Chapter 14
Birth of Software Industry and Human Computer Interaction

Key Topics
Software industry
Software contractors industry
Corporate software products
Mass market for software
Software as a service
Text-based interface
Graphical user interface
Usability standards
Mouse
Microsoft Office

14.1 Introduction

The market participants in the early days of computing consisted of a small number of computer companies such as IBM, which was a giant corporation and smaller companies such as Burroughs, Sperry, NCR, and so on. IBM was the dominant player in the market, and the computer industry at that time was described as Snow white (i.e., IBM) and the seven dwarfs (i.e., the competitors to IBM in the market).

The software produced in the early days of computing was proprietary and developed by commercial vendors such as IBM and its competitors. Once a customer made a decision to purchase a particular computer from a computer company, it was dependent on that company providing it with proprietary software to meet its needs.

The hardware of the various computers of the different vendors were incompatible, and this meant that if a customer changed vendor, then there was a need to rewrite software for the new computer architecture.

The early computers were not user friendly and users needed to be skilled to operate them. Human–computer interaction is a branch of computer science that is concerned with the design, evaluation, and implementation of interactive computing systems for human use. It is focused on the interfaces between people and computers and has grown over the decades to include text-based interaction systems, graphical user interfaces, and voice user interfaces.

The development of home computers from the mid-1970s meant that everyone in the world was now a potential computer user, and it was clear that there was a need to

G. O'Regan, *A Brief History of Computing*,
https://doi.org/10.1007/978-3-030-66599-9_14

improve the usability of machines. Humans interact with computers in many ways, and so it is important to understand the interface between them to facilitate the interaction.

14.2 Birth of Software Industry

The vast majority of software produced in the early days of computing was proprietary and developed by commercial vendors such as IBM and UNIVAC. The computer companies provided a total solution to their clients including both hardware and software, and there was a very limited independent software sector that developed application software for specific clients. Computer companies were essentially in the hardware business, and so they bundled software with their machines, that is, the software was essentially given away free and provided with the mainframe computer. Another words, operating systems, system software, and application programs were provided with the mainframe computer, and software was not priced separately from the computer hardware.

IBM was the dominant player in the computer field in the 1960s. The company had approximately a 70% market share of the computer industry in the 1960s and 1970s, and this led it being regarded as a monopoly with excessive powers. The US Department of Justice (DOJ) launched a major antitrust case against IBM in 1969, as it viewed the company as a dominant monopoly. Its goal was to eliminate IBM's excessive power over the industry by breaking it into smaller business units that would compete against one another. The antitrust case continued for 13 years (with over 30 million pages of documents produced as part of the case) up to the introduction of the IBM personal computer in the early 1980s. The eventual ruling in 1982 was that the case was "without merit."

IBM decided in 1968 to unbundle many of its software programs, and it introduced a new pricing strategy for its software programs and support and training services. IBM had several motivations for unbundling its software and services including:

- Anticipation of upcoming DOJ antitrust case.
- The cost of software development and support (especially its experience with the development of the System 360).
- The growth of independent software vendors meant that IBM was no longer obliged to provide all software solutions to its customers.
- The introduction of the System 360 meant that vendors could now provide a software product to a whole family of architecturally compatible machines.

The IBM decision meant that the computer industry changed forever, with software changing from being a giveaway item to becoming a commercial product and industry in its own right. The IBM unbundling decision led in time to the software and services industry that we see today, and the quality of software and its usability became increasingly important. Chapter 16 discusses the important field of software engineering, which emerged in the late 1960s as a response to the crisis in software development, and Sect. 14.4 is concerned with human–computer interaction and software usability.

Today, the software and services industry is immense, and the largest companies in the industry are IBM, Microsoft, Oracle, and SAP. Cambell–Kelly [CK:04] proposed a division of the software industry into three sectors: software contractors (from the mid-1950s); producers of corporate software products (from the mid-1960s), and mass market of software products (from the late 1970s). We expand on this classification a little to include more recent developments.

14.2.1 Software Contractors Industry

The software contractors industry consists of the first programming companies, with the earliest contractors dating from the mid-1950s. Their role was to develop complete systems or application programs for their clients. Their early customers included the US military, government organizations, and large corporations, and their role was to provide the appropriate expertise in designing, developing, and testing of the software. The selection of a software contractor was generally based on its project management capability, as well as the scope and cost of the proposed solution.

The selection process aimed to identify the most capable contractor that proposed the most appropriate and cost-effective solution. The software was produced for one customer only, as a mass market for software did not exist at that time, and so the solution was tailored to meet the needs of the customer.

Systems Development Corporation (SDC) was founded in 1955 as the systems engineering group for the SAGE project at the RAND Corporation. SDC developed the software for the SAGE system (IBM did not do the software part of the project), and SDC later developed the timesharing system for ARPA's mainframe computer in the 1960s, and it also developed the JOVIAL programming language for real-time applications. RAND spun off SDC in 1957 as a not-for-profit organization that provided expertise in software development to the US military, and SDC later became a for-profit organization offering its services to all types of organizations from the late 1960s.

Computer Sciences Corporation (CSC) was founded in 1959, and it initially provided software development services to companies such as IBM and Honeywell and later with publicly funded organizations such as NASA. It became a global provider of information technology services with operations in the United States, Europe, and Asia. It merged with HP Enterprise (formerly EDS) in 2017 to form DXC Technology.

Today, there are many international consulting companies in the United States, Europe, and Asia, such as Infosys, Wipro, and Tata in India; Accenture and Cap Gemini in Europe; and Cognizant and IBM in the United States.

14.2.2 Corporate Software Products

The IBM System/360 (see Chap. 8) was a family of small to large computers, and it was a paradigm shift from the traditional "one size fits all" philosophy of the computer industry. The computers employed the same user instruction set, and there were 14 models in the family. There was strict compatibility within the family, and so a program written for one model would work on another model. This led to a market for software products for the System/360, and several software companies were formed to develop software products.

This includes Informatics General, which was founded by Walter F. Bauer in California in 1962. The company was initially involved in the software contracting business, but it later built its own software products. These include the Mark IV, which was a file management system/report generator for the IBM System/360. There were over 1000 installations of this product. Informatics also developed other software products.

Applied Data Research was founded in New Jersey in 1959, and it was initially in the software contracting business. It later developed several of its own software products, and the most widely used of these include Autoflow for automatic flowcharting; Meta COBOL which is a macro processor for COBOL, and Librarian for source code control management.

Advanced Computer Techniques (ACT) was founded by Charles Lecht in New York in 1962, and the company was active in creating compiler related tools in its early years. It developed compilers for Fortran and COBOL in the 1960s, and it developed a compiler for Pascal and Ada in the 1970s and 1980s. It became a public company in 1968, and it diversified into other areas such as education and training, service bureaus to handle data processing needs of clients, and packaged software business of compilers and related tools.

Today, there are many companies providing corporate software products. For example, SAP is a German software corporation that makes enterprise software to manage business operations and customer relations.

14.2.3 Personal Computer Software Industry

The invention of the microprocessor led to the development of home and personal computers, and led to a massive demand for software for home and personal use. This included software such as editors, compilers, spreadsheets, and games, and it led to a mass market for software. The number of units sold for a software product for mainframes and minicomputers was in the hundreds (or thousands), but in the brave new world of personal computing, the number of units sold was in the millions of units.

MITS developed the prototype of the Altair 8800 home computer in late 1974, and the released product came as a home computer kit version (which was assembled by the customer) or a more expensive fully assembled version. One of the earliest products for the home/personal computing market was Altair Basic, which was produced in 1975 by a small company called Microsoft. It was Microsoft's first product, and computer hobbyists began producing software to run on the Altair and the emerging home computers.

Early software programs for home computers were often provided in a book or magazine, and the user would type in the entire program to the computer. However, this was a slow process and could only deal with small programs. Further, if the user mistyped, then the program either did not work as intended or was inoperable. The cassette tape was another popular means of distributing early software programs for home computers.

The emergence of the IBM personal computer in the early 1980s fundamentally changed the computing field, and it rapidly took market share in the home/personal computer market. It led to a massive demand (in millions of units) for these computers, as well as a massive demand for application software to run on the machines. Floppy disks became available for distributing software for personal computers in the 1980s.

The demand for software led to the growth of several large software companies in the 1980s that were providing application software for the IBM PC. These include Lotus Software (later part of IBM) that developed the popular 1-2-3-spreadsheet program (spreadsheet calculations, database functionality and graphical charts). This was the dominant spreadsheet software in the 1980s, but it was eclipsed by Microsoft Excel in the 1990s. WordPerfect Corporation (now part of the Corel Corporation) was the dominant player in the word-processor market in the 1980s. It created the popular WordPerfect (WP) word processor in the late 1970s/early 1980s, which remained the dominant word processor until it was eclipsed by Microsoft Word in the 1990s.

Microsoft has dominated the personal computer software industry since the 1990s, and Microsoft Office is a suite of office applications for the Microsoft Windows operating system. It consists of well-known programs such as Microsoft Word, which is a word processor; Microsoft Excel, which is a spreadsheet program; Microsoft PowerPoint, which is used to create slideshows for presentations; Microsoft Access, which is a database management system for Windows; and Microsoft Outlook, which is a personal information manager. We discuss Microsoft Office in more detail in Sect. 14.3.

14.2.4 Software as a Service

The idea of software as a service (SaaS) is that the software may be hosted remotely on a server (or servers), and access provided to it over the Internet through a web browser. The functionality is provided at the remote server with client access provided through the web browser. The cost of hosting and management of the service is transferred to the service provider, with the initial set up costs for users significantly less than for traditional software.

The software is licensed to the user on a subscription basis. Occasionally, the software is free to use with funding for the service provided through advertisements, or there may be a free basic service provided with charges applied for the more advanced version.

14.2.5 Open-Source Software

Open-source development is an approach to software development in which the source code is published, and thousands of volunteer software developers from around the world participate in developing and improving the software. The idea is that the source code is not proprietary, and that it is freely available for software developers to use and modify as they wish. One useful benefit is that it may potentially speed up development time, thereby shortening time to market.

The roots of open source development are in the Free Software Foundation (FSF). This is a nonprofit organization founded by Richard Stallman [ORg:13-b] to promote the free software movement, and it has developed a legal framework for the free software movement.

The Linux operating system is a well-known open-source product, and other products include mySQL, Firefox, and Apache HTTP server. The quality of software produced by the open-source movement is good, and defects are generally identified and fixed faster than with proprietary software development.

14.2.5.1 Free Software Foundation

Richard Stallman (Fig. 14.1) is the prophet of the free software movement, and he launched the Free Software Foundation (FSF) in 1985. Stallman joined the Artificial Intelligence Laboratory at MIT as a programmer, and he later became a critic of restricted computer access at the lab. He believed that software users should have the freedom to share software with others and to be able to study and make changes to the software that they use. He left his position at MIT to launch the free software movement, and he explains his concept of free software as:

> Free software is a matter of liberty, not price. To understand the concept, you should think of free as in free speech, not as in free beer.

He launched the GNU project in 1984, which is a free software movement, and involves the participation of volunteer software programmers from around the world. He formed the Free Software Foundation (FSF) to promote the free software movement, and he is the nonsalaried president of the organization. FSF has developed a legal framework for the free software movement, which provides a legal means to protect the modification and distribution rights of free software. The

Fig. 14.1 Richard Stallman. (Creative Commons)

meaning of the term “free software” is defined in the GNU manifesto, and he lists four key freedoms essential to software development [Sta:02], and a program is termed “free” if it satisfies these properties. These are:

1. Freedom to run the program for any purpose
2. Freedom to access, study and to improve the code, and to modify it to suit your needs
3. Freedom to make copies of the program and to redistribute them to others
4. Freedom to distribute copies of the modified program so that others can benefit from your improvements

The GNU project uses software that is free for users to copy, edit, and distribute. It is free in the sense that users can change the software to fit individual needs. Stallman has written many essays on software freedom and is a key campaigner for the free software movement. The legal framework for the free software movement provides protection to the modification and distribution rights of free software. Stallman introduced the concept of “copyleft,” which is a form of *licensing of free software*. It makes a program or product free and requires that all modified or extended versions of the program are also free.

Stallman has argued against intellectual property such as patent law and copyright law. He has argued against patenting software ideas, stating that a patent is an absolute monopoly on the use of an idea. He states that while 20 years may not seem like a long period of time, that in the software field it is essentially a generation, due to the pace at which technology changes in the world. Further, *patents act a barrier to competition and lead to monopolies*. They make it difficult for new companies to enter a market place, due to the restrictions and costs associated with the licensing of patents. In recent times, we have seen large companies acquire others for their intellectual property (e.g., the Google acquisition of Motorola Mobility was due to the latter’s valuable collection of patents), and today there are major intellectual property wars in the corporate world.

Stallman argues that copyright law places Draconian restrictions on the public and takes away freedoms that they would otherwise have. They protect the businesses of the copyright owner, and he suggests that alternative approaches should be considered in the digital age.

14.2.6 App Stores

Applications for mobile phones and tablets are termed "apps," and sales of apps are made through an App store, which vets the app and may take a percentage of every app that is sold. Apple's App store is used for apps that run on Apple's iOS operating system for iPhones, and Google Play is a popular App store for Android phones (there are multiple App stores available for the Android platform).

The Apps may be created by companies, by organizations or by individuals, and some are free to the user while others are subject to a payment.

14.3 Microsoft Office Software

Microsoft Office is a suite of office applications for the Microsoft Windows operating system. It consists of well-known programs such as Microsoft Word, which is a word processor; Microsoft Excel, which is a spreadsheet program; Microsoft PowerPoint, which is used to create slideshows for presentations; Microsoft Access, which is a database management system for Windows; and Microsoft Outlook, which is a personal information manager.

Microsoft's first Office application was a spreadsheet program initially called Multiplan when it was released in 1982. It was developed as a competitor to VisiCalc (Apple's spreadsheet program), and it renamed to *Excel* when it was released on the Macintosh in 1985. Excel is a spreadsheet program consisting of a grid of cells in rows and columns that may be used for data manipulation and arithmetic operations. It includes functionality for statistical, engineering, and financial applications, and it can display lines, histograms, and charts. It also provides support for user-defined macros.

Microsoft Word is the leading word processor, and the first version of the program was released on the MS/DOS operating system in 1983. It was designed for use with a mouse, and it provides "*What you see is what you get*" functionality. The first version of Word for Windows was released in 1989, and Microsoft Word began to dominate the market from the early 1990s.

Microsoft PowerPoint is a popular presentation program, and it enables the user to create a presentation consisting of several slides. Each slide may contain text, graphics, audio, movies, and so on. PowerPoint has made it easier to create presentations. It was originally developed for the Macintosh computer in 1987, and it was released for Windows in 1990.

The first version of *Microsoft Access* was released in 1992, and this database management system enables users to create tables, queries, forms, and reports. It includes a graphical user interface that allows users to build queries without knowledge of the query language. *Microsoft Outlook* is a personal information manager, and it is used mainly as an email application, but it also includes a calendar, task manager, note taking, and web browsing.

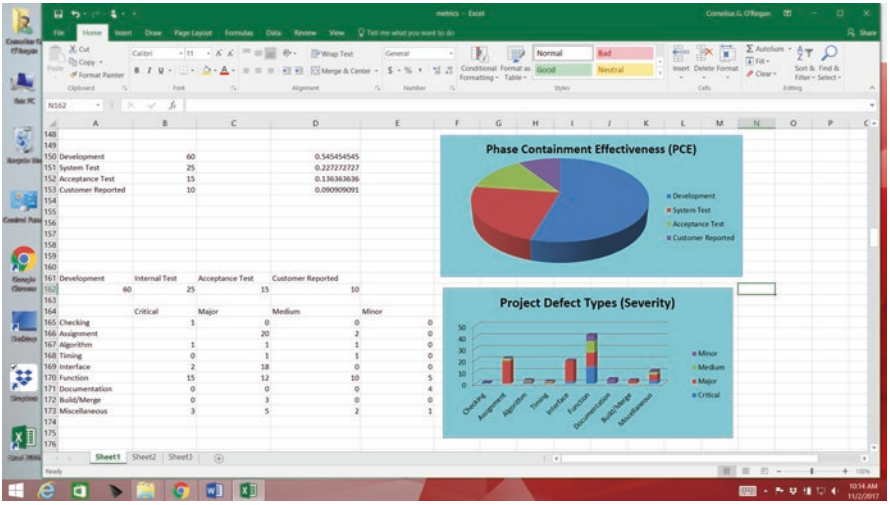
The various Microsoft application programs such as Word, Excel, and PowerPoint were all available individually, until they were bundled together into the Microsoft Office suite in 1989.

14.3.1 *Microsoft Excel*

Microsoft Excel is a spreadsheet program, and it consists of a grid of cells in rows and columns that may be used for data manipulation and arithmetic operations. It includes functionality for statistical, engineering, and financial applications, and it has graphical functionality to display lines, histograms, and charts (Fig. 14.2).

This spreadsheet program was initially called Multiplan when it was released in 1982, and it was Microsoft's first Office application. It was developed as a competitor to Apple's VisiCalc, and it was initially released on computers running the CP/M operating system. It was renamed to *Excel* when it was released on the Macintosh in 1985, and the first version of Excel for the IBM PC was released in 1987.

It provides support for user-defined macros, and it also allows the user to employ Visual Basic for Applications (VBA) to perform numeric computation and report the results back to the Excel spreadsheet. Lotus 1-2-3 was the leading Spreadsheet tool of the 1980s, but Excel overtook it from the early 1990s.



	A	B	C	D	E
148					
149					
150	Development	60		0.545454545	
151	System Test	25		0.227272727	
152	Acceptance Test	15		0.136363636	
153	Customer Reported	10		0.090909091	
154					
155					
156					
157					
158					
159					
160					
161	Development	Internal Test	Acceptance Test	Customer Reported	
162	60	25	15	10	
163					
164		Critical	Major	Medium	Minor
165	Checking	1	0	0	0
166	Assignment		20	2	0
167	Algorithm	1	1	1	0
168	Timing	0	1	1	0
169	Interface	2	18	0	0
170	Function	15	12	10	5
171	Documentation	0	0	0	4
172	Build/Merge	0	3	0	0
173	Miscellaneous	3	5	2	1
174					
175					
176					

Fig. 14.2 Microsoft Excel

14.3.2 Microsoft PowerPoint

Microsoft PowerPoint is a popular presentation program that allows the user to create a presentation consisting of several slides. Each slide may contain text, graphics, audio, movies, and so on, and PowerPoint has made it easier to create and deliver presentations. The user may customize slideshows and show the slides in a different order from the original order. It has advanced features for animating text and graphics, video editing, and even broadcasting the presentation.

Microsoft PowerPoint was initially called Presenter, and Forethought Inc. originally developed it in 1987 for the Macintosh computer. Microsoft acquired Forethought for $14 million in 1987, and the first Windows version of PowerPoint was released in 1990. PowerPoint has many features to enable professional presentations to be made (Fig. 14.3).

14.3.3 Microsoft Word

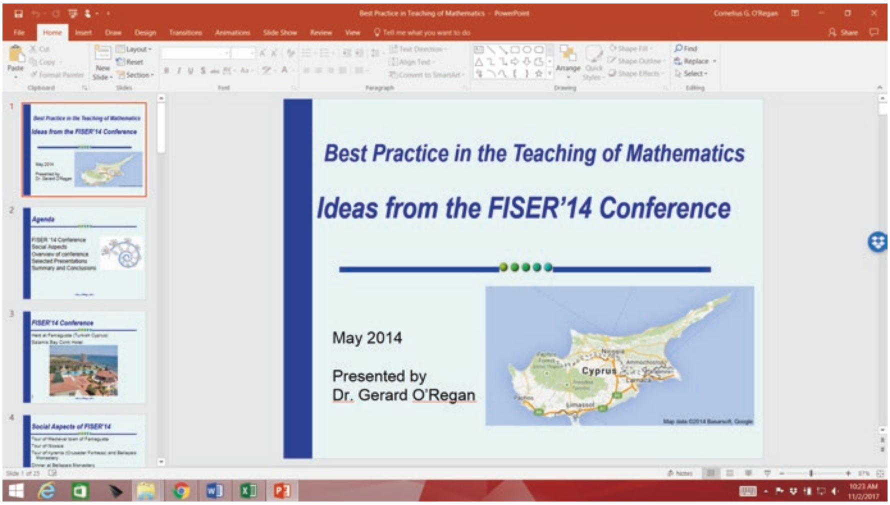
Microsoft Word is used for word processing tasks such as creating and editing documents. Charles Simonyi and Richard Brodie developed it for the MS/DOS operating system in the early 1980s. Simonyi and Brodie were former Xerox PARC employees who had worked on the Xerox Bravo WYSIWYG GUI word processor (the first such word processor), and they joined Microsoft in 1981. The first version of Microsoft Word was released in 1983.

Wordstar and WordPerfect were the leading word processors at the time, and it took some time for Microsoft Word to gain popularity. Word was designed for use with a mouse, and it provided “What you see is what you get” (WYSIWYG) functionality. Microsoft continued to improve the product, and it was ported to the MAC operating system in 1985. The first version for Windows was released in 1989, and Word began to dominate the word processing market shortly after the release of Windows 3.0 (Fig. 14.4).

14.3.4 Microsoft Access and Outlook

Microsoft Access is a database management system that allows users to create tables, queries, forms and reports, and connects them together with macros. It includes a graphical user interface that allows users to build queries without knowledge of the query language, or the user can create the query using the SQL database query language.

Microsoft Outlook is a powerful e-mail program and a personal information manager. It allows user to schedule meetings and to book meeting rooms and other resources, and the main Outlook sections include Mail, Calendar, Contacts, Tasks,



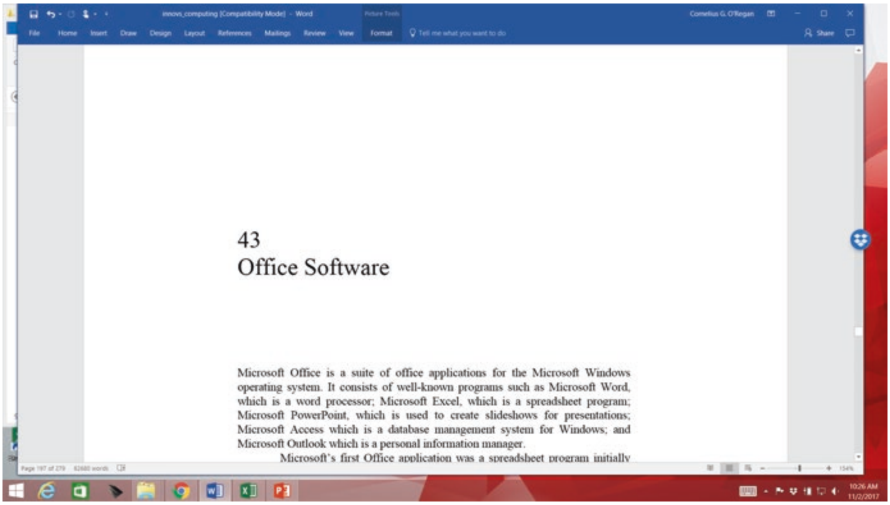
Best Practice in the Teaching of Mathematics

Ideas from the FISER'14 Conference

May 2014

Presented by
Dr. Gerard O'Regan

Fig. 14.3 Microsoft PowerPoint



43
Office Software

Microsoft Office is a suite of office applications for the Microsoft Windows operating system. It consists of well-known programs such as Microsoft Word, which is a word processor; Microsoft Excel, which is a spreadsheet program; Microsoft PowerPoint, which is used to create slideshows for presentations; Microsoft Access which is a database management system for Windows; and Microsoft Outlook which is a personal information manager.

Microsoft's first Office application was a spreadsheet program initially

Fig. 14.4 Microsoft Word

Notes, and Journal. Users may create and send e-mail messages, and manage their e-mails by creating e-mail rules; they may create e-mail auto-reply messages to automatically reply when they are out of the office; manage meetings, events, and appointments; maintain and manage contacts; and to define tasks that the user needs to perform (including their priority).

14.4 Human–Computer Interaction

The interaction between humans and machines was mainly limited to information technology professionals from the early days of computing up to the mid/late 1970s. This changed after the invention of the microprocessor in the early 1970s, which led to an explosion of interest from computer hobbyists, and the subsequent development of home computers from the mid-1970s. The introduction of the IBM personal computer in the early 1980s meant that everyone in the world was now was a potential computer user, and it led to a new market of personal applications and tools to support the user. However, it was clear that there were serious deficiencies with respect to the usability of computers in carrying out the tasks that users wished to perform.

Humans interact with computers in many ways, and so it is important to understand the interface between human and machines to facilitate an effective interaction. The early computer systems were *batch processing* (running programs in batches without human intervention) on a large expensive mainframe computer. The interaction between the human (operator) and computer was limited, and it consisted of placing the punched cards (encoded instructions to the computer) on the card reader, and the computer would then process the cards overnight. These computers were slow and expensive, and it was important that they be used efficiently 24 h a day. The computer could run only one program at a time, and programmers were unable to interact with the computer while it was running. This made it difficult and time consuming to identify and correct errors.

Licklider wrote an influential paper “Man-Computer Symbiosis” in 1960 [Lic:60], in which he outlined the need for a simple interaction system between users and computers. This paper mentioned ideas such as sharing computers among many users; interactive information processing and programming; large-scale storage and retrieval; and speech and handwriting recognition.

Doug Engelbart was one of the main developers of NLS (On Line System) in the late 1960s, and this online word processor system had revolutionary features such as the first computer mouse; time-sharing; and a command line interface. User trials and testing was employed in its development as part of a *philosophy toward a system adapting to people rather than people adapting to a system.*

Human–computer interaction (HCI) is a branch of computer science that is concerned with the design, evaluation, and implementation of interactive computing systems for human use. It is focused on the interfaces between people and computers and involves several different fields including computer science, cognitive psychology, design, and communication. The human–computer interaction field has evolved over the decades to text-based interaction systems, to graphical user interfaces (GUI), and voice user interfaces (VUI) for speech recognition and speech synthesis.

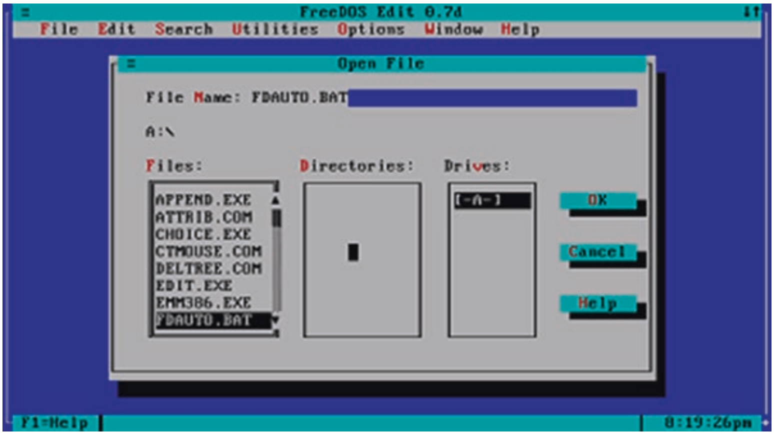
A *text-based interface* (also known as a command line interface) is where the system interaction (input and output) and navigation is text based. These interfaces are easier to use than punched card programming, but they require skilled users. This is due to the difficulty in remembering long lists of system commands.

One of the most well-known text-based operating system was Microsoft's MS/DOS operating system for IBM compatible personal computers, which was introduced in 1981 (Fig. 14.5). Text-based interfaces are effective for expert users but are more difficult for users with an average level of knowledge. They have a steep learning curve, as it is difficult to remember long system commands. The fact that they are not very intuitive or user-friendly motivated research into alternative approaches to human–computer interaction.

The *graphical user interface* (GUI) is an interface that uses graphical icons, menus, and windows to represent information and action to the user. It was a revolution in human and computer interaction, and the GUI was intuitive and user friendly. They have made computers and electronic devices attractive to nontechnical users, and the usability of the GUI has allowed a large range of users with varying ability and expertise to successfully interact with computers.

Early work on graphical user interfaces took place at Xerox PARC in the 1970s with their work on the Xerox Alto personal workstation (Fig. 11.1). This was the first computer to use a mouse-driven graphical user interface, and it was essentially a small minicomputer rather than a personal computer (it was not based on the microprocessor). Its significance is that it had a major impact on the user interface design, and especially on the design of the Apple Macintosh computer.

The Xerox Star was introduced in the early 1980s, and it followed sound usability principles (prototyping and analysis, iterative development, and testing with users) in its development. Steve Jobs visited Xerox PARC in late 1979, and he realized that the future of personal computing was with computers that employed a graphical user interface (such as in the Xerox Alto). Jobs was amazed that Xerox had not commercialized the technology, as he saw its graphical user interface as a revolution in computing and a potential goldmine in the future of computing. The



File Name: FDAUTO.BAT

A:\

APPEND.EXE
ATTRIB.COM
CHOICE.EXE
CTMOUSE.COM
DELTREE.COM
EDIT.EXE
EMM386.EXE
FDAUTO.BAT

[-A-]

Fig. 14.5 FreeDOS text editing

design of the Apple Macintosh was heavily influenced by the design of the Xerox Alto.

The Macintosh was a much easier machine to use than the existing IBM personal computer. Its friendly and intuitive graphical user interface was a revolutionary change from the command driven operating system of the IBM PC, which required the users to be familiar with its operating system commands. It was 1990 before Microsoft introduced its Windows 3.0 GUI-driven operating system (Fig. 14.6).

Today, the prevalent paradigm in human computer interaction is the WIMP (windows, icons, menus, and pointers) paradigm, which is comprised of a graphic and text interface navigated by a mouse and keyboard. The future of HCI is predicted to be the SILK (Speech, Image, Language, Knowledge) paradigm, where communication between humans and machine will be more natural and intuitive.

14.4.1 HCI Principles

The success of computer systems is critically influenced by the design of the human–computer interaction, and in the achievement of end-user computing satisfaction. Human–computer interaction is concerned with the study of humans and machines, and so it needs knowledge of both to be effective. The study of machines

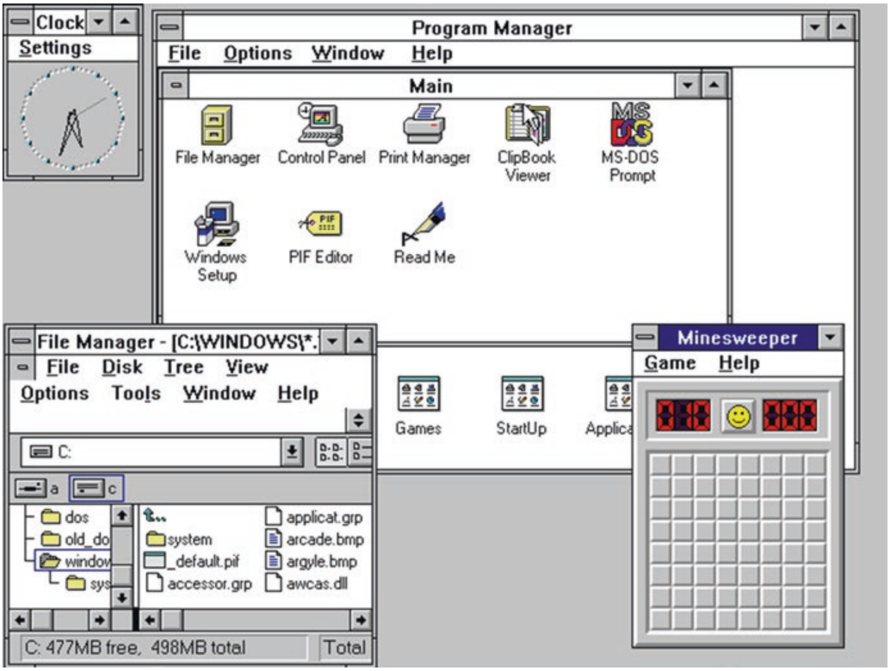


Fig. 14.6 Microsoft Windows 3.11 (1993). (Used with permission from Microsoft)

requires knowledge of computer graphics, programming languages, capabilities of current technology, and so on, whereas on the human side, it requires knowledge of cognitive psychology, ergonomics, and other human factors such as usability and end-user satisfaction. Table 14.1 summarizes Shneiderman's "Eight Golden Rules of Interface Design" [Shn:05]:

There are several fundamental principles and models underlying HCI. It is essential to understand the user and their characteristics, as well as their diversity in age, experience, physical and intellectual abilities, and so on. It is customary to distinguish between two types of user knowledge (IT and domain knowledge), and the user's proficiency in each type of knowledge yields several user categories that range between novice and expert.

– Interface knowledge (knowledge of the IT technology)
– Domain/task knowledge of the real-world system

The software will generally support multiple user categories, where novices get opportunities to learn about the system and have fewer opportunities for error. It is important to understand the domain in which the software will be used and to identify the tasks to be performed, as well as the frequency in which they will be performed.

Table 14.1 Eight golden rules of interface design

Principle	Description
Strive for consistency	Consistent terminology, sequences of action, and commands throughout the system
Enable frequent users to use shortcuts	The user will naturally desire to reduce the number of interactions as the frequency of use increases
Provide informative feedback	There should be appropriate system feedback
Design dialog to yield closure	Sequences of actions should be organized into groups with a beginning, middle and end
Offer simple error handling	Design the system (as far as possible) to prevent the user from making a serious error. The system should be able to detect an error and provide a handling mechanism
Permit easy reversal of actions	This is important to the user as it means that errors can be easily undone
Support internal locus of control	The system should be designed to make the users initiators of actions rather than responders to actions
Reduce short-term memory load	There are limitations to human processing in short-term memory, and so displays should be kept simple

14.4.2 Software Usability

Usability has become an important area in software engineering, and especially since the emergence of the World Wide Web in the early 1990s. The usability of the software is the perception that a user or group of users has of its quality and ease of use (i.e., is the software easy to use and easy to learn?), and its efficiency and effectiveness. Usability is a multidisciplinary field, and psychological testing may be employed to evaluate the perception that users have of the computer system. Usability is defined in the ISO 9241 standard as:

> **Usability** is the degree to which software can be used by specified consumers to achieve quantified objectives with effectiveness, efficiency, and satisfaction in a quantified context of use.

There are several standards for usability including the ISO 9241 and ISO 16982 standards, and the IEC 62366-1 standard (Applications of Usability Engineering to Medical Devices) from the International Electrotechnical Commission (IEC).

Usability, like quality, needs to be built into the software product rather than adding it later, and it needs to be considered from the earliest stages of the software development process. It requires an analysis of the user population and the tasks that they perform, as well as their knowledge and experience. The specification of the user and system requirements needs to include the usability requirements, as these are an integral part of the system.

There will often be a variety of different viewpoints to be considered, and this leads to multiple design solutions and an evaluation of these against the requirements. An iterative software development lifecycle is often employed, with active user involvement during the development process. Prototyping is employed to give the users a flavor of the proposed system and to get early user feedback on its usability. User acceptance testing (including usability testing) provides confidence that the software satisfies the usability, accessibility, and quality expectations of the users (Table 14.2).

Table 14.2 Software development lifecycle (including usability)

Phase	Description
Requirements	Interviews with the different categories of users
Prototype	Initial prototype developed and structured feedback given by users (usually via questionnaire)
Spiral design/ development	Design a little; code a little; test a little; formal review & user feedback prior to new spiral
Acceptance	Final acceptance testing by users

14.4.3 User-Centered Design

User-centered design (UCD) is a design process that is focused on the usability of and accessibility of the system to be developed, and it places the users at the center of the software development process. The users are actively involved from the beginning of the project, and regular feedback is obtained from them at each stage of the process. UCD follows well-established techniques for analysis and design, and it is focused on understanding the characteristics of users and their needs (Table 14.3).

The UCD design activities focus on the user, including understanding the tasks that they perform, their needs, and their experience. The users clarify what they want from the product, and the environment in which the software will be used. The designers then determine how the users are currently performing their tasks, and what they like and dislike about the ways in which the tasks are currently done. This helps the designer to design a product that will be fit for purpose that will satisfy the usability expectations of users, as well as being competitive in the market.

The software development team produces an initial version (or prototype) of the product, and the prototype has sufficient functionality to test some parts of the design. The design and development proceeds in cycles of modification, testing, and a user review of the current version, until the software satisfies functional, usability, and accessibility requirements. The approach is to design a little; code a little; test a little; evaluate and decide on whether to proceed with subsequent cycles.

A prerelease of the software may be created and sent to a restricted set of users for their evaluation, and the user feedback is then used to finalize the product prior to its actual release.

Table 14.3 UCD principles

Principle	Description
User understanding	The design is based on an explicit understanding of users, tasks, and environments (i.e., Who are the users? What are their tasks and needs? What is their experience?).
User involvement	The users are involved throughout the design and development (and user feedback shapes the design and development).
User evaluation	The design is driven and refined by user evaluation (and the user acceptance testing confirms that the usability and functional requirements are properly implemented).
Iterative development	The software development process is iterative, and the approach is to design and develop a little, get feedback from the user evaluation, modify accordingly and proceed to the next cycle in the iteration.
Design	The design addresses the whole user experience.
Multidisciplinary	The design team includes multidisciplinary skills.

14.5 The Mouse

The computer mouse was invented by Douglas Engelbart of the Augmentation Research Center (ARC) at the Stanford Research Institute (SRI) in the mid-1960s. It consisted of a wooden shell, a circuit board, and two metal wheels that came into contact with the surface that it was being used on (Fig. 14.7). Engelbart had been investigating ways for individuals to improve their capability in solving complex problems, and the mouse was part of ARC's oNline System (NLS).

Engelbart envisaged problem-solvers using computer-aided work stations using some sort of device to move a cursor around a screen. Engelbart and Bill English developed the first prototype of the mouse in 1964, and it worked on an early windows graphical user interface. They christened the device "mouse" as the early prototypes had a cord attached to the rear part of the device that looked like a tail and resembled an actual mouse.

The 1967 patent application described the mouse as an X-Y position indicator for a display system. It was publicly demonstrated at a famous computer conference in 1968, where Engelbart and a group of 17 other researchers of the ARC group gave a public demonstration of their NLS System.

The public demonstration took place at the Fall Joint Computer Conference held in San Francisco, and the mouse was just one of several innovations presented by Engelbart on that day. The goal of the NLS system was to act as an instrument to help humans operate within the domain of complex information structures.

The demonstration included the mouse, hypertext,[1] a precursor to today's graphical user interfaces; networked computers with shared screen collaboration involving two people at different sites communicating over a network with audio and video interface. The public demonstration introduced several fundamental computing concepts taken for granted today, and it later became known as "The Mother of all Demos."

The mouse operates on the principle that the computer determines the distance and speed that the mouse has traveled and converts that information into coordinates that it can plot on a display screen. The original mouse was used by Engelbart to navigate the NLS system, and this was also the first system to use hypertext.

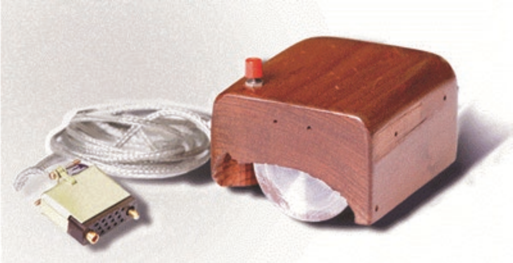


Fig. 14.7 SRI First Mouse

[1] Ted Nelson coined the term hypertext in the early 1960s.

Bill English moved to Xerox PARC in 1971, and the "Ball Mouse" was developed by English at PARC in 1972. It replaced the external wheels with a single ball that could rotate in any direction. The ball mouse became an important part of the graphical user interface of the Xerox Alto computer system, which was developed in Xerox and used at several universities. Xerox eventually commercialized a version of the Alto (the Xerox Star 8010) in 1981, and this was one of the earliest computers to be sold with a mouse.

The term "mouse" became an accepted term of the modern computer lexicon when it was introduced as a standard part of the Apple Macintosh in 1984 (Fig. 14.8). Steve Jobs had visited PARC to see the Xerox Alto, and he licensed the technology from Xerox. The Apple Lisa and Macintosh both used a graphical user interface and a mouse, and Microsoft made the MS/DOS Microsoft Word program mouse compatible, and the first Microsoft mouse for the PC appeared in 1983. The mouse became pervasive after the release of the Apple Macintosh and later Atari and Amiga personal computers in the mid-1980s, and the release of Microsoft Windows 3.0 in the early 1990s.

A mouse is a pointing device that detects two-dimensional motion relative to a surface, and it generally involves the motion of a pointer on a display. It is held in the user's hand and generally has one or more buttons and a scroll wheel (Fig. 14.9).

An optical mouse was invented in 1980, which eliminated the need for the use of the rolling ball. The latter often became dirty from rolling around leading to a negative impact on its performance. It was several years before optical mice became commercially viable, but today they have replaced ball-based mice and are supplied as a standard part of new computers.

An optical mouse is an advanced computer-pointing device that uses a light-emitting diode (LED), an optical sensor, and digital signal processing (DSP) instead of the traditional ball mouse technology. Movement is detected by sensing changes in reflected light rather than the interpretation of a rolling sphere. Steve Kirsch of MIT and Mouse Corporation and Richard Lyon of Xerox invented the first optical mouse independently of each other in 1980.

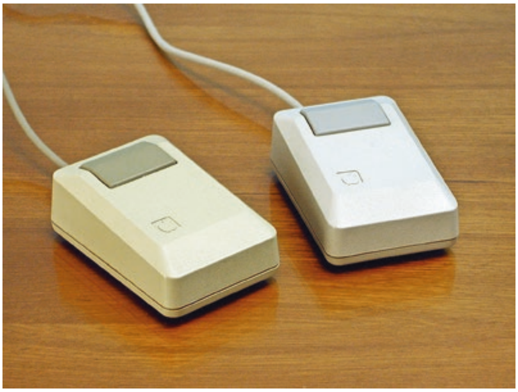


Fig. 14.8 Two Macintosh Plus Mice. 1984

Fig. 14.9 A. Computer mouse with two buttons and a scroll wheel

14.6 Review Questions

1. Describe the evolution of the software industry.
2. Describe the suite of applications in Microsoft Office.
3. What is a text-based interface?
4. What is a graphical user interface?
5. Explain the importance of software usability.
6. Investigate the various usability standards such as ISO 9241 and ISO 16982.
7. Explain user-centered design.
8. Describe the evolution of human computer interfaces.

14.7 Summary

IBM decided in 1968 to unbundle many of its software programs, and this decision changed the computer industry forever, with software changing from being a give-away item to becoming a commercial product and industry in its own right. This decision led in time to the software and services industry that we see today, and the quality of software and its usability became increasingly important.

Human–computer interaction is a branch of computer science that is concerned with the design, evaluation, and implementation of interactive computing systems for human use. It is focused on the interfaces between people and computers and has grown over the decades to include text-based interaction systems, graphical user interfaces, and voice user interfaces.

Humans interact with computers in many ways, and so it is important to understand the interface between them to facilitate the interaction. The early interaction between humans and computers was via batch processing with limited interaction between the operator and computer. These were followed by text-based interfaces (also known as a command line interface), where the system interaction (input and output) and navigation is text based.

The graphical user interface is a human computer interface that uses graphical icons, menus, and windows to represent information and action to the user. They are intuitive and user friendly and have made computers and electronic devices attractive to nontechnical users.

The success of modern software systems is related to the usability of the software, and user-centered design has become a key paradigm in building usability in the software. It places the user at the center of the software development process with active user involvement and evaluation employed.