Our Motto: 124 seats, 124 tickets.

Having the number of passengers and the number of seats come out the same has not always been easy.

(Major airlines have ticket offices all over the country. To err was inevitable.)

So we got together with IBM to make sure the seat we reserve for you is reserved for you.

And you can imagine what a collaboration like this would come up with. A computer. (A giant that took 10 years and 30 million dollars to develop. We call it Sabre.)

It not only "memorizes" every seat on every flight we have—it also memorizes the name and address of everybody on a waiting basis.

The moment there's a cancellation, it tells us you're next on the list—and even gives us your number to call.

(And it doesn't wait until we ask. It barges right in and tells us there's an empty seat on Flight 61 and to get hold of Paul Zoellner in Riverdale, New York.)

There it is. 124 seats. 124 tickets. Want one?

American Airlines

The SABRE airline reservation system was a path-breaking application that gave American Airlines a great commercial advantage over competitors.

1

The Software Industry

In January 1952, half a century ago, *Fortune* magazine ran an article titled "Office Robots." It was one of the very first general articles on computers. No mention was made of programming or software—indeed, the latter term had not yet been invented. However, the general-purpose nature of the computer was well understood. Having described the use of the Univac computer at the US Census Bureau the writer explained: "At the flip of a few switches, UNIVAC can be turned from such mass statistical manipulations to solving differential equations for scientists or handling payroll lists, computations, and check writing for businessmen."[1] The slightly inappropriate metaphor "flipping a few switches" suggests that programming was either something *Fortune*'s writer did not understand or something he thought his readers did not need to know about.

It would be nearly 15 years before a major business magazine devoted a feature article to software. In November 1966 *Business Week* carried a report titled "Software Gap—A Growing Crisis for Computers."[2] The article bemoaned the shortage of programmers, but hinted at the glorious opportunities ahead for the nascent software industry. For almost another 15 years, however, the software industry remained a hidden world—known mainly to computer-industry professionals, investors, and analysts. Not until September 1980 did *Business Week* carry a special report on the software business, its first in-depth look at the industry since 1966. That this article appeared at all was a tribute to the Association of Data Processing Service Organizations (ADAPSO), the trade association of the US software industry, whose Image Committee had worked tirelessly behind the scenes to get the industry noticed by the media. Titled "Missing Computer Software," the 1980 report (like its 1966 predecessor) highlighted the shortages of software applications and programmers and trumpeted the recent spectacular growth of the industry.[3] The article is

especially interesting now because, like a fly in amber, it is caught in a time when computing still meant "big iron" and the only kind of software most business people knew about was bought for huge sums by corporations. Yet even as this article was going to press the personal computer was changing everyone's perception of information processing.

Business Week's next special report on the software industry appeared in February 1984, and it could not have been more different in tone from the 1980 article.[4] In the intervening 3½ years, the rise of the personal computer had made the world at large aware of software, and brand names such as VisiCalc, WordStar, and Lotus 1-2-3 had entered the lingua franca of many office workers. There was no longer any need for *Business Week* to explain as it had in 1980—like a kindergarten teacher explaining to the class—that software was "the long lists of commands or instructions that tell the computer what to do."[5] Instead, the 1984 article, titled "Software: The New Driving Force," spoke of a $10 billion industry with boundless opportunities.[6] No longer was software only for corporations; now it was in the shopping malls too.

Writing at the very end of the twentieth century, the authors of a book titled *Secrets of Software Success* claimed: "Life without software is hard to imagine. Without software, paper letters would be the fastest form of written correspondence. No fax, no e-mail, and no business voice mail. But that's just the beginning of the impact of software. Across industries, software now enables and fuels economic growth. . . . Software tasks today range from controlling nuclear power plants, recognizing customer purchasing patterns, enabling stock trading, and running banking systems all the way to running cell phone systems and exploring for oil."[7] Warming to their theme, the authors continued: "Software—nothing but pure knowledge in codified form—largely drives and enables today's economy."

If the writers seem somewhat hyperbolic, they should be forgiven. From its first glimmerings in the 1950s, the software industry has evolved to become the fourth largest industrial sector in the US economy.[8] This book is the story of that evolution.

Understanding the Software Industry

Although today most people are aware of the software industry, not many would claim to "understand" it. In contrast, most people have a sense of knowing, say, the automobile industry—they are familiar with its products, they know or can envision the production processes, and they

understand the links between producer and consumer. Perhaps this understanding is naive and illusory, for beneath the surface there is a fantastically intricate set of industrial networks. However, in recent times no one has felt the need to "explain" the automobile industry. The same is true of most other producer industries, whether they be chemicals, airplanes, building materials, or food.

Yet when it comes to software, people are much less comfortable. This is due in part to the intangible nature of software, evocatively described by one prominent software scientist as "only slightly removed from pure thought-stuff."[9] But it is also attributable to the fact that traditional industries have been around for so long that we have unconsciously internalized a great deal of knowledge about them. The software industry is relatively new. Twenty-five years ago it was invisible and unacknowledged; today it is ubiquitous.

The aim of this book is to explain the software industry by a historical account of its evolution. Because no simple one-dimensional framework is adequate for this purpose, I use three main vectors of explanation. The first vector is that of time—the historical development and periodization of the industry. This vector informs the whole structure of the book, which traces the evolution of the industry from its first glimmerings in the mid 1950s to the mid 1990s in a series of partially overlapping but chronologically progressing narratives. The second vector of explanation is the sectorization of the industry, which can be divided into three main types of firm: software contractors, producers of corporate software products, and makers of mass-market software products. The third vector is that of products and markets. Software comes in many prices, sizes, and genres; sometimes one copy is sold, sometimes 100, sometimes 10 million. Clearly this range of possibilities leads to a significant variety that suggests an explanation though classification or taxonomy.[10]

Periodization, Sectorization, and Capabilities

The software industry can be divided into three sectors: software contracting, corporate software products, and mass-market software products.[11] Each of these three sectors emerged at a moment when contemporary computer technology created a business opportunity for a new mode of software delivery. Rather neatly (though purely coincidentally), the three sectors arrived at intervals of a decade.

Software contracting developed alongside the corporate mainframe computer in the mid 1950s. A software contractor wrote a one-of-a-kind

program for a corporate or government customer. Custom-written programs were hugely expensive, $1 million being not untypical.

Corporate software products emerged after the launch of the IBM System/360 computer family in the mid 1960s. The new IBM computer was relatively inexpensive, sold in large numbers, and thus created a much broader market for lower-cost software than could ever have been satisfied by software contractors. A software product was a program that could be used without modification by a large number of corporate users. Software products typically automated common business functions, such as payroll or inventory management, or ran an entire medium-size business, such as a manufacturing operation or a savings bank. They were typically priced between $5,000 and $100,000. The more successful ones sold in the hundreds, and a few in the thousands.

The arrival of the personal computer in the mid 1970s created an opportunity for mass-market software. The most characteristic form of distribution was a shrink-wrapped box of software sold in a retail store or by mail order. Software for personal computers was relatively cheap (typically between $100 and $500) and sold in large volumes, often several hundred thousand copies. In parallel with the personal computer revolution, there was a revolution in software-based home entertainment. Entertainment software was a major subsector of the mass-market software industry.

The terminology used to describe each of the three sectors is somewhat problematic, or at least ahistorical, because all three sectors have continued to flourish since their inception and have adopted the preferred terminology of the day. For example, "software contracting," which began in 1955, pre-dated the invention of the word "software"; it originally went by such names as "custom programming" and "programming services." Similarly, the first pre-packaged programs were simply called "software products," no further distinction being necessary. With the rise of the personal computer software industry, it became necessary to distinguish between the markets for corporate software and personal computer software by introducing terms such as "enterprise software" and "shrink-wrapped software."

The division of the software industry into three sectors is natural both in market terms and in terms of the distinctive business models that firms evolved. The software firms' competencies and their knowledge of their specialized markets enabled the more successful firms to maintain dominant positions in their own sector but made it difficult for them to cross over into either of the other sectors. Thus, the very strengths that

enabled a firm to succeed in one market segment became institutional rigidities in another. This is the main reason why few firms have successfully escaped the confines of their particular sector.[12]

Software Contractors

The defining event for the software contracting industry came in 1956, when the US-government-owned RAND Corporation created the Systems Development Corporation (SDC) to develop the computer programs for the huge SAGE air defense project. This was the first of several multi-billion-dollar defense projects in the 1950s and the 1960s, known as the L-Systems, that provided an important market for early software contractors. At the same time, computer manufacturers and private corporations were also creating a demand for software, albeit on a smaller scale. In response to the latter demand, small startup firms such as the Computer Usage Company (CUC) and the Computer Sciences Corporation (CSC) came into existence. These firms ultimately developed into major corporations that competed successfully for the largest software contracts.

The business model consciously or unconsciously adopted by custom programming firms was that of an engineering or construction contractor. They existed by bidding for and winning contracts executed on a time-and-materials basis or a fixed-price basis. The critical capabilities for a software contracting firm were exploitation of scope, cost estimation, and project management. A successful software contractor exploited the economies of scope by specializing in particular submarkets. For example, SDC specialized in real-time defense projects, while CSC focused on systems software for computer manufacturers. By concentrating on these narrow markets, firms could reduce costs by reusing software from one project in the next and could develop specialized human resources by working in a consistent application domain. Specialized domestic knowledge enabled non-American firms to survive against multi-national competitors. The profits on software contracting were surprisingly low, typically less than 15 percent of sales, so cost-estimation and project-management skills were essential. Accurate cost estimation was needed to prepare a price-competitive bid, and project-management skills were needed to ensure completion within time and cost constraints. In contrast, marketing was a relatively unimportant competence, since most of the selling was done through the personal contacts of senior staff members or by responding to openly published requests for quotation.

Corporate Software Products

Two packaged programs, Applied Data Research's Autoflow and Informatics' Mark IV (announced in 1965 and 1967, respectively), are generally agreed to be, if not the first, certainly the most influential of the early software products. These products and a few others had already proved viable in the market in January 1970, when IBM implemented its "unbundling" decision. Previously, IBM had provided programs free of charge to customers on request, as had the other computer manufacturers. This made it difficult for software entrepreneurs to establish a market. Therefore, the software products that succeeded were ones that satisfied needs not yet anticipated by the computer manufacturers. Under antitrust pressure (perhaps assisted by an independent lawsuit from Applied Data Research), IBM decided to charge separately for software and other services. Unbundling had the effect of establishing a vibrant market for software products, which previously had been merely embryonic. It was a turning point for the industry.

At first, because of the analogy between the low incremental costs of reproducing programs and recorded music, the software products business was likened to the recorded-music industry. This turned out to be an illusion. Because of their high marketing costs and the need for sales support, corporate software products were classic capital goods. Thus, the business model adopted by the software products firms, often quite consciously, was that of a producer of capital goods—and the firms often looked to computer manufacturers, particularly IBM, for role models. The critical capabilities that the firms developed were exploitation of scale, corporate marketing, quality assurance, and pre- and after-sale support. Exploitation of scale was the most important of these capabilities, because selling in volume was the only way to recover the high initial development costs of a generalized software product, which were much higher than for custom software. Because sales volume was so important, it was necessary to develop quota-based sales operations, typically on the IBM model, and for this reason firms often recruited former IBM salespeople. Software products, such as database programs or industrial applications, were usually "mission critical," and for this reason product reliability was paramount. The software firms developed skills in quality assurance, using such techniques as beta testing to ensure that programs were ruggedly "productized" and reliable in use. Finally, as with all capital goods, pre- and after-sale support was needed to establish a long-term relationship with the customer. In the case of software products, this took the forms of product customization, user training, and regular upgrades.

These services turned out to be unexpected sources of income for which the pioneers of the industry had not initially planned.

Mass-Market Software Products

The personal computer software industry began in the late 1970s with the establishment of hundreds of very small software firms, almost none of which had any connection with the existing software industry. Microsoft is one of the few firms from this early period to have survived. The industry really took off in 1979–80, with the arrival of mass-market software such as Software Arts' VisiCalc spreadsheet and MicroPro's WordStar word processor. In many popular histories, VisiCalc is credited as the "killer app" that kicked off the personal computer revolution. The concept of the "killer app" is attractive and superficially plausible, but it has been neither taken up nor refuted by academic economists. The view taken in this book is that the "killer app" hypothesis probably confuses cause and effect. Thus, one could argue that the personal computer established a platform on which many software products could exist, and that VisiCalc was simply a prominent example. Had VisiCalc not existed, the personal computer revolution would still have happened, and perhaps another software product would have earned the epithet "killer app."

Along with VisiCalc and WordStar came a slew of popular products that became, if not exactly household names, certainly well-recognized brands: Supercalc, Lotus 1-2-3, dBase II, WordPerfect, and many others. Closely related to the personal computer software industry (whose products were used in corporations and homes) was the recreational software industry (whose products were used exclusively in domestic and learning environments). Recreational software products tended to be cheaper (typically $50) and more ephemeral. Early firms active in the recreational software industry included Activision and Broderbund.

The personal computer software products industry was completely disjoint from the corporate software products industry. The essential difference between the two was that their markets differed by two orders of magnitude in numbers of units sold. For example, in 1984 the world's best-selling corporate software product was Informatics' Mark IV, with 3,000 installations; the best-selling personal computer software product was WordStar, with sales of 700,000.

The analogy to the recorded-music business held, and the business model adopted by the industry was that of a producer of information goods. Another parallel was drawn with the pharmaceutical industry,

which had a similar cost structure based on high R&D inputs, low production costs, and high marketing expenses.

The critical capabilities developed by the personal computer software firms included exploitation of scale, mass marketing, and ease of use. Exploitation of scale through high volumes was the defining characteristic of the mass-market software industry, whose product cost structure was entirely different from that of the corporate software industry. For example, the cost of a Mark IV installation was about $100,000, while WordStar cost $495. Personal computer software firms targeted their products at the end user rather than at the corporate information systems manager, making use of low-cost distribution channels such as retail outlets and mail order. This required the development of a set of marketing competencies much different from that of corporate software firms, with their IBM-type sales forces. In time, personal computer software firms with strong corporate sales, such as Lotus and Microsoft, developed conventional sales forces. The sales messages, as expressed in advertising, continued to be largely directed toward end users, however.

So that personal computer software products could be used by many thousands of customers without any after-sale support, programs had to have intuitive interfaces and had to require no customization. This again required the development of a set of skills different from that of corporate software makers, who could rely on training courses and third parties to install and customize software.

Figure 1.1 illustrates the three-sector structure of the software industry, populated with a few of the firms that will feature in this book. Though this view of the industry is widely accepted by academics and industry analysts, it is an artificial construction designed to bring some coherence to what would otherwise be a star field of random firms. Though the great majority of firms can be unequivocally placed in a particular sector, this is not true of every firm. Most of the software contractors established in the 1950s and the 1960s subsequently sold software products in the 1960s and the 1970s as a subsidiary activity. For example, Informatics began as software contractor in 1962, and programming services remained its primary business, even though its Mark IV became the biggest-selling independent software product in the industry. Hence, Informatics can be properly located in both software contracting and corporate software products. Such multiple activities were usually reflected in a firm's organizational structure. In Informatics, for example, there was a separate "Software Products Division" that ran the Mark IV operation. Later in the development

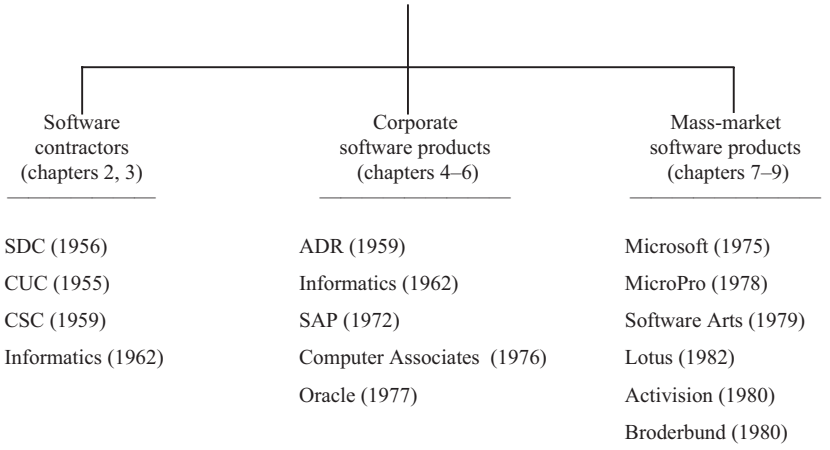

Figure 1.1
A taxonomy of the software industry.

of the industry, producers of corporate software, such as Computer Associates and Oracle, attempted to establish personal computer software operations.

An inestimable advantage of structuring this book around the three-sector model is that it ensures an allocation of space that bears some relation to the importance of any given subject. Thus, Microsoft, the leading player in personal computer software, gets a rightful and prominent place in the chapters devoted to mass-market software, but nowhere else in the book. Microsoft deservedly dominates about 10 percent of the book, just as it dominates 10 percent of the software industry. If this book serves no other purpose, I hope it will at least provide a corrective to the common misconception that Microsoft is the center of the software universe around which all else revolves.

Scope: Exclusions and Limitations

Packaged software often has a list of exclusions and limitations that, when read carefully, explain to a disappointed user why the software does not do the very thing he or she bought it for. In the spirit of exclusion and limitation, I should declare up front what this book is not about.

The most severe limitation is that the book has a strong US focus. To some extent this is justified in a single-volume work. The United States dominates the world's software industry, especially in software products.

However, other countries do have indigenous software industries, and readers may feel that these countries have been slighted. In fact, I am British, I wrote this book in Britain with only occasional visits to the United States, and I probably have a better knowledge of the British software industry than of the American. However, I felt unable to incorporate much material on the British software industry in this book because it would have been disproportionate, would have appeared chauvinistic, and would have raised this question: What about the software industries of Germany, France, Japan, Israel, Ireland, the former Eastern Bloc countries, and so on? All these countries certainly deserve attention, but I think this would be best done by a set of monographic studies written from the viewpoint of the individual software nation. Indeed, there is a fine study of the Indian software industry by my colleague Richard Heeks at Manchester University.[13] In 1996, when Heeks published *India's Software Industry*, the exports of the entire Indian software industry were estimated at about $700 million. At the time, the total revenues of Microsoft were $9 billion and those of Computer Associates were $3.5 billion, about half from overseas sales. Thus, India's entire software exports were less than the software exports of any of the top five US producers, and only a fraction of 1 percent of world output. Purely on a space argument, it is hard to make a case for a long discussion of Indian software here. Many industry observers foresee a glittering future for the Indian software industry, but it is a complex industry with its own rich history, its own economic and political shaping, and its own structure (for example, it has a distinctive "offshore" software writing sector). Indeed, it is so different from the American software industry that it might almost be on another planet. The Japanese software industry is completely different again, with a strong software contracting sector but a weak software products sector (with the exception of videogames).[14] I believe that the same is true of the other nations, and one could not begin to incorporate these different histories into a single volume except in the most superficial way. These other software nations all deserve to be studied on their merits, not as a set of potted plants set out for the inspection of the hurried Western reader.

A second limitation of this book is the cutoff date of 1995. Naturally, historians have a professional reluctance to write about very recent events on which they lack a proper perspective, so I have no fear of criticism from other historians on that score. However, any self-respecting industry analyst or software journalist would bring the story up to date and would, for good measure, project a few years into the future. This

involves a set of skills different from that of the historian. It is not mere pusillanimity that makes me reluctant to attempt to do the same, but the fact that such projections are often wrong and therefore that contemporary obsessions often miss the real drama and turning points. For example, in the last 5 years there has been an enormous amount of press coverage of the Java programming language, the Linux operating system, and open-source software. I have no idea whether these will turn out to be turning points in the industry or not, and my opinion is certainly no better than the average pundit's. On the other hand, I find it quite fascinating that in the business press of the early 1990s the Internet was one of the least-written-about subjects, getting perhaps one-tenth the column inches devoted to Microsoft Windows or the tribulations of WordPerfect. I don't know what it is, but I bet there is something much more important going on right now than Java, Linux, or open-source software, and that it will be 2010 before it becomes fully apparent.

The year 1995 also seems a good cutoff point because there is a sense that in the mid 1990s, with the rise of the Internet, the software industry entered a new phase of its development. For example, the electronic delivery of software has made the metaphor of "shrink wrapped" software inappropriate. More significantly, software firms appear to be extending their reach as the boundaries between corporate and consumer software become more diffuse. Thus, since 1995, Microsoft has increasingly strayed from its traditional desktop software into corporate networks at one end of the spectrum and into videogame consoles at the other. Likewise, Oracle, once an archetypal producer of corporate software, has been making forays into desktop software and video entertainment. In *Secrets of Software Success*, Detlev Hoch and his co-authors project the "Internet Era," a new period in the development of the software industry, and suggest boundary dates of 1994 and 2008.[15] My expectation is that we will indeed, as historians, move toward such a periodization in due time. However, my aim in this book is to set out a history that is robust enough to make sense 10 or 15 years from now.

Numbers: Software Industry Statistics and Trade Associations

Recently I read an article in the business press that referred to "the $300 billion software industry." Three years of researching the software industry had made me cautious about interpreting such statements. Before I became immersed in the subject, I had naively assumed that making software was not much different from making photocopiers, refrigerators, or

automobiles. I assumed it would be possible to determine the number sold, the total revenues of the industry, and the value of the market. And, allowing for distribution costs, it would be possible to reconcile these numbers. Unfortunately, this is just not possible for the software industry. Indeed, it is not even possible to find out the total revenues of the industry from any public-domain source of which I am aware. Here are some of the reasons.

According to *Software Magazine*, the sales of the top 500 software firms in the year 2000 amounted to $259 billion. However, the authors of *Secrets of Software Success* cite two estimates for the total number of firms in the software industry worldwide, one stating that there are 35,000 firms with more than five employees and the other stating that there are 150,000 firms "regardless of size."[16] Take your pick. The US Census Bureau stated that in 1997 there were 12,000 software publishing "establishments" and a further 31,000 establishments engaged in programming services in the United States. Thus, the figure given by *Software Magazine*—a well-regarded proxy for industry size—speaks only for the biggest firms; it leaves out many thousands of small firms. This has always been the case. In the late 1970s, analysts computed revenues for the top 50 firms, then the top 100, then the top 200, then the top 500.

Another problem with measuring the software industry is that it is in one respect (though only in one respect) like the chemical industry: its products are sometimes consumed within the industry and sometimes by end users. (You might occasionally buy a couple of pounds of washing soda, but when did you last buy a retail gallon of sulfuric acid?) For example, a major activity in the software industry is the installation of Enterprise Resource Planning (ERP) software made by firms such as SAP and Oracle. When SAP or Oracle uses its in-house personnel to install the software for an end user, the combined cost of software and consulting services shows up in its year-end revenues. However, when a computer services firm such as EDS or CSC undertakes the same activity, the cost of the ERP software bought in from SAP or Oracle is passed on to the end user together with the installation charges, and the original ERP software appears in the accounting books of both firms and adds to their year-end revenues. This is not an isolated example. Today most software products are constructed using at least some tools or components bought from other software vendors, or some development activity may be subcontracted to a specialist software house. Because of this "double counting," total sales to end-users is arguably a better measure of the software industry than total industry revenues.[17]

Table 1.1 presents data on the US software market in the period 1970–2000. I believe this is the only 30-year time series for the software industry in the public domain, and it appears here courtesy of the industry analyst INPUT. INPUT has a rigorous, proprietary methodology—which I have seen but cannot disclose—based on questionnaires and interviews with end users and subsequent reconciliation with vendors.[18] Table 1.1 covers only the US market, because INPUT did not begin to capture worldwide data until much later. However, the value of table 1.1 is the trend. The time series gives a real insight into the growth of the industry. This is demonstrated visually in figure 1.2, which shows the classic "hockey stick" growth of the industry—growth that has not yet begun to flatten. Figure 1.3 reveals some interesting subplots. It shows that software products became the dominant mode of US of software consumption around 1980, and that the growth of programming services then began to slow, relatively. Figure 1.3 also shows that software products did not really take off until the early 1980s, long after IBM liberated the industry by unbundling.

There are two main time series for US software industry revenues in the public domain. One series (which lapsed in 1990) was published by ADAPSO; statistics supplied by INPUT were used in its later years. The other series is produced at taxpayers' expense by the US Census Bureau.

Table 1.2 gives the ADAPSO-INPUT industry statistics for the years 1966–1990.[19] This is the only long-term time series for US software industry revenues in the public domain. ADAPSO, founded in 1961, became the leading trade association for the US computer services and software industries.[20] In its first 25 years of existence, the number of member firms grew from about 40 to 850; these tended to be the larger firms, and between them they represented approximately half the sales of the industry. ADAPSO was like other trade associations in that its role was to promote the industry's interests through lobbying, public relations activities, education, standards setting, and so on. In 1991 it was renamed the Information Technology Association of America (ITAA).

ADAPSO's many activities were complemented by a program to gather statistics on the industry. Annual industry surveys were published from 1966 to 1990. Early on, before the emergence of professional software industry analysts, ADAPSO made use of an academic consultant. Beginning in 1970, it employed the International Data Corporation (IDC) and Quantum Sciences, two early providers of information on the computer industry. At first, ADAPSO simply tracked the two main classes of industry participant: processing services firms and software houses.

Table 1.1
User expenditures, US software market.

	Systems software products	Applications software products	Total software products	Programming services	Total software
1970	$150,000,000	$100,000,000	$250,000,000	$744,000,000	$994,000,000
1971	$210,000,000	$140,000,000	$350,000,000	$856,000,000	$1,206,000,000
1972	$270,000,000	$170,000,000	$440,000,000	$952,000,000	$1,392,000,000
1973	$330,000,000	$210,000,000	$540,000,000	$1,072,000,000	$1,612,000,000
1974	$390,000,000	$270,000,000	$660,000,000	$1,200,000,000	$1,860,000,000
1975	$490,000,000	$320,000,000	$810,000,000	$1,352,000,000	$2,162,000,000
1976	$590,000,000	$390,000,000	$980,000,000	$1,528,000,000	$2,508,000,000
1977	$720,000,000	$480,000,000	$1,200,000,000	$1,728,000,000	$2,928,000,000
1978	$890,000,000	$590,000,000	$1,480,000,000	$1,984,000,000	$3,464,000,000
1979	$1,150,000,000	$720,000,000	$1,870,000,000	$2,346,000,000	$4,216,000,000
1980	$1,401,000,000	$1,325,000,000	$2,726,000,000	$2,985,000,000	$5,711,000,000
1981	$1,974,000,000	$2,191,000,000	$4,165,000,000	$3,994,000,000	$8,159,000,000
1982	$2,685,000,000	$3,080,000,000	$5,765,000,000	$4,335,000,000	$10,100,000,000
1983	$3,534,000,000	$4,168,000,000	$7,702,000,000	$5,023,000,000	$12,725,000,000
1984	$5,333,000,000	$5,741,000,000	$11,074,000,000	$5,387,000,000	$16,461,000,000
1985	$6,322,000,000	$6,964,000,000	$13,286,000,000	$6,233,000,000	$19,519,000,000
1986	$8,022,000,000	$9,015,000,000	$17,037,000,000	$6,833,000,000	$23,870,000,000

1987	$9,880,000,000	$10,670,000,000	$20,550,000,000	$7,540,000,000	$28,090,000,000
1988	$12,095,000,000	$12,970,000,000	$25,065,000,000	$8,805,000,000	$33,870,000,000
1989	$14,512,000,000	$16,208,000,000	$30,720,000,000	$10,185,000,000	$40,905,000,000
1990	$16,390,000,000	$17,676,000,000	$34,066,000,000	$10,402,000,000	$44,468,000,000
1991	$18,370,000,000	$18,923,000,000	$37,293,000,000	$10,872,000,000	$48,165,000,000
1992	$19,825,000,000	$21,582,000,000	$41,407,000,000	$11,657,000,000	$53,064,000,000
1993	$21,702,000,000	$24,176,000,000	$45,878,000,000	$12,663,000,000	$58,541,000,000
1994	$23,845,000,000	$28,003,000,000	$51,848,000,000	$13,627,000,000	$65,475,000,000
1995	$26,200,000,000	$32,111,000,000	$58,311,000,000	$15,319,000,000	$73,630,000,000
1996	$28,753,000,000	$36,998,000,000	$65,751,000,000	$19,774,000,000	$85,525,000,000
1997	$31,618,000,000	$42,724,000,000	$74,342,000,000	$23,100,000,000	$97,442,000,000
1998	$33,426,000,000	$50,365,000,000	$83,790,000,000	$27,250,000,000	$111,040,000,000
1999	$37,217,000,000	$61,227,000,000	$98,444,000,000	$31,910,000,000	$130,354,000,000
2000	$41,689,000,000	$63,000,000,000	$104,689,000,000	$33,400,000,000	$138,089,000,000

Courtesy of INPUT.

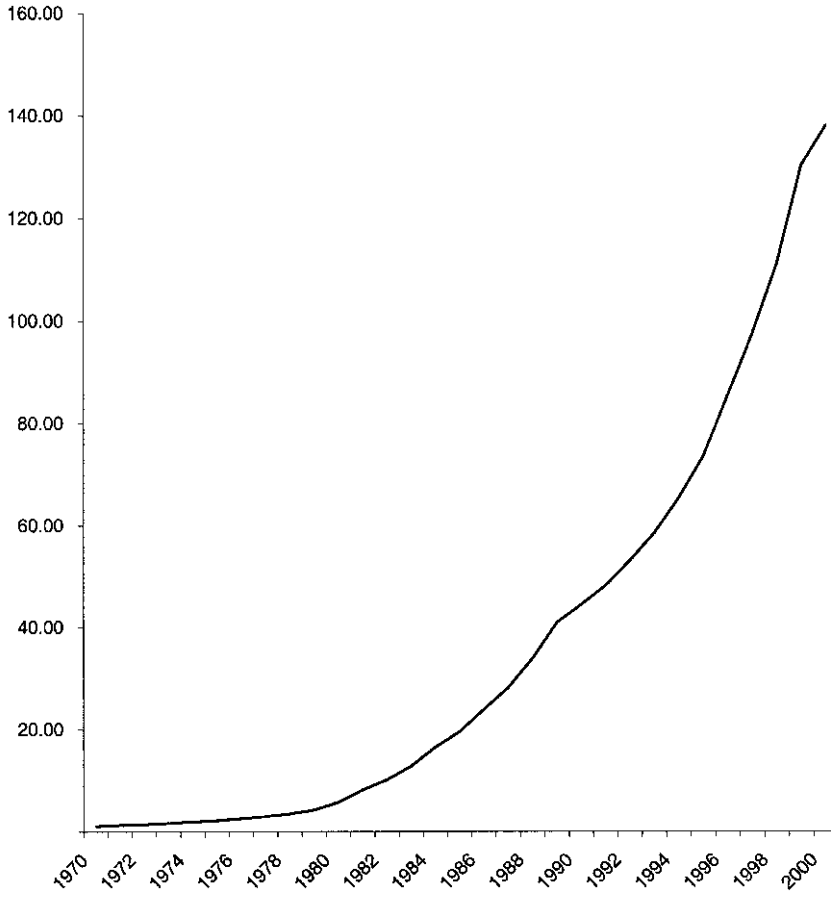

Figure 1.2
The total US software market (user expenditures in billions of dollars), 1970–2000. Courtesy of INPUT.

Processing services firms included organizations (such as ADP and Computers & Software Inc.) that undertook data processing activity for client firms and whose programming activities, if any, were incidental. Software houses were organizations that undertook custom programming for client firms or, beginning in the late 1960s, sold software products.

In the absence of an official census of the software industry, information providers used ad hoc collection methods based on surveys of member firms and on sampling. For example, ADAPSO's fourth annual survey (1970) estimated the number of firms in the industry "from examining the firms listed under 'Data Processing Services' in the yellow pages for

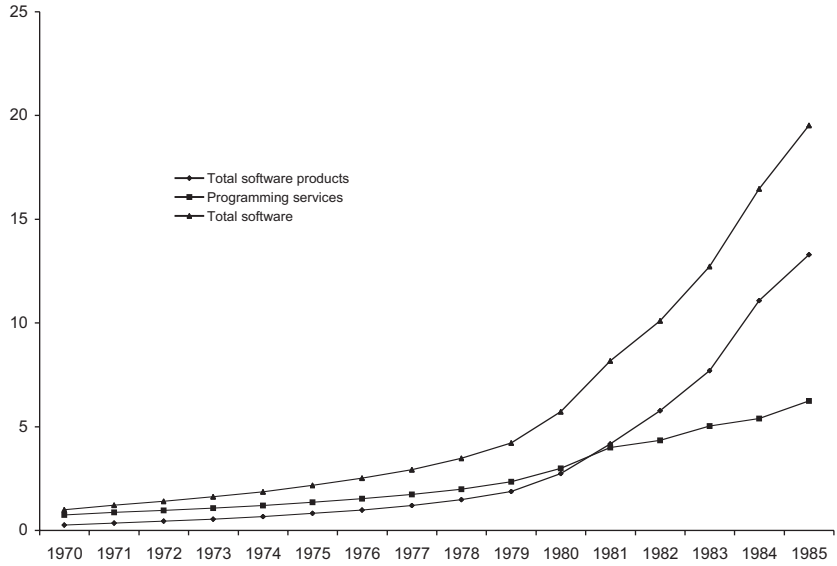

Figure 1.3
The US software market (user expenditures in billions of dollars), 1970–1985. Courtesy of INPUT.

31 cities."[21] From 1976 to 1990, the computer industry analyst INPUT produced all ADAPSO's statistics. Beginning in 1977, INPUT disaggregated software houses into programming services firms and software products firms. Beginning in 1980, integrated systems (also known as turnkey systems) were included. INPUT's figures are estimates of the total domestic and international revenues of all US-owned software producers, excluding recreational software firms.

The statistics produced by INPUT for ADAPSO were the best-regarded in the industry, and they were adopted by the US Department of Commerce and by the Organization for Economic Cooperation and Development for their official reports on the software industry in the mid 1980s.[22] Since 1991, ADAPSO's successor, ITAA, has no longer published annual surveys of the industry, primarily because the ITAA's board decided to suspend funding for the statistics program; once terminated, it was never resumed.

Two other trade associations also published statistics on the industry. One of these was the Software Publishers Association (SPA), established in 1984 to exclusively represent the mass-market software industry, including producers of business, educational, domestic, and recreational

Table 1.2
Revenues, US computer services and software industry. ("Software" column includes software products and programming services.)

	Total		Processing services		Software		Software products		Programming services		Turnkey		Systems integrators	
	Firms	$ million	Firms	$ million	Firms	$ million	Firms	$ million	Firms	$ million	Firms	$ million	Firms	$ million
1966	700	540		440		100								
1967	840	740		560		180								
1968	1,300	1,040		770		270								
1969	1,300	1,550		1,100		450								
1970	1,400	1,900		1,460		440								
1971	1,500	2,350		1,900		450								
1972	1,600	2,760		2,300		460								
1973	1,700	3,230		2,750		480								
1974	1,900	4,410		3,760		650								
1975	2,550	4,580	1,557	3,290	993	1,290								
1976	2,584	5,325	1,556	3,605	1,028	1,720								
1977	2,977	6,300	1,942	4,700	1,035	1,600	618	600	417	1,000				
1978	3,391	7,750	2,089	5,580	1,302	2,170	752	940	550	1,230				
1979	4,055	9,446	2,140	6,706	1,915	2,760	1,095	1,210	820	1,550				
1980	4,336	14,903	2,132	8,800	2,203	6,103	1,225	2,631	978	3,472				
1981	6,178	22,311	2,259	11,220	2,889	8,335	1,605	3,765	1,284	4,570	1,030	2756		

1982	6,470	26,430	2,130	12,484	3,227	10,624	1,879	5,295	1,348	5,329	1,113	3,322		
1983	7,000	32,600	2,150	14,600	3,650	13,900	2,250	7,500	1,400	6,400	1,200	4,100		
1984		41,200		14,800		20,000		11,100		8,900		6,400		
1985	7,313	47,000	2,121	16,300	3,963	23,600	2,488	13,000	1,475	10,600	1,229	7,100		
1986	7,532	54,000	2,110	21,300	4,260	25,800	2,705	14,800	1,555	11,000	1,162	6,900		
1987		67,400		21,800		33,300		20,600		12,700		8,500		5,500
1988	8,070	79,400	2,225	25,000	4,510	40,200	2,970	25,100	1,540	15,100	1,255	9,500	80	6,900
1989		89,800		22,700		45,900		30,700		15,200		9,500		7.500
1990		100,400		25,100		51,300		34,500		16,800		10,300		8,900

Source: ADAPSO annual reports and internal sources. Data attributed to unnamed consultants, IDC, Quantum Sciences (various years), and INPUT (1976–1990).

software. Statistics were published from 1984 to 1997, when publication was discontinued owing to funding problems. The SPA's activities were primarily directed at reducing software piracy.[23] This activity was always politically difficult, not least because the Association's strong-arm tactics risked alienating the industry's customers.[24] For this reason, another trade association, the Business Software Alliance (BSA), was formed by a group of the largest personal computer software firms in 1987. Its overt mission was to promote anti-piracy legislation and enforcement, particularly overseas, in a more politically adept way. In its lobbying activities, the BSA has used census data on the US software industry to demonstrate that the software products industry is a large sector of the US economy, is a major net exporter, and deserves anti-piracy legislation.

Table 1.3 shows the US Census Bureau statistics for the computer and software industries, which it began to publish in the mid 1980s. Standard Industry Classification 737 consists of "computer programming, data processing, and other computer related services." In 1987, the SIC 737 classification for the computer services and software industries was subdivided into nine subclasses, numbered 7371 through 7379. The first three of these constituted the "core" software industry—programming services (SIC 7371), packaged software (SIC 7372), and integrated systems (SIC 7373). Statistics on these three classes have been published since 1990 (table 1.4). Statistics on the software industry are collected as follows: Every five years (those ending in 2 and 7), all significant establishments are required by law to complete a schedule detailing their sources of income, while smaller establishments are sampled or their contributions estimated from other sources. In other years, firms are sampled in lieu of a full census. Respondents are required to complete a detailed questionnaire allocating their revenues to programming services, packaged software (including recreational software), and integrated systems. The resulting statistics eventually enter the public domain through publication in the *Statistical Abstract of the United States*. Because the Census Bureau carries out much more exhaustive data collection than commercial industry analysts (who do not have the force of law to compel responses or the machinery to analyze them), the census statistics are probably more accurate, but they are not necessarily more useful in understanding the industry.

It is important to appreciate that the ADAPSO-INPUT and Census Bureau statistics on the software industry are not really comparable. The ADAPSO statistics, which will mostly be used in this book, measure the total software revenues of US software firms, wherever located; the Census

Table 1.3
Annual receipts (in billions) of US computer and data processing services industries (Standard Industrial Classification 737), 1985–1998.

1985	1986	1987	1988	1989	1990	1991	1992	1993	1994	1995	1996	1997	1998
$45.1	$50.6	$56.0	$67.7	$78.7	$88.3	$94.4	$104.7	$117.9	$135.6	$156.4	$184.4	$215.3	$258.8

Source: US Census Bureau, Statistical Abstract of the United States, various years.

Table 1.4
Annual receipts (in billions) of US core software industry.

SIC		1990	1991	1992	1993	1994	1995	1996	1997	1998
7371	Programming services	$21.3	$23.4	$25.0	$27.4	$31.1	$35.1	$42.1	$50.1	$64.2
7372	Packaged software	$16.5	$18.3	$21.2	$25.2	$28.9	$33.2	$39.3	$43.1	$50.5
7373	Integrated systems	$12.9	$13.8	$15.2	$16.2	$17.0	$17.5	$20.2	$26.1	$31.8
7371–7373	Total software	$50.8	$55.6	$61.6	$68.6	$77.0	$85.8	$101.6	$119.5	$160.9

Source: US Census Bureau, Statistical Abstract of the United States, various years.

Bureau statistics report the worldwide revenues of software producers arising from their operations located in the United States, regardless of ownership (though in fact there are few non-American software firms with operations in the United States). Thus, the ADAPSO data would capture all the software revenues of Microsoft (say), while the Census Bureau data would capture only the part derived from domestic operations. Similarly, CSC's programming services and consulting operations in many non-US countries are not captured by the census data.

Although statistics for the software industries of individual nations exist, there are few time series, there is no universal basis for defining the sectors of the industry, and there are similar problems in interpreting industrial census statistics. Table 1.5 shows two of the more plausible estimates for the worldwide software industry in 1982 and 1990. The 1982 statistics were provided to the US Department of Commerce by INPUT and IDC. There is no estimate for Germany—then a major software producer, comparable to France or Britain—because "information was not available from government or private-sector sources [although] we presume, that based on the size of West Germany's computer equipment market, that the size of the country's software industry would place it at a level near the four shown."[25] This speaks eloquently for the poverty of the available statistics. More reliance can, perhaps, be placed on the 1990 data, attributed to the IDC.[26]

As table 1.5 shows dramatically, the United States dominates the world software industry, with a 70 percent market share in 1982 and 57 percent in 1990. What the table does not convey is the relative importance of the three sectors of the software industry in different countries. In 1990, an authoritative source estimated that Japan had a $7.9 billion market for programming services but only a $1.4 billion market for packaged software, and that most of the latter was imported from the United States.[27] Western Europe was much closer to the American model. In Britain, for example, the market for programming services was slightly smaller than that for package software, but the latter was heavily penetrated by US imports.[28] Without doubt, the packaged software industry was and remains an American phenomenon.

Sources

This book is perhaps the first attempt at writing a full-length history of the software industry broader than the study of an individual firm. To date, most histories of software firms, particularly the numerous books

Table 1.5
Revenues (in US dollars) of world software industry, 1982 and 1990.

	1982	1990
US	$10.3 billion (70%)	$62.7 billion (57%)
Japan	$1.2 billion (8%)	$14.3 billion (13%)
France	$1.3 billion (9%)	$8.8 billion (8%)
Germany	—	$7.7 billion (7%)
UK	$0.7 billion (5%)	$6.6 billion (6%)
Other		$9.9 billion (9%)
Total	$14.7 billion (100%)	$110.0 billion (100%)

Sources: 1982 data from US Department of Commerce, *A Competitive Assessment of the United States Software Industry,* p. 32-6 (data attributed to IDC and INPUT); 1990 data from Richard Brandt, "Can the US Stay Ahead in Software?" *Business Week,* March 11, 1991: 62–67 (data attributed to IDC).

about Microsoft, have been journalistic endeavors based on interviews and press clippings. These books are usually lively and full of incident, but historically flawed because they lack wider context and because they rely on hearsay not supported by documentation. Much more satisfactory have been the two major corporate histories written to date: Claude Baum's history of SDC, *The System Builders*, and Richard Foreman's history of Informatics, *Fulfilling the Computer's Promise.*[29] The authors of these two fine histories made extensive use of a major corporate archive in addition to oral testimony and the general business literature.

Since the present book does not focus on any one firm, heavy use of corporate archives was not appropriate, nor indeed was it possible. Instead, it is based largely on monographic studies of the software industry, the periodical literature, and reports by industry analysts. These sources are discussed in detail below. It is hoped, first, that this analysis of the literature will prove of value to future historians of the software industry, whether of individual firms or of broader sectors and themes. A second purpose of this analysis is to discuss the shortcomings of the literature, which have inevitably impinged on this book. Readers who have no interest in either concern may safely advance to chapter 2.

In discussing the monographic literature, one should perhaps begin with the bad news: There are more books written about Microsoft and Bill Gates than about the rest of the industry put together. Most of this literature has been produced by journalists seeking to satisfy the immense curiosity about how Bill Gates came to be the richest man in the

world. The result has been a gross distortion of the public perception of the structure of the software industry and Microsoft's place in it. That Microsoft generates at least half of the literature on the software industry but accounts for no more than 10 percent of traded software perhaps says everything.

In contrast with the torrent of Microsoft histories, there are barely a dozen worthwhile accounts of other software companies. Apart from the excellent histories of SDC and Informatics mentioned above, the majority of the rest are autobiographical accounts by movers and shakers in the software industry. They include Sandra Kurtzig's *CEO*, W. E. Peterson's *AlmostPerfect*, John Walker's *The Autodesk File*, Douglas Carlston's *Software People*, John Imlay's *Jungle Rules*, and Ben Voth's *A Piece of the Computer Pie*.[30] All these books contain valuable firsthand accounts of individual firms. Some of the better books by journalists (and not about Microsoft) are Mike Wilson's account of Oracle, *The Difference between God and Larry Ellison*, Gerd Meissner's *SAP: Inside the Software Power*, and Tristan Gaston-Breton's *La Saga Cap Gemini*.[31] There are also a few article-length reminiscences by industry pioneers, including Elmer Kubie's recollections of the early years of the Computer Usage Company and J. Lesourne and R. Armand's description of the first decade of the French software house SEMA.[32] Most histories of individual firms focus on the largest companies—those with hundreds or thousands of employees and with revenues of $1 billion or more. Since the average software company has fewer than 30 employees, this literature is highly unrepresentative. Hence, for me one of the hidden gems of software history is the little-known privately published volume *The MacNeal-Schwendler Corporation: The First Twenty Years,* written by that corporation's founding president, Richard MacNeal, in 1988.[33]

The biographies of individual firms, however many they number, are not truly representative of the software industry, any more than a random collection of biographies—of say Mozart, Marconi, and Kissinger—are representative of the human race. The best sources for the broader industrial scene are the reports resulting from government-sponsored software policy studies, a few academic monographs, and the publications of market-research organizations. Three major national policy reports related to software were written in the mid 1980s, when the software industry experienced its most dramatic growth spurt. These reports were published by the US Department of Commerce, by the Organization for Economic Cooperation and Development, and by the UK government's Advisory Council for Applied Research and Development.[34]

These reports, all in the public domain, give an excellent 1980s view of the software industry. During the 1990s, the software industry has had some attention from mainstream economists and business analysts. Publications in this category include David Mowery's edited volume *The International Computer Software Industry*, Salvatore Torrisi's *Industrial Organisation and Innovation: An International Study of the Software Industry*, Stephen Siwek and Harold Furchtgott-Roth's *International Trade in Computer Software*, and *Secrets of Software Success* by Detlev Hoch et al.[35] Although these works are not primarily historical, they are all informed by history, and in time they will become important historical sources in their own right.

The periodical literature of the software industry comprises, in order of usefulness, the trade press, general business periodicals, and newspapers. The trade press includes several long-running and well-respected general computer periodicals, such as *Datamation*, *Computer World*, and *Byte*, and many narrower titles. Among the latter are two that focused on the software industry: *Software News* and *Business Software Review*. *Software News* commenced publication in 1984 and was renamed *Software Magazine* in 1988; its profiles of individual companies and its industry rankings were particularly useful.[36] In the 1980s, *Business Software Review* was published by International Computer Programs as a spinoff of its *ICP Quarterly* software catalog. Its most interesting features were early rankings of the industry and reports of ICP's annual "million dollar awards" celebrating cumulative sales achievements of individual software products.[37] Among the general computer periodicals, by far the most useful is *Datamation*, which was published in print from 1958 until 1997 (when it became web based). *Datamation* is the only periodical to span almost the entire history of the software industry. For its first 30 years, when it was aimed squarely at senior data processing managers, it was the best source on the computer industry; for example, the annual Datamation 100 was perhaps the best survey of the computer industry ever produced. *Datamation* became less useful in the early 1990s, when it began to focus more on low-level technical issues and dropped costly features such as the Datamation 100. It is a loss that future computer historians will mourn.

The trade press has published dozens of titles over the years, journals flourishing for a few years before vanishing into oblivion. There are few complete holdings of any serials other than *Datamation* and *Byte*, and none of them are indexed. However, in the age of the press release, the trade papers often carried the same story, at the same time, in almost the

same words. For the historian, there are few gems in this ephemeral literature that justify the difficulty of excavation.

In the general business press, the best sources on the history of the software industry are *Business Week* and *Fortune*. Alas, neither is indexed, but there are less pleasant occupations than leafing through the pages of these journals in a warm library on a cold afternoon. The long and authoritative articles *Fortune* ran until about 1980 are important sources for historians of the computer industry; since then, *Fortune* has published shorter articles that have proved less useful for the historian. *Business Week* tracked the software industry only intermittently from 1966 through the 1970s. After 1980, however, as software became a significant sector of the economy, articles on it appeared with increasing frequency,. Today it is a rare issue of *Business Week* that does not contain at least one reference to the industry. Newspapers and the financial press have covered the software industry since the software stock boom of 1966–1969. The articles generally are brief and contain little more than financial details; however, newspapers are well indexed, so at least the articles are easy to find.

Business historians traditionally make much use of company archives, which typically contain annual reports, minutes of board meetings, planning documents, product literature, the correspondence of senior officers, and ephemera of various kinds. Usually a company first thinks about creating an archive on the occasion of commissioning a history for a 25th or a 50th anniversary. Few software firms have reached this stage of maturity. Indeed, the only major corporate archive available to scholars is that of SDC, now housed by the Charles Babbage Institute at the University of Minnesota. The SDC Collection covers the period from 1956 to 1981, when SDC was acquired by the Burroughs Corporation. The impressive volume of information in the SDC collection hints at the riches that will become available when software companies organize their archives. For the moment, we can only be patient.[38]

The software industry came into being at about the same time as the industry analyst. The software industry attracted a number of general industry analysts, such as Frost & Sullivan (established in 1961), Creative Strategies International (formed in 1969), and Business Communications (formed in 1971). Several analysts were established in the 1960s and the 1970s to track, exclusively, the computer software and service industries. These include the International Data Corporation (formed in 1964), International Computer Programs (formed in 1967), the Yankee Group (formed in 1970), and INPUT (formed in 1976). And there have been numerous other industry watchers reporting on software, including

Auerbach Publications, Communications Trends, Electronic Trend Publications, the Gartner Group, International Resource Development, and Knowledge Industry Publications.[39]

Industry and market-research reports were produced for subscribers only (primarily software firms and large users), and few have migrated to the public domain. In the last 30 years, thousands of such reports have been published. For example, a catalog of INPUT reports published between 1976 and 1993 lists about 1,500 titles. And INPUT was just one of more than a dozen industry analysts. Of this huge volume of material, no more than 50 reports survive in the Library of Congress. Although so few reports survive, they were indispensable to the writing of this book.[40] It is hard to believe that all this information has simply vanished, but it truly has—few of the analysts who responded to my inquiry reported holding materials more than 10 years old. There are, however, a few bright spots. In the summer of 2000, in time for the final draft of this book, ICP transferred its archives to the Charles Babbage Institute. INPUT is establishing a corporate archive of its 25 years of industry reports. In February 2000, the Software History Center was incorporated (in Benicia, California) with the support of seed money from Computer Associates International. The Software History Center is dedicated to preserving the history of the software industry by ensuring that records of the companies, the individuals, and the events that shaped the industry's growth are preserved and made accessible to anyone seeking to understand how the industry evolved.[41]