CHAPTER

2

The Wheel: UX Processes, Lifecycles, Methods, and Techniques

Highlights

- The need for process.
- Basic process components for UX:
  - The UX design lifecycle, the idea of the Wheel.
  - Lifecycle activities.
  - UX design lifecycle process.
  - Lifecycle subactivities.
  - UX design methods.
  - UX design techniques.
- The fundamental UX lifecycle activities:
  - Understand Needs.
  - Design Solutions.
  - Prototype Candidates.
  - Evaluate UX.
- UX design techniques as life skills.
- Choosing and appropriating UX processes, methods, and techniques.

2.1 INTRODUCTION

2.1.1 Where Are We Heading?

Now that you are getting into the process-oriented chapters, it is important to remember that this is all about UX design, not at all about software. What we produce is a UX design, usually represented as a prototype. These designs will be realized in software by developers, software engineers, and programmers, using a corresponding software engineering lifecycle (Section 29.3).

2.1.2 The Need for Process

Long ago, software engineering folks recognized that having a process is necessary for developing complex systems, and it's something they invest enormous resources (Paulk, Curtis, Chrissis, & Weber, 1993) into defining, verifying, and following. On the UX side, Wixon and Whiteside were ahead of their time while at Digital Equipment Corp in the 1980s and put it this way (Whiteside & Wixon, 1985), as quoted in Macleod, Bowden, Bevan, and Curson (1997):

> *Building usability into a system requires more than knowledge of what is good. It requires more than an empirical method for discovering problems and solutions. It requires more than support from upper management and an openness on the part of all system developers. It even requires more than money and time. Building usability into a product requires an explicit engineering process. That engineering process is not logically different than any other engineering process. It involves empirical definition, specification of levels to be achieved, appropriate methods, early delivery of a functional system, and the willingness to change that system. Together these principles convert usability from a "last minute add on" to an integral part of product development. Only when usability engineering is as much part of software development as scheduling can we expect to regularly produce products in which usability is more than an advertising claim.*

Without guidance from a UX design process, practitioners are forced to make it up as they go along. If you have seen this happen in your projects, you are not alone. An approach without a process will be idiosyncratic. What practitioners do will be dictated and limited by their own experience. They will emphasize their own favorite ways to do things while other important process activities fall through the cracks. Finally, as Holtzblatt (1999) puts it, following a process for product development is a useful hedge against "the relentless drive of an organization to ship 'something' by a given date."

2.1.3 What Do You Get by Having a Process?

Process is a guiding structure. A process is a guiding structure that helps both novices and experts deal with the complex details of a project. A process enforces a systematic approach, bringing order to what could be very chaotic, especially within a large and complex project.

Process acts as a framework to ensure novice designers are on track to a quality product and on the path to becoming experts. For experts, a process acts as a checklist to make sure they do not miss any important aspects of the problem in the heat of productivity. It helps designers answer questions such as "Where are we now?" and "What can/should we do next?"

Process offers reliability and consistency. A documented process offers a way to use basically the same approach from project to project and from one team member to another.

Process provides scaffolding for learning. Design is all about learning. A process provides a fabric on which you can build a knowledge base of what you have learned, applying organizational memory from similar previous efforts to incorporate lessons learned in the past. This fabric, in turn, helps train novice designers in the ways of UX at that organization or in the discipline at large.

Process provides a shared conception of what you are doing. A documented process lets everyone know how a product or system (software plus UX) is being developed. Process also helps your team coordinate and communicate by externalizing the state of development for observation, measurement, analysis, and control—otherwise, communication among the project roles about what they are doing is difficult because they don't have a shared concept of what they should be doing.

Reliability, UX Evaluation

Refers to the repeatability of a UX method or technique from one UX practitioner to another and from one time to another for the same practitioner. Section 21.2.5.2.

2.2 THE BASIC PROCESS COMPONENTS FOR UX

2.2.1 UX Design Lifecycle

A lifecycle is just what it says: *It's a cycle of the life of a UX design, from inception to deployment and beyond.*

2.2.2 UX Lifecycle Activities

Lifecycle activities are the high-level things you do during a lifecycle (Fig. 2-1):

- Understand Needs (of users).
- Design Solutions.
- Prototype Candidates (for promising designs).
- Evaluate UX.

In Section 2.3, we'll follow up on these fundamental lifecycle activities in more detail.

2.2.3 UX Design Lifecycle Process

A UX lifecycle process is a representation of how you put the lifecycle activities together in a sequence over time and how the lifecycle activities—the boxes of Fig. 2-1—are connected in the flow of the process, usually represented in the form of a flow

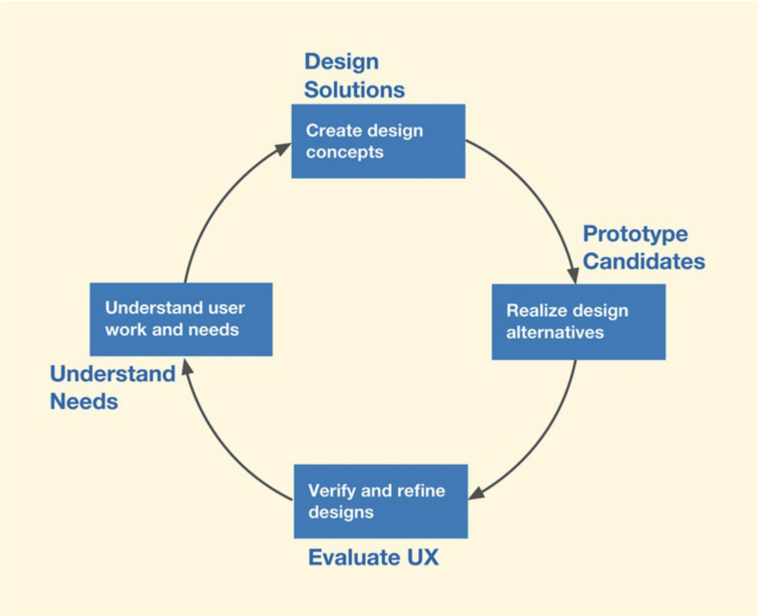
Design
Solutions
Create design concepts
Prototype
Candidates
Realize design alternatives
Verify and refine designs
Evaluate UX
Understand user work and needs
Understand
Needs

Fig. 2-1

The basic UX design lifecycle process in terms of elemental lifecycle activities.

chart diagram. Fine distinctions are unnecessary here, so we use the terms "process," "lifecycle," and "lifecycle process" more or less interchangeably.

2.2.4 The Wheel: A Model of the UX Lifecycle

If we expand this abstract cycle a bit to include feedback and iteration, we get a kind of UX lifecycle template of Fig. 2-2, which, as an analogy, we call "the Wheel." This is because it goes around in circles, and with each rotation it brings you closer to your destination.

This basic picture is the blueprint for the process common to almost any kind of design; it applies whether the design scope is just a small piece (chunk) of a product/system or the whole system.

2.2.5 Lifecycle Subactivities

Each lifecycle activity is substantial enough to be described in terms of its own set of subactivities. *Lifecycle subactivities are the things you do during a single lifecycle activity.*

Example subactivities for the Understand Needs (Section 2.3.1) lifecycle activity include:

- Data elicitation.
- Data analysis.

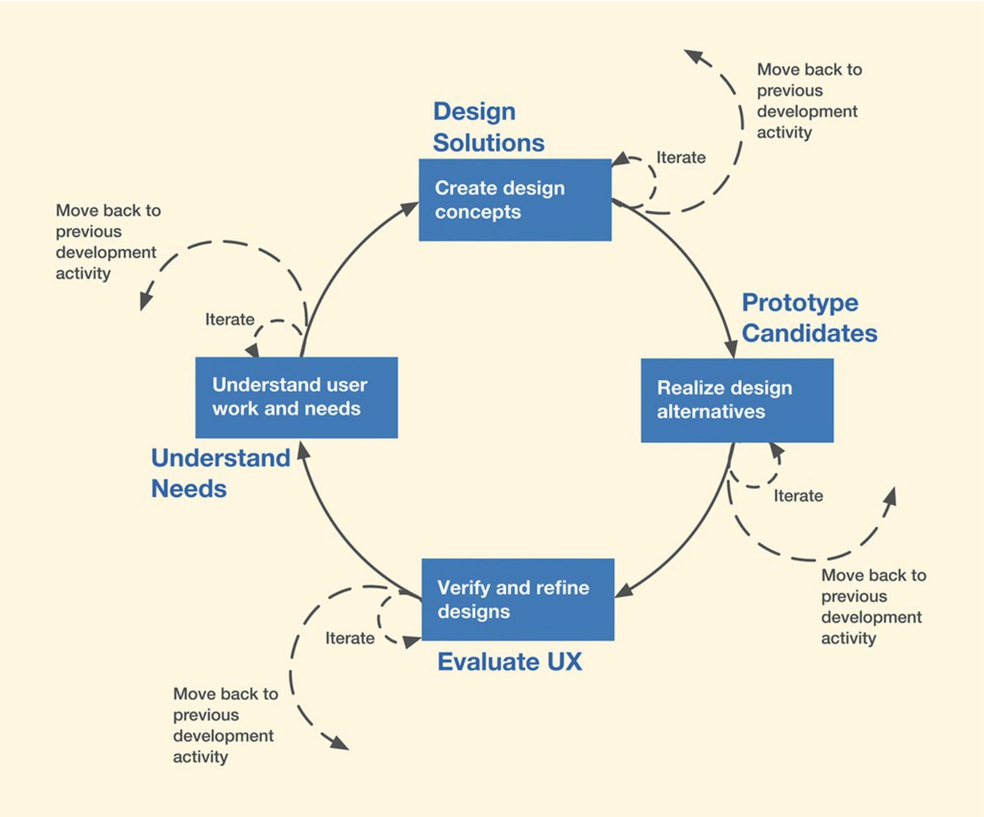
Design Solutions
Create design concepts
Iterate
Move back to previous development activity
Prototype Candidates
Realize design alternatives
Iterate
Move back to previous development activity
Evaluate UX
Verify and refine designs
Iterate
Move back to previous development activity
Understand Needs
Understand user work and needs
Iterate
Move back to previous development activity

Fig. 2-2

The Wheel: A lifecycle template illustrating the process part of this book.

- Data modeling.
- Requirements extraction.

2.2.6 UX Methods

In our vocabulary, *a method is a way one can carry out the whole or part of a given lifecycle activity or subactivity*. An example of a method for the Understand Needs lifecycle activity is Usage Research (which we will describe in Part 2 of this book).

2.2.7 UX Techniques

Finally, in our usage, *a UX technique is a specific detailed practice you can use to perform a step within an activity, subactivity, or method*. A given UX design technique could be useful in a variety of different lifecycle activities and is not tied to a specific UX method. Examples of UX techniques for the data elicitation activity within the usage research UX method are:

- User interviews.
- Observation of users at work.

2.2.8 A Hierarchy of Terms

To distinguish the levels among descriptions of ways things can be done in UX design, we end up with a kind of hierarchy of terminology:

- Process, or UX lifecycle process.
- UX lifecycle activities and subactivities.
- UX methods.
- UX techniques.

These terms, despite their common appearance in the literature, are usually vague and ill-defined within the discipline. We have chosen to shade the definitions of these closely related terms to reflect the meanings we think are most commonly understood within this loose kind of hierarchical relationship, examples of which are shown in Table 2-1.

2.3 THE FUNDAMENTAL UX LIFECYCLE ACTIVITIES

In this section, we look more in depth at the individual UX design lifecycle activities and subactivities. Much of this book is about these topics.

The four basic UX lifecycle activities of Figs. 2-1 and 2-2 are:

- **Understand Needs**, to understand users, work practice, usage, the subject-matter domain, and, ultimately, needs for the design.
- **Design Solutions**, to create designs as solutions.
- **Prototype Candidates** (of promising solutions) to realize and envision promising design candidates.
- **Evaluate UX**, to verify and refine designs with respect to the user experience they afford.

Waterfall Lifecycle Process

One of the earliest formal software engineering lifecycle processes; an ordered linear sequence of lifecycle activities, each of which flowed into the next like a set of cascading tiers of a waterfall. Section 4.2.

Table 2-1

Informal hierarchy of process, methods, and techniques with simple examples

Lifecycle process	Traditional waterfall process (Section 4.2)
Lifecycle activity	Understand Needs (Part 2)
Subactivity	Elicit usage information (Chapter 7), analyze usage information (Chapter 8), model system or product usage (Chapter 9), codify needs (Chapter 10)
Method	Usage research, surveys, and competitive analyses (for elicit information subactivity); usage research analysis (for analyze information subactivity); flow, sequence, task models (for model usage subactivity); formal requirements (for codify needs subactivity)
Technique	Interviews, observations, affinity diagramming, etc.

For a given iteration, each box of the figure represents a method for carrying out the corresponding lifecycle activity. The choice of which method is to be used depends on the design situation (Section 2.5).

The depiction of UX lifecycle activities in distinct boxes is a convenient way to highlight each activity for discussion and for mapping to chapters in this book. In practice, however, these activities do not have such clear-cut boundaries; there can be significant intertwining and overlap (Section 5.2).

2.3.1 The Understand Needs UX Lifecycle Activity

The Understand Needs lifecycle activity is used to understand the business domain, users, work practice, usage, and the overall subject-matter domain. The most popular method is some variation of usage research and the most rigorous version includes these subactivities, each of which is detailed within its own chapter:

- Data elicitation (Chapter 7): Interview and observe users at work and gather data about work practice, users, usage, and needs.
- Data analysis (Chapter 8): Distill and organize usage research data.
- Data modeling (Chapter 9): Create representations of user characteristics, information flow, tasks, and work environments (for collaboration, sharing, archival, rehearsal, immersion).
- Requirements extraction (Chapter 10): Codify needs and requirements.

In Fig. 2-3, we illustrate the data elicitation subactivity as it is done with possible methods and techniques.

Usage Research

A method for performing the Understand Needs UX lifecycle activity in which users are interviewed and observed in their work context. Pertaining to gathering detailed descriptions of work domain knowledge and existing work practice for the purpose of understanding work activities and user needs to inform design to support the work practice. (Section 7.2).

Immersion

A form of deep thought and analysis of the problem at hand—to "live" within the context of a problem and to make connections among the different aspects of it (Section 2.4.7).

2.3.2 The Design Solutions UX Lifecycle Activity

Design Solutions is perhaps the most important lifecycle activity and the one with the broadest purview. Typical subactivities of this activity change dramatically over time as the project and the product evolve and mature through these basic "stages" (Fig. 2-4):

- **Generative design:** Ideation and sketching to create design ideas (Chapter 14), low-fidelity prototyping (Chapter 20), and critiquing for design exploration (Section 14.4).
- **Conceptual design:** Creating mental models, system models, storyboards, low fidelity prototypes of conceptual design candidates (Chapter 15).
- **Intermediate design:** Developing ecological, interaction, emotional design plans for most promising candidates (Chapters 16, 17, and 18), creating illustrated scenarios, wireframes, medium fidelity mockups of design forerunners, and identifying design tradeoffs to compare design candidates.

Storyboard

A visual scenario in the form of a series of sketches or graphical clips, often annotated, in cartoon-like frames, illustrating the interplay between a user and an envisioned ecology or device (Section 17.4.1).

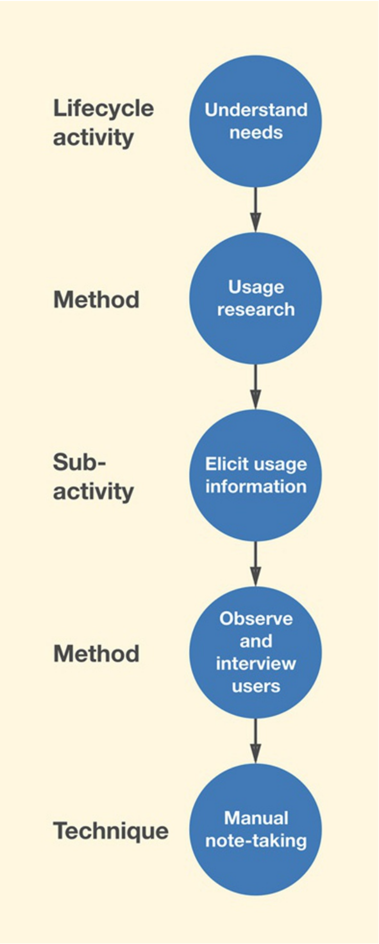

Fig. 2-3
The data elicitation subactivity of the Understand Needs lifecycle activity.

- **Design production:** Specifying detailed design plans for implementation of the emerging design choice (Chapters 17 and 18).

The relative importance of each of these subactivities depends on the design situation, especially the kind of product or system being created.

2.3.2.1 Interpretation of "design": broad versus narrow

A point of potential confusion arises from ambiguity in the way the term "design" is used in UX and other fields. On one hand, the whole of Fig. 2-1 is called the UX design lifecycle. So, one might conclude that this whole diagram depicts the answer to: what is UX design?

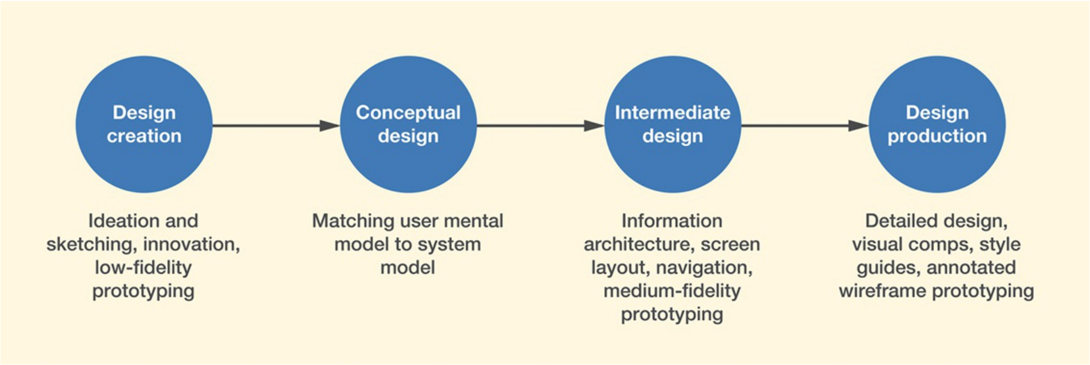
Design creation
Conceptual design
Intermediate design
Design production
Ideation and sketching, innovation, low-fidelity prototyping
Matching user mental model to system model
Information architecture, screen layout, navigation, medium-fidelity prototyping
Detailed design, visual comps, style guides, annotated wireframe prototyping

Fig. 2-4

Typical Design Solutions subactivities.

But the astute reader might note there is also a box at the top called Design Solutions within this lifecycle. Maybe that's what UX design is. In fact, both ways of using the term are useful, but we lack the vocabulary to distinguish them.

We sidestep the ambiguity trap by following the loose conventions of the field and using "design" with both meanings, hoping that context will provide clarity. If context doesn't disambiguate our meaning, we'll call out the specific meaning.

Broad interpretation. In the broad interpretation, "design" is the overall UX design lifecycle process. In simple terms, the UX process is a UX design process.

Narrow interpretation. In the narrow interpretation, "design" is just a single activity within the UX lifecycle and its subactivities are shown in Fig. 2-4. This narrower view also allows us to keep design separate from discussion of other lifecycle activities, such as Understand Needs or Evaluate UX. As a lifecycle activity on its own, design is important enough to have its own definitions, activities, theory, and practice.

The narrow view has led to misconceptions. The narrow view, considered alone, may have contributed to a long-held misunderstanding of the role of UX designers in projects. As Tim Brown (2008, p. 84) says, "Historically, design has been treated as a downstream step in the development process—the point where designers, who have played no earlier role in the substantive work of innovation, come along and put a beautiful wrapper around the idea. Not seeing design, especially UX design, in the broad view, project teams have been slow to invite designers into the whole process."

Different views of design have led to different views of prototyping. The different views at different levels of UX design offer a good way to distinguish the two kinds of prototyping and evaluation that Buxton (2007b) points out. Prototyping, evaluation, and iteration for getting the *right design* is part of the narrow view within the design creation subactivity. Getting the *design right* is the goal of prototyping, evaluation, and iteration within the broader overall UX lifecycle process. In the narrow context, prototyping is used within the design-creation subactivity as sketching and rapid low-fidelity prototypes, while higher-fidelity prototypes occur in the subactivity for the Prototype Candidates activity in the overall UX design lifecycle.

2.3.3 The Prototype Candidates UX Lifecycle Activity

Here, prototyping is a full-fledged lifecycle activity to realize and envision promising design candidates. The main subactivity is to create representations of design to required fidelity in the form of:

- Paper prototypes.
- Wireframes and wireflows.
- Click-through wireframe prototypes.
- Physical prototypes.

Like sketching, prototype building is usually done in parallel with, and in conjunction with, design; a prototype is an extension of the idea of a sketch. As designs evolve in designers' minds, they produce various kinds of prototypes as external design representations. Because prototypes are made for many different purposes, there are many kinds of prototypes, each with its own methods and techniques, as discussed in Chapter 20.

Prototypes are made at many different levels of fidelity, including low fidelity (especially paper prototypes such as printouts of static wireframes, for design exploration and early design reviews), medium fidelity (such as click-through wireframe prototypes), and high fidelity (programmed functional prototypes), and "visual comps" for pixel-perfect look and feel.

2.3.4 The Evaluate UX Lifecycle Activity

This activity is about verifying and refining the UX design to ensure we are getting the design right. Subactivities and possible alternative methods for the Evaluate UX activity to assess, verify, and refine designs might include:

- **Collect evaluation data**: Evaluate designs with empirical or analytic methods to simulate or understand actual usage and produce evaluation data.
- **Analyze evaluation data** (for identifying critical incidents, root causes).
- **Propose redesign solutions**.
- **Report results**.

Methods abound for doing the activities and subactivities of UX evaluation. From lightweight and rapid to thorough and laborious, from full empirical studies to quick and dirty inspections, depending on the design situation. Also, there are different methods and techniques for evaluating the different components of UX: usability, usefulness, emotional impact, and meaningfulness. These will be covered in detail in Part 5.

Inspection (UX)

An analytical evaluation method in which a UX expert evaluates an interaction design by looking at it or trying it out, sometimes in the context of a set of abstracted design guidelines. Expert evaluators are both participant surrogates and observers, asking themselves questions about what would cause users problems and giving an expert opinion predicting UX problems. Section 25.4.

2.4 UX DESIGN TECHNIQUES AS LIFE SKILLS

In the context of this book, UX techniques are used in UX design situations. Because these techniques are also used for problem-solving in our everyday lives, we also think of them as *life skills*—basic generic skills for solving problems, helping designers and nondesigners thrive as human beings.

In addition, some techniques appearing in our process chapters are very specific to UX processes. Here are just a few examples:

- Card sorting (Section 8.6.1): A technique for organizing data collected, for example, in usage research or in UX evaluation so you can understand and make sense of them.
- Think-aloud: A data collection technique used within the lab-based evaluation method for the Evaluate UX lifecycle activity where participants are prompted to verbalize their thoughts and plans as they interact with a design prototype or system.
- Note taking: A technique for gathering raw user data in the *elicit information* subactivity (Chapter 7) of the Understand Needs lifecycle activity that includes variations such as audio recording, video recording, hand written notes, and notes typed on a laptop.

There are many UX design techniques out there and more are being described in the literature and used in practice over time (Martin & Hanington, 2012).

Here we describe the most important techniques, viewed as generic skills, according to our experience as researchers and practitioners in this field. Most should seem familiar because you will have encountered them or used them in other contexts. Here you can think of them as a kind of preview of things that will come up in later chapters.

2.4.1 Observation

***Observation** is the practice of witnessing an ongoing activity with the objective of understanding underlying phenomenon.* Things to look for include exceptions, surprises, generalities, patterns, workflows, sequencing, what works and what doesn’t, problems and barriers, and how people react to problems (or if they do). Observation provides the inputs for reasoning and deduction, but the ability to observe effectively can be elusive.

It can take practice to develop the skill to observe. UX professionals have to train themselves to know when they are seeing something important and not let it slip away because it doesn't register. As George Perkins Marsh (1864, p. 10) said (and Sherlock Holmes persistently demonstrated), "Sight is a faculty; [but] seeing is an art."

Example: Observation of a Car Wash Workflow

Here is an example of the observation technique in a real-life setting, an example about observing a car wash workflow problem. On a recent trip to Buster's Auto Spa, our local car wash, I had to sit in line almost half an hour to get into the car wash. I guess most people would read the paper or listen to music to pass the time. But, being blessed/cursed with UX designer skills, I was curious about the cause of the delay.

In Fig. 2-5, we show a simplified sketch of the entrance to the car wash, pretty much like the entrance to any car wash. Note that this is a practical example of at least two other techniques, abstraction (next section) and sketching (Section 2.4.9).

Most of the time the car wash works well, but occasionally it has a work flow problem. When a customer wants the interior of the car cleaned, the attendants use a large vacuum cleaner mounted in a fixed location at the entrance.

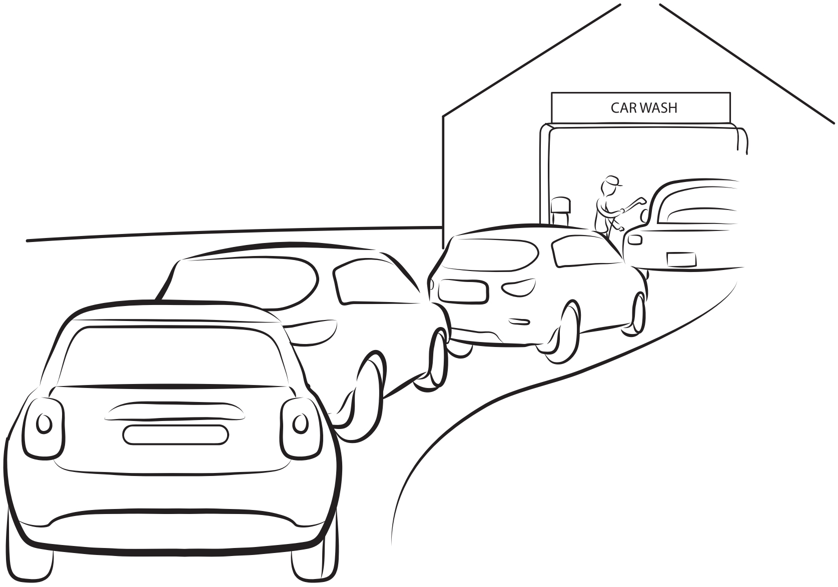

Fig. 2-5

Entrance to Buster's Auto Spa.

Vacuuming can take 10 minutes, during which time no cars are moving through the car wash—bad for the car wash company and bad for customers who have to wait in line behind the car being vacuumed. Customer waiting is especially bad because getting your car washed is something you usually do on the way to something else. Further, the narrow lane leading up to the entrance makes it almost impossible for customers to bail. Of course, I began to think about different designs of the facility that would solve the problem. What would you recommend?

Exercise 2.1: Make Some Deeper Observations

Your first exercise is here. Active learning means learning by doing. The best way to learn the processes described in this book is by doing the exercises. We have organized your participation in the process chapters at three levels: examples for you to follow in the text, a more or less parallel set of exercises to do with a group or on your own, and a set of extensively specified team project assignments (http://www.theuxbook.com/). We start you off here with an easy one.

Observe a customer service counter at Costco or similarly busy department store and prepare findings. Include descriptions of what is taking place, the flow, breakdowns, etc.

2.4.2 Abstraction

Abstraction *is the practice of removing detail irrelevant to a given objective.* "Abstraction is considered to be a key skill underlying most activities in computer science (CS) and software engineering (SE)" (Hazzan & Kramer, 2016). The result is a clearer picture of what is important without the distraction of extraneous matter. In other words, abstraction is the separation of the wheat from the chaff. Abstraction also entails the ability to generalize from an example. You have to be able to comprehend and extract the essence of a particular observed incident or phenomenon as an instance of a more general case or principle.

Consider the case of building a house where you as a designer (architect) are interviewing users (residents) of a house you are going to build. Different users mention different needs:

- User A: I would like to have a fan in the kitchen to clear any cooking smells.
- User B: I like to open my windows to the study and let in fresh air.
- User C: I do a lot of gluing in my workshop and need to have large windows and doors so the chemical smells don't overwhelm me.

All these specific instances can be abstracted under a generic idea of ventilation for the house that will lead to a design that solves all the individual problems in the list

2.4.3 Note Taking

Note taking *is the practice of efficiently capturing descriptions of observations.* It includes a set of techniques for qualitative data collection.

Techniques for note taking include making hand written notes, typing notes on a laptop, recording the essence on audio, or recording on video. However you do it, note taking should be done as an almost subconscious activity without distracting your cognitive processes from the observation activity. This usually means using the simplest means possible, like hand written notes or notes typed on a laptop. I recorded my notes about the car wash problem on a pocket digital audio recorder that I have with me always and transcribed them when I got home.

To be efficient, you must apply abstraction during note taking to capture the essential points while keeping the verbiage to a minimum. Notes can include sketches and/or models, analogies, or any other descriptive mechanism, bringing additional techniques into the mix.

2.4.4 Data/Idea Organization

Data organization *is the practice of sorting data by category to make raw data understandable.* Techniques for data organization include:

- Card sorting.
- Affinity diagrams.
- Mind-mapping: "A mind map is a diagram used to visually organize information. A mind map is often created around a single concept, drawn as an image in the center of a blank landscape page, to which associated representations of ideas such as images, words and parts of words are added. Major ideas are connected directly to the central concept, and other ideas branch out from those."[1]
- Concept mapping: "A concept map or conceptual diagram is a diagram that depicts suggested relationships between concepts. It is a graphical tool that instructional designers, engineers, technical writers, and others use to organize and structure knowledge."[2]

[1] https://en.wikipedia.org/wiki/Mind_map
[2] https://en.wikipedia.org/wiki/Concept_map

Card sorting is commonly used to organize menu structures in desktop applications. Designers list all the actions the system should support and each action is printed on a card. Users are then asked to organize them into groups. Affinity diagrams, hierarchical schemes for organizing larger sets of data, will be covered in detail in Chapter 8. Mind-mapping and concept-mapping are techniques used to externalize ideas and data that are loosely structured and connected.

2.4.5 Modeling

Modeling *is the practice of representing complex and abstract phenomenon along particular dimensions to simplify and aid understanding.* It is a way to explain or categorize aspects of the problem space.

Modeling is a specific kind of abstraction, usually to identify and represent objects, relationships, actions, operations, variables, and dependencies. Modeling is a way to organize and present information for deeper understanding. It's a way to draw generalizations and relationships from raw data.

As an example of modeling as a life skill, consider the personal research you do before buying a car. You will encounter all kinds of information about cars. In order to compare one car against another, you need to organize this information with some kind of model. Your model might include dimensions for consideration, which might be style, aesthetics, and technical specifications. Under styles, you might consider convertibles, SUVs, and sedans. Under aesthetics, you might consider body shapes, aerodynamic appearance, and colors available. Under technical specifications you might consider cost, MPG, horsepower, torque, and all-wheel drive versus front-wheel drive.

2.4.6 Storytelling

Storytelling *is the practice of using narrative to explain aspects of a phenomenon or design with the objective of immersing the audience in the phenomenon.*

Storytelling is a technique often used in the field of advertising. It can be more compelling to tell stories of people who use a product and who get pleasure and/or utility from it in their lives, than just to list advantages of the product.

Storytelling is also a good life skill that can be used in a wide variety of situations. A real estate agent telling the story of a house, when it was constructed, who lived there before, and who lives there now is more compelling than just talking about square footage and other features. The stories enable us to envision how we might live in this house and make our own memories there.

A more practical and useful example for the readers of this book comes from what we've noticed in our years of experience doing job interviews with aspiring UX designers and researchers. We noticed that job candidates who resorted to storytelling when they introduced themselves or when they did design portfolio walkthroughs were always more interesting and relatable. Storytelling helped these candidates communicate the context of a project, including the design brief, the challenge, the politics and the culture in a rich and engaging way. In contrast, candidates who did a "presentation" of their portfolio one design slide at a time were not as effective because there was no glue to put together the disconnected design snapshots in each slide into a narrative we could relate to.

2.4.7 Immersion

***Immersion** is a form of deep thought and analysis of the problem at hand—to "live" within the context of a problem and to make connections among the different aspects of it.*

Immersion is about surrounding yourself in your UX work area (see UX design studio in Section 5.3) with the artifacts of creative design (posters, notes, sketches, photos, diagrams, quotations, goal statements) as in a war room. You close yourself off from outside distractions and everything you see acts as a kind of cognitive scaffolding and a catalyst that helps spawn design ideas.

The artifacts act as stimuli that trigger framings and bring to the surface connections and relationships. Everything you see acts as a kind of cognitive scaffolding and a catalyst that help spawn ideas.

Consider this example of immersion in a non-UX setting. When a police detective is trying to solve a difficult crime, she might isolate herself in a "war room" set apart from other distractions of the job or its surroundings. She will become steeped in the problem and surround herself with artifacts, such as photos and sketches of the crime scene, police and witness reports. She might also post a timeline on the wall (a kind of flow model of how things unfolded) along with sketches of the life stories of people involved. She studies the problem so intensely that she "becomes" the perpetrator.

Finally, although much of your immersion will necessarily occur in your UX studio, immersion on-site (where the system you are building will be used) can be an effective supplemental immersion space for analysis and design (Schleicher, Jones, & Kachur, 2010).

Framing

The practice of posing a problem within a particular perspective based on a pattern or theme that structures the problem and highlights the aspects you will explore (Section 2.4.10).

2.4.8 Brainstorming

***Brainstorming** is the practice of interactive group discussion for exploring different ideas, problems, and solutions*:

- Must be done as a group activity. Each person's inputs and discussion stimulates, triggers thoughts, and inspires the others.
- Is a major skill in the Design Solutions lifecycle activity to highlight different perspectives and generate different framings of a phenomenon or a problem.
- Can be used in the Evaluate UX lifecycle activity to create solutions to identified UX problems.
- Can be used in any situation where the problem is open ended. For example, who are potential users of this system? Where can we find participants for evaluation?

More about brainstorming as part of ideation, sketching, and critiquing in UX design can be found in Chapter 14 on generative design.

2.4.9 Sketching and Drawing

***Sketching** in UX is the practice of drawing simple pictures and diagrams depicting the essence of problems and solutions.*

It is a way to externalize analysis and exploration of objects, their relationships, and an emerging understanding of the problems and solutions. The most important point about sketching is that it is not about art or aesthetics. It's about communication of ideas. So, don't worry if your sketches are not perfectly proportioned and artistic. See Section 14.3.1.1 for a description of how sketching is used as an integral part of generative design.

A sketch is a kind of prototype. It uses an abstract representation, highlighting the salient features to aid visualization. Sketching actually helps your thinking by the embodiment of engaging the hand-eye connection to cognition in the brain (Graves, 2012). This can boost cognition in the creative act (Section 14.3.1.3). Sketching must always occur with ideation. As Buxton (2007b) says, "If you are doing design, you are sketching."

Sketching is a life skill with wide range of applications in everyday life. Reorganizing furniture in your home? First make a sketch of the envisioned configuration or layout. Substantiate it with a model of your workflows for that room and immerse yourself in the usage context of that room.

2.4.10 Framing and Reframing

Framing and reframing *comprise the practice of posing a problem within a particular perspective.*

Framing builds a perspective that structures the problem and highlights the aspects you will explore. A framing is a pattern or a particular theme from which we view everything as we are in the process of finding solutions. In the specific context of a framing we can ask "what if?" and "why don't we do this?" Framings are reusable and gain strength with reuse.

To create a framing, you must go back to the basic elements of the problem, the underlying abstract phenomenon, identify what is really going on, the essence of the problem, and ignore all the rest as noise. Because framing is a specific kind of brainstorming, it is best done as a team activity.

Nigel Cross (2001) describes a book by Kees Dorst (2015) as providing "a practical new approach to design-led innovation. His frame creation approach enables the addressing of difficult and wicked problems through the use of design thinking."

Design Thinking

The mindset of getting organizations to think like designers, applying design principles and practice to businesses and business processes (Section 1.8.1).

Here is a non-UX example of how different framings lead to different solutions in any kind of problem solving in almost any setting. For example, suppose you have occasional flooding of fields along the flood plains of a river near a town. If you frame it as a problem of too much water coming down the river, it will lead you to focus on design solutions in the form of dams or other ways to control the flow into the valley. Alternatively, if you frame it as a problem of the river overflowing its banks, you might turn to a system of dikes to keep the water within the banks. Or, if you frame it as a natural occurrence to work with instead of fighting against, you will require landowners to build homes only above the flood plain.

Coming back to the problem at Buster's Auto Spa, we can see it as a practical example of framing. Clearly, designers of the car wash had framed the design mainly in terms of the simplicity of only the normal ("happy path") flow. However, careful observation of the workflow in other car washes might have led them to a problem framing that included the case of vacuuming the interior and a design that accommodated that operation in a different space that didn't block the normal flow.

2.4.11 Reasoning and Deduction

Reasoning and deduction *is a long-standing practice of applying logic to process observed facts, fit them together, and arrive at a logical conclusion.*

The observations are the predicates of the logic and the conclusions are deductions. Reasoning and deduction are a way of synthesizing new facts through the use of logic as applied to existing facts.

In UX, reasoning and deduction are often used to arrive at user needs based on usage research, design features based on needs, tradeoffs and constraints based on insights from the work domain. We will see examples of this in Chapter 10 where we deduce system requirements from usage data.

2.4.12 Prototyping and Envisioning

Prototyping *is the practice of producing or building a model or mockup of a design that can be manipulated and used at some level to manifest or simulate a user experience, which can be evaluated.*

Prototyping extends the idea of sketching. As the main output of UX design, a prototype is a platform for envisioning and evaluating efficacy of a design as a problem solution. See Chapter 20 for much more about prototyping.

2.4.13 Critical Thinking

Critical thinking *is the practice of "objective analysis of facts to form a judgment. The subject is complex, and there are several different definitions which generally include the rational, skeptical, unbiased analysis or evaluation of factual evidence."*[3]

Critical thinking is the essential core of UX evaluation for testing, reviewing, diagnosing, verifying, or validating a candidate design solution. This kind of evaluation requires skills for observation, abstraction, data collection, note taking, and reasoning and deduction, plus the ability to make judgements, rankings, and ratings.

See Part 5 for much more about UX evaluation.

2.4.14 Iteration

Iteration *is the practice of repeating a cycle of analysis, design, prototyping, and evaluation to refine an understanding of a concept or to improve a design as a problem solution.*

A simple non-UX example of iteration is seen in the rereading and reediting of a paper or report that an author might do before submitting it.

2.4.15 UX Techniques Are Used in Combination

When used in UX design or as life skills, these techniques are usually combined within methods. For example, a police detective must combine skills to solve crimes, including observation, note taking, storytelling, immersion, brainstorming, sketching, framing, and reasoning and deduction.

[3] https://en.wikipedia.org/wiki/Critical_thinking

We will have more to say about these techniques wherever we use them in the later process chapters, especially in the ideation, sketching, and critiquing of Chapter 14.

2.5 CHOOSING UX PROCESSES, METHODS, AND TECHNIQUES

Within any given project, you have to choose UX processes, methods, and techniques for the UX lifecycle activities and subactivities.

Agile Lifecycle Process

A small-scope lifecycle process (UX or SE) in which all lifecycle activities are performed for one feature of the product or system, and then the lifecycle is repeated for the next feature. An agile process is driven by needs formulated as user stories of capabilities instead of abstract system requirements and is characterized by small and fast deliveries of releases to get early usage-based feedback. Section 4.3.

2.5.1 The UX Lifecycle Process Choice

The UX lifecycle process choice is made at the highest level. The way things have worked out in the world has had a large influence on that choice. The software engineering (SE) world has adopted an agile lifecycle process almost universally and we, in UX, are making the same choice of an agile lifecycle process, to keep pace with our SE project partners. As in SE, an agile UX process is one in which you manage change during the process by delivering UX designs in small chunks.

While much of this book can apply to nonagile UX lifecycle processes, we intend this book to be about agile UX. But, as we will soon see, that can mean many things depending on multiple factors and where you are in the overall process. In Chapters 4 and 29, we will tell you the full story of what this means in practice.

2.5.2 The Idea of Appropriating Methods and Techniques

There is an abundance of UX design methods. “HCI is awash in methods and the theories that underlie them” (Harrison, Back, & Tatar, 2006). So, how can you make sense of what you need for your project?

The usual practice in a book like this, or in a course on UX design, is for the author or instructor to take a method-oriented approach to lay out his or her favorite UX methods and proceed to show why that is *the* way to do UX design. In this edition, we think it makes more sense to start with the idea of a design situation, especially the product or system being designed, and talk about how to match it by appropriating an approach that can achieve it, based on goals and expected outcomes.

This idea of appropriating design methods comes from Harrison and Tatar (2011) and it simply means that you take “standard” methods you have learned about and modify and adapt them to your specific design situations, making them your own methods (appropriating them).

The Harrison and Tatar "method of methods" idea is so simple on one level that it can easily get lost in the voluminous literature on UX design methods. But their method of methods idea is also powerful and important, especially for teaching design, both to new students and to experienced practitioners.

2.5.2.1 Design situations: Dependencies that govern lifecycle activity, method, and technique choices

Harrison and Tatar (2011) describe a ***design situation*** *as the circumstances under which a design method will be applied and appropriated.* "Design situation" is a good umbrella term because it includes the target product or system and the project and all of its context, including the type of product or system, the client, the users, the market, the subject-matter domain and its complexity plus the designer's familiarity with it, and the project team and their capabilities, skills, and experience.

2.5.2.2 Choosing methods and techniques

Whenever you need to design for a lifecycle activity, you will have a set of methods and techniques to choose from. For example, suppose you need to choose a method to carry out the Understand Needs lifecycle activity. As an example, one of the most popular of such methods is called usage research, a method for interviewing and observing real users to understand their work activity (Part 2 of this book).

Early method and technique choices constrain later ones. Earlier choices of methods and techniques can constrain later choices by suggesting, eliminating, or dictating appropriate methods and techniques for subsequent choices. For example, methods and techniques used for data analysis in a given situation will depend on what kind of data you have, and how the data were collected.

2.5.2.3 Mapping project parameters to lifecycle activity, method, and technique choices

To summarize, in Fig. 2-6 we show the mapping from project parameters to possible choices of UX methods. While there are some general guidelines for making these mapping choices, fine-tuning is the job of project teams, especially the project manager or product owner (the person responsible for success of the product and for pursuing business goals, Section 5.4.1.5). Much of it is intuitive and straightforward.

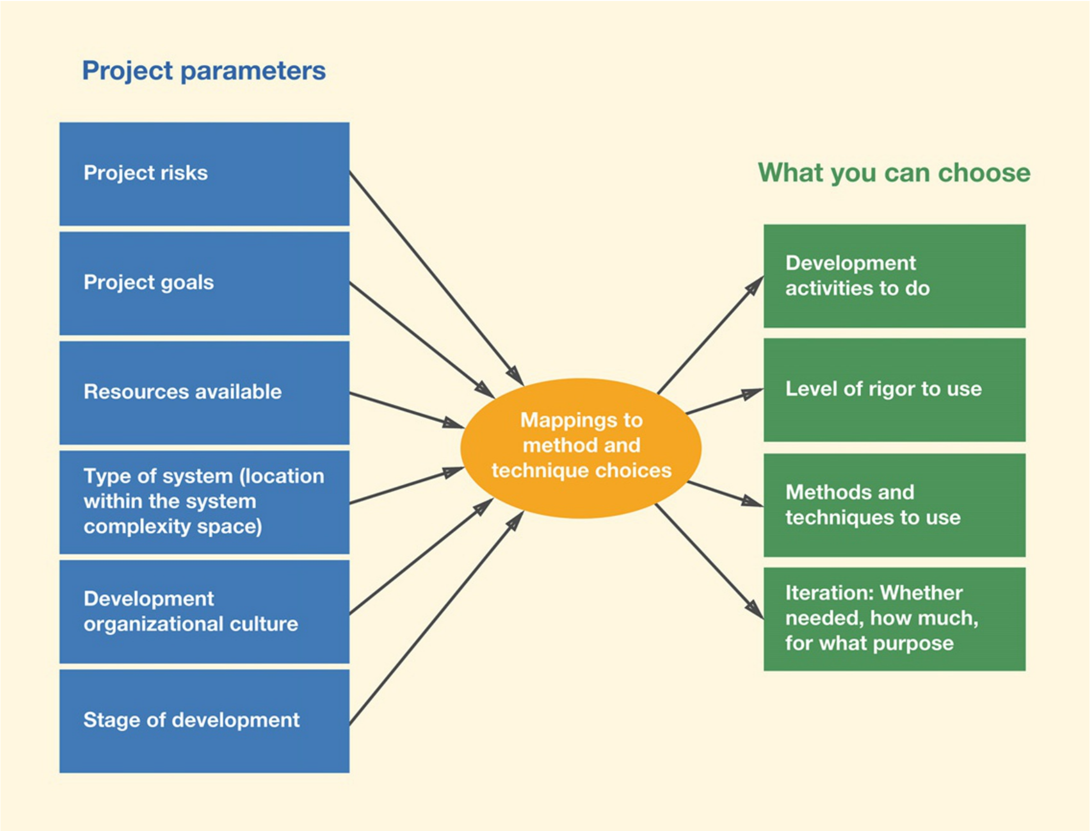
Project parameters
Project risks
Project goals
Resources available
Type of system (location within the system complexity space)
Development organizational culture
Stage of development
Mappings to method and technique choices
What you can choose
Development activities to do
Level of rigor to use
Methods and techniques to use
Iteration: Whether needed, how much, for what purpose

Fig. 2-6

Mapping project parameters to agile UX method choices.