PART

Two

The User Interface Design Process

Part 2 presents an extensive series of guidelines for the interface design process. It is organized in the order of the development steps typically followed in creating a graphical system's or Web site's screens and pages. In total, 14 steps are presented, beginning with "Know Your User or Client" and ending with a discussion of testing. Other topics addressed include considerations in screen design, navigation, screen-based controls, writing messages and text, color, and graphics. This organization scheme enables all the interface design activities to be addressed easily, clearly, and sequentially.

Let's first look at several critical general aspects of the design process. "Obstacles and Pitfalls in the Development Path" points out the realities of designing for people, and some reasons why design may not live up to expectations. "Designing for People: The Seven Commandments" lists the guidelines that are the cornerstones of the entire design process. Then, the concept of usability, the primary objective on any development effort, is defined and discussed.

Obstacles and Pitfalls in the Development Path

Developing a computer system is never easy. The path is littered with obstacles and traps, many of them human in nature. Gould (1988) has made these general observations about design:

- Nobody ever gets it right the first time.
- Development is chock-full of surprises.

- Good design requires living in a sea of changes.
- Making contracts to ignore change will never eliminate the need for change.
- Even if you have made the best system humanly possible, people will still make mistakes when using it.
- Designers need good tools.
- You must have behavioral design goals like performance design goals.

The first five conditions listed will occur naturally because people are people, both as users and as developers. These kinds of behavior must be understood and accepted in design. User mistakes, while they will always occur, can be reduced. Guidelines in the various design steps address this problem. Behavioral design goals are reviewed in Step 2, "Understand the Business Function."

Pitfalls in the design process exist because of a flawed design process, including a failure to address critical design issues, an improper focus of attention, or development team organization failures. Common pitfalls are:

- No early analysis and understanding of the user's needs and expectations.
- A focus on using design features or components that are "neat" or "glitzy."
- Little or no creation of design element prototypes.
- No usability testing.
- No common design team vision of user interface design goals.
- Poor communication between members of the development team.

"Know Your User or Client" is addressed in Step 1. Prototypes and testing are addressed in Step 14, "Test, Test, and Retest."

Designing for People: The Seven Commandments

The complexity of a graphical or Web interface will always magnify any problems that do occur. While obstacles to design will always exist, pitfalls can be eliminated if the following design commandments remain foremost in the development process.

1. **Provide a multidisciplinary design team**. Provide a balanced design team, including specialists in:
   - Development, including system analysis and software design.
   - Interface Design.
   - Visual design.
   - Usability assessment.
   - Documentation.
   - Training.

   Effective design and development requires the application of very diverse talents. No one person possesses all the skills to perform all the necessary tasks; the best that can be hoped for is that one person may possess a couple of skills. A balanced design team with very different talents must be established. Needed are

specialists in development to define requirements and write the software, human factors specialists to define behavioral requirements and apply behavioral considerations, and people with good visual design skills. Also needed are people skilled in testing and usability assessment, documentation specialists, and training specialists.

All designers, however, should be strongly user-oriented. A study (Bailey, 1993) revealed that user-oriented designers are superior to computer-oriented designers when making user interface design decisions. User-oriented designers should be responsible for making the majority of user interface design decisions

Also, select team members who can effectively work and communicate with one another. To optimize communication, locate the team members in close proximity to one another.

2. **Solicit early and ongoing user involvement**. Involving the users in requirements determination and/or testing from the beginning provides a direct conduit to the knowledge they possess about jobs, tasks, system goals, and needs. Different types of users may exist:
   - **End users**. Sometimes simply called users, these are the people who actually use the system to perform tasks and jobs. One caution, however: user involvement of this kind should be based on job or task knowledge, not status or position. The boss seldom knows what is really happening in the office. Throughout the remainder of this text the term "users" will be used to designate end users.
   - **Customers**. These are the people within the using organization who pay for and usually specify the overall objectives and goals of the system.
   - **Other interested parties**. These are people within the user organization who also have an interest in the development of the system.

   Involvement of these different kinds of users also enables the developer to confront people's resistance to change, a common human trait. People dislike change for a variety of reasons, among them fear of the unknown and lack of identification with the system. Involvement in design removes the unknown and gives the user a stake in the system or identification with it.

   It has long been a belief among designers that involving users in the design phase of a system is beneficial in terms of system quality, efficiency, and effectiveness. Kujala (2003) performed a research literature review to clarify the relationship between user involvement during design and development and final system success. Reviewing more than three dozen studies, she reported both positive and negative findings. Bailey (2005a), in reviewing Kujala's findings, reached the following conclusions regarding user involvement:
   - A more accurate set of requirements will be obtained.
   - In some situations there may be improved user acceptance of the system.
   - There is little evidence that systems are either more effective or efficient when users are closely involved in making design decisions.
   - During testing, users can be effectively used as participants. Emphasis should be on obtaining quantitative data.

It is helpful, then, for users to be involved at every stage in the interface design, development, and implementation cycle:

- **Early in the design process when requirements are being determined**. Users can help by providing design requirements and specifications, testing early design prototypes, and by allowing themselves to be observed performing their current tasks. Users can also provide feedback concerning their current system and the prototypes being tested.
- **Throughout prototyping to test designs and options**. Feedback and suggestions can made for each prototype tested in the development process.
- **During training**. Opinions can be gathered and any additional problems described.
- **After system delivery**. Opinions can be gathered and feedback concerning any additional problems encountered during actual system use can be provided.

3. **Gain a complete understanding of users and their tasks**. All users, including customers and other interested parties, today expect a level of design sophistication from all user interfaces, including Web sites. The product, system or Web site must be geared to people's needs and the system's goal, not those of the developers. A wide gap in technical abilities, objectives, and attitudes often exists between users and developers. A failure by developers to understand the differences will doom a product or system to failure.

   Usability goals in the form of measurable objectives must also be established. Set performance goals such as success rates and the time it takes to complete tasks. Set preference goals that address satisfaction and acceptance by users. Design success cannot be determined without quantitative values to compare system performance against.

4. **Create the appropriate design**. The total user experience must be created, including an appropriate allocation of function between the user and the system. Consider as many user interface issues as possible during the design process. A design methodology that has been found to be successful is called *parallel design.* Using this concept, proposed by Ovaska and Raiha (1995), multiple developers independently evaluate design requirements and issues and propose design solutions. Then, to find the best ideas, individual solutions are presented to, and discussed among, all developers. Two studies (Macbeth et al., 2000; McGrew, 2001) have found this process works exceptionally well. More design ideas are presented and considered, and developers responded to good ideas no matter who had proposed them. As Bailey (2002) suggests, the objective is to "saturate the design space." Interface designers should consider as many alternative designs as possible before selecting the best among them to begin the iterative design process.

   Begin utilizing design standards and guidelines at the start of the design process. All interface design decisions must be made as design proceeds, not after design is complete. This helps ensure that the best possible design decisions are made and that design consistency is achieved. This also avoids problems later on in the development process.

5. **Perform rapid prototyping and testing**. Prototyping and testing the product will quickly identify problems and allow solutions to be developed. The design process is complex and human behavior is still not well understood. While the design guidelines that follow go a long way toward achieving ease of use, all problems cannot possibly be predicted. Prototyping and testing must be continually performed during all stages of development to uncover all potential defects.

   If thorough testing is not performed before product release, the testing will occur in the user's office. Encountering a series of problems early in system use will create a negative first impression in the customer's mind, and this may harden quickly, creating attitudes that may be difficult to change. It is also much harder and more costly to fix a product after its release. In many instances, people may adapt to, or become dependent upon, a design, even if it is inefficient. This also makes future modifications much more difficult.

6. **Modify and iterate the design as much as necessary**. Design will be an *iterative* process. Design prototypes will be developed and tested, and changes will be made on the basis of the test results. The process will be repeated, fine-tuning occurring, until all usability goals are achieved. The substantial value of iterative design has been confirmed by several studies (Tan et al., 2001; Bailey and Wolfson, 2005; LeDoux et al., 2005). Each of these studies found that system modifications based upon the results of one test yielded performance improvements on a follow-up test. For example:
   - A 28% faster average task completion time (Tan et al., 2001).
   - A 37% reduction in usability problems (Tan et al., 2001).
   - Nine of ten task scenarios took less time (Bailey and Wolfson, 2005).
   - User satisfaction score increased from 63 to 73 (Bailey and Wolfson, 2005).
   - The average time to complete task scenarios was reduced from 68 to 51 seconds (25% improvement) (LeDoux et al., 2005).
   - The overall user satisfaction score improved from 49 to 82 (67% improvement) (LeDoux et al., 2005).

   Test, modify, and retest has been proven to work well. This process is repeated until all usability goals are achieved. Then the iterative process ends.

7. **Integrate the design of all the system components**. The software, the documentation, the help function, and training needs are all important elements of a graphical system or Web site and all should be developed concurrently. A system is being constructed, not simply software. Concurrent development of all pieces will point out possible problems much earlier in the design process, allowing them to be more effectively addressed. Time will also exist for design trade-offs to be thought out more carefully.

Usability

Much of the development process will focus on the concept of system *usability*. Usability is a quality attribute that assesses how easy a user interface is to use. The term *usability* also refers to methods for improving ease-of-use throughout the entire design process.

Bennett (1979) was the first to use the term usability to describe the effectiveness of human performance. In the following years a more formal definition was proposed by Shackel (1981) and modified by Bennett (1984). Shackel (1991) simply defined usability as "the capability to be used by humans easily and effectively, where,

easily = to a specified level of subjective assessment,

effectively = to a specified level of human performance."

In recent years more specific descriptions have been presented. Nielsen (2003) suggests usability possess these five quality components:

- **Learnability**: How easy is it for users to accomplish basic tasks the first time they encounter the design?
- **Efficiency**: Once users have learned the design, how quickly can they perform tasks?
- **Memorability**: When users return to the design after a period of not using it, how easily can they reestablish proficiency?
- **Errors**: How many errors do users make, how severe are these errors, and how easily can they recover from the errors?
- **Satisfaction**: How pleasant is it to use the design?

The following dimensions of usability have been described by Quesenbery (2003).

- **Effective.** The completeness and accuracy with which users achieve their goals.
- **Efficient.** The speed (with accuracy) with which users can complete their tasks.
- **Engaging.** The degree to which the tone and style of the interface makes the product pleasing or satisfying to use.
- **Error tolerant.** How well the design prevents errors and helps with recovery from those that do occur.
- **Easy to learn.** How well the product supports both initial orientation and deepening understanding of its capabilities.

Usability is one of an interface's most important qualities. For systems or products whose use is discretionary, such as Web sites, a difficult to use interface can cause people to stop using it. For business applications, whose use is usually mandatory, the result is lowered worker productivity. Usability, however, cannot be looked at independently of another system quality, *utility*. Utility refers to a system or product's functionality. Does it do what people want it to do? An entity may have a high level of usability but does not accomplish anything of value for its user. Conversely, an entity may be capable of performing many valuable functions for its user but, because it is

not easy to use, the functions cannot be accomplished. Usability and utility are equally important in design.

MYTH **Usability is nothing but common sense.**

Usability also has a relationship with flexibility in design. In general, as the flexibility of a design increases, its usability decreases. Flexible designs can perform more functions than specialized designs, but they perform them less efficiently. The flexibility-usability trade-off exists because accommodating flexibility entails satisfying a greater number of design requirements. This leads to more design compromises and more complexity in the design. Flexibility generally pays most dividends when users cannot clearly anticipate their future needs. Then, flexible designs that enable people to address future contingencies are usually needed. That flexibility will usually result in a reduction in usability, however, and should always be considered.

Usability Assessment in the Design Process

Usability assessment should begin in the early stages of the product development cycle and should be continually applied throughout the process. The assessment should include the user's entire experience, and all the product's important components. Usability assessment methods are discussed in more detail in Step 14, "Test, Test, and Retest."

Common Usability Problems

Mandel (1994) lists the 10 most common usability problems in graphical systems as reported by IBM usability specialists. They are:

1. Ambiguous menus and icons.
2. Languages that permit only single-direction movement through a system.
3. Input and direct manipulation limits.
4. Highlighting and selection limitations.
5. Unclear step sequences.
6. More steps to manage the interface than to perform tasks.
7. Complex linkage between and within applications.
8. Inadequate feedback and confirmation.
9. Lack of system anticipation and intelligence.
10. Inadequate error messages, help, tutorials, and documentation.

The Web, with its dynamic capabilities and explosive entrance into our lives, has unleashed what seems like more than its own share of usability problems. Many are similar to those outlined previously. One study (Ceaparu et al., 2004) found users spend almost 40% of their computer time trying to get things to work or work better. Difficult installations, viruses, and connectivity troubleshooting challenge people. The

systems that slow them down the most are operating systems, e-mail, and Web browsing problems.

Web usability characteristics particularly wasteful of people's time, and often quite irritating, are:

Visual clutter. A lack of "white space," meaningless graphics, and unnecessary and wasteful decoration often turn pages into jungles of visual noise. Meaningful content lies hidden within the unending forest of vines and trees, forcing the user to waste countless minutes searching for what is relevant. Useless displayed elements are actually a form of visual noise.

Impaired information readability. Page readability is diminished by poor developer choices in typefaces, colors, and graphics. Use of innumerable typefaces and kaleidoscopic colors wrestle meaning from the screen. A person's attention is directed toward trying to understand why the differences exist, instead of being focused toward identifying and understanding the page's content. Backgrounds that are brightly colored or contain pictures or patterns greatly diminish the legibility of the overwritten text.

Incomprehensible components. Some design elements give the user no clue as to their function, leaving their purpose not at all obvious. Some icons and graphics, for example, are shrouded in mystery, containing no text to explain what they do. Some buttons don't look at all like command buttons, forcing the user to "minesweep" the screen with a mouse to locate the objects that can be used to do something. Command buttons or areas that give no visual indication that they are clickable often won't be clicked. Language is also often confusing, with the developer's terminology being used, not that of the user.

Annoying distractions. Elements constantly in motion, scrolling marquees or text, blinking text, or looping continually running animations compete with meaningful content for the user's eye's and attention—and destroy a page's readability. Automatically presented music or other sounds interrupt one's concentration, as do nonrequested pop-up widows, which must be removed, wasting more of the user's time. A person's senses are under constant attack, and the benefits afforded by one's peripheral vision are negated.

Confusing navigation. A site's structure often resembles a maze of twisting pages into which the user wanders and is quite soon lost. Poor, little, or no organization exists among pages. The size and depth of many Web sites can eventually lead to a "lost in space" feeling as perceived site structure evaporates as one navigates. Embarking on a side trip can lead to a radical change in context or a path with no signposts or landmarks. Navigation links lead to dead-ends from which there is no return, or boomerang you right back to the spot where you are standing without you being aware of it. Some navigation elements are invisible. (See mystery icons and minesweeping above.) Confusing navigation violates expectations and results in disturbing unexpected behavior.

Inefficient navigation. A person must transverse content-free pages to find what is meaningful. One whole screen is used to point to another. Large graphics waste screen space and add to the page count. The path through the navigation maze is

often long and tedious. Reams of useless data must be sifted through before a need can be fulfilled. Massive use of short pages with little content often creates the feeling that one is "link drunk."

Inefficient operations. Time is wasted doing many things. Page download times can be excessive. Pages that contain, for example, large graphics and maps, large chunky headings, or many colors, take longer to download than text. Excessive information fragmentation can require navigation of long chains of links to reach relevant material, also accelerating user disorientation.

Excessive or inefficient page scrolling. Long pages requiring scrolling frequently lead to the user's losing context as related information's spatial proximity increases and some information entirely disappears from view and, therefore, from memory. Out of sight is often out of mind. If navigation elements and important content are hidden below the page top, they may be missed entirely. To have to scroll to do something important or complete a task can be very annoying; especially if the scrolling is caused by what the user considers is an irrelevancy or noise.

Information overload. Poorly organized or large amounts of information tax one's memory and can be overwhelming. Heavy mental loads can result from making decisions concerning which links to follow and which to abandon, given the large number of choices available. Or from trying to determine what information is important, and what is not. Or from trying to maintain one's place in a huge forest of information trees. One easily becomes buried in decisions and information. Requiring even minimal amounts of learning to use a Web site adds to the mental load.

Design inconsistency. Design inconsistency has not disappeared with the Web. It has been magnified. The business system user may visit a handful of systems in one day, the Web user may visit dozens, or many more. It is expected that site differences will and must exist because each Web site owner strives for its own identity. For the user's sake, however, some consistency must exist to permit a seamless flow between sites. Consistency is needed in, for example, navigation element location on a page and the look of navigation buttons (raised). The industry is diligently working on this topic and some "common practices" are already in place. The learning principle of rote memorization, however, is still being required within many sites. For example, the industry practice of using different standard link colors for unvisited sites (blue) and visited sites (purple) is often violated. This forces users to remember different color meanings in different places, and this also causes confusion between links and underlined text. Design guidelines for graphical user interfaces have been available for many years. Too often they are ignored (or the designer is unaware of them). Examples of inappropriate uses abound in design. The use of check boxes instead of radio buttons for mutually exclusive options, for example. Or the use of drop-down list boxes instead of combination boxes when the task mostly requires keyboard form fill-in. The Web is a form of the graphical user interface, and GUI guidelines should be followed.

MAXIM **The usability of a system is improved when similar parts are expressed in similar ways.**

Outdated or undated information. One important value of a Web site is its "currentness." Outdated or undated information destroys a site's credibility in the minds of many users, and therefore its usefulness. A useless site is not very usable.

Stale design caused by emulation of printed documents and past systems. The Web is a new medium with expanded user interaction and information display possibilities. While much of what we have learned in the print world and past information systems interface design can be ported to the Web, all of what we know should not be blindly moved from one to the other. Web sites should be rethought and redesigned using the most appropriate and robust design techniques available.

Some of these usability problems are a result of the Web's "growing pains." For other problems developers themselves can only be blamed, for they too often have created a product to please themselves and "look cool," not to please their users. Symptoms of this approach include overuse of bleeding edge technology, a focus on sparkle, and jumping to implement the latest Internet technique or buzzword. These problems, of course, did not start with the Web. They have existed since designers began building user interfaces.

Some Practical Measures of Usability

Usability, or the lack thereof, can often be sensed by a simple observation of, or talking to, people using an interface. While these measures lack scientific rigor, they do provide an indication that there may be usability problems.

Are people asking a lot of questions or often reaching for a manual? Many questions or frequent glances at manuals are signs that things are not as clear and intuitive as they should be. When in doubt, the first reaction of many people is to ask someone for assistance. When no one is around, then we look in a manual.

Are frequent exasperation responses heard? "Oh damn!" or similar reactions are usually used to express annoyance or frustration. Their frequency, and loudness, may foretell a strong rejection of a product. The absence of exasperation, however, may not represent acceptance. Some people are not as expressive in their language, or are better able to smother their feelings.

Are there many irrelevant actions being performed? Are people doing things the hard way? Are there incidental actions required for, but not directly related to, doing a job? These include excessive mouse clicks or keyboard strokes to accomplish something, or going through many operations to find the right page in a manual or the right window or page in the display.

Are there many things to ignore? Are there many elements on the screen that the user must disregard? Are there many "doesn't pertain to me" items? If so,

remember, they still consume a portion of a person's visual or information-processing capacities, detracting from the capacities a person could devote to relevant things.

Do a number of people want to use the product? None of us goes out of our way to make our own lives more difficult. (Unfortunately, other people may, however.) We tend to gravitate to things easy to work with or do. If a lot of people want to use it, it probably has a higher usability score. Attitudes may be a very powerful factor in a system's or Web site's acceptance.

Some Objective Measures of Usability

Tyldesley (1988) and Shackel (1991) have both presented possible objective criteria for measuring usability. Tyldesley's criteria are shown in Table II.1 Shackel's criteria are as follows:

How *effective* is the interface? Can the required range of tasks be accomplished:

- At better than some required level of performance (for example, in terms of speed and errors)?
- By some required percentage of the specified target range of users?
- Within some required proportion of the range of usage environments?

How *learnable* is the interface? Can the interface be learned:

- Within some specified time from commissioning and start of user training?
- Based on some specified amount of training and user support?
- Within some specified relearning time each time for intermittent users?

How *flexible* is the interface? Is it flexible enough to:

- Allow some specified percentage variation in tasks and/or environments beyond those first specified?

What are the *attitudes* of the users? Are they:

- Within acceptable levels of human cost in terms of tiredness, discomfort, frustration, and personal effort?
- Such that satisfaction causes continued and enhanced usage of the system?

The selection of the most appropriate measurements will be dependent upon the type of system and/or application being tested. For Web sites some terminology may have to be modified to reflect page elements (e. g. Links, not commands).

Human performance goals in system use, like any other design goal, should be stated in quantitative and measurable ways. Without performance goals it will not be known if they are achieved, or how successful the system really is. Clear and concrete goals also provide objectives for usability testing and ensure that a faulty or unsatisfactory product will not be released.

Values for the various criteria should be specified in absolute terms. An absolute goal might be "Task A must be performed by a first-time user in 12 minutes with no

errors with 30 minutes of training and without referring to a manual." Goals may also be set in relative terms. For example, "Task B must be performed 50 percent faster than it was using the previous system."

The level of established goals will depend on the capabilities of the user, the capabilities of the system, and the objectives of the system. In addition to providing commitments to a certain level of quality, goals become the foundation for the system test plan.

Table II.1 Possible Usability Measurement Criteria

1. Time to complete a task.
2. Percentage of task completed.
3. Percentage of task completed per unit time (speed metric).
4. Ratio of successes to failures.
5. Time spent on errors.
6. Percentage of number of errors.
7. Percentage of number of competitors that do this better than the current product.
8. Number of commands used.
9. Frequency of help or documentation use.
10. Time spent using help or documentation.
11. Percentage of favorable to unfavorable user commands.
12. Number of repetitions of failed commands.
13. Number of runs of success and of failures.
14. Number of times the interface misleads the user.
15. Number of good and bad features recalled by users.
16. Number of available commands not invoked.
17. Number of regressive behaviors.
18. Number of users preferring your system.
19. Number of times users need to work around a problem.
20. Number of times the user is distracted from a work task.
21. Number of times the user loses control of a system.
22. Number of times the user expresses frustration or satisfaction.

From Tyldesley (1988).