CHAPTER SIX

DESIGN THINKING

The Double-Diamond Model of Design

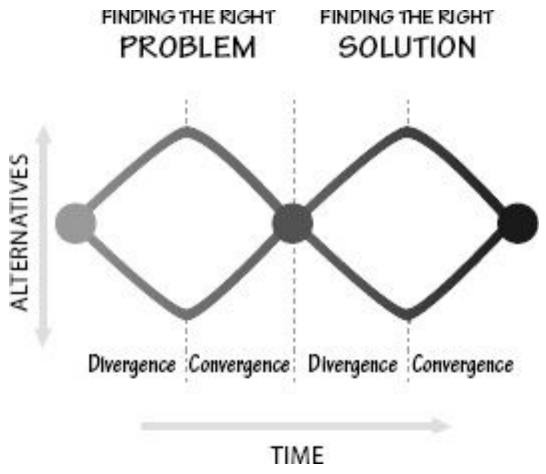

FIGURE 6.1. The Double-Diamond Model of Design. Start with an idea, and through the initial design research, expand the thinking to explore the fundamental issues. Only then is it time to converge upon the real, underlying problem. Similarly, use design research tools to explore a wide variety of solutions before converging upon one. (Slightly modified from the work of the British Design Council, 2005.)

Designers often start by questioning the problem given to them: they expand the scope of the problem, diverging to examine all the fundamental issues that underlie it. Then they converge upon a single problem statement. During the solution phase of their studies, they first expand the space of possible solutions, the divergence phase. Finally, they converge upon a proposed solution (Figure 6.1). This double diverge-converge pattern was first introduced in 2005 by the British Design Council, which called it the *double-diamond design process model*. The Design Council divided the design process into four stages: “discover” and “define”—for the divergence and convergence phases of finding the right problem, and “develop” and “deliver”—for the divergence and convergence phases of finding the right solution.

The double diverge-converge process is quite effective at freeing designers from unnecessary restrictions to the problem and solution spaces. But you can sympathize with a product manager who, having given the designers a problem to solve, finds them questioning the assignment and insisting on traveling all over the world to seek deeper understanding. Even when the designers start focusing upon the problem, they do not seem to make progress, but instead develop a wide variety of ideas and thoughts, many only half-formed, many clearly impractical. All this can be rather unsettling to the product manager who, concerned about meeting the schedule, wants to see immediate convergence. To add to the frustration of the product manager, as the designers start to converge upon a solution, they may realize that they have inappropriately formulated the problem, so the entire process must be repeated (although it can go more quickly this time).

This repeated divergence and convergence is important in properly determining the right problem to be solved and then the best way to solve it. It looks chaotic and ill-structured, but it actually follows well-established principles and procedures. How does the product manager keep the entire team on schedule despite the apparent random and divergent methods of designers? Encourage their free exploration, but hold them to the schedule (and budget) constraints. There is nothing like a firm deadline to get creative minds to reach convergence.

The Human-Centered Design Process

The double-diamond describes the two phases of design: finding the right problem and fulfilling human needs. But how are these actually done? This is where the human-centered design process comes into play: it takes place within the double-diamond diverge-converge process.

There are four different activities in the human-centered design process (Figure 6.2):

1. Observation
2. Idea generation (ideation)
3. Prototyping
4. Testing

These four activities are iterated; that is, they are repeated over and over, with each cycle yielding more insights and getting closer to the desired solution. Now let us examine each activity separately.

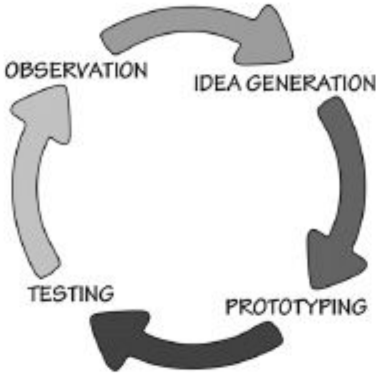

FIGURE 6.2. The Iterative Cycle of Human-Centered Design. Make observations on the intended target population, generate ideas, produce prototypes and test them. Repeat until satisfied. This is often called the *spiral method* (rather than the circle depicted here), to emphasize that each iteration through the stages makes progress.

OBSERVATION

The initial research to understand the nature of the problem itself is part of the discipline of design research. Note that this is research about the customer and the people who will use the products under consideration. It is not the kind of research that scientists do in their laboratories, trying to find new laws of nature. The design researcher will go to the potential customers, observing their activities, attempting to understand their interests, motives, and true needs. The problem definition for the product design will come from this deep understanding of the goals the people are trying to accomplish and the impediments they experience. One of its most critical techniques is to observe the would-be customers in their natural environment, in their normal lives, wherever the product or service being designed will actually be used. Watch them in their homes, schools, and offices. Watch them commute, at parties, at mealtime, and with friends at the local bar. Follow them into the shower if necessary, because it is essential to understand the real situations that they encounter, not some pure isolated experience. This technique is called *applied ethnography*, a method adapted from the field of anthropology. Applied ethnography differs from the slower, more methodical, research-oriented practice of academic anthropologists because the goals are different. For one, design researchers have the goal of determining human needs that can be addressed through new products. For another, product cycles are driven by schedule and budget, both of which require more rapid assessment than is typical in academic studies that might go on for years.

It's important that the people being observed match those of the intended audience. Note that traditional measures of people, such as age, education, and income, are not always important: what matters most are the activities to be performed. Even when we look at widely different cultures, the activities are often surprisingly similar. As a result, the studies can focus upon the activities and how they get done, while being sensitive to how the local environment and culture might modify those activities. In some cases, such as the products widely used in business, the activity dominates. Thus, automobiles, computers, and phones are pretty standardized across the world because their designs reflect the activities being supported.

In some cases, detailed analyses of the intended group are necessary. Japanese teenage girls are quite different from Japanese women, and in turn, very different from German teenage girls. If a product is intended for subcultures like these, the exact population must be studied. Another way of putting it is that different products serve different needs. Some products are also symbols of status or group membership. Here, although they perform useful functions, they are also fashion statements. This is where teenagers in one culture differ from those of another, and even from younger children and older adults of the same culture. Design researchers must carefully adjust the focus of their observations to the intended market and people for whom the product is intended.

Will the product be used in some country other than where it is being designed? There is only one way to find out: go there (and always include natives in the team). Don't take a shortcut and stay home, talking to students or visitors from that country while remaining in your own: what you will learn is seldom an accurate reflection of the target population or of the ways in which the proposed product will actually be used. There is no substitute for direct observation of and interaction with the people who will be using the product.

Design research supports both diamonds of the design process. The first diamond, finding the right problem, requires a deep understanding of the true needs of people. Once the problem has been defined, finding an appropriate solution again requires deep understanding of the intended population, how those people perform their activities, their capabilities and prior experience, and what cultural issues might be impacted.

DESIGN RESEARCH VERSUS MARKET RESEARCH

Design and marketing are two important parts of the product development group. The two fields are complementary, but each has a different focus. Design wants to know what people really need and how they actually will use the product or service under consideration. Marketing wants to know what people will buy, which includes learning how they make their purchasing decisions. These different aims lead the two groups to develop different methods of inquiry. Designers tend to use qualitative observational methods by which they can study people in depth, understanding how they do their activities and the environmental factors that come into play. These methods are very time consuming, so designers typically only examine small numbers of people, often numbering in the tens.

Marketing is concerned with customers. Who might possibly purchase the item? What factors might entice them to consider and purchase a product? Marketing traditionally uses large-scale, quantitative studies, with heavy reliance on focus groups, surveys, and questionnaires. In marketing, it is not uncommon to converse with hundreds of people in focus groups, and to question tens of thousands of people by means of questionnaires and surveys.

The advent of the Internet and the ability to assess huge amounts of data have given rise to new methods of formal, quantitative market analysis. "Big data," it is called, or sometimes "market analytics." For popular websites, A/B testing is possible in which two potential variants of an offering are tested by giving some randomly selected fraction of visitors (perhaps 10 percent) one set of web pages (the A set); and another randomly selected set of visitors, the other alternative (the B set). In a few hours, hundreds of thousands of visitors may have been exposed to each test set, making it easy to see which yields better results. Moreover, the website can capture a wealth of information about people and their behavior: age, income, home and work addresses, previous purchases, and other websites visited. The virtues of the use of big data for market research are frequently touted. The deficiencies are seldom noted, except for concerns about invasions of personal privacy. In addition to privacy issues, the real problem is that numerical correlations say nothing of people's real needs, of their desires, and of the reasons for their activities. As a result, these numerical data can give a false impression of people. But the use of big data and market analytics is seductive: no travel, little expense, and huge numbers, sexy charts, and impressive statistics, all very persuasive to the executive team trying to decide which new products to develop. After all, what would you trust—neatly presented, colorful charts, statistics, and significance levels based on millions of observations, or the subjective impressions of a motley crew of design researchers who worked, slept, and ate in remote villages, with minimal sanitary facilities and poor infrastructure?

The different methods have different goals and produce very different results. Designers complain that the methods used by marketing don't get at real behavior: what people say they do and want does not correspond with their actual behavior or desires. People in marketing complain that although design research methods yield deep insights, the small number of people observed is a concern. Designers counter with the observation that traditional marketing methods provide shallow insight into a large number of people.

The debate is not useful. All groups are necessary. Customer research is a tradeoff: deep insights on real needs from a tiny set of people, versus broad, reliable purchasing data from a wide range and large number of people. We need both. Designers understand what people really need. Marketing understands what people actually buy. These are not the same things, which is why both approaches are required: marketing and design researchers should work together in complementary teams.

What are the requirements for a successful product? First, if nobody buys the product, then all else is irrelevant. The product design has to provide support for all the factors people use in making purchase decisions. Second, once the product has been purchased and is put into use, it must support real needs so that people can use, understand, and take pleasure from it. The design specifications must include both factors: marketing and design, buying and using.

IDEA GENERATION

Once the design requirements are determined, the next step for a design team is to generate potential solutions. This process is called *idea generation*, or *ideation*. This exercise might be done for both of the double diamonds: during the phase of finding the correct problem, then during the problem solution phase.

This is the fun part of design: it is where creativity is critical. There are many ways of generating ideas: many of these methods fall under the heading of "brainstorming." Whatever the method used, two major rules are usually followed:

- **Generate numerous ideas.** It is dangerous to become fixated upon one or two ideas too early in the process.
- **Be creative without regard for constraints.** Avoid criticizing ideas, whether your own or those of others. Even crazy ideas, often obviously wrong, can contain creative insights that can later be extracted and put to good use in the final idea selection. Avoid premature dismissal of ideas.

I like to add a third rule:

- **Question everything.** I am particularly fond of "stupid" questions. A stupid question asks about things so fundamental that everyone assumes the answer is obvious. But when the question is taken seriously, it often turns out to be profound: the obvious often is not obvious at all. What we assume to be obvious is simply the way things have always been done, but now that it is questioned, we don't actually know the reasons. Quite often the solution to problems is discovered through stupid questions, through questioning the obvious.

PROTOTYPING

The only way to really know whether an idea is reasonable is to test it. Build a quick prototype or mock-up of each potential solution. In the early stages of this process, the mock-ups can be pencil sketches, foam and cardboard models, or simple images made with simple drawing tools. I have made mock-ups with spreadsheets, PowerPoint slides, and with sketches on index cards or sticky notes. Sometimes ideas are best conveyed by skits, especially if you're developing services or automated systems that are difficult to prototype.

One popular prototype technique is called "Wizard of Oz," after the wizard in L. Frank Baum's classic book (and the classic movie) *The Wonderful Wizard of Oz*. The wizard was actually just an ordinary person but, through the use of smoke and mirrors, he managed to appear mysterious and omnipotent. In other words, it was all a fake: the wizard had no special powers.

The Wizard of Oz method can be used to mimic a huge, powerful system long before it can be built. It can be remarkably effective in the early stages of product development. I once used this method to test a system for making airline reservations that had been designed by a research group at the Xerox Corporation's Palo Alto Research Center (today it is simply the Palo Alto Research Center, or PARC). We brought people into my laboratory in San Diego one at a time, seated them in a small, isolated room, and had them type their travel requirements into a computer. They thought they were interacting with an automated travel assistance program, but in fact, one of my graduate students was sitting in an adjacent room, reading the typed queries and typing back responses (looking up real travel schedules where appropriate). This simulation taught us a lot about the requirements for such a system. We learned, for example, that people's sentences were very different from the ones we had designed the system to handle. Example: One of the people we tested requested a round-trip ticket between San Diego and San Francisco. After the system had determined the desired flight to San Francisco, it asked, "When would you like to return?" The person responded, "I would like to leave on the following Tuesday, but I have to be back before my first class at 9 AM." We soon learned that it wasn't sufficient to understand the sentences: we also had to do problem-solving, using considerable knowledge about such things as airport and meeting locations, traffic patterns, delays for getting baggage and rental cars, and of course, parking—more than our system was capable of doing. Our initial goal was to understand language. The studies demonstrated that the goal was too limited: we needed to understand human activities.

Prototyping during the problem specification phase is done mainly to ensure that the problem is well understood. If the target population is already using something related to the new product, that can be considered a prototype. During the problem solution phase of design, then real prototypes of the proposed solution are invoked.

TESTING

Gather a small group of people who correspond as closely as possible to the target population—those for whom the product is intended. Have them use the prototypes as nearly as possible to the way they would actually use them. If the device is normally used by one person, test one person at a time. If it is normally used by a group, test a group. The only exception is that even if the normal usage is by a single person, it is useful to ask a pair of people to use it together, one person operating the prototype, the other guiding the actions and interpreting the results (aloud). Using pairs in this way causes them to discuss their ideas, hypotheses, and frustrations openly and naturally. The research team should be observing, either by sitting behind those being tested (so as not to distract them) or by watching through video in another room (but having the video camera visible and after describing the procedure). Video recordings of the tests are often quite valuable, both for later showings to team members who could not be present and for review.

When the study is over, get more detailed information about the people's thought processes by retracing their steps, reminding them of their actions, and questioning them. Sometimes it helps to show them video recordings of their activities as reminders.

How many people should be studied? Opinions vary, but my associate, Jakob Nielsen, has long championed the number five: five people studied individually. Then, study the results, refine them, and do another iteration, testing five different people. Five is usually enough to give major findings. And if you really want to test many more people, it is far more effective to do one test of five, use the results to improve the system, and then keep iterating the test-design cycle until you have tested the desired number of people. This gives multiple iterations of improvement, rather than just one.

Like prototyping, testing is done in the problem specification phase to ensure that the problem is well understood, then done again in the problem solution phase to ensure that the new design meets the needs and abilities of those who will use it.

ITERATION

The role of iteration in human-centered design is to enable continual refinement and enhancement. The goal is rapid prototyping and testing, or in the words of David Kelly, Stanford professor and cofounder of the design firm IDEO, “Fail frequently, fail fast.”

Many rational executives (and government officials) never quite understand this aspect of the design process. Why would you want to fail? They seem to think that all that is necessary is to determine the requirements, then build to those requirements. Tests, they believe, are only necessary to ensure that the requirements are met. It is this philosophy that leads to so many unusable systems. Deliberate tests and modifications make things better. Failures are to be encouraged —actually, they shouldn't be called failures: they should be thought of as learning experiences. If everything works perfectly, little is learned. Learning occurs when there are difficulties.

The hardest part of design is getting the requirements right, which means ensuring that the right problem is being solved, as well as that the solution is appropriate. Requirements made in the abstract are invariably wrong. Requirements produced by asking people what they need are invariably wrong. Requirements are developed by watching people in their natural environment.

When people are asked what they need, they primarily think of the everyday problems they face, seldom noticing larger failures, larger needs. They don’t question the major methods they use. Moreover, even if they carefully explain how they do their tasks and then agree that you got it right when you present it back to them, when you watch them, they will often deviate from their own description. “Why?” you ask. “Oh, I had to do this one differently,” they might reply; “this was a special case.” It turns out that most cases are “special.” Any system that does not allow for special cases will fail.

Getting the requirements right involves repeated study and testing: iteration. Observe and study: decide what the problem might be, and use the results of tests to determine which parts of the design work, which don’t. Then iterate through all four processes once again. Collect more design research if necessary, create more ideas, develop the prototypes, and test them.

With each cycle, the tests and observations can be more targeted and more efficient. With each cycle of the iteration, the ideas become clearer, the specifications better defined, and the prototypes closer approximations to the target, the actual product. After the first few iterations, it is time to start converging upon a solution. The several different prototype ideas can be collapsed into one.

When does the process end? That is up to the product manager, who needs to deliver the highest-possible quality while meeting the schedule. In product development, schedule and cost provide very strong constraints, so it is up to the design team to meet these requirements while getting to an acceptable, high-quality design. No matter how much time the design team has been allocated, the final results only seem to appear in the last twenty-four hours before the deadline. (It’s like writing: no matter how much time you are given, it’s finished only hours before the deadline.)

ACTIVITY-CENTERED VERSUS HUMAN-CENTERED DESIGN

The intense focus on individuals is one of the hallmarks of human-centered design, ensuring that products do fit real needs, that they are usable and understandable. But what if the product is intended for people all across the world? Many manufacturers make essentially the same product for everyone. Although automobiles are slightly modified for the requirements of a country, they are all basically the same the world round. The same is true for cameras, computers, telephones, tablets, television sets, and refrigerators. Yes, there are some regional differences, but remarkably little. Even products specifically designed for one culture—rice cookers, for example—get adopted by other cultures elsewhere.

How can we pretend to accommodate all of these very different, very disparate people? The answer is to focus on activities, not the individual person. I call this *activity-centered design*. Let the activity define the product and its structure. Let the conceptual model of the product be built around the conceptual model of the activity.

Why does this work? Because people’s activities across the world tend to be similar. Moreover, although people are unwilling to learn systems that appear to have arbitrary, incomprehensible requirements, they are quite willing to learn things that appear to be essential to the activity. Does this violate the principles of human-centered design? Not at all: consider it an enhancement of HCD. After all, the activities are done by and for people. Activity-centered approaches are human-centered approaches, far better suited for large, nonhomogeneous populations.

Take another look at the automobile, basically identical all across the world. It requires numerous actions, many of which make little sense outside of the activity and that add to the complexity of driving and to the rather long period it takes to become an accomplished, skilled driver. There is the need to master foot pedals, to steer, use turn signals, control the lights, and watch the road, all while being aware of events on either side of and behind the vehicle, and perhaps while maintaining conversations with the other people in the auto. In addition, instruments on the panel need to be watched, especially the speed indicator, as well as the water temperature, oil pressure, and fuel level. The locations of the rear-and side-view mirrors require the eyes to be off the road ahead for considerable time.

People learn to drive cars quite successfully despite the need to master so many subcomponent tasks. Given the design of the car and the activity of driving, each task seems appropriate. Yes, we can make things better. Automatic transmissions eliminate the need for the third pedal, the clutch. Heads-up displays mean that critical instrument panel and navigation information can be displayed in the space in front of the driver, so no eye movements are required to monitor them (although it requires an attentional shift, which does take attention off the road). Someday we will replace the three different mirrors with one video display that shows objects on all sides of the car in one image, simplifying yet another action. How do we make things better? By careful study of the activities that go on during driving.

Support the activities while being sensitive to human capabilities, and people will accept the design and learn whatever is necessary.

ON THE DIFFERENCES BETWEEN TASKS AND ACTIVITIES

One comment: there is a difference between task and activity. I emphasize the need to design for activities: designing for tasks is usually too restrictive. An activity is a high-level structure, perhaps “go shopping.” A task is a lower-level component of an activity, such as “drive to the market,” “find a shopping basket,” “use a shopping list to guide the purchases,” and so forth.

An activity is a collected set of tasks, but all performed together toward a common high-level goal. A task is an organized, cohesive set of operations directed toward a single, low-level goal. Products have to provide support for both activities and the various tasks that are involved. Well-designed devices will package together the various tasks that are required to support an activity, making them work seamlessly with one another, making sure the work done for one does not interfere with the requirements for another.

Activities are hierarchical, so a high-level activity (going to work) will have under it numerous lower-level ones. In turn, low-level activities spawn “tasks,” and tasks are eventually executed by basic “operations.” The American psychologists Charles Carver and Michael Scheier suggest that goals have three fundamental levels that control activities. Be-goals are at the highest, most abstract level and govern a person’s being: they determine why people act, are fundamental and long lasting, and determine one’s self-image. Of far more practical concern for everyday activity is the next level down, the do-goal, which is more akin to the goal I discuss in the seven stages of activity. Do-goals determine the plans and actions to be performed for an activity. The lowest level of this hierarchy is the motor-goal, which specifies just how the actions are performed: this is more at the level of tasks and operations rather than activities. The German psychologist Marc Hassenzahl has shown how this three-level analysis can be used to guide in the development and analysis of a person's experience (the user experience, usually abbreviated UX) in interacting with products.

Focusing upon tasks is too limiting. Apple's success with its music player, the iPod, was because Apple supported the entire activity involved in listening to music: discovering it, purchasing it, getting it into the music player, developing playlists (that could be shared), and listening to the music. Apple also allowed other companies to add to the capabilities of the system with external speakers, microphones, all sorts of accessories. Apple made it possible to send the music throughout the home, to be listened to on those other companies' sound systems. Apple's success was due to its combination of two factors: brilliant design plus support for the entire activity of music enjoyment.

Design for individuals and the results may be wonderful for the particular people they were designed for, but a mismatch for others. Design for activities and the result will be usable by everyone. A major benefit is that if the design requirements are consistent with their activities, people will tolerate complexity and the requirements to learn something new: as long as the complexity and the new things to be learned feel appropriate to the task, they will feel natural and be viewed as reasonable.

ITERATIVE DESIGN VERSUS LINEAR STAGES

The traditional design process is linear, sometimes called the *waterfall method* because progress goes in a single direction, and once decisions have been made, it is difficult or impossible to go back. This is in contrast to the iterative method of human-centered design, where the process is circular, with continual refinement, continual change, and encouragement of backtracking, rethinking early decisions. Many software developers experiment with variations on the theme, variously called by such names as Scrum and Agile.

Linear, waterfall methods make logical sense. It makes sense that design research should precede design, design precede engineering development, engineering precede manufacturing, and so on. Iteration makes sense in helping to clarify the problem statement and requirements; but when projects are large, involving considerable people, time, and budget, it would be horribly expensive to allow iteration to last too long. On the other hand, proponents of iterative development have seen far too many project teams rush to develop requirements that later prove to be faulty, sometimes wasting huge amounts of money as a result. Numerous large projects have failed at a cost of multiple billions of dollars.

The most traditional waterfall methods are called *gated* methods because they have a linear set of phases or stages, with a gate blocking transition from one stage to the next. The gate is a management review during which progress is evaluated and the decision to proceed to the next stage is made.

Which method is superior? As is invariably the case where fierce debate is involved, both have virtues and both have deficits. In design, one of the most difficult activities is to get the specifications right: in other words, to determine that the correct problem is being solved. Iterative methods are designed to defer the formation of rigid specifications, to start off by diverging across a large set of possible requirements or problem statements before convergence, then again diverging across a large number of potential solutions before converging. Early prototypes have to be tested through real interaction with the target population in order to refine the requirements.

The iterative method, however, is best suited for the early design phases of a product, not for the later stages. It also has difficulty scaling its procedures to handle large projects. It is extremely difficult to deploy successfully on projects that involve hundreds or even thousands of developers, take years to complete, and cost in the millions or billions of dollars. These large projects include complex consumer goods and large programming jobs, such as automobiles; operating systems for computers, tablets, and phones; and word processors and spreadsheets.

Decision gates give management much better control over the process than they have in the iterative methods. However, they are cumbersome. The management reviews at each of the gates can take considerable time, both in preparation for them and then in the decision time after the presentations. Weeks can be wasted because of the difficulty of scheduling all the senior executives from the different divisions of the company who wish to have a say.

Many groups are experimenting with different ways of managing the product development process. The best methods combine the benefits of both iteration and stage reviews. Iteration occurs inside the stages, between the gates. The goal is to have the best of both worlds: iterative experimentation to refine the problem and the solution, coupled with management reviews at the gates.

The trick is to delay precise specification of the product requirements until some iterative testing with rapidly deployed prototypes has been done, while still keeping tight control over schedule, budget, and quality. It may appear impossible to prototype some large projects (for example, large transportation systems), but even there a lot can be done. The prototypes might be scaled objects, constructed by model makers or 3-D printing methods. Even well-rendered drawings and videos of cartoons or simple animation sketches can be useful. Virtual reality computer aids allow people to envision themselves using the final product, and in the case of a building, to envision living or working within it. All of these methods can provide rapid feedback before much time or money has been expended.

The hardest part of the development of complex products is management: organizing and communicating and synchronizing the many different people, groups, and departmental divisions that are required to make it happen. Large projects are especially difficult, not only because of the problem of managing so many different people and groups, but also because the projects' long time horizon introduces new difficulties. In the many years it takes to go from project formulation to completion, the requirements and technologies will probably change, making some of the proposed work irrelevant and obsolete; the people who will make use of the results might very well change; and the people involved in executing the project definitely will change.

Some people will leave the project, perhaps because of illness or injury, retirement or promotion. Some will change companies and others will move on to other jobs in the same company. Whatever the reason, considerable time is lost finding replacements and then bringing them up to the full knowledge and skill level required. Sometimes this is not even possible because critical knowledge about project decisions and methods are in the form we call *implicit knowledge*; that is, within the heads of the workers. When workers leave, their implicit knowledge goes with them. The management of large projects is a difficult challenge.

Deliberately Making Things Difficult

> *How can good design (design that is usable and understandable) be balanced with the need for "secrecy" or privacy, or protection? That is, some applications of design involve areas that are sensitive and necessitate strict control over who uses and understands them. Perhaps we don't want any user-in-the-street to understand enough of a system to compromise its security. Couldn't it be argued that some things shouldn't be designed well? Can't things be left cryptic, so that only those who have clearance, extended education, or whatever, can make use of the system? Sure, we have passwords, keys, and other types of security checks, but this can become wearisome for the privileged user. It appears that if good design is not ignored in some contexts, the purpose for the existence of the system will be nullified.* (A computer mail question sent to me by a student, Dina Kurktchi. It is just the right question.)

In Stapleford, England, I came across a school door that was very difficult to open, requiring simultaneous operation of two latches, one at the very top of the door, the other down low. The latches were difficult to find, to reach, and to use. But the difficulties were deliberate. This was good design. The door was at a school for handicapped children, and the school didn't want the children to be able to get out to the street without an adult. Only adults were large enough to operate the two latches. Violating the rules of ease of use is just what was needed.

Most things are intended to be easy to use, but aren't. But some things are deliberately difficult to use—and ought to be. The number of things that should be difficult to use is surprisingly large:

- Any door designed to keep people in or out.
- Security systems, designed so that only authorized people will be able to use them.
- Dangerous equipment, which should be restricted.
- Dangerous operations that might lead to death or injury if done accidentally or in error.
- Secret doors, cabinets, and safes: you don't want the average person even to know that they are there, let alone to be able to

work them.

- Cases deliberately intended to disrupt the normal routine action (as discussed in Chapter 5). Examples include the acknowledgment required before permanently deleting a file from a computer, safeties on pistols and rifles, and pins in fire extinguishers.
- Controls that require two simultaneous actions before the system will operate, with the controls separated so that it takes two people to work them, preventing a single person from doing an unauthorized action (used in security systems or safety-critical operations).
- Cabinets and bottles for medications and dangerous substances deliberately made difficult to open to keep them secure from children.
- Games, a category in which designers deliberately flout the laws of understandability and usability. Games are meant to be difficult; in some games, part of the challenge is to figure out what is to be done, and how.

Even where a lack of usability or understandability is deliberate, it is still important to know the rules of understandable and usable design, for two reasons. First, even deliberately difficult designs aren't entirely difficult. Usually there is one difficult part, designed to keep unauthorized people from using the device; the rest of it should follow the normal principles of good design. Second, even if your job is to make something difficult to do, you need to know how to go about doing it. In this case, the rules are useful, for they state in reverse just how to go about the task. You could systematically violate the rules like this:

- Hide critical components: make things invisible.
- Use unnatural mappings for the execution side of the action cycle, so that the relationship of the controls to the things being controlled is inappropriate or haphazard.
- Make the actions physically difficult to do.
- Require precise timing and physical manipulation.
- Do not give any feedback.
- Use unnatural mappings for the evaluation side of the action cycle, so that system state is difficult to interpret.

Safety systems pose a special problem in design. Oftentimes, the design feature added to ensure safety eliminates one danger, only to create a secondary one. When workers dig a hole in a street, they must put up barriers to prevent cars and people from falling into the hole. The barriers solve one problem, but they themselves pose another danger, often mitigated by adding signs and flashing lights to warn of the barriers. Emergency doors, lights, and alarms must often be accompanied by warning signs or barriers that control when and how they can be used.

Design: Developing Technology for People

Design is a marvelous discipline, bringing together technology and people, business and politics, culture and commerce. The different pressures on design are severe, presenting huge challenges to the designer. At the same time, the designers must always keep foremost in mind that the products are to be used by people. This is what makes design such a rewarding discipline: On the one hand, woefully complex constraints to overcome; on the other hand, the opportunity to develop things that assist and enrich the lives of people, that bring benefits and enjoyment.