

Chapter 2

THE PROCESS OF INTERACTION DESIGN

2.1 Introduction

2.2 What Is Involved in Interaction Design?

2.3 Some Practical Issues

Objectives

The main goals of this chapter are to accomplish the following:

- Reflect on what interaction design involves.
- Explain some of the advantages of involving users in development.
- Explain the main principles of a user-centered approach.
- Introduce the four basic activities of interaction design and how they are related in a simple lifecycle model.
- Ask some important questions about the interaction design process and provide the answers.
- Consider how interaction design activities can be integrated into other development lifecycles.

2.1 Introduction

Imagine that you have been asked to design a cloud-based service to enable people to share and curate their photos, movies, music, chats, documents, and so on, in an efficient, safe, and enjoyable way. What would you do? How would you start? Would you begin by sketching how the interface might look, work out how the system architecture should be structured, or just start coding? Or, would you start by asking users about their current experiences with sharing files and examine the existing tools, for example, Dropbox and Google Drive, and based on this begin thinking about how you were going to design the new service? What would you do next? This chapter discusses the process of interaction design, that is, how to design an interactive product.

There are many fields of design, such as graphic design, architectural design, industrial design, and software design. Although each discipline has its own approach to design, there

are commonalities. The Design Council of the United Kingdom captures these in the double-diamond of design, as shown in Figure 2.1. This approach has four phases which are iterated:

- *Discover*: Designers try to gather insights about the problem.
- *Define*: Designers develop a clear brief that frames the design challenge.
- *Develop*: Solutions or concepts are created, prototyped, tested, and iterated.
- *Deliver*: The resulting project is finalized, produced, and launched.

Interaction design also follows these phases, and it is underpinned by the philosophy of user-centered design, that is, involving users throughout development. Traditionally, interaction designers begin by doing user research and then sketching their ideas. But who are the users to be researched, and how can they be involved in development? Will they know what they want or need if we just ask them? From where do interaction designers get their ideas, and how do they generate designs?

In this chapter, we raise and answer these kinds of questions, discuss user-centered design, and explore the four basic activities of the interaction design process. We also introduce a lifecycle model of interaction design that captures these activities and the relationships among them.

2.2 What Is Involved in Interaction Design?

Interaction design has specific activities focused on discovering requirements for the product, designing something to fulfill those requirements, and producing prototypes that are then evaluated. In addition, interaction design focuses attention on users and their goals.

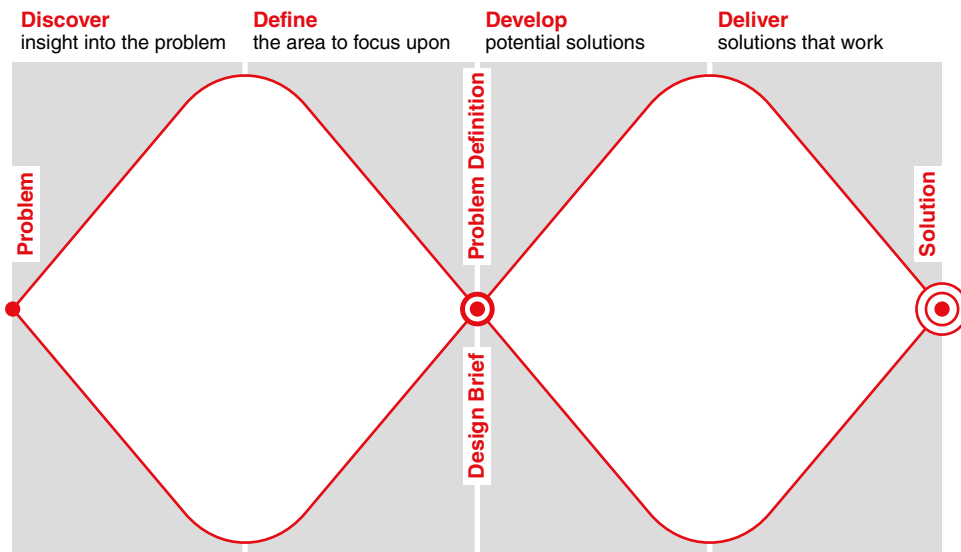


Figure 2.1 The double diamond of design

Source: Adapted from <https://www.designcouncil.org.uk/news-opinion/design-process-what-double-diamond>

For example, the artifact's use and target domain are investigated by taking a user-centered approach to development, users' opinions and reactions to early designs are sought, and users are involved appropriately in the development process itself. This means that users' concerns direct the development rather than just technical concerns.

Design is also about trade-offs—about balancing conflicting requirements. One common form of trade-off when developing a system to offer advice, for example, is deciding how much choice will be given to the user and how much direction the system should offer. Often, the division will depend on the purpose of the system, for example, whether it is for playing music tracks or for controlling traffic flow. Getting the balance right requires experience, but it also requires the development and evaluation of alternative solutions.

Generating alternatives is a key principle in most design disciplines and one that is also central to interaction design. Linus Pauling, twice a Nobel Prize winner, once said, “The best way to get a good idea is to get lots of ideas.” Generating lots of ideas is not necessarily hard, but choosing which of them to pursue is more difficult. For example, Tom Kelley (2016) describes seven secrets for successful brainstorming, including sharpening the focus (having a well-honed problem statement), having playful rules (to encourage ideas), and getting physical (using visual props).

Involving users and others in the design process means that the designs and potential solutions will need to be communicated to people other than the original designer. This requires the design to be captured and expressed in a form that allows review, revision, and improvement. There are many ways of doing this, one of the simplest being to produce a series of sketches. Other common approaches are to write a description in natural language, to draw a series of diagrams, and to build a prototype, that is, a limited version of the final product. A combination of these techniques is likely to be the most effective. When users are involved, capturing and expressing a design in a suitable format is especially important since they are unlikely to understand jargon or specialist notations. In fact, a form with which users can interact is most effective, so building prototypes is an extremely powerful approach.

ACTIVITY 2.1

This activity asks you to apply the double diamond of design to produce an innovative interactive product for your own use. By focusing on a product for yourself, the activity deliberately de-emphasizes issues concerned with involving other users, and instead it emphasizes the overall process.

Imagine that you want to design a product that helps you organize a trip. This might be for a business or vacation trip, to visit relatives halfway around the world, or for a bike ride on the weekend—whatever kind of trip you like. In addition to planning the route or booking tickets, the product may help to check visa requirements, arrange guided tours, investigate the facilities at a location, and so on.

1. Using the first three phases of the double diamond of design, produce an initial design using a sketch or two, showing its main functionality and its general look and feel. This activity omits the fourth phase, as you are not expected to deliver a working solution.
2. Now reflect on how your activities fell into these phases. What did you do first? What was your instinct to do first? Did you have any particular artifacts or experiences upon which to base your design?

(Continued)

Comment

1. The first phase focuses on discovering insights about the problem, but is there a problem? If so, what is it? Although most of us manage to book trips and travel to destinations with the right visas and in comfort, upon reflection the process and the outcome can be improved. For example, dietary requirements are not always fulfilled, and the accommodation is not always in the best location. There is a lot of information available to support organizing travel, and there are many agents, websites, travel books, and tourist boards that can help. The problem is that it can be overwhelming.

The second phase is about defining the area on which to focus. There are many reasons for travelling—both individual and family—but in my experience organizing business trips to meetings worldwide is stressful, and minimizing the complexity involved in these would be worthwhile. The experience would be improved if the product offers advice from the many possible sources of information and tailors that advice to individual preferences.

The third phase focuses on developing solutions, which in this case is a sketch of the design itself. Figure 2.2 shows an initial design. This has two versions of the product—one as an app to run on a mobile device and one to run on a larger screen. The assumptions underlying the choice to build two versions are based on my experience; I would normally plan the details of the trip at my desk, while requiring updates and local information while traveling. The mobile app has a simple interaction style that is easy to use on the go, while the larger-screen version is more sophisticated and shows a lot of information and the various choices available.

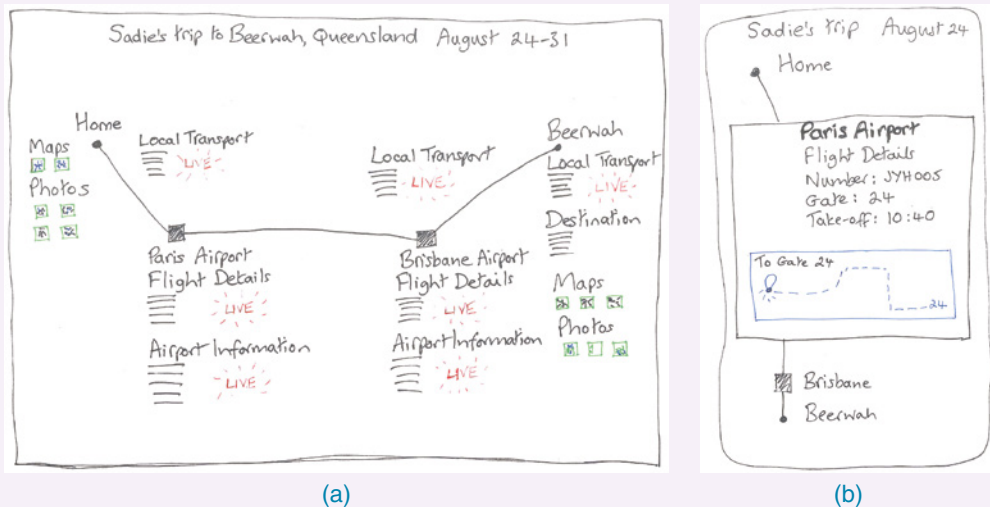


Figure 2.2 Initial sketches of the trip organizer showing (a) a large screen covering the entire journey from home to Beerwah in Australia and (b) the smartphone screen available for the leg of the journey at Paris (Charles de Gaulle) airport

2. Initially, it wasn't clear that there was a problem to address, but on reflection the complexity of the available information and the benefit of tailoring choices became clearer. The second phase guided me toward thinking about the area on which to focus. Worldwide business trips are the most difficult, and reducing the complexity of information sources through customization would definitely help. It would be good if the product learned about my preferences, for example, recommending flights from my favorite airline and finding places to have a vegan meal.

Developing solutions (the third phase) led me to consider how to interact with the product—seeing detail on a large screen would be useful, but a summary that can be shown on a mobile device is also needed. The type of support also depends on where the meeting is being held. Planning a trip abroad requires both a high-level view to check visas, vaccinations, and travel advice, as well as a detailed view about the proximity of accommodation to the meeting venue and specific flight times. Planning a local trip is much less complicated.

The exact steps taken to create a product will vary from designer to designer, from product to product, and from organization to organization (see Box 2.1). Capturing concrete ideas, through sketches or written descriptions, helps to focus the mind on what is being designed, the context of the design, and what user experience is to be expected. The sketches can capture only some elements of the design, however, and other formats are needed to capture everything intended. Throughout this activity, you have been making choices between alternatives, exploring requirements in detail, and refining your ideas about what the product will do. ■

2.2.1 Understanding the Problem Space

Deciding what to design is key, and exploring the problem space is one way in which to decide. This is the first phase in the double diamond, but it can be overlooked by those new to interaction design, as you may have discovered in Activity 2.1. In the process of creating an interactive product, it can be tempting to begin at the nuts and bolts level of design. By this we mean working out how to design the physical interface and what technologies and interaction styles to use, for example, whether to use multitouch, voice, graphical user interface, heads-up display, augmented reality, gesture-based, and so forth. The problem with starting here is that potential users and their context can be misunderstood, and usability and user experience goals can be overlooked, both of which were discussed in Chapter 1, “What Is Interaction Design?”

For example, consider the augmented reality displays and holographic navigation systems that are available in some cars nowadays (see Figure 2.3). They are the result of decades of research into human factors of information displays (for instance, Campbell et al., 2016), the driving experience itself (Perterer et al., 2013; Lee et al., 2005), and the suitability of different technologies (for example, Jose et al., 2016), as well as improvements in technology. Understanding the problem space has been critical in arriving at workable solutions that are safe and trusted. Having said that, some people may not be comfortable using a holographic navigation system and choose not to have one installed.



(a)



(b)

Figure 2.3 (a) Example of the holographic navigation display from WayRay which overlays GPS navigation instructions onto the road ahead and gathers and shares driver statistics (b) an augmented reality navigation system available in some cars today

Sources: (a) Used courtesy of WayRay, (b) Used courtesy of Muhammad Saad

While it is certainly necessary at some point to choose which technology to employ and decide how to design the physical aspects, it is better to make these decisions after articulating the nature of the problem space. By this we mean understanding what is currently the user experience or the product, why a change is needed, and how this change will improve the user experience. In the previous example, this involves finding out what is problematic with existing support for navigating while driving. An example is ensuring that drivers can continue to drive safely without being distracted when looking at a small GPS display mounted on the dashboard to figure out on which road it is asking them to “turn left.” Even when designing for a new user experience, it still requires understanding the context for which it will be used and the possible current user expectations.

The process of articulating the problem space is typically done as a team effort. Invariably, team members will have differing perspectives on it. For example, a project manager is likely to be concerned about a proposed solution in terms of budgets, timelines, and staffing costs, whereas a software engineer will be thinking about breaking it down into specific technical concepts. The implications of pursuing each perspective need to be considered in relation to one another. Although time-consuming and sometimes resulting in disagreements among the design team, the benefits of this process can far outweigh the associated costs: there will be much less chance of incorrect assumptions and unsupported

claims creeping into a design solution that later turn out to be unusable or unwanted. Spending time enumerating and reflecting upon ideas during the early stages of the design process enables more options and possibilities to be considered. Furthermore, designers are increasingly expected to justify their choice of problems and to be able to present clearly and convincingly their rationale in business as well as design language. Being able to think and analyze, present, and argue is valued as much as the ability to create a product (Kolko, 2011).

BOX 2.1

Four Approaches to Interaction Design

Dan Saffer (2010) suggests four main approaches to interaction design, each of which is based on a distinct underlying philosophy: User-centered design, Activity-centered design, Systems design, and Genius design.

Dan Saffer acknowledges that the purest form of any of these approaches is unlikely to be realized, and he takes an extreme view of each in order to distinguish among them. In user-centered design, the user knows best and is the guide to the designer; the designer's role is to translate the users' needs and goals into a design solution.

Activity-centered design focuses on the behavior surrounding particular tasks. Users still play a significant role, but it is their behavior rather than their goals and needs that is important. Systems design is a structured, rigorous, and holistic design approach that focuses on context and is particularly appropriate for complex problems. In systems design, it is the system (that is, the people, computers, objects, devices, and so on) that the center of attention, while the users' role is to set the goals of the system.

Finally, genius design is different from the other three approaches because it relies largely on the experience and creative flair of a designer. Jim Leftwich, an experienced interaction designer interviewed by Dan Saffer (2010, pp. 44–45), prefers the term *rapid expert design*. In this approach, the users' role is to validate ideas generated by the designer, and users are not involved during the design process itself. Dan Saffer points out that this is not necessarily by choice, but it may be because of limited or no resources for user involvement.

Different design problems lend themselves more easily to different approaches, and different designers will tend to gravitate toward using the approach that suits them best. Although an individual designer may prefer a particular approach, it is important that the approach for any one design problem is chosen with that design problem in mind. ■

2.2.2 The Importance of Involving Users

Chapter 1 stressed the importance of understanding users, and the previous description emphasizes the need to involve users in interaction design. Involving users in development is important because it's the best way to ensure that the end product is usable and that it indeed will be used. In the past, it was common for developers to talk only to managers, experts, or

proxy users, or even to use their own judgment without reference to anyone else. While others involved in designing the product can provide useful information, they will not have the same perspective as the target user who performs the activity every day or who will use the intended product on a regular basis.

In commercial projects, a role called the *product owner* is common. The product owner's job is to filter user and customer input to the development cycle and to prioritize requirements or features. This person is usually someone with business and technical knowledge, but not interaction design knowledge, and they are rarely (if ever) a direct user of the product. Although the product owner may be called upon to assess designs, they are a proxy user at best, and their involvement does not avoid the need for user involvement.

The best way to ensure that developers gain a good understanding of users' goals, leading to a more appropriate, more usable product, is to involve target users throughout development. However, two other aspects unrelated to functionality are equally as important if the product is to be usable and used: expectation management and ownership.

Expectation management is the process of making sure that the users' expectations of the new product are realistic. Its purpose is to ensure that there are no surprises for users when the product arrives. If users feel they have been cheated by promises that have not been fulfilled, then this will cause resistance and even rejection. Marketing of the new arrival must be careful not to misrepresent the product, although it may be particularly difficult to achieve with a large and complex system (Nevo and Wade, 2007). How many times have you seen an advertisement for something that you thought would be really good to have, but when you actually see one, you discover that the marketing hype was a little exaggerated? We expect that you felt quite disappointed and let down. This is the kind of feeling that expectation management tries to avoid.

Involving users throughout development helps with expectation management because they can see the product's capabilities from an early stage. They will also understand better how it will affect their jobs and lives and why the features are designed that way. Adequate and timely training is another technique for managing expectations. If users have the chance to work with the product before it is released through training or hands-on demonstrations of a prerelease version, then they will understand better what to expect when the final product is available.

A second reason for user involvement is ownership. Users who are involved and feel that they have contributed to a product's development are more likely to feel a sense of ownership toward it and support its use (Bano et al., 2017).

How to involve users, in what roles, and for how long, needs careful planning, as discussed in the next Dilemma box.

DILEMMA

Too Much of a Good Thing?

Involving users in development is a good thing, but what evidence is there that user involvement is productive? How much should users be involved and in what role(s)? Is it appropriate for users to lead a technical development project, or is it more beneficial for them to focus on evaluating prototypes?

Uli Abelein et al. (2013) performed a detailed review of the literature in this area and concluded that, overall, the evidence indicates that user involvement has a positive effect on user satisfaction and system use. However, they also found that even though the data clearly indicates this positive effect, some links have a large variation, suggesting that there is still no clear way to measure the effects consistently. In addition, they found that most studies with negative correlations involving users and system success were published more than 10 years previously.

Ramanath Subrayaman et al. (2010) investigated the impact of user participation on levels of satisfaction with the product by both developers and users. They found that for new products, developer satisfaction increased as user participation increased. On the other hand, user satisfaction was higher if their participation was low, and satisfaction dropped as their participation increased. They also identified that high levels of user involvement can generate conflicts and increased reworking. For maintenance projects, both developers and users were most satisfied with a moderate level of participation (approximately 20 percent of overall project development time). Based just on user satisfaction as an indicator of project success, then, it seems that low user participation is most beneficial.

The kind of product being developed, the kind of user involvement possible, the activities in which they are involved, and the application domain all have an impact on the effectiveness of user input (Bano and Zowghi, 2015). Peter Richard et al. (2014) investigated the effect of user involvement in transport design projects. They found that involving users at later stages of development mainly resulted in suggestions for service improvement, whereas users involved at earlier stages of innovation suggested more creative ideas.

Recent moves toward an agile way of working (see Chapter 13, “Interaction Design in Practice”) has emphasized the need for feedback from customers and users, but this also has its challenges. Kurt Schmitz et al. (2018) suggests that in tailoring their methods, teams consider the distinction between frequent participation in activities and effective engagement.

User involvement is undoubtedly beneficial, but the levels and types of involvement require careful consideration and balance. ■

2.2.3 Degrees of User Involvement

Different degrees of user involvement are possible, ranging from fully engaged throughout all iterations of the development process to targeted participation in specific activities and from small groups of individual users in face-to-face contexts to hundreds of thousands of potential users and stakeholders online. Where available, individual users may be co-opted onto the design team so that they are major contributors to the development. This has pros and cons. On the downside, full-time involvement may mean that they become out of touch with their user community, while part-time involvement might result in a high workload for them. On the positive side, having a user engaged full or part-time does mean that the input is available continually throughout development. On the other hand, users may take part in specific activities to inform the development or to evaluate designs once they are available. This is a valuable form of involvement, but the users’ input is limited to that particular activity. Where the circumstances around a project limit user involvement in this way, there are techniques to keep users’ concerns uppermost in developers’ minds, such as through personas (see Chapter 11, “Discovering Requirements”).

Initially, user involvement took the form of small groups or individuals taking part in face-to-face information-gathering design, or evaluation sessions, but increasing online connectivity has led to a situation in which many thousands of potential users can contribute to product development. There is still a place for face-to-face user involvement and *in situ* studies, but the range of possibilities for user involvement is now much wider. One example of this is online feedback exchange (OFE) systems, which are increasingly used to test design concepts with millions of target users before going to market (Foong et al., 2017).

In fact, design is becoming increasingly participative through crowdsourcing design ideas and examples, for instance (Yu et al., 2016). Where crowdsourcing is used, a range of different people are encouraged to contribute, and this can include any and all of the stakeholders. This wide participation helps to bring different perspectives to the process, which enhances the design itself, produces more user satisfaction with the final product, and engenders a sense of ownership. Another example of involving users at scale is citizen engagement, the goal of which is to engage a population—civic or otherwise—with the aim of promoting empowerment through technology. The underlying aim is to involve members of the public in helping them make a change in their lives where technology is often viewed as an integral part of the process.

Participatory design, also sometimes referred to as *cooperative design* or *co-design*, is an overarching design philosophy that places those for whom systems, technologies, and services are being designed, as central actors in creation activities. The idea is that instead of being passive receivers of new technological or industrial artifacts, end users and stakeholders are active participants in the design process. Chapter 12, “Design, Prototyping, and Construction,” provides more information on participatory design.

The individual circumstances of the project affect what is realistic and appropriate. If the end-user groups are identifiable, for example, the product is for a particular company, then it is easier to involve them. If, however, the product is intended for the open market, it is unlikely that users will be available to join the design team. In this case, targeted activities and online feedback systems may be employed. Box 2.2 outlines an alternative way to obtain user input from an existing product, and Box 2.5 discusses A/B testing, which draws on user feedback to choose between alternative designs.

BOX 2.2

User Involvement After Product Release

Once a product has been released, a different kind of user involvement is possible—one that captures data and user feedback based on day-to-day use of the product. The prevalence of customer reviews has grown considerably in recent years, and they significantly affect the popularity and success of a product (Harman et al., 2012). These reviews provide useful and far-ranging user feedback. For example, Hammad Khalid et al. (2015) studied reviews of mobile apps to see what reviewers complained about. They identified 12 complaint types, including privacy and ethics, interface, and feature removal. Customer reviews can provide useful insight to help improve products, but detailed analysis of feedback gathered this way is time-consuming.

Error reporting systems (ERSs, also called *online crashing analysis*) automatically collect information from users that is used to improve applications in the longer term. This is done with users' permission, but with a minimal reporting burden. Figure 2.4 shows two dialog boxes for the Windows error reporting system that is built into Microsoft operating systems. This kind of reporting can have a significant effect on the quality of applications. For example, 29 percent of the errors fixed by the Windows XP (Service Pack 1) team were based on information collected through their ERS (Kinshumann et al., 2011). While Windows XP is no longer being supported, this statistic illustrates the impact ERSs can have. The system uses a sophisticated approach to error reporting based on five strategies: automatic aggregation of error reports; progressive data collection so that the data collected (such as abbreviated or full stack and memory dumps) varies depending on the level of data needed to diagnose the error; minimal user interaction; preserving user privacy; and providing solutions directly to users where possible. By using these strategies, plus statistical analysis, effort can be focused on the bugs that have the highest impact on the most users. ■

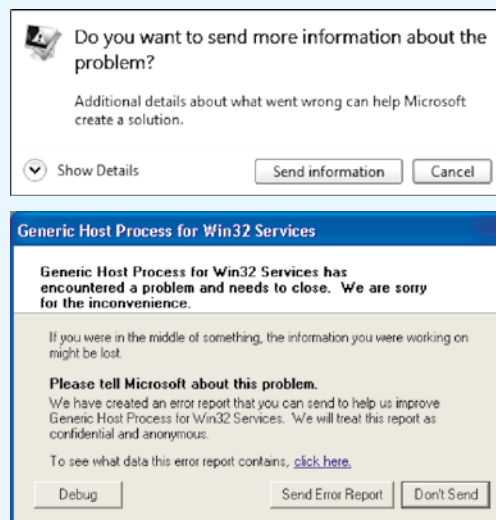


Figure 2.4 Two typical dialog boxes from the Windows error reporting system

2.2.4 What Is a User-Centered Approach?

Throughout this book, we emphasize the need for a user-centered approach to development. By this we mean that the real users and their goals, not just technology, are the driving force behind product development. As a consequence, a well-designed system will make the most of human skill and judgment, will be directly relevant to the activity in hand, and will support rather than constrain the user. This is less of a technique and more of a philosophy.

When the field of HCI was being established, John Gould and Clayton Lewis (1985) laid down three principles that they believed would lead to a “useful and easy to use computer system.” These principles are as follows:

1. *Early focus on users and tasks.* This means first understanding who the users will be by directly studying their cognitive, behavioral, anthropomorphic, and attitudinal characteristics. This requires observing users doing their normal tasks, studying the nature of those tasks, and then involving users in the design process.
2. *Empirical measurement.* Early in development, the reactions and performance of intended users to printed scenarios, manuals, and so forth, are observed and measured. Later, users interact with simulations and prototypes, and their performance and reactions are observed, recorded, and analyzed.
3. *Iterative design.* When problems are found in user testing, they are fixed, and then more tests and observations are carried out to see the effects of the fixes. This means that design and development are iterative, with cycles of design-test-measure-redesign being repeated as often as necessary.

These three principles are now generally accepted as the basis for a user-centered approach. When this paper was written, however, they were not accepted by most developers. We discuss these principles in more detail in the following sections.

Early Focus on Users and Tasks

This principle can be expanded and clarified through the following five further principles:

1. *Users’ tasks and goals are the driving force behind the development.*
While technology will inform design options and choices, it is not the driving force. Instead of saying “Where can we deploy this new technology?” say “What technologies are available to provide better support for users’ goals?”
2. *Users’ behavior and context of use are studied, and the system is designed to support them.*
This is not just about capturing users’ tasks and goals. How people perform their tasks is also significant. Understanding behavior highlights priorities, preferences, and implicit intentions.
3. *Users’ characteristics are captured and designed for.*
When things go wrong with technology, people often think it is their fault. People are prone to making errors and have certain limitations, both cognitive and physical. Products designed to support people should take these limitations into account and try to prevent mistakes from being made. Cognitive aspects, such as attention, memory, and perception issues are introduced in Chapter 4, “Cognitive Aspects.” Physical aspects include height, mobility, and strength. Some characteristics are general, such as color blindness, which affects about 4.5 percent of the population, but some characteristics are associated with a particular job or task. In addition to general characteristics, those traits specific to the intended user group also need to be captured.
4. *Users are consulted throughout development from earliest phases to the latest.*
As discussed earlier, there are different levels of user involvement, and there are different ways in which to consult users.
5. *All design decisions are taken within the context of the users, their activities, and their environment.*
This does not necessarily mean that users are actively involved in design decisions, but that is one option.

ACTIVITY 2.2

Assume you are involved in developing a novel online experience for buying garden plants. Although many websites exist for buying plants online, you want to produce a distinct experience to increase the organization's market share. Suggest ways of applying the previous principles in this task.

Comment

To address the first three principles, you would need to find out about the tasks and goals, behavior, and characteristics of potential customers of the new experience, together with any different contexts of use. Studying current users of existing online plant shops will provide some information, and it will also identify some challenges to be addressed in the new experience. However, as you want to increase the organization's market share, consulting existing users alone would not be enough. Alternative avenues of investigation include physical shopping situations—for example, shopping at the market, in the local corner shop, and so on, and local gardening clubs, radio programs, or podcasts. These alternatives will help you find the advantages and disadvantages of buying plants in different settings, and you will observe different behaviors. By looking at these options, a new set of potential users and contexts can be identified.

For the fourth principle, the set of new users will emerge as investigations progress, but people who are representative of the user group may be accessible from the beginning. Workshops or evaluation sessions could be run with them, possibly in one of the alternative shopping environments such as the market. The last principle could be supported through the creation of a design room that houses all of the data collected, and it is a place where the development team can go to find out more about the users and the product goals. ■

Empirical Measurement

Where possible, specific usability and user experience goals should be identified, clearly documented, and agreed upon at the beginning of the project. They can help designers choose between alternative designs and check on progress as the product is developed. Identifying specific goals up front means that the product can be empirically evaluated at regular stages throughout development.

Iterative Design

Iteration allows designs to be refined based on feedback. As users and designers engage with the domain and start to discuss requirements, needs, hopes, and aspirations, then different insights into what is needed, what will help, and what is feasible will emerge. This leads to a need for iteration—for the activities to inform each other and to be repeated. No matter how good the designers are and however clear the users may think their vision is of the required artifact, ideas will need to be revised in light of feedback, likely several times. This is particularly true when trying to innovate. Innovation rarely emerges whole and ready to go. It takes time, evolution, trial and error, and a great deal of patience. Iteration is inevitable because designers never get the solution right the first time (Gould and Lewis, 1985).

2.2.5 Four Basic Activities of Interaction Design

The four basic activities for interaction design are as follows:

1. Discovering requirements for the interactive product.
2. Designing alternatives that meet those requirements.
3. Prototyping the alternative designs so that they can be communicated and assessed.
4. Evaluating the product and the user experience it offers throughout the process.

Discovering Requirements

This activity covers the left side of the double diamond of design, and it is focused on discovering something new about the world and defining what will be developed. In the case of interaction design, this includes understanding the target users and the support an interactive product could usefully provide. This understanding is gleaned through data gathering and analysis, which are discussed in Chapters 8–10. It forms the basis of the product's requirements and underpins subsequent design and development. The requirements activity is discussed further in Chapter 11.

Designing Alternatives

This is the core activity of designing and is part of the Develop phase of the double diamond: proposing ideas for meeting the requirements. For interaction design, this activity can be viewed as two subactivities: conceptual design and concrete design. Conceptual design involves producing the conceptual model for the product, and a conceptual model describes an abstraction outlining what people can do with a product and what concepts are needed to understand how to interact with it. Concrete design considers the detail of the product including the colors, sounds, and images to use, menu design, and icon design. Alternatives are considered at every point. Conceptual design is discussed in Chapter 3, and more design issues for specific interface types are in Chapter 7; more detail about how to design an interactive product is in Chapter 12.

Prototyping

Prototyping is also part of the Develop phase of the double diamond. Interaction design involves designing the behavior of interactive products as well as their look and feel. The most effective way for users to evaluate such designs is to interact with them, and this can be achieved through prototyping. This does not necessarily mean that a piece of software is required. There are different prototyping techniques, not all of which require a working piece of software. For example, paper-based prototypes are quick and cheap to build and are effective for identifying problems in the early stages of design, and through role-playing users can get a real sense of what it will be like to interact with the product. Prototyping is covered in Chapter 12.

Evaluating

Evaluating is also part of the Develop phase of the double diamond. It is the process of determining the usability and acceptability of the product or design measured in terms of a variety of usability and user-experience criteria. Evaluation does not replace activities concerned with quality assurance and testing to make sure that the final product is fit for its intended purpose, but it complements and enhances them. Chapters 14–16 cover evaluation.

The activities to discover requirements, design alternatives, build prototypes, and evaluate them are intertwined: alternatives are evaluated through the prototypes, and the results are fed back into further design or to identify alternative requirements.

2.2.6 A Simple Lifecycle Model for Interaction Design

Understanding what activities are involved in interaction design is the first step to being able to do it, but it is also important to consider how the activities are related to one another. The term *lifecycle model* (or *process model*) is used to represent a model that captures a set of activities and how they are related. Existing models have varying levels of sophistication and complexity and are often not prescriptive. For projects involving only a few experienced developers, a simple process is adequate. However, for larger systems involving tens or hundreds of developers with hundreds or thousands of users, a simple process just isn't enough to provide the management structure and discipline necessary to engineer a usable product.



Source: Fran / Cartoon Stock

Many lifecycle models have been proposed in fields related to interaction design. For example, software engineering lifecycle models include the waterfall, spiral, and V models (for more information about these models, see Pressman and Maxim [2014]). HCI has been less associated with lifecycle models, but two well-known ones are the Star (Hartson and Hix, 1989) and an international standard model ISO 9241-210. Rather than explaining the details of these models, we focus on the classic lifecycle model shown in Figure 2.5. This model shows how the four activities of interaction design are related, and it incorporates the three principles of user-centered design discussed earlier.

Many projects start by discovering requirements from which alternative designs are generated. Prototype versions of the designs are developed and then evaluated. During prototyping or based on feedback from evaluations, the team may need to refine the requirements or to redesign. One or more alternative designs may follow this iterative cycle in parallel. Implicit in this cycle is that the final product will emerge in an evolutionary fashion from an initial idea through to the finished product or from limited functionality to sophisticated functionality. Exactly how this evolution happens varies from project to project. However many times through the cycle the product goes, development ends with an evaluation activity that ensures that the final product meets the prescribed user experience and usability criteria. This evolutionary production is part of the Delivery phase of the double diamond.

In recent years, a wide range of lifecycle models has emerged, all of which encompass these activities but with different emphases on activities, relationships, and outputs. For example, Google Design Sprints (Box 2.3) emphasize problem investigation, solution development, and testing with customers all in one week. This does not result in a robust final product, but it does make sure that the solution idea is acceptable to customers. The in-the-wild approach (Box 2.4) emphasizes the development of novel technologies that are not necessarily designed for specific user needs but to augment people, places, and settings. Further models are discussed in Chapter 13.

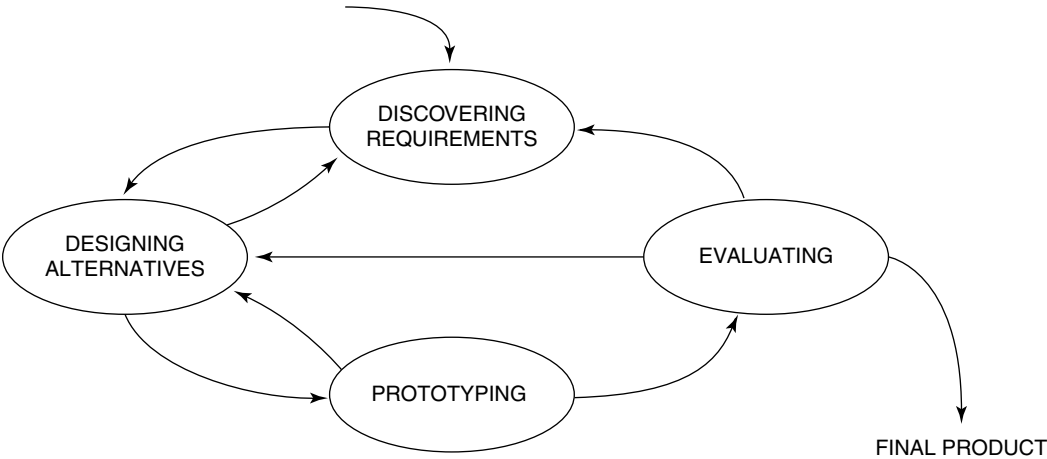


Figure 2.5 A simple interaction design lifecycle model

BOX 2.3

Google Design Sprints (Adapted from Knapp et al. (2016))

Google Ventures has developed a structured approach to design that supports rapid idea- tion and testing of potential solutions to a design challenge. This is called the *Google Design Sprint*. A sprint is divided into five phases, and each phase is completed in a day. This means that in five days, you can go from a design challenge to a solution that has been tested with customers. As the authors say, “You won’t finish with a complete, detailed, ready-to-ship product. But you will make rapid progress, and know for sure if you’re headed in the right direction” (Knapp et al., 2016, p16–17). Teams are encouraged to iterate on the last two phases and to develop and re-test prototypes. If necessary, the first idea can be thrown away and the process started again at Phase 1. There is preparation to be done before the sprint begins. This preparation and the five phases are described next (see Figure 2.6).

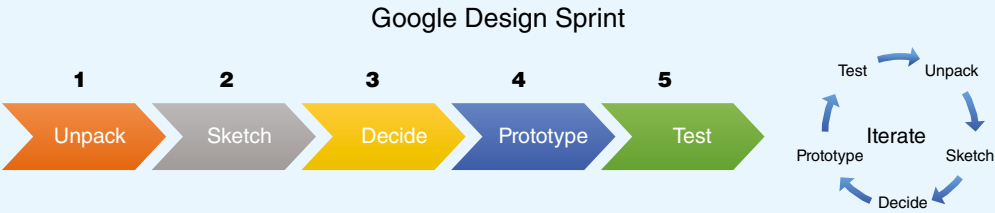


Figure 2.6 The five phases of the Google Design Sprint

Source: www.agilemarketing.net/google-design-sprints. Used courtesy of Agile Marketing

Setting the Stage

This time is used to choose the right design challenge, gather the right team, and organize the time and space to run the sprint (that is, full-time for everyone for five days). The sprint can help in high-stake challenges, when you're running out of time, or if you're just stuck. The team composition depends on the product, but it has about seven people including a decider (who chooses the design to show to the customer), customer expert, technical expert, and anyone who will bring a disruptive perspective.

Unpack

Day 1 focuses on making a map of the challenge and choosing a target, that is, a part of the challenge that can be achieved in a week.

Sketch Competing Solutions

Day 2 focuses on generating solutions, with an emphasis on sketching and individual creativity rather than group brainstorming.

Decide on the Best

Day 3 focuses on critiquing the solutions generated on Day 1, choosing the one most likely to meet the sprint's challenge, and producing a storyboard. Whichever solution is chosen, the decider needs to support the design.

Build a Realistic Prototype

Day 4 focuses on turning the storyboard into a realistic prototype, that is, something on which customers can provide feedback. We discuss prototyping further in Chapter 12.

Test with Target Customers

Day 5 focuses on getting feedback from five customers and learning from their reactions.

The Google Design Sprint is a process for answering critical business questions through design, prototyping, and testing ideas with customers. Marta Rey-Babarro, who works at Google as a staff UX researcher and was the cofounder of Google's internal Sprint Academy, describes how they used a sprint to improve the experience of traveling for business.

We wanted to see if we could improve the business travel experience. We started by doing research with Googlers to find out what experiences and what needs they had when they traveled. We discovered that there were some Googlers who traveled over 300 days a year and others who traveled only once or twice a year. Their travel experiences and needs were very different. After this research, some of us did a sprint in which we explored the whole travel experience, from the planning phase to coming back home and submitting receipts. Within five days we came up with a vision of what that experience could be. On the fifth day of the sprint, we presented that vision to higher-level execs. They loved it and sponsored the creation of a new team at Google that has developed new

tools and experiences for the traveling Googler. Some of those internal online experiences made it also to our external products and services outside of Google.

Marta Rey-Babarro ■

To see a more detailed description of the Google Design Sprint and to access a set of five videos that describe what happens on each day of the sprint, go to **www.gv.com/sprint/#book**.

BOX 2.4

Research in the Wild (Adapted from Rogers and Marshall (2017))

Research in the wild (RITW) develops technology solutions in everyday living by creating and evaluating new technologies and experiences *in situ*. The approach supports designing prototypes in which researchers often experiment with new technological possibilities that can change and even disrupt behavior, rather than ones that fit in with existing practices. The results of RITW studies can be used to challenge assumptions about technology and human behavior in the real world and to inform the re-thinking of HCI theories. The perspective taken by RITW studies is to observe how people react to technology and how they change and integrate it into their everyday lives.

Figure 2.7 shows the framework for RITW studies. In terms of the four activities introduced earlier, this framework focuses on designing, prototyping, and evaluating technology and ideas and is one way in which requirements may be discovered. It also considers relevant theory since often the purpose of an RITW study is to investigate a theory, idea, concept, or observation. Any one RITW study may emphasize the elements of the framework to a different degree.

Technology: Concerned with appropriating existing infrastructures/devices (e.g., Internet of Things toolkit, mobile app) *in situ* or developing new ones for a given setting (e.g., a novel public display).

Design: Covers the design space of an experience (e.g., iteratively creating a collaborative travel planning tool for families to use or an augmented reality game for playing outdoors).

In situ study: Concerned with evaluating *in situ* an existing device/tool/service or novel research-based prototype when placed in various settings or given to someone to use over a period of time.

Theory: Investigating a theory, idea, concept or observation about a behavior, setting or other phenomenon using existing ones or developing a new one or extending an existing one. ■

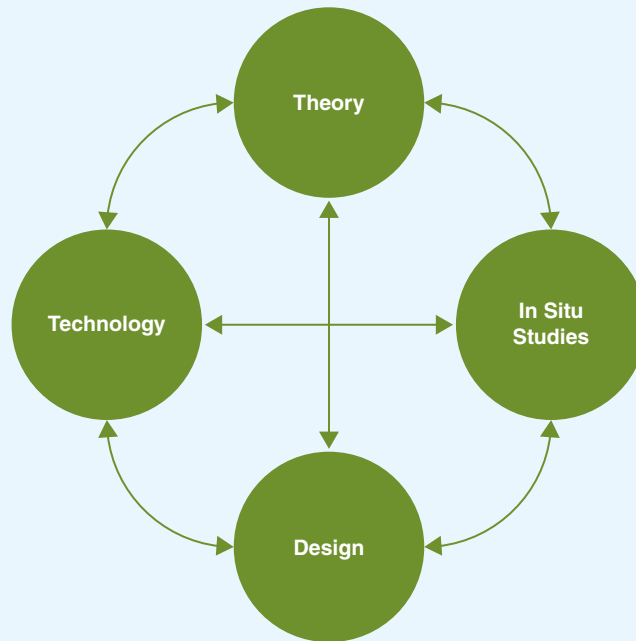


Figure 2.7 A framework for research in the wild studies

Source: Rogers and Marshall (2017), p. 6. Used courtesy of Morgan & Claypool

2.3 Some Practical Issues

The discussion so far has highlighted some issues about the practical application of user-centered design and the simple lifecycle of interaction design introduced earlier. These issues are listed here:

- Who are the users?
- What are the users' needs?
- How to generate alternative designs
- How to choose among alternatives
- How to integrate interaction design activities with other lifecycle models

2.3.1 Who Are the Users?

Identifying users may seem like a straightforward activity, but it can be harder than you think. For example, Sha Zhao et al. (2016) found a more diverse set of users for smartphones than most manufacturers recognize. Based on an analysis of one month's smartphone app

usage, they discovered 382 distinct types of users, including Screen Checkers and Young Parents. Charlie Wilson et al. (2015) found that little is understood about who the users of smart homes in general are expected to be, beyond those focused on health-related conditions. In part, this is because many products nowadays are being developed for use by large sections of the population, and so it can be difficult to determine a clear description. Some products (such as a system to schedule work shifts) have more constrained user communities, for example a specific role (shop assistant) within a particular industrial sector (retail). In this case, there may be a range of users with different roles who relate to the product in different ways. Examples are those who manage direct users, those who receive outputs from the system, those who test the system, those who make the purchasing decision, and those who use competitive products (Holzblatt and Jones, 1993).

There is a surprisingly wide collection of people who all have a stake in the development of a successful product. These people are called *stakeholders*. Stakeholders are the individuals or groups that can influence or be influenced by the success or failure of a project. Alan Dix et al. (2004) observed that is pertinent to a user-centered view of development: “It will frequently be the case that the formal ‘client’ who orders the system falls very low on the list of those affected. Be very wary of changes which take power, influence or control from some stakeholders without returning something tangible in its place.”

The group of stakeholders for a particular product will be larger than the group of users. It will include customers who pay for it; users who interact with it; developers who design, build, and maintain it; legislators who impose rules on the development and operation of it; people who may lose their jobs because of its introduction; and so on (Sharp et al., 1999).

Identifying the stakeholders for a project helps to decide who to involve as users and to what degree, but identifying relevant stakeholders can be tricky. Ian Alexander and Suzanne Robertson (2004) suggest using an onion diagram to model stakeholders and their involvement. This diagram shows concentric circles of stakeholder zones with the product being developed sitting in the middle. Soo Ling Lim and Anthony Finkelstein (2012) developed a method called StakeRare and supporting tool called StakeNet that relies on social networks and collaborative filtering to identify and prioritize relevant stakeholders.

ACTIVITY 2.3

Who are the stakeholders for an electricity smart meter for use in the home to help households control their energy consumption?

Comment

First, there are the people who live in the house, such as older adults and young children, with a range of abilities and backgrounds. To varying degrees, they will be users of the meter, and their stake in its success and usability is fairly clear and direct. Householders want to make sure that their bills are controlled, that they can easily access suppliers if they want to, and that their electricity supply is not interrupted. On the other hand, the entire family will

want to continue to live in the house in comfort, for example, with enough heat and light. Then there are the people who install and maintain the meter. They make sure that the meter is installed correctly and that it continues to work effectively. Installers and maintainers want the meter to be straightforward to install and to be robust and reliable to reduce the need for return visits or maintenance calls. Outside of these groups are electricity suppliers and distributors who also want to provide a competitive service so that the householders are satisfied and to minimize maintenance costs. They also don't want to lose customers and money because the meters are faulty or are providing inaccurate information. Other people who will be affected by the success of the meter include those who work on the powerlines and at electricity generation plants, those who work in other energy industries, and ultimately the government of the country that will want to maintain steady supply for its industry and population. ■

2.3.2 What Are the Users' Needs?

If you had asked someone in the street in the late 1990s what they needed, their answer probably wouldn't have included a smart TV, a ski jacket with an integrated smartphone, or a robot pet. If you presented the same person with these possibilities and asked whether they would buy them if they were available, then the answer may have been more positive. Determining what product to build is not simply a question of asking people "What do you need?" and then supplying it, because people don't necessarily know what is possible. Suzanne and James Robertson (2013) refer to "un-dreamed-of" needs, which are those that users are unaware they might have. Instead of asking users, this is approached by exploring the problem space, investigating the users and their activities to see what can be improved, or trying out ideas with potential users to see whether the ideas are successful. In practice, a mixture of these approaches is often taken—trying ideas in order to discover requirements and decide what to build, but with knowledge of the problem space, potential users, and their activities.

If a product is a new invention, then identifying the users and representative tasks for them may be harder. This is where in-the-wild studies or rapid design sprints that provide authentic user feedback on early ideas are valuable. Rather than imagining who might want to use a product and what they might want to do with it, it's more effective to put it out there and find out—the results might be surprising!

It may be tempting for designers simply to design what they would like to use themselves, but their ideas would not necessarily coincide with those of the target user group, because they have different experiences and expectations. Several practitioners and commentators have observed that it's an "eye-opening experience" when developers or designers see a user struggling to complete a task that seemed so clear to them (Ratcliffe and McNeill, 2012, p. 125).

Focusing on people's goals, usability goals and user experience goals is a more promising approach to interaction design than simply expecting stakeholders to be able to articulate the requirements for a product.

2.3.3 How to Generate Alternative Designs

A common human tendency is to stick with something that works. While recognizing that a better solution may exist, it is easy to accept the one that works as being “good enough.” Settling for a solution that is good enough may be undesirable because better alternatives may never be considered, and considering alternative solutions is a crucial step in the process of design. But where do these alternative ideas come from?

One answer to this question is that they come from the individual designer’s flair and creativity (the genius design described in Box 2.1). Although it is certainly true that some people are able to produce wonderfully inspired designs while others struggle to come up with any ideas at all, very little in this world is completely new. For example, the steam engine, commonly regarded as an invention, was inspired by the observation that steam from a kettle boiling on the stove lifted the lid. An amount of creativity and engineering was needed to make the jump from a boiling kettle to a steam engine, but the kettle provided inspiration to translate this experience into a set of principles that could be applied in a different context. Innovations often arise through cross-fertilization of ideas from different perspectives, individuals, and contexts; the evolution of an existing product through use and observation; or straightforward copying of other, similar products.

Cross-fertilization may result from discussing ideas with other designers, while Bill Buxton (2007) reports that different perspectives from users generated original ideas about alternative designs. As an example of evolution, consider the cell phone and its descendant, the smartphone. The capabilities of the phone in your pocket have increased from the time they first appeared. Initially, the cell phone simply made and received phone calls and texts, but now the smartphone supports a myriad of interactions, can take photos and record audio, play movies and games, and record your exercise routine.

Creativity and invention are often wrapped in mystique, but a lot has been uncovered about the process and of how creativity can be enhanced or inspired (for example, see Rogers, 2014). For instance, browsing a collection of designs will inspire designers to consider alternative perspectives and hence alternative solutions. As Roger Schank (1982, p. 22) puts it, “An expert is someone who gets reminded of just the right prior experience to help him in processing his current experiences.” And while those experiences may be the designer’s own, they can equally well be others’.

Another approach to creativity has been adopted by Neil Maiden et al. (2007). They ran creativity workshops to generate innovative requirements in an air traffic management (ATM) application domain. Their idea was to introduce experts in different fields into the workshop and then invite stakeholders to identify analogies between their own field and this new one. For example, they have invited an Indian textile expert, a musician, a TV program scheduler, and a museum exhibit designer. Although not all obviously analogical domains, they sparked creative ideas for the air traffic management application. For example, participants reported that one textile design was elegant, that is, simple, beautiful, and symmetrical. They then transferred these properties to a key area of the ATM domain—that of aircraft conflict resolution. They explored the meaning of elegance within this context and realized that elegance is perceived differently by different controllers. From this they generated the requirement that the system should be able to accommodate different air traffic controller styles during conflict resolution.

A more pragmatic answer to this question, then, is that alternatives come from seeking different perspectives and looking at other designs. The process of inspiration and creativity can be enhanced by prompting a designer's own experience and studying others' ideas and suggestions. Deliberately seeking out suitable sources of inspiration is a valuable step in any design process. These sources may be very close to the intended new product, such as competitors' products; they may be earlier versions of similar systems; or they may be from a completely different domain.

Under some circumstances, the scope to consider alternative designs is limited. Design is a process of balancing constraints and trading off one set of requirements with another, and the constraints may mean that there are few viable alternatives available. For example, when designing software to run under the Windows operating system, the design must conform to the Windows look and feel and to other constraints intended to make Windows programs consistent for the user. When producing an upgrade to an existing system, keeping familiar elements of it to retain the same user experience may be prioritized.

ACTIVITY 2.4

Consider the product introduced in Activity 2.1. Reflecting on the process again, what inspired your initial design? Are there any innovative aspects to it?

Comment

For our design, existing sources of information and their flaws were influential. For example, there is so much information available about travel, destinations, hotel comparisons, and so forth, that it can be overwhelming. However, travel blogs contain useful and practical insights, and websites that compare alternative options are informative. We were also influenced by some favorite mobile and desktop applications such as the United Kingdom's National Rail smartphone app for its real-time updating and by the Airbnb website for its mixture of simplicity and detail.

Perhaps you were inspired by something that you use regularly, like a particularly enjoyable game or a device that you like to use? I'm not sure how innovative our ideas were, but the main goal was for the application to tailor its advice for the user's preferences. There are probably other aspects that make your design unique and that may be innovative to a greater or lesser degree. ■

2.3.4 How to Choose Among Alternative Designs

Choosing among alternatives is mostly about making design decisions: Will the device use keyboard entry or a touch screen? Will the product provide an automatic memory function or not? These decisions will be informed by the information gathered about users and their tasks and by the technical feasibility of an idea. Broadly speaking, though, the decisions fall into two categories: those that are about externally visible and measurable features

BOX 2.5

A Box Full of Ideas

The innovative product design company IDEO was mentioned in Chapter 1. Underlying some of its creative flair is a collection of weird and wonderful engineering housed in a large flatbed filing cabinet called the TechBox. The TechBox holds hundreds of gizmos and interesting materials, divided into categories such as Amazing Materials, Cool Mechanisms, Interesting Manufacturing Processes, Electronic Technologies, and Thermal and Optical. Each item has been placed in the box because it represents a neat idea or a new process. The staff at IDEO take along a selection of items from the TechBox to brainstorming meetings. The items may be chosen because they provide useful visual props or possible solutions to a particular issue or simply to provide some light relief.

Each item is clearly labeled with its name and category, but further information can be found by accessing the TechBox's online catalog. Each item has its own page detailing what the item is, why it is interesting, where it came from, and who has used it or knows more about it. Items in the box include an example of metal-coated wood and materials with and without holes that stretch, bend, and change shape or color at different temperatures.

Each of IDEO's offices has a TechBox, and each TechBox has its own curator who is responsible for maintaining and cataloging the items and for promoting its use within the office. Anyone can submit a new item for consideration. As items become commonplace, they are removed from the TechBox to make way for the next generation of fascinating curios. ■

and those that are about characteristics internal to the system that cannot be observed or measured without dissecting it. For example, in a photocopier, externally visible and measurable factors include the physical size of the machine, the speed and quality of copying, the different sizes of paper it can use, and so on. Underlying each of these factors are

DILEMMA

Copying for Inspiration: Is It Legal?

Designers draw on their experience of design when approaching a new project. This includes the use of previous designs that they know work—both designs that they have created themselves and those that others have created. Others' creations often spark inspiration that also leads to new ideas and innovation. This is well known and understood. However, the expression of an idea is protected by copyright, and people who infringe on that copyright can be taken to court and prosecuted. Note that copyright covers the expression of an idea and not the idea itself. This means, for example, that while there are numerous smartphones all with similar functionality, this does not represent an infringement of copyright as the idea has been expressed in different ways and it is the expression that has been copyrighted. Copyright is free and is automatically invested in the author, for instance, the writer of a book or a programmer who develops a program, unless they sign the copyright over to someone else. Employment contracts often include a statement that the copyright relating to anything produced in the course of that employment is automatically assigned to the employer and does not remain with the employee.

Patenting is an alternative to copyright that does protect the idea rather than the expression of the idea. There are various forms of patenting, each of which is designed to allow the inventor to capitalize on their idea. For example, Amazon patented its one-click purchasing process, which allows regular users simply to choose a purchase and buy it with one mouse click (US Patent No. 5960411, September 29, 1999). This is possible because the system stores its customers' details and recognizes them when they access the Amazon site again.

In recent years, the creative commons community (<https://creativecommons.org/>) has suggested more flexible licensing arrangements that allow others to reuse and extend a piece of created work, thereby supporting collaboration. In the open source software development movement, for example, software code is freely distributed and can be modified, incorporated into other software, and redistributed under the same open source conditions. No royalty fees are payable on any use of open source code. These movements do not replace copyright or patent law, but they provide an alternative route for the dissemination of ideas.

So, the dilemma comes in knowing when it is OK to use someone else's work as a source of inspiration and when you are infringing copyright or patent law. The issues are complex and detailed and well beyond the scope of this book, but Bainbridge (2014) is a good resource to understand this area better. ■

other considerations that cannot be observed or studied without dissecting the machine. For example, the choice of materials used in a photocopier may depend on its friction rating and how much it deforms under certain conditions. In interaction design, the user experience is the driving force behind the design and so externally visible and measurable behavior is the main focus. Detailed internal workings are still important to the extent that they affect external behavior or features.

One answer to the previous question is that choosing between alternative designs is informed by letting users and stakeholders interact with them and by discussing their experiences, preferences, and suggestions for improvement. To do this, the designs must be in a form that can be reasonably evaluated by users, not in technical jargon or notation that seems impenetrable to them. Documentation is one traditional way to communicate a design, for example, a diagram showing the product's components or a description of how it works. But a static description cannot easily capture the dynamics of behavior, and for an interactive product this needs to be communicated so that users can see what it will be like to operate it.

Prototyping is often used to overcome potential client misunderstandings and to test the technical feasibility of a suggested design and its production. It involves producing a limited version of the product with the purpose of answering specific questions about the design's feasibility or appropriateness. Prototypes give a better impression of the user experience than simple descriptions; different kinds of prototyping are suitable for different stages of development and for eliciting different kinds of feedback. When a deployable version of the product is available, another way to choose between alternative designs is to deploy two different variations and collect data from actual use that is then used to inform the choice. This is called *A/B testing*, and it is often used for alternative website designs (see Box 2.5).

Another basis on how to choose between alternatives is quality, but that requires a clear understanding of what quality means, and people's views of quality vary. Everyone has a notion of the level of quality that is expected, wanted, or needed from a product. Whether this is expressed formally, informally, or not at all, it exists and informs the choice between alternatives. For example, one smartphone design might make it easy to access a popular music channel but restrict sound settings, while another requires more complicated key sequences to access the channel but has a range of sophisticated sound settings. One user's view of quality may lean toward ease of use, while another may lean toward sophisticated sound settings.

Most projects involve a range of different stakeholder groups, and it is common for each of them to define quality differently and to have different acceptable limits for it. For example, although all stakeholders may agree on goals for a video game such as "characters will be appealing" or "graphics will be realistic," the meaning of these statements can vary between different groups. Disputes will arise if, later in development, it transpires that "realistic" to a stakeholder group of teenage players is different from "realistic" to a group of parent stakeholders or to developers. Capturing these different views clearly clarifies expectations, provides a benchmark against which products and prototypes can be compared, and forms a basis on which to choose among alternatives.

The process of writing down formal, verifiable—and hence measurable—usability criteria is a key characteristic of an approach to interaction design called usability engineering. This has emerged over many years and with various proponents (Whiteside et al., 1988; Nielsen, 1993). Most recently, it is often applied in health informatics (for example, see Kushniruk et al., 2015). Usability engineering involves specifying quantifiable measures of product performance, documenting them in a usability specification, and assessing the product against them.

BOX 2.5

A/B Testing

A/B testing is an online method to inform the choice between two alternatives. It is most commonly used for comparing different versions of web pages or apps, but the principles and mathematics behind it came about in the 1920s (Gallo, 2017). In an interaction design context, different versions of web pages or apps are released for use by users performing their everyday tasks. Typically, users are unaware that they are contributing to an evaluation. This is a powerful way to involve users in choosing between alternatives because a huge number of users can be involved and the situations are authentic.

On the one hand, it's a simple idea—give one set of users one version and a second set the other version and see which set scores more highly against the success criteria. But dividing up the sets, choosing the success criteria, and working out the metrics to use are nontrivial (for example, see Deng and Shi, 2016). Pushing this idea further, it is common to have “multivariate” testing in which several options are tried at once, so you end up doing A/B/C testing or even A/B/C/D testing. ■

ACTIVITY 2.5

Consider your product from Activity 2.1. Suggest some usability criteria that could be applied to determine its quality. Use the usability goals introduced in Chapter 1—effectiveness, efficiency, safety, utility, learnability, and memorability. Be as specific as possible. Check the criteria by considering exactly what to measure and how to measure its performance.

Then try to do the same thing for some of the user experience goals introduced in Chapter 1. (These relate to whether a system is satisfying, enjoyable, motivating, rewarding, and so on.)

Comment

Finding measurable characteristics for some of these is not easy. Here are some suggestions, but there are others. Where possible, criteria that are measurable and specific are preferable.

- *Effectiveness*: Identifying measurable criteria for this goal is particularly difficult since it is a combination of the other goals. For example, does the system support travel organization, choosing transport routes, booking accommodation, and so on? In other words, is the product used?
- *Efficiency*: Is it clear how to ask for recommendations from the product? How quickly does it identify a suitable route or destination details?

(Continued)

- *Safety*: How often does data get lost or is the wrong option chosen? This may be measured, for example, as the number of times this happens per trip.
- *Utility*: How many functions offered are used for every trip, how many every other trip, and how many are not used at all? How many tasks are difficult to complete in a reasonable time because functionality is missing or the right subtasks aren't supported?
- *Learnability*: How long does it take for a novice user to be able to do a series of set tasks, for example, to book a hotel room in Paris near the meeting venue for the meeting dates, identify appropriate flights from Sydney to Wellington, or find out whether a visa is needed to go to China?
- *Memorability*: If the product isn't used for a month, how many functions can the user remember how to perform? How long does it take to remember how to perform the most frequent task?

Finding measurable characteristics for the user experience criteria is harder. How do you measure satisfaction, fun, motivation, or aesthetics? What is entertaining to one person may be boring to another; these kinds of criteria are subjective and so cannot be measured as objectively. ■

2.3.5 How to Integrate Interaction Design Activities Within Other Lifecycle Models

As illustrated in Chapter 1 (Figure 1.4), many other disciplines contribute to interaction design, and some of these disciplines have lifecycles of their own. Prominent among them are those associated with software development, and integrating interaction design activities within software development has been discussed for many years; for example, see Carmelo Ardito et al. (2014) and Ahmed Seffah et al. (2005).

The latest attempts to integrate these practices focus on agile software development. Agile methods began to emerge in the late 1990s. The most well-known of these are eXtreme Programming (Beck and Andres, 2005), Scrum (Schwaber and Beedle, 2002), and Kanban (Anderson, 2010). The Dynamic Systems Development Method (DSDM) (DSDM, 2014), although established before the current agile movement, also belongs to the agile family as it adheres to the agile manifesto. These methods differ, but they all stress the importance of iteration, early and repeated user feedback, being able to handle emergent requirements, and striking a good balance between flexibility and structure. They also all emphasize collaboration, face-to-face communication, streamlined processes to avoid unnecessary activities, and the importance of practice over process, that is, of getting work done.

The opening statement for the *Manifesto for Agile Software Development* (www.agilemanifesto.org/) reads as follows:

We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:

- *Individuals and interactions over processes and tools*
- *Working software over comprehensive documentation*
- *Customer collaboration over contract negotiation*
- *Responding to change over following a plan*

This manifesto is underpinned by a series of principles, which range from communication with the business to excellence of coding and maximizing the amount of work done. The agile approach to development is particularly interesting from the point of view of interaction design because it incorporates tight iterations and feedback and collaboration with the customer. For example, in Scrum, each sprint is between one and four weeks, with a product of value being delivered at the end of each sprint. Also, eXtreme¹ Programming (XP) stipulates that the customer should be on-site with developers. In practice, the customer role is usually taken by a team rather than by one person (Martin et al., 2009), and integration is far from straightforward (Ferreira et al., 2012). Many companies have integrated agile methods with interaction design practices to produce a better user experience and business value (Loranger and Laubheimer, 2017), but it is not necessarily easy, as discussed in Chapter 13, “Interaction Design in Practice.”

In-Depth Activity

These days, timepieces (such as clocks, wristwatches, and so on) have a variety of functions. Not only do they tell the time and date, but they can speak to you, remind you when it's time to do something, and record your exercise habits among other things. The interface for these devices, however, shows the time in one of two basic ways: as a digital number such as 11:40 or through an analog display with two or three hands—one to represent the hour, one for the minutes, and one for the seconds.

This in-depth activity is to design an innovative timepiece. This could be in the form of a wristwatch, a mantelpiece clock, a sculpture for a garden or balcony, or any other kind of timepiece you prefer. The goal is to be inventive and exploratory by following these steps:

- (a) Think about the interactive product that you are designing: What do you want it to do? Find three to five potential users, and ask them what they would want. Write a list of requirements for the clock, together with some usability criteria and user experience criteria based on the definitions in Chapter 1.
- (b) Look around for similar devices and seek out other sources of inspiration that you might find helpful. Make a note of any findings that are interesting, useful, or insightful.
- (c) Sketch some initial designs for the timepiece. Try to develop at least two distinct alternatives that meet your set of requirements.
- (d) Evaluate the two designs by using your usability criteria and by role-playing an interaction with your sketches. Involve potential users in the evaluation, if possible. Does it do what you want? Is the time or other information being displayed always clear? Design is iterative, so you may want to return to earlier elements of the process before you choose one of your alternatives.

¹The method is called *extreme* because it pushes a key set of good practices to the limit; that is, it is good practice to test often, so in XP the development is test-driven, and a complete set of tests is executed many times a day. It is good practice to talk to people about their requirements, so rather than having weighty documentation, XP reduces documentation to a minimum, thus forcing communication, and so on.

Summary

In this chapter, we looked at user-centered design and the process of interaction design. That is, what is user-centered design, what activities are required in order to design an interactive product, and how are these activities related? A simple interaction design lifecycle model consisting of four activities was introduced, and issues surrounding the involvement and identification of users, generating alternative designs, evaluating designs, and integrating user-centered concerns with other lifecycles were discussed.

Key Points

- Different design disciplines follow different approaches, but they have commonalities that are captured in the double diamond of design.
- It is important to have a good understanding of the problem space before trying to build anything.
- The interaction design process consists of four basic activities: discover requirements, design alternatives that meet those requirements, prototype the designs so that they can be communicated and assessed, and evaluate them.
- User-centered design rests on three principles: early focus on users and tasks, empirical measurement, and iterative design. These principles are also key for interaction design.
- Involving users in the design process assists with expectation management and feelings of ownership, but how and when to involve users requires careful planning.
- There are many ways to understand who users are and what their goals are in using a product, including rapid iterations of working prototypes.
- Looking at others' designs and involving other people in design provides useful inspiration and encourages designers to consider alternative design solutions, which is key to effective design.
- Usability criteria, technical feasibility, and users' feedback on prototypes can all be used to choose among alternatives.
- Prototyping is a useful technique for facilitating user feedback on designs at all stages.
- Interaction design activities are becoming better integrated with lifecycle models from other related disciplines such as software engineering.

Further Reading

ASHMORE, S. and RUNYAN, K. (2015) *Introduction to Agile Methods*, Addison Wesley. This book introduces the basics of agile software development and the most popular agile methods in an accessible way. It touches on usability issues and the relationship between agile and marketing. It is a good place to start for someone new to the agile way of working.

KELLEY, T., with LITTMAN, J. (2016) *The Art of Innovation*, Profile Books. Tom Kelley is a partner at IDEO. In this book, Kelley explains some of the innovative techniques used at

IDEO, but more importantly he talks about the culture and philosophy underlying IDEO's success. There are some useful practical hints in here as well as an informative story about building and maintaining a successful design company.

PRESSMAN, R.S. and MAXIM, B.R. (2014) *Software Engineering: A Practitioner's Approach (Int'l Ed)*, McGraw-Hill Education. If you are interested in pursuing the software engineering aspects of the lifecycle models section, then this book provides a useful overview of the main models and their purpose.

SIROKER, D. and KOOMEN, P. (2015) *A/B Testing: The Most Powerful Way to Turn Clicks into Customers*, John Wiley. This book is written by two experienced practitioners who have been using A/B testing with a range of organizations. It is particularly interesting because of the example cases that show the impact that applying A/B testing successfully can have.

ROGERS, Y. (2014) *Secrets of Creative People* (PDF available from www.id-book.com/). This short book summarizes the findings from a two-year research project into creativity. It emphasizes the importance of different perspectives to creativity and describes how successful creativity arises from sharing, constraining, narrating, connecting, and even sparring with others.

