an impact on this, but it's still too vague. Why is it important to generate excitement? How does that excitement translate into meeting a business need? And how can you tell if you've been successful?

Business Advocate

"Our objective is to increase the amount of traffic on our website."

Now we're getting there. This one is easy to measure, but it's too focused on an intermediate step. Suppose you do generate more traffic: It may not help you if people don't perform the actions you want once they get there.

Fuzzy objectives can give you a sense of a client's desires and larger goals. From these you can craft more solid project objectives, such as

- Increase the revenue from online sales by 10 percent
- Increase the revenue from online advertising by 20 percent
- Increase the number of current and potential customers in our customer database to at least 20,000
- Deliver highly rated and highly referenced content to our primary users (Note that this one requires some work to decide how to measure "highly rated" and "highly referenced," but the elements are there to build from)

Each of these can be measured and affected by your project. They can also map pretty closely to your designs and the features offered. For example, it's very common to offer an online newsletter as a way to meet an objective of growing the customer database: To deliver the newsletter you'll need to capture customer e-mail addresses, which will be added to the database. Objectives may also bring out new requirements. For example, if you're measuring success by the average rating given to articles on your site, you'll need a feature that allows users to give ratings. In these ways, objectives help you focus as you gather ideas for the site, and these may later become project requirements.

If there are multiple objectives, be sure to create a prioritized list with your business sponsor and project team. Objectives sometimes conflict with each other during design, and the team will need to know what takes precedence. The final prioritized list of objectives should come from your project sponsor, but you can be a key part of the discussion. Let's talk about how.

How Can a UX Designer Help?

If you find the project objectives are unclear at the beginning of a project, you can bring your facilitation skills to bear. Help the project team understand the business-related context of the project by holding a workshop with key stakeholders (see the next chapter for more on identifying the right stakeholders). Your goal in this session, which usually lasts two to four

hours, is to bring out information on the company's strengths, weaknesses, opportunities, and threats. Called a *SWOT analysis*, this is a common business analysis technique and one way to discuss a company's position in the market. You can also use this time to discuss the company's competition.

Understand Strengths and Weaknesses

The *SW* in a SWOT analysis are the company's current strengths and weaknesses as they pertain to the project. Strengths and weaknesses could include internal processes as well as external perceptions—and often they influence each other. For example, a company with a large research and development (R&D) department could have access to a large source of original research that is published (a strength), but there may be no one to help make that content more accessible to the average user, leading to the perception that the company is "too academic" (a weakness).

Identify Opportunities and Threats

The *OT* is the future-facing half of the SWOT. Considering the things that differentiate the company from its competitors, what future initiatives could it pursue that will open up a new niche or strengthen a current one? What situations could threaten those plans?

For example, our R&D company may decide to hire writers to publish more accessible feature articles around its original research (an opportunity), but if the current site toolset doesn't have robust content-management features, the publishing process may be prohibitively slow. That could give competitors a chance to respond more quickly (a threat).

Compare Competitors

What is the company's main competition? Who are the competitors for the site being developed? They can be different, especially for large companies or brand new sites.

Are there sites that aren't necessarily direct competitors but that represent interesting models to consider? You can learn a lot from reviewing other e-commerce sites to see whether and how they sell what you're selling.

Pull It Together

SWOT and competitors are good topics to discuss at the same time because they interact with each other. It's hard to talk about future threats without knowing who your competitors are—and once you start talking about future opportunities, new competitors may come to mind.

Once you have a full picture here of the company's competitors and SWOT, your project objectives—as well as the overall fit of your project within the company strategy—should become easier to define, and the priorities among them should become clear.

Solidifying project objectives helps you understand expectations of what the project is going to accomplish. Next, let's talk about expectations concerning how the project will be run. Understanding the project approach will help you collaborate effectively and involve the right people at the right time.

Understand the Project Approach

Knowing the overall approach, or *methodology*, of a project is an important part of understanding when and how you'll be involved and how you should be involving others, such as your project team and business stakeholders.

Sometimes there seem to be as many project approaches as there are projects. How to choose the right approach for a project is a large topic in itself. The methodology you choose can depend on many things, including the structure and location of the project team, the technologies being used on the project, and the degree to which collaboration is a part of the company's culture. For the purposes of this book, we're assuming that you've joined a project where the approach has largely been determined by those responsible for the project's success, such as the project sponsor and project manager. In this situation, your main goal will be to understand the approach and help make it effective for the business stakeholders and your users.

Here we'll focus on two of the most common types of approach, as well as a third that shows a possible variation you might encounter on a project. The important thing to note is that most approaches involve the same steps:

- **Plan** the overall strategy, approach, and team structure
- **Define** the project requirements

- **Design** interaction and visual concepts and evolve them into detailed specifications
- **Develop**, test, and refine the solution
- **Deploy** the solution via messaging, training, and a planned launch
- **Extend** the project by making recommendations for improvements

The names for these steps may vary, as may the degree to which they overlap and the way information is documented. But the general activities in each step are common to most projects and to all three models presented here.

Waterfall Approach

A *waterfall approach* (**Figure 4.3**) involves treating the steps of a project as separate, distinct *phases*, where approval of one phase is needed before the next phase begins. For example, the Design phase does not begin in earnest until requirements have been approved by business stakeholders, who sign off on one or more requirements documents at the end of the Define phase.

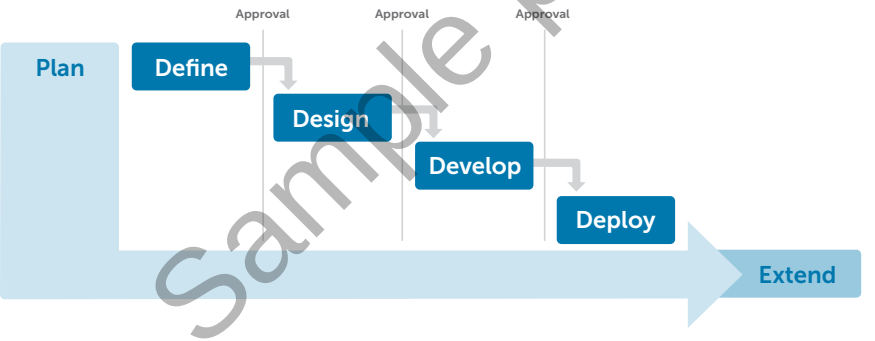

Figure 4.3 *Example of a waterfall approach, where each phase "falls" into the next*

The problem with a pure waterfall approach is that it assumes that each phase can be completed with minimal changes to the phase before it. So if you come up with new requirements in the Design phase, which is common, you must suggest changes to documents that were approved at the end of the Define phase, which can throw off the plan and the schedule.

Agile Approaches

Because change is constant, project teams are continually looking for more flexible approaches than the waterfall model. Many methodologies follow a more fluid approach, with some steps happening alongside each other; for example, versions of the website could be released on a rapid, iterative schedule using an *agile* or *rapid development* approach (**Figure 4.4**). An agile approach generally has a greater focus on rapid collaboration and a reduced focus on detailed documentation and formal sign-off.

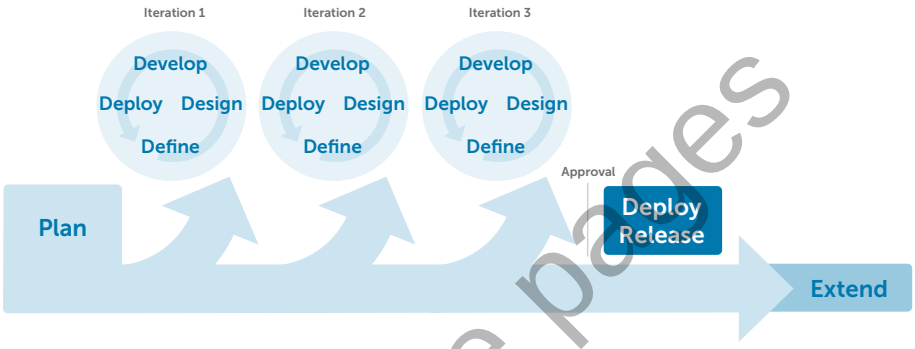

Figure 4.4 *Example of an agile approach*

A true agile approach (following the best practices developed by members of the Agile Alliance, for example) calls for small teams whose members are located next to each other physically, with little focus on defining formal roles between team members. Working this way allows a very high degree of collaboration, which reduces the need for heavy documentation between the stages of design, development, and testing. A team member can pose a question, come to the answer together with other team members during a quick whiteboarding session, and implement a solution without the delay of detailed documentation and approval. Stakeholder reviews occur with a fully functioning system when one of the many iterations is released, and the resulting input is taken into account as the next iteration is planned. (*Iterations* are draft versions of a particular site or application and may also be called *sprints*.)

Designers moving to an agile approach for the first time often face a conundrum. How do you go from a waterfall approach (which favors detailed documentation and sign off, taking weeks or months per phase), to an agile

approach (which favors conversations and quick decision making over the course of days or weeks) and still make time for design thinking and user research? To see how some designers have made the transition, let's dive deeper into a particular kind of approach called Lean UX.

Lean UX

Lean UX is an agile project approach that's well-suited to products being developed in the face of great uncertainty (as most products for startups are). It reduces waste in the project's process by removing effort spent on features that don't really matter at the time of each iteration. For example, spending time designing an entire set of categories and subcategories of products may be wasteful if the team has not yet proven that they're offering products that their target users are willing to purchase.

Some of the principles of Lean UX include:

- **A focus on validated learning.** Iterations of the product are not seen as simply working versions of the product, but as the presentation of a hypothesis that can be tested with users. The goal is to *learn* as quickly as possible, by validating design decisions with customers and incorporating the subsequent changes that will help the team learn the next important lesson.

The Origins of Lean UX

Eric Ries developed an approach called the Lean Startup after studying Toyota's lean manufacturing processes, and Steve Blank's Customer Development model, which emphasizes the need for startups to have an early focus on customers. A Lean UX approach builds on this direction with its customer inclusion, its reduction of waste in the process, and its definition of product iterations as experiments.

Entrepreneurs can find out how experimentation and a focus on learning can help startup teams in the face of uncertainty, in *The Lean Startup* by Eric Ries (Crown Business, 2011).

- **A continuous loop of Build—Measure—Learn.** Lean processes prioritize the building of a testable iteration of the product as quickly as possible, in order to test assumptions that the team is making about how users will react to the product. Tests fail or succeed based on qualitative user feedback during research, and on quantitative measures that are put in place to track success. These measures should pull from actual user behavior—for example, the number of registrations for a site, the number of products purchased, and so on. Care should be taken that the measures put in place really test the assumptions of that iteration. For example, if people are registering for your site, but not taking any important actions in it afterwards, you've just learned that your post-registration experience needs to be more compelling! Incorporate that learning into your decisions on what goes into the next build, and complete the loop (**Figure 4.5**).

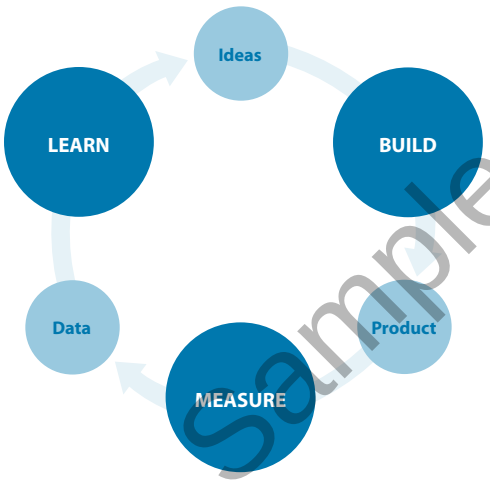

Figure 4.5 *Lean approaches focus on a loop of Build—Measure—Learn. The process is meant to increase the speed by which teams cycle through the loop, maximizing learning and allowing for quicker adjustments in strategy based on customer response.*

- **The importance of developing the Minimum Viable Product (MVP) at each iteration.** In a lean process, the focus is on testing a hypothesis about user behavior, rather than building a fully functioning product at each iteration. Teams don't need to create a fully functioning digital version of their product to test a hypothesis (although a more robust digital version will eventually exist after several cycles, if things are going well on the validation front). Especially in beginning stages teams can focus on developing a Minimum Viable Product, which Eric Ries defines as "that version of product that enables a full turn of the Build—Measure—Learn

Ask the Experts: Jeff Gothelf

Jeff Gothelf is the founder of Proof, a product innovation and design studio, and is the author of Lean UX: Getting Out of the Deliverables Business *(O'Reilly, 2012.)*

You've said that Lean UX is an approach that gets designers out of the deliverables business. But many UX designers use deliverables—documentation like detailed wireframes—in order to communicate design recommendations to their teams. Does Lean UX skip this step?

Deliverables aren't skipped in Lean UX. They're still used, to the extent necessary, to communicate to target audiences like stakeholders, developers, and customers. But they're not the focus of the project, and they don't create the bottleneck they would in waterfall where the team sits back and waits for the design to be "done" and then signed off before anyone else begins working. Also, there's still a large amount of work that you're doing as a designer outside of the actual documentation, because the goal is to validate your design hypotheses as soon as possible—so the more tests you're running with customers, the more right you'll be in the product you're building. The trick to getting to a level of efficiency with it is to build a shared understanding across your entire cross-functional team. The more you're out there with your team having conversations with them, discussing what you're seeing and why you think that is, the more they get a clear sense of why you're making the design decisions you're making and what direction the design is heading in. When they have a clear sense of that, they need less documentation in order

loop with a minimum amount of effort and the least amount of development time." For example, elements of the experience that should eventually become automated—like confirmation emails when a purchase has been made—may be completed more manually by a team member while the test is being run in an earlier iteration.

- **A move away from formal deliverables and detailed documentation.** This is consistent with the overall agile approach. Deliverables like detailed wireframes and use cases, which may become replacements for direct communication and fast implementation of ideas, are removed from the process in favor of faster methods like sketching. Conceptual wireframes

to start building those experiences because they have the foundation and shared understanding from which to work.

How does a Lean UX approach affect the role of a UX designer on the project?

It assumes a leadership quality to the UX role on the team. The Lean UX process forces you to constantly communicate out to the team and solicit feedback from them. But it's not design by committee, either. Yes, you're out there soliciting feedback from your team early and often, but it isn't your job to take all their feedback and make sure it all makes it into the next iteration of the design. It's your job to prioritize the feedback based on what's needed to get to the next level of learning with your customers. Then you incorporate that and communicate the result out to the team so they're aware of what design decisions are being made, and why.

Is there anything the team should have from the beginning to ensure a Lean UX approach is successful?

There needs to be a freedom to fail in the organization. If people feel they have to get it right the first time, right out of the gate, than the whole process fails. This is a hypothesis validation process, and by the very nature of it you're going to come up with some wrong answers. The idea is to figure out what those wrong answers are as quickly as possible and minimize the wasted effort going down those paths. If you don't have the freedom to be wrong in your organization, then you will not be able to execute this process with any kind of success.

are often still used, but are meant to illustrate quickly as an aid to communication, and do not "live on" as records of design decisions.

When an agile approach is working as it's designed to, it's a beautiful thing. At most companies and within most consulting engagements, however, teams rarely follow a pure agile approach. In part, this is because companies often have distributed teams and remote workers, which makes it difficult to maintain the high degree of collaboration needed to take best advantage of the pure agile approach. However, a greater prevalence of virtual collaboration tools and digital sketching tools makes this distributed agile approach increasingly possible, as long as teams commit to clear communication, high availability, and effective decision-making.

Surfing

LUXr is a company founded on Lean UX practices. You can see an introduction to Lean UX and more detail on its principles on their site at http://luxr.co/lean-ux/9-principles-for-lean-ux/.

All UX designers should be focused on reducing waste in the process, and prioritizing communication over documentation regardless of the specific approach of the team, as Whitney Hess points out here: http://whitneyhess.com/blog/2011/02/27/why-i-detest-the-term-lean-ux/

Modified Approaches

Many projects try to follow an approach that marries elements of waterfall and agile approaches, with enough structure and documentation to reduce the risks posed by distributed teams and turnover of team members, but enough collaboration and iteration to respond to changes in a relatively nimble way. For example, a project may follow a waterfall model but include an overlap in phases so that there are key collaboration points from team to team. This allows potential changes to surface earlier in each phase. This may also include an early release (such as a beta release to a particular user group) with a shorter iteration cycle. Feedback from that release can then be incorporated before the full deployment of the new site. (**Figure 4.6**)

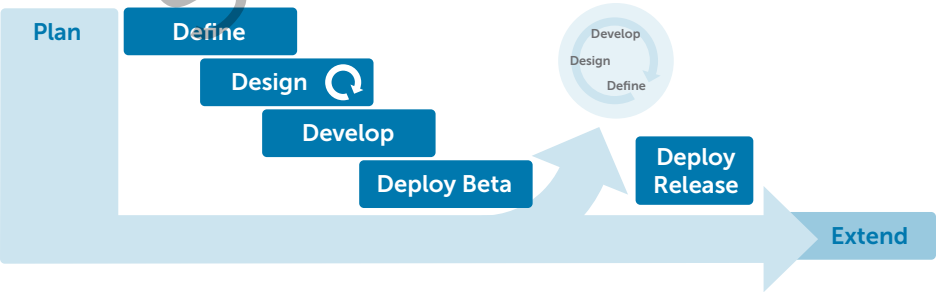

Figure 4.6 *Modified waterfall with beta release*

Notice the smaller iterations within the Design phase in Figure 4.5. That's one of the greatest values you bring to your team as a UX designer. Tools such as wireframes (Chapter 12) and prototypes (Chapter 13) can allow you to gather feedback on quick iterations of ideas, before a lot of development time has been put in.

This book loosely follows a modified waterfall approach like the one shown in Figure 4.6. However, many of the topics covered here will apply to your project regardless of the specifics of your approach, because the basic activities behind them—defining and designing, for example—are still necessary.

Deep Diving

If your project is using an agile approach, you'll have unique needs during requirements gathering, such as the writing of "user stories" as a way to capture requirements. We recommend *User Stories Applied: For Agile Software Development* by Mike Cohn (Addison-Wesley Professional, 2004).

How Does the Approach Affect Me?

Knowing your approach helps you understand a number of things:

- **What questions you should be asking, and when.** For example, if you're working with a pure waterfall approach, you'll need to put in extra effort to make sure the requirements captured in the Define phase contain all the information you need for the Design phase. (We'll be discussing requirements in the next chapter.)
- **Expectations on how project team members will collaborate and how close that collaboration will be.** For example, an agile approach requires very close collaboration. A waterfall approach may involve individual work most of the time, with touchpoints once or several times per week.
- **The level of detail needed in your documentation and the level of formality.** Documents submitted at sign-off points need to be formal, almost like legal contracts. Typically, you'll need more formal documents

in a waterfall approach, where sign-off is required before you move on to the next phase. However, you may also have some formal sign-off documents when using an agile approach—for example, to capture information at major decision points, such as when a particular iteration is prepared for full release and deployment.

- **Important milestones that involve approval from stakeholders and deployment to different groups.** The approach will determine what different audiences need to provide at various points in the project, including approvals from stakeholders at sign-off points and feedback from potential users during a beta release.

Now that you've solidified your project objectives and gained an understanding of the project approach, in the next chapter we'll start with the primary work in the Define phase: gathering requirements.