

1

Introducing User Interface Design

1

Introduction

1 Why the User Interface Matters

Human-computer interaction (HCI) is the study of how humans interact with computer systems. Many disciplines contribute to HCI, including computer science, psychology, ergonomics, engineering, and graphic design. HCI is a broad term that covers all aspects of the way in which people interact with computers. In their daily lives, people are coming into contact with an increasing number of computer-based technologies. Some of these computer systems, such as personal computers, we use directly. We come into contact with other systems less directly — for example, we have all seen cashiers use laser scanners and digital cash registers when we shop. And, as we are all too aware, some systems are easier to use than others.

When users interact with a computer system, they do so via a **user interface (UI)**. This book explores how to design good user interfaces — interfaces that are easy to use and easy to understand, that meet the needs of the intended users, and that support users in the tasks they wish to undertake. In this part of the book, we introduce you to user interface design and evaluation. In particular, we explain why good user interface design is important and highlight the consequences of poor or bad user interface design. More important, we will get you to start thinking about users — and why and how to involve them in the design and evaluation of the user interface.

2 Computers Are Ubiquitous

Technology has advanced so much that computer systems are used on an everyday basis by almost everyone. A **computer system** (or an **interactive computer system** or just a **system**) is the combination of hardware and software components that receive input from, and communicate output to, a user in order to support his or her performance of a task (see Figure 1.1). Computer systems may be used directly, as in the case of personal computers (PCs) in use at work or at home. Often, though, we use **embedded computer systems** where the technology is invisible to us. For

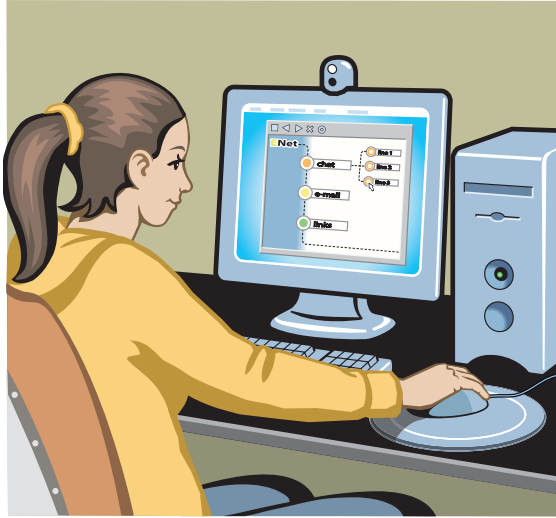


Figure 1.1 The interface is the part of the computer system with which the user interacts in order to use the system and achieve his or her goal.

example, computer-based microchip technology can be found embedded in personal goods such as digital watches and mobile phones, in domestic appliances such as microwave ovens, washing machines, and video recorders, and in the instrument panels of cars. Again, but less directly, computers are used when we shop; many stores use laser scanners that “swipe” the bar codes on goods to record both the goods we purchase and total the amounts we spend. Behind the scenes, the scanning of goods assists with automated stock control

and stock reordering. When we take money from our bank accounts using an automated teller machine (ATM) or when we use ATM debit cards to buy goods electronically, our bank details are accessed via the bank’s computer system. The list of everyday ways in which we use computer-based systems seems endless.

Whether we are aware of it or not, computers pervade our life. Computer applications are used either by us, or for us, in some way almost every day. The user interface (or just **interface**) is that part of the computer system with which a user interacts in order to undertake his or her tasks and achieve his or her goals.

The user interface and the ways of interacting with computer-based systems are different for each system. For example, digital watches generally have buttons that users press to set the time or use the stopwatch facility. Microwave ovens might have dials to turn or a digital display and a touchpad of buttons to set the cooking time. PCs have a screen, a keyboard, and a mouse (or sometimes a trackball or a joystick) that enable **interaction** to take place. So each user interface is different. Depending on the design of the interface, each of these systems will either be usable — that is, easy to learn and easy to use — or problematic for users.

Earlier we described a computer system as the combination of hardware and software components that receive input from, and communicate output to, a user to support his or her performance of a task. Although the user interface is simply the part of the computer system that enables interaction and serves as a bridge between users and the system, to users the interface often *is* the system (Constantine and Lockwood, 1999). The user’s view of a computer system is often limited to and based solely on his or her experience of the user interface (see Figure 1.2).

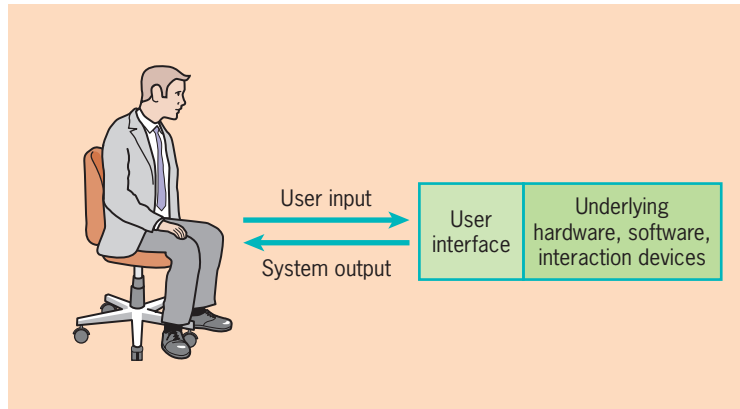


Figure 1.2 To the user, the interface is the computer system. (From Constantine and Lockwood, 1999.)

The design of controls, and the selection of interaction devices for input and output, will be discussed further in Chapters 12 through 14.

For example, when you use the controls on the panel of a washing machine, the controls form the interface between you and the machine — you are not concerned with the underlying technology or the software of the washing machine itself. What is important to you is that the controls and their settings are intuitive and easy to understand and use so that you will achieve your goal of laundering clothes. Similarly, when you surf the Internet, the pages of a web site displayed on your PC's monitor form the interface between you and the site. The web page UI may contain controls like scroll bars, clickable hot spots, or links in the form of text or images. These items are all part of the interface.

3

The Importance of Good User Interface Design

You will learn more about command-line interfaces and other interaction styles in Chapter 11.

Good **user interface design** is important because, as we have discussed, computer use permeates everyday life. Early computer systems were expensive and were developed mainly for particular tasks, like advanced number-crunching; as such, these systems were employed only by specialist computer users. Often the systems had command-line interfaces, with obscure commands known only by these specialist users. Thus, the user had to adapt to the system, and learning how to use the system required much effort.

Computing systems, however, are no longer the province of the specialist user. As the price of PCs and computer-based technologies has fallen, the ownership of these types of goods by nonspecialists has widened. In August 2000, 51% of households in the United States had access to one or more home computers, and 42% of households had access to the Internet (U.S. Census Bureau, 2001). In 2002, 54% of households in the United Kingdom had access to some form of home computer, and 44% had access to the Internet (National Statistics, 2004). Therefore, the need for the design and development of user interfaces that support the tasks people want to do and that can be used easily by a variety of people with varying abilities has become

an important issue. Users are more comfortable with computer systems that are easy to use, easy to understand, and enable them to attain their goals with minimum frustration.

One way of demonstrating the importance of good user interface design is by showing tangible benefits that can be discussed in cash terms. For businesses, good user interfaces can lead to benefits such as higher staff productivity, lower staff turnover, higher staff morale, and higher job satisfaction. Economically, these benefits should translate into lower operating costs. In addition, computer systems that are easy to use and easy to understand require less training, again saving employers money. Bad user interfaces, on the other hand, may result in stress and unhappiness among staff, leading to high staff turnover, reduced productivity, and, consequently, financial losses for the business. As you will see later, it is easy to give examples of the effects of bad design, but showing the financial benefits of good user interface design can be more difficult. Invariably, many factors are involved and this makes it difficult to attribute success directly to good user interface design.

3.1 What Is a Good User Interface Design?

A good user interface design encourages an easy, natural, and engaging interaction between a user and a system, and it allows users to carry out their required tasks. With a good user interface, the user can forget that he or she is using a computer and get on with what he or she wants to do. Just as knowledge of the transmission mechanism of a car is of little concern to most motorists, knowledge of the internal workings of a computer system should be of little consequence to its users.

Although we have used the adjectives “good,” “poor,” and “bad” to describe user interfaces, it is worth noting that each of these terms is subjective: they have different meanings for different people and their use to rate various aspects of a user interface will vary. You may have used the terms “good” or “bad” to describe, for example, the colors used in an interface, the pictures on the icons, or how attractive or eye-catching the interface was. These attributes describe the overall look or aesthetics of the UI. Nevertheless, they are only a part of our focus in this book. Our real concern is whether a user interface is good, bad, or poor in relation to its **usability**.

► What Is Usability?

Usability is defined in Part 11 of the **ISO 9241** standard (BSI, 1998) as “the extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use.” *Effectiveness* is the accuracy and completeness with which specified users can achieve specified goals in particular environments. *Efficiency* is defined as the resources expended in relation to the accuracy and completeness of the goals achieved. *Satisfaction* is the comfort and acceptability of the work system to its users and other people affected by its use.

Note two key aspects of this definition of usability. First, to be usable an interface should be perceived as being usable by the *specified* users — users for whom the system has been designed and developed. Next, the scope of focus for the design of

We discuss some alternative definitions of usability in Chapter 6.

You will learn more about domains, tasks, and environments in Chapter 4.

the interface is extended by looking beyond the users' immediate work environment and looking at the wider *context* or situation within which the system is expected to operate (i.e., the domain, tasks, and the environment that make up an organization). Thus, usability is concerned with the extent to which users of an application are able to work effectively, efficiently, and with satisfaction in their particular contexts.

A computer system that is usable in one context may be unusable in another. As a user interface designer, it is important to consider the context in which the system will be used. A UI that users find pleasurable is likely to be more acceptable than one that annoys them. Users are more likely to use a computer system that they enjoy than one that irritates them. Contented users are likely to be more productive, so usability is clearly related to user satisfaction (Constantine and Lockwood, 1999).

3.2 The Problems of Poor or Bad User Interfaces

► User Frustration and Dissatisfaction

Problems for users and the public in general arise as a result of poorly designed user interfaces. The term “computer rage” was coined in 1999 following a Market & Opinion Research International (MORI) poll conducted on behalf of Compaq Computer Limited, UK and Ireland. The study, *Rage against the Machine* (Compaq, 1999), (www.mori.com/polls/2002/bthomecomputing.shtml) found that, for one reason or another, stress and frustration levels with workplace technology are rising. Workers, it reports, have started to become both verbally and physically abusive toward the information technology (IT) in use (see Figure 1.3 and Box 1.1). Concerning monetary matters, the study indicates that

[t]he cost to business of this increase in stress levels of employees is not only based on sick days or under-performance, but also the working time lost through waiting for IT problems to be solved. Confederation of British Industry (CBI) statistics currently evaluate this at a staggering £25,000 (\$40,000) per person in lost business each year (based on one hour a day being spent sorting out IT problems). (p. 4)

In October 2002, research for British Telecom (BT) Home Computing, (www.mori.com/polls/2002/bthome-topline.shtml) again conducted by MORI, found that 70% of personal computer users suffered from “PC rage” — that is, the users surveyed admitted to shouting, swearing, or being violent to their computers when problems like crashing or virus infections arise.



Figure 1.3 Computer rage: Workers have started to become physically (and verbally) abusive toward IT.

Box 1.1**Man Shoots Laptop**

A 48-year-old man, George Doughty, was allegedly so frustrated by his laptop crashing that he took a handgun and shot it four times. According to police he apparently hung the destroyed laptop on a wall as if it were a “hunting trophy.” Lafayette police officer Rick Bashor told local newspapers, “It’s sort of funny, because everybody always threatens their computers, [but] it’s the first time someone shot a computer because he was upset with it.” The man admitted to police that he should not have shot his laptop, but that it seemed appropriate to at the time.

From <http://news.bbc.co.uk/1/hi/world/americas/2826587.stm>, reported March 6, 2003, downloaded June 1, 2004

Box 1.2**Survey Highlights Computer Rage**

One in five Scots suffers from “Internet rage” and some feel like hurling their computers through a window, according to a survey undertaken in February 2004. Around 1000 Scots were asked to tick a box with several options about their pet hates in everyday life for the survey this month. Some 45% of those polled blamed sluggish Internet connections for making their blood boil. This was more than twice the number of people (20%) who said watching their favourite soccer team get beaten drove them mad. . . . One in 10 surfers confessed they sometimes felt like punching their keyboard, whacking the monitor with a hammer and even throwing their PCs out the window. A third of people quizzed said they had to walk away to cool down. Additionally, one fifth of Scots feel that slow Internet connections at work make them lose up to an hour a day.

From Jude Sheerin, PA News, as reported at news.scotland.com, <http://news.scotsman.com/latest.cfm?id=2522311>, February 12, 2004.

Despite more than two decades of HCI research, it remains an unfortunate fact that many computer systems do not do what users want them to do. Users often describe their difficulties in system use as “computer problems,” which is nonspecific as to the source of the problems. There could be several explanations. For example, the problems could be related to buggy software or to the use of older, less efficient hardware or technology that slows the processing of information. Or maybe there was no clear understanding about the work environments in which the new computer systems were expected to operate. Box 1.3 looks at problems that occurred when the UK Passport Agency introduced a new computer system.

Equally, a poorly designed user interface could have contributed to the problems. While there is no direct evidence in any of the news reports to suggest that poor user

Box 1.3**Passport Agency Delays**

In May 1999 the UK Passport Agency hit the headlines as the waiting time for a passport applied for by post lengthened from a target time of two weeks to between seven and ten weeks. While this increase in waiting times was partly due to a larger than expected increase in passport applications (because of a new requirement for children to have their own passports), a second reason cited was “computer problems” caused by the introduction of new computer systems at two of the Passport Agency’s offices. Applicants anxious to ensure that they would have their passports in time to holiday abroad queued by the hundreds in Glasgow, Liverpool, and London.

*Drawn from the BBC News web site,
June 15, June 24, June 28, and June 29, 1999.*

interface design was to blame, it is likely that user interface problems contributed to the difficulties that users had with these systems.

EXERCISE 1.1 (Allow 10 minutes)

Think for a moment about the situation outlined in Box 1.3. Suggest what the consequences of the Passport Agency’s computer problems may have been for the following groups of people:

- The general public
- The workers at the Passport Agency

DISCUSSION

The consequences to the general public were enormous. People had great difficulty getting their passports in time for their vacations. People waiting for passports may have suffered from stress and anxiety related to the possibility that they may not be able to go away. Some may have lost money as a result of being unable to travel. Business travelers would have been affected too. Both groups probably felt anger and frustration with what they would have perceived as computers being in control and staffed by unorganized, incompetent government administrators. Furthermore, many people had to take time off work to line up for hours at passport offices. The passport agency workers, too, suffered consequences. They would have been under great stress to meet the demands of the public, and they would have felt anger and frustration because the system did not enable them to do their jobs of processing passport applications and issuing passports.

► Loss of Productivity, Efficiency, and Money

Computer systems with poor user interfaces can have a financial cost. Take the crisis at the Passport Agency. It was reported that the cost of this fiasco was \$20 million

(£12.6 million), which included nine million dollars (six million pounds) on staff overtime and at least \$242,000 (£161,000) in compensation to the hundreds of people who missed their vacations as a result of not receiving their passports on time. The Passport Agency also spent \$24,000 (£16,000) on umbrellas for people who had to wait outside of passport offices in the rain to get their passports over the counter. Subsequently the price of a passport was increased. The supplier of the computer system had agreed to pay \$3.7 million (£2.45 million) of the costs, leaving the remainder to be paid by the taxpayer. In these days of payment for productivity and efficiency, wages may have been lost if agency workers' earnings were linked to a level of productivity they were unable to meet because the computer system was unfit for its purpose.

3.3 Safety and the User Interface

So far we have considered the problems of poor user interfaces in terms of user frustration and dissatisfaction, and the loss of productivity and efficiency to business. There is another important aspect to consider: the issue of safety, both for computer systems users and the general public.

Systems in which human or environmental safety is of paramount concern are referred to as safety-critical systems. These systems include aircraft, aircraft flight decks, air traffic control consoles, nuclear power plants, control systems, and medical devices. The Three Mile Island nuclear power plant disaster (see Figure 1.4 and Box 1.4) illustrated that safety can be severely compromised by poor user interface design, with potentially serious consequences.

Box 1.4

The Three Mile Island Nuclear Power Plant Disaster

One of the most discussed issues during the early 1980s was the Three Mile Island nuclear power plant disaster. The incident nearly resulted in a meltdown of the nuclear reactor. The cause of the incident was never conclusively determined, but experts, official bodies, and the media all blamed a combination of operator error and bad interface design. In particular, much media attention and several official reports focused on the design of the control panels in the process plant. The incident could have been prevented if the control panels had been designed to provide the operators with the necessary information to enable them to perform their tasks efficiently and correctly. The following are just some of the interface problems that were identified:

- A light indicated that a valve had been closed when in fact it had not.
- The light indicator was obscured by a caution tag attached to another valve controller.



Figure 1.4 The Three Mile Island nuclear power plant.

- The control room alarm system provided audible and visual indication for more than 1500 alarm conditions. Evidently this number of alarms was intended to facilitate control of the entire plant during normal operating conditions. However, the layout and grouping of controls on the control panel had not been well thought out and so enhanced, rather than minimized, operator error (Brookes, 1982; cited in Leveson, 1995).
- A single “acknowledge” button silenced *all* the alarms at the same time, but it was not used because the operators knew they would lose information if they silenced some of the alarms. There was simply no way for the operators to cancel the less important signals so that they could attend to the important ones.

The root of the problem, therefore, seemed to be that the control panels did not support the task of serious error and incident recovery. The control panels misinformed the operators. They did not indicate to the operators the true state of affairs in the reactor plant, and they did not provide the necessary information in a form that the operators could understand and use to rectify the situation.

In November 2000, the topic of user interface design suddenly became international news when the outcome of the U.S. presidential election hung on the results of one county in Florida. In this election, an apparently minor aspect of equipment design turned out to have major consequences. Many voters in Palm Beach County felt that their vote went to the wrong person (see Box 1.5).

Voting Instructions

Step 1. Using Both Hands,
Insert the Ballot into the slot
and pull the chain down until
you feel the chain catch.

Step 2. Be sure the top center
of the slot is open and the
chain can't feel tight and stiff.

Step 3. To mark the ballot,
insert the ballot into the
slot and pull the chain down.
Do not use the chain to
pull the ballot out of the slot.

Step 4. When the ballot is
out of the slot, the chain
will be loose. Do not pull
the chain down. Do not pull
the chain up. Do not pull
the chain out of the slot.

**NEUTRALIZING MACHINES
TO BOTTLES**

LEGAL TIME LIMIT FOR VOTING BOOTH IS 5 MINUTES

Figure 1.5 The type of ballot booth used in Palm Beach County. © Steve Krug 2004, used with permission.

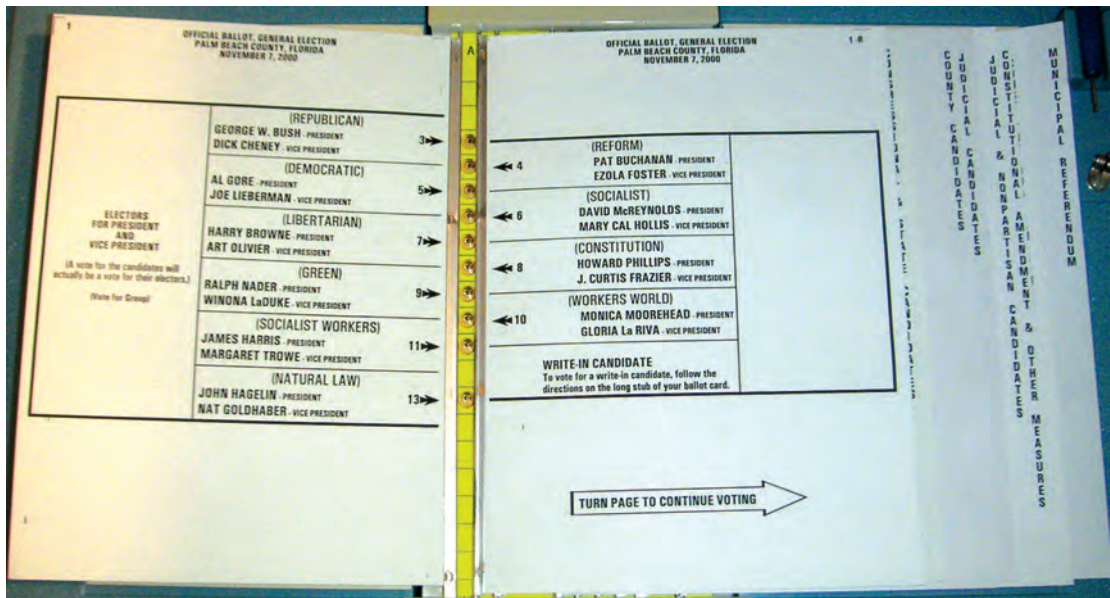


Figure 1.6 The problematic page of the ballot in the booth. © Steve Krug 2004, used with permission.

Box 1.5**A Palm Beach Voter Comments on the Disputed Ballot**

Tuesday at the polls, the ballot information was very confusing if one was voting for the Democrat. It was hard to know whom I was voting for, the way the ballot was printed. I did not know whether I was voting for my choice, Al Gore, or for Pat Buchanan.

That was very scary and upsetting. I had to take the ballot out a couple of times and place it back again to be sure that the arrows pointed to the right hole; even after the third try, I was not sure whom I was voting for, and that makes me very mad. Many other citizens have complained regarding this situation. I am sure this was extremely confusing for senior citizens especially.

Delia Pinto-Houbrick

*From the Palm Beach Post, letters to the editor, November 10, 2000,
www.palmbeachpost.com, visited July 8, 2003.*

for Gore (which required them to punch the third hole) but they mistakenly punched the second hole down because Gore was listed second on the left.

► Small Irritations Are Also a Problem

If you have found it difficult to relate to the “catastrophic” examples we have discussed so far, there are many less disastrous but still problematic examples that may be more familiar to you. Take, for instance, the process of shutting down your computer from Microsoft Windows. To do this, you have to press the Start button on the task bar, and find the command Shut Down on the **menu**. Intuitively, is that where you would expect the Shut Down command to be? What other domestic appliance, or any other type of device, is stopped by starting it? Although the Start button may not have been the obvious place to look for the Shut Down command when you first used Windows, once you have used Windows for some time you adapt to what it makes you do to shut down your computer and you just do it.

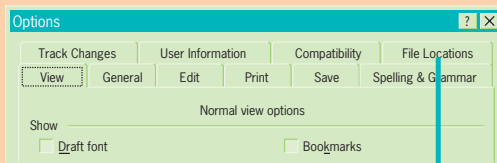
EXERCISE 1.2 (Allow five minutes)

Think about your use of the different software applications provided in the Microsoft Office Suite or in another suite of office applications that you use (e.g., StarOffice from Sun). Choose one application, and think about a particular feature that you find confusing when you use it.

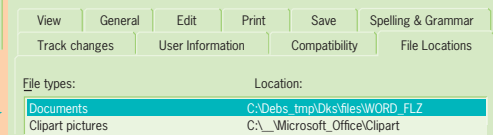
DISCUSSION

Debbie writes: The application I use most often from the Microsoft Office Suite is Word. For me, a confusing feature in more recent versions of Word is tabbed

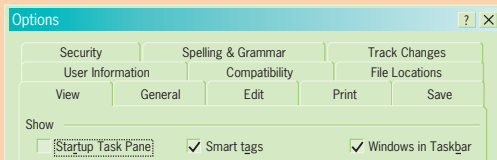
The “Options” dialogue box from the “Tools” menu in Word 97



Choosing the “File Locations” tab only makes the two rows of tabs swap positions. The ordering of the tabs from left to right does not change.



The “Options” dialogue box from the “Tools” menu in Word 2002



Choosing the “Spelling & Grammar” tab makes the three rows of tabs rotate their positions. When this is done in real time the tabs seem to scramble, and it is hard to see that the ordering of the tabs from left to right does not actually change.

Thus it is far easier to see and follow the movement of the tabs in Word ‘97 than in Word 2002, the more current version.

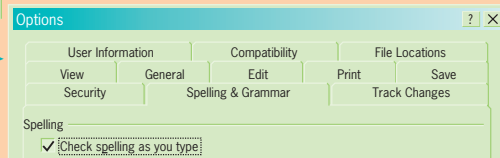


Figure 1.7 A comparison of tabbed dialog boxes for Word 97 / Word 2002.

dialogs. Specifying document settings is done via the Options tabbed dialog box, which is accessed from the Tools menu.

Before my work computer was upgraded, I had been using Word 97 for a number of years. In Word 97, there are only two rows of tabs for setting document options. As there are more tabs in one row than in the other, it is easier to see and understand how the tabs move when a tab is clicked on.

With greater word processing functionality, the Options dialog box became more complex in Word 2002; there are eleven tabs in the Options dialog box arranged in three rows of tabs. Clicking on any tab causes a puzzling rearrangement of the tabs. In fact, each row of tabs moves as a whole; only the row positions are changed rather than the positions of the tabs themselves, so there is some reason to it (see Figure 1.7).

4 Designing for Users

15
Part 1

With the more widespread use of computers, the knowledge, skills, and experience of computer users have become very broad. A good user interface caters to end users and supports them in the tasks they wish to undertake. A computer system that is developed without a good knowledge of the users and what they want to do with the system may be usable in that it can be used to do *something*, but it may not do what the users want to do in order to achieve their goals. The system will be *usable*, but not necessarily *useful*. This is not to say that all computer systems have to be designed to accommodate everyone. Computer systems should be designed for the needs and capabilities of the users for whom they are intended. Ultimately, a user should not have to think unnecessarily about the intricacies of how to use a computer unless, of course, that itself is the user's task.

4.1 User-Centered Design

User-centered design (UCD) is an approach to user interface design and development that involves users throughout the design and development process. User-centered design not only focuses on understanding the users of a computer system under development but also requires an understanding of the tasks that users will perform with the system and of the **environment** (organizational, social, and physical) in which they will use the system. Taking a user-centered design approach should optimize a computer system's usability.

Earlier we provided the ISO 9241:11 definition of usability. ISO 13407, *Human-Centered Design Processes for Interactive Systems* (ISO, 1997), provides guidance on and lists the main principles and essential activities for human (user)-centered design, for achieving usability in systems. Briefly, the four main principles of human-centered design are (ISO, 1997 p. 7):

1. The active involvement of users
2. An appropriate allocation of function between user and system
3. The iteration of design solutions
4. Multidisciplinary design teams

The four essential human-centered design activities are (ISO, 1997 p. 10):

1. Understand and specify the context of use
2. Specify the user and organizational requirements
3. Produce design solutions (prototypes)
4. Evaluate designs with users against requirements

Adopting the approach prescribed by ISO 13407 ensures that the users' perspectives form part of the HCI design and development process, which will positively influence the usability of the final product.

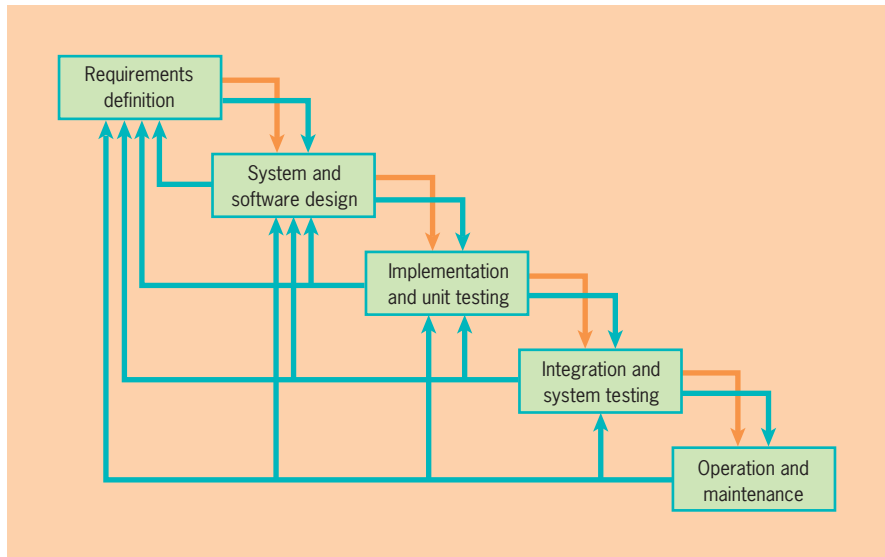


Figure 1.8 The classic life cycle. (From Sommerville, 1995.)

4.2 The Classic Life Cycle

User-centered design and traditional software engineering take very different approaches to computer system design. Traditionally, software developers have treated each phase of the software design life cycle as an independent part of software development, which must be completely satisfied before moving on to the next phase. This is particularly so in relation to the **classic life cycle** (also known as the waterfall model, so named because of the cascade from one phase to another; see Figure 1.8). It prescribes a predominantly sequential transition between the successive software life cycle phases, where each phase is completely satisfied before the next begins (this is represented in Figure 1.8 by the red arrows).

This view is, of course, simplistic. Software engineers readily accept that although the design is guided and regulated by this top-down somewhat linear model, in practice there are many iterations up and down between stages. Sommerville (1992), for example, has the following to say on the matter:

In practice, however, the development stages overlap and feed information to each other. During design, problems with requirements are identified; during coding, design problems are found; and so on. The software process is not a simple linear model but involves a sequence of iterations of the development activities. (p. 7)

Therefore, within the software design life cycle there is a need for the phases to feed information to each other, and for iteration, rather than the development proceeding from start to finish in a simple linear fashion. This iteration is represented in Figure 1.8 by the blue arrows. The essential difference between the classic life cycle

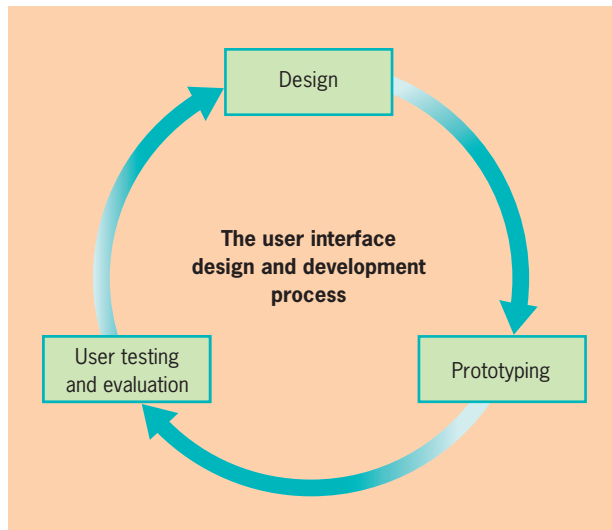


Figure 1.9 The iterative user interface design and evaluation process. (From Greenberg, 1996.)

and user-centered interface design is that user interface design and development is based on the premise that users should be involved throughout the design life cycle. Additionally, the process should be highly iterative, so that the design can be tested (or evaluated) with users to make sure it meets the users' requirements. Unlike this **iterative design** process, the waterfall life cycle generally leaves evaluation to the end. Let us look at these aspects further. Figure 1.9 illustrates the iterative user interface design and development process.

4.3 Involving Users

The way to be user-centered is to involve users and to pay attention to their views. This can include a variety of approaches, from simply observing users' working practices as part of collecting system requirements, to using psychologically based user-modeling techniques, to including user representatives on the design team. More important, users should be involved in the testing and evaluation of the system during its design and development. But who are the users?

► Who Are the Users?

In a user interface design project, developers generally refer to several types of people as users:

- Customers, who pay for and perhaps specify the computer system under development
- Other people within the users' organizations who have an interest in the development of the system
- Users or end users — the people who actually use the system directly to undertake tasks and achieve goals

You will learn more about constraints and trade-offs in Part 2.

You will learn more about persuading others to make changes in Chapters 28 and 29.

To avoid confusion, from this point on we will reserve the term “users” for the end users of a computer system.

The difficulty in having so many different people involved is that everyone works from his or her own perspective and has a particular agenda. For example, managers will want to ensure that their organization's effectiveness is not impaired by the introduction of a new system. They will want to be sure that the final solution is cost-effective, does not threaten efficiency, and is safe and reliable. Professional bodies or trade unions will want to ensure that working agreements are not broken and that workers' rights in relation to employment and IT-related health and safety issues are protected.

Additionally, there is the design team, which potentially includes, as well as users, design team managers, marketing personnel, software engineers and designers, programmers, graphic designers, systems analysts, HCI specialists, and other research and development staff. Again, each works from his or her own perspective. A software engineer, for example, may provide advice about optimal configurations of computer system architecture and the performance of the software. A graphic designer will focus on the visual appeal of the UI and may wish to establish whether graphics are appropriate to all application areas or whether there are any potential problems in selecting iconic representations for certain situations. Marketing people want to sell products on time and as cost-effectively as possible and want the right image for the application. Users need to feel confident that the computer system will offer the right facilities to ensure that their work is carried out at least as effectively as with previous methods or computer systems.

It can be difficult to reconcile all the views, and there will be constraints and trade-offs related to decisions made for the final design.

EXERCISE 1.3 (Allow 10 minutes)

Think about a recent project you have been involved with at work or an important meeting that you have attended (for example, a meeting at a club you belong to or a local community meeting). Various people were there, all expressing their own views. In the project or at the meeting, was it easy/necessary/appropriate to accommodate all the views expressed when arriving at a decision?

DISCUSSION

Many people have been involved in the development of this book. The following is a partial list:

- The academic members of the editorial team
- The authors and others involved in the Open University course that this book has been drawn from
- Academic and professional reviewers
- Our publisher and its editing staff

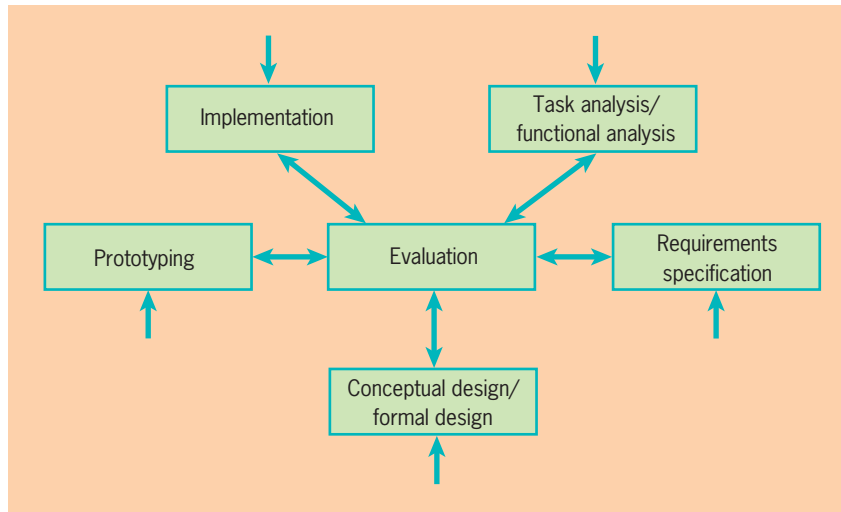


Figure 1.10 The star life cycle. (From Hix and Hartson, 1993.)

It has sometimes been challenging to work as a team as we are scattered across different time zones and some people have moved on to other work since they made their original contributions.

4.4 Making the Design Process Iterative

Making the user interface design and development process iterative is a way of ensuring that users can get involved in design and that different kinds of knowledge and expertise related to the computer system can be brought into play as needed. We shall use an approach adapted from a model proposed by Hix and Hartson (1993). It is known as the star life cycle for obvious reasons, as you can see from its appearance in Figure 1.10.

The star life cycle encourages iteration. First, you will notice that the central point of the star is **evaluation**, which is viewed as being relevant at all stages in the life cycle and not just at the end of product development as the classic life cycle tends to suggest. Evaluation is concerned with gathering data about the usability of a design or product by a specified group of users for a particular activity within a specified environment or work context. Not surprisingly, a host of different evaluation techniques are needed to support the different stages of design and to reflect the design needs of different kinds of products. These include **interviews** with users and others, observing users in their workplace, and getting users' opinions from questionnaires or other types of surveys. Second, the star life cycle is "intended to be equally supportive of both top-down and bottom-up development, plus inside-out and outside-in development" (Hix and Hartson, 1993, p. 101). Thus, a design can start with any process in the star life cycle.

You will learn more about evaluation in Part 4.

► When and How to Involve Users

Users should be involved in every part of the user interface design and development life cycle.

- *Early in the design process when the requirements are being specified.* Users could help in defining the requirements for the system by contributing a specification or by testing early mockups. Users can get involved by allowing themselves to be observed and giving feedback about the problems of the current system.
- *During prototyping, to test designs and options.* Users could test versions of the interface to provide feedback and make suggestions to the designers.
- *Just before delivery of the product.* Users again could test the product by using it or completing surveys about the various features. At this point, however, only minimal changes would be allowable.
- *During training/after delivery of the system.* Again, users would use the product and give their opinions and detail any problems. Revisions at this stage would be included in the next version.

EXERCISE 1.4 (Allow 10 minutes)

Think about the approach you take to software development at work. If this is not applicable to you, think about how you might approach a task like preparing a meal for friends. Think about where you would involve users and where you might work iteratively (that is, where you might repeat steps).

DISCUSSION

If I were to prepare a meal for friends, I would certainly involve them by first asking when they could come — there's no point in cooking a great meal and then asking them whether they are available! I would also consult them in advance over particular food likes and dislikes or perhaps what they would want on the menu. In cooking, there are dozens of things you do repeatedly (or iteratively): lots of tasting goes on, and we tinker with ingredients until the taste is how we want it. My grandmother owned a restaurant and my father was a chef, so when I make, for example, a sauce for spaghetti or lasagne, it's an all-day affair. I start with tomatoes and vegetable stock, I add some spices, and then let it simmer (stirring occasionally) before tasting it. Usually it will need more salt or more stock, and once this is added I let it simmer a while longer. Somewhere along the way I add vegetables and meat (if there are no vegetarians dining with us). The sauce is usually just right about five hours later, after lots of iterations to the cooking sauce! Essentially, my approach is no different to that in Figure 1.10 — I *design* the sauce from my recipe, I test *prototypes* of the sauce as it simmers, and then I *evaluate* how good (or not!) it is.

5

The Two Types of Knowledge Needed for UI Design

You will look at the information-gathering activities in Part 2 and the UI design knowledge in Part 3.

In addition to the need for knowing about the intended users of a system and for designing iteratively, there is other information you need to know about and take into account when designing a user interface. This information comes from two sources:

- Information-gathering activities and analyses that form part of the user interface design and development process
- User interface design knowledge — for example, design principles and design rules.

6

Evaluation

This book is text based. The limitation of it being a static text-based volume of work is that the knowledge we are trying to impart can only be presented sequentially: chapter by chapter, topic by topic, in a given order. However, while you are studying this book, it is important to keep in mind the star model and to remember that there is no one single-user interface design approach or life cycle: it can start with any of the processes detailed in the star life cycle, and it should be highly iterative. Iterative design provides the opportunity for evaluation and user testing, the reviewing of the results of user testing, and the consideration of how that feeds into the cycle. In addition, it allows you to consider what design alternatives would improve the user interface design.

Don't be intimidated by the thought of *doing* evaluation. For most of us, evaluation is something we actually already do, but without formally saying, "I'll just do an evaluation here." We undertake evaluation every day, in many ways. For example, we shop, we buy different foods, we eat them, and then we decide whether they are good enough to buy again. If you drive a car, then you may be thinking about things like how many miles per gallon you are getting from your car or how well it is running. And if your car is running badly, then you may be skeptical about how well the mechanic has done his job — you are evaluating the mechanic's skill in servicing your car in relation to how well the car is performing. When we buy consumer goods, we may go into a retail outlet and play with (or evaluate) several different manufacturer's models of the item we are interested in purchasing; we press buttons, turn knobs or dials, and generally try to decide which model seems to do best what we want to purchase the item for.

EXERCISE 1.5 (Allow 10 minutes)

Think about a recent purchase you have made. It might be a new PC, a new car, or a domestic appliance for the home; or perhaps you were involved in making the decision about buying some equipment at work. Think about how you made the decision. Did you buy the first item you saw, or did you look for specific features to judge alternatives by and to guide your choice?

DISCUSSION

There is no right or wrong answer here, as decisions in relation to purchases are often personal to the individual and specific to the item being purchased. For example, I recently bought a microwave oven. I was looking for an oven without a turntable. This feature was important to me, as my previous model had a rotating microwave antenna under the floor of the oven (rather than an internal turntable). I had become used to being able to put any size or shape of cookware into my now defunct 15-plus-year-old model without worrying about cookware getting wedged, as often happens in ovens with turntables. Although there were other features to consider, like cost, color, design style, power levels, cooking programs, and the type of controls (dials and knobs versus touchpad and digital display), these were less important to me personally. In the end, as I was unable to find a microwave oven without a turntable, I evaluated the dozen or so models available on the other features before deciding which model to purchase.

Although I didn't find the microwave oven I wanted, in the end I made an okay choice: the oven looks good, it is only one-third the size of my old oven (which freed up some worktop space), and it was reasonably priced. However, the biggest flaw in my decision making was that I did not (or rather could not) assess (evaluate) how usable it was. It is a limitation that you cannot test certain domestic appliances in showrooms before you buy them. Now I own a microwave where I need to have the manual in front of me to set a cooking program, as the sequence of button presses is complicated and there are no reminders anywhere on the interface to tell me what the steps are (i.e., the visibility is poor).

At one meeting we discussed how each of us decided which washing machine to buy. One person was clear that he would only buy from a particular manufacturer. Another person's concern was style, another person's was spin rpm. Not one person mentioned the interface, interface controls, or usability as a prime determiner in their choice. The point here is that choices are made for many reasons, which often have nothing to do with the actual interface. Nevertheless, in designing your interfaces you should ensure that you follow an iterative user-centered design process both to increase the usability of your interfaces and to make it less of a gamble for the users.

Consciously or unconsciously we “do evaluation” in everyday life. We gather information, compare alternatives, reflect on outcomes, and make choices based on the information gathered and how well our requirements could be met. Evaluation undertaken for UI design is no different in essence. What differs is when the evaluation is done, and how the information (or knowledge) gathered is used to inform the design and development process.

6.1 When and How Do You Evaluate?

Evaluation is a way of finding out whether your systems work, and it is an ongoing activity throughout the life cycle. You can look for as many problems as possible

(**diagnostic evaluation**) or you can try to measure the performance of the system (**measurement evaluation**). The role of the evaluation is always to inform the design and to improve it at all stages. Measurement evaluation frequently contributes to subsequent versions of a product. Different kinds of evaluation may be carried out at different design stages and for different reasons.

► Evaluation Early in the Life Cycle

Evaluation during the early design stages is undertaken to validate the users' requirements, predict the usability of the product or the usability of an aspect of the product, and to assess how well the interface meets the users' needs.

The earliest user evaluations may be best done using paper-based prototypes and mockups. These are very low cost but can yield a great amount of information and feedback about a design. Findings from early evaluations can be taken on board and fed back into the design process before the design is set.

► Evaluation Later in the Life Cycle

Evaluation later in the design cycle is also undertaken to assess how well the user interface meets the users' needs. It is carried out to check the usability of the nearly completed system and to ensure that the system meets the specified usability requirements. At this point, findings are unlikely to be fed into the UI design and development process, as generally by this point the system is more or less ready. These findings might be used for the next version or release of a system rather than for changing the near-finished product.

► How Do You Evaluate?

Choosing what to do will depend not only on the questions that you want to answer but also on logistical factors, such as the time available to do the evaluation, the availability of suitable expertise and equipment, access to users, and so on. Often the choice comes down to money: How much will it cost and how will we benefit from doing it?

There are many different techniques for evaluation. Here are just a few to get you started. The first two will be discussed further in Chapter 2, while the process of evaluation will be covered in depth in Part 4.

- *Observing the organization and how people work.* Several kinds of evaluation depend on some form of observation or monitoring of the way in which users interact with a product or **prototype**. The observation may take place informally in the field or in a laboratory as part of more formal usability testing.
- *Interviewing, talking, and asking questions.* As well as examining users' performance, it is important to find out what they think about using the technology. No matter how good users' performance scores are when using technology, if for some reason they do not like using it, then it will not be used. Surveys using questionnaires and interviews provide ways of collecting users' attitudes to the system.

- *Making predictions.* The aim of this kind of evaluation is to predict the types of problems that users will encounter without actually testing the system with them.

As a general rule, any kind of user testing is better than none. You will learn something valuable from even the most informal evaluations — watching a single user interacting with a UI under design will provide interesting and informative feedback about the design.

7 Summary

In this chapter, we introduced you to user interface design, discussed why it is important, and explored what can happen when UIs are badly designed. We then introduced you to user-centered design, emphasized the importance of user involvement throughout design and development, and stressed the need for an iterative approach to design and frequent evaluations of the design in progress. We then discussed when to evaluate user interface designs and provided a short introduction to evaluation techniques.