Number Crunching without Programming: The Evolution of Spreadsheet Usability

Martin Campbell-Kelly
Warwick University

The first personal-computer spreadsheet, VisiCalc, was launched in May 1979. During the next decade, the spreadsheet evolved from a simple calculating aid to an indispensable tool of modern business, employed by tens of millions of users who had little or no direct computer experience. This article describes the development of spreadsheet usability from VisiCalc through Lotus 1-2-3 to Microsoft Excel.

The role of users in shaping technologies has become a major strand in the history of technology.[1] Users were crucial in the development and sustenance of many important technology platforms. For example, broadcast radio was largely invented by radio amateurs in the years following World War I,[2] and the personal computer was the creation of computer hobbyists in the western United States.[3] Once these technology platforms were commercialized—by users-turned-entrepreneurs as well as by incumbent manufacturers—users continued to shape the technologies by exercising market power. Thus, radio programs, funded by advertising, were selected and shaped by listeners exercising power through their presence or absence in the radio audience.[4] With personal computer software—especially video games and entertainment software—users selected software that provided a rewarding experience in terms of attributes such as aesthetics, function, and price.[5]

This article is a detailed study of how one particular software genre, the spreadsheet program, was shaped by manufacturers responding to users' wants, largely as they were expressed through market preferences. Of course, software producers also engaged in other tactics to keep their users hooked (such as proprietary file formats) much as broadcasters used devices such as serials and cliffhangers to retain their listeners. In this article, however, the focus is on the role of usability in building market share.

At all times in its history, the spreadsheet evolved by increasing its ease of use, broadly interpreted. For example, one of the original motivations of VisiCalc was to save users having to solve problems by writing programs in Basic. In June 1979 Bob Frankston, coinventor of VisiCalc, explained this motivation in a paper to a small audience (consisting mainly of family and friends) at the National Computer Conference in New York. In his paper Frankston observed that, in the 1930s, projections of telephone growth showed that by the 1950s the system would need an unsupportable number of operators; of course, the technology of the dial emerged so that everyone became their own telephone operator. Frankston analogized that computing faced a similar bottleneck—that for computer usage to expand, everyone would have to learn programming. Frankston argued that, like the telephone dial, "VisiCalc made everyone a programmer."[6]

In this article, the usability story is told, without significant loss of generality, by focusing on the dominant spreadsheets VisiCalc, Lotus 1-2-3, and Microsoft Excel (see Table 1).

The making of VisiCalc

The VisiCalc concept predated the personal computer revolution. It was conceived as a user-friendly financial analysis tool, and its design was not "accidental" but was deeply informed by computer science knowledge and financial practice.[7] It was designed and implemented by two graduate computer scientists, one of whom was also an MBA student at the time. Although conceived in a world

 Published by the IEEE Computer Society 1058-6180/07/$25.00

Table 1. Market-leading spreadsheets.

Software Product	Date Released	Retail Price
VisiCalc	May 1979	$249
Lotus 1-2-3	Jan. 1983	$495
Lotus 1-2-3 release 2	Sept. 1985	$495
Microsoft Excel 2.0	Oct. 1987	$395
Lotus 1-2-3 release 3	June 1989	$595
Microsoft Excel 3.0	May 1990	$495

where time-sharing was the dominant mode of online computing, when the Apple II personal computer arrived, function perfectly fitted form.

The primary inventor of VisiCalc was Dan Bricklin. Bricklin came from an entrepreneurial background, his grandfather and father both having run small printing shops.[8] In 1969 he enrolled at the Massachusetts Institute of Technology (MIT) as a math major and also took classes in computer science, in which he excelled. During his time at MIT, he wrote an online calculator program for the Multics time-sharing system which was then under development, and also worked on an APL (A Programming Language) compiler. At this time, he met up with Frankston with whom he formed a lasting friendship. Their relationship matured into something like a mildly bickering marriage—Bricklin dominant, talkative, and intuitive; Frankston reflective, thorough, and occasionally nettled. After graduating in 1973, Bricklin became a programmer with the Digital Equipment Corporation (DEC) in Maynard, Massachusetts. There he worked on a typesetting system and then became project leader of what was later sold as the WPS-8 word processing system, which ran on the PDP-8 minicomputer. The word processing system was designed for relatively naïve users, such as secretaries, and had a simple intuitive interface—several features of which (such as the ''status line'' and ''scrolling'') found their way into VisiCalc. In 1976, Bricklin left DEC because it was relocating to New Hampshire, and he obtained a senior programming job with a small manufacturer of point-of-sale terminals—again honing his usability insights with inexperienced users.

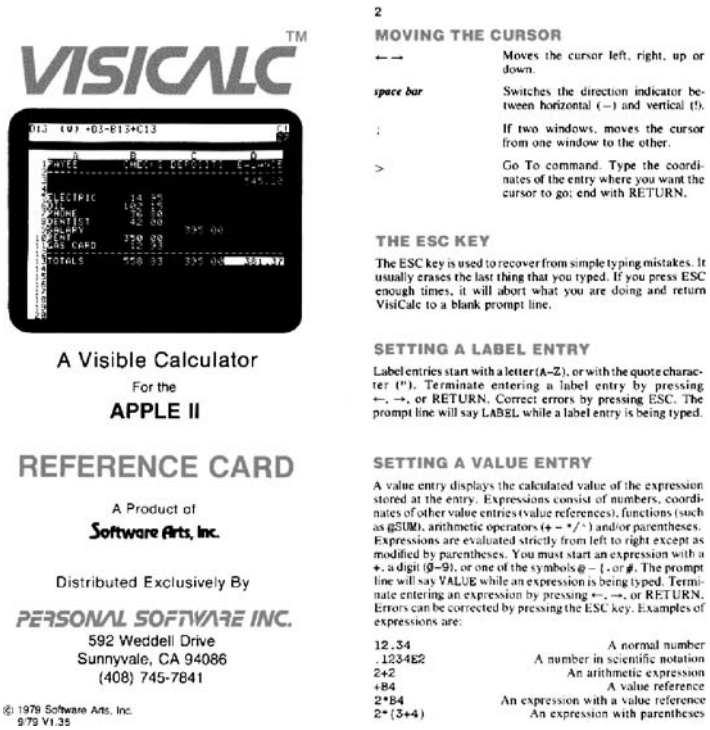
In 1978, Bricklin enrolled in the Harvard University MBA program, with the aim of equipping himself with a wider skill set than programming—he did not want to suffer the same fate as many programmers he had encountered who had become unemployed in their fifties. In the course of the MBA, students routinely had to make financial projections for which they had access to Basic on a PDP-10 time-sharing system. Not surprisingly, given his background, Bricklin went a little further and devised a rudimentary spreadsheet program. For his thesis project he developed a fully functioning system. The Harvard time-sharing system had addressable CRT-terminals, which gave much the same facilities as an early personal computer, except for the latter's instant responsiveness.

Bricklin showed the program to classmates and faculty, and their reactions were encouraging. These interactions provided Bricklin with significant user insights. For example, a finance professor explained the necessity of including financial functions such as Net Present Value (NPV). Another emphasized ''Whatever you do, your competition is the back of the envelope. Senior executives don't do this stuff. You've got to make it simpler, simpler.''[9] One professor cautioned that there were already many financial packages for mainframe and time-sharing computers on the market. Another professor, however, was aware of the fast-breaking personal computing industry and introduced him to Dan Fylstra, who had just started a software publishing venture called Personal Software.

Fylstra was another Harvard MBA graduate (class of 1977), and his thesis project had been a business plan for a personal computer software publishing company ''in much the same way that books and records are produced and marketed.''[10] Fylstra was exceedingly well informed about personal computers, whereas Bricklin and Frankston had barely encountered them. Fylstra had been an associate founding editor of *Byte* magazine in 1977 while still at Harvard, and was a founding editor of *Computer Dealer*, a trade publication for the emerging personal computer retail stores.[11]

Fylstra had established Personal Software in early 1978, operating in the classic manner of a software publishing startup of the period—duplicating cassettes in his apartment and stuffing them into postal envelopes. A program called Microchess proved to be a best-seller in a stable of also-rans—an early indication of the winner-takes-all nature of the personal computer software industry (rather like books and records).

Bricklin and Fylstra first met in the spring of 1978. It was Bricklin's first exposure to personal computers, and he immediately saw the



VISICALC™

A Visible Calculator
For the
APPLE II

REFERENCE CARD

A Product of
Software Arts, Inc.

Distributed Exclusively By
PERSONAL SOFTWARE INC.
592 Weddell Drive
Sunnyvale, CA 94086
(408) 745-7841

© 1978 Software Arts, Inc.
9/79 V1.35

2

MOVING THE CURSOR

←→ Moves the cursor left, right, up or down.

space bar Switches the direction indicator between horizontal (–) and vertical (!).

: If two windows, moves the cursor from one window to the other.

> Go To command. Type the coordinates of the entry where you want the cursor to go; end with RETURN.

THE ESC KEY

The ESC key is used to recover from simple typing mistakes. It usually erases the last thing that you typed. If you press ESC enough times, it will abort what you are doing and return VisiCalc to a blank prompt line.

SETTING A LABEL ENTRY

Label entries start with a letter (A–Z), or with the quote character ("). Terminate entering a label entry by pressing ←, →, or RETURN. Correct errors by pressing ESC. The prompt line will say LABEL while a label entry is being typed.

SETTING A VALUE ENTRY

A value entry displays the calculated value of the expression stored at the entry. Expressions consist of numbers, coordinates of other value entries (value references), functions (such as @SUM), arithmetic operators (+ – */ ^) and/or parentheses. Expressions are evaluated strictly from left to right except as modified by parentheses. You must start an expression with a +, a digit (0–9), or one of the symbols @ – (. or #. The prompt line will say VALUE while an expression is being typed. Terminate entering an expression by pressing ←, →, or RETURN. Errors can be corrected by pressing the ESC key. Examples of expressions are:

12.34	A normal number
.1234E2	A number in scientific notation
2+2	An arithmetic expression
+B4	A value reference
2*B4	An expression with a value reference
2*(3+4)	An expression with parentheses

Figure 1. VisiCalc reference card, 1979. (Courtesy of Dan Bricklin and IBM.)

potential of high-volume sales to desktop computer users, compared with the $5,000-plus professional financial analysis packages that sold in small numbers to businesses. He was aware that his software would have to be exceptionally user-friendly and intuitive in order to eliminate the need for costly support.

Fylstra loaned Bricklin and Frankston an Apple II computer, and Frankston started programming in earnest in November 1978. The Apple II had no development software apart from Basic, and an application program written in Basic would have been much too slow and lacking in responsiveness. VisiCalc was therefore developed in 6502-assembler on a commercial time-sharing system, and the resulting object code downloaded to the Apple II. The program would then be debugged. During this development period the closely coupled interaction between machine and user was perfected, and this would be one of the overwhelming advantages of a personal computer over a sluggish time-sharing system. At the same time. Bricklin wrote the user documentation and a reference card (see Figure 1). During this period the program was constantly refined so that features that were non-intuitive or could not be clearly communicated in the documentation were dropped. The reference card further constrained VisiCalc from incorporating features that could not be concisely explained.

In January 1979, Bricklin and Frankston incorporated their operation as Software Arts and funded the venture with a sum of $20,000 each plus a bank loan of $65,000—the latter funded a Prime 550 minicomputer for software development. Incidentally, one aspect of personal computer history often overlooked is that early software development for consumer machines needed access to a commercial time-sharing system or an expensive minicomputer. This was a significant barrier to entry into personal computer software during the brief period that VisiCalc dominated its market.

VisiCalc was launched at the May 1979 Comdex at a price of $99 and sold rather slowly at first, at a rate of about 1,000 copies per month. It is often said that VisiCalc was the "killer application" that launched the personal computer revolution. While this assertion may be overstated, VisiCalc certainly helped to legitimize the business acceptance of personal computers. VisiCalc was often sold to professional users as a "turn-key" system, where a computer, a printer, and VisiCalc were sold as a package. VisiCalc was never a true consumer item. Encouraged by reasonable sales to business users, Fylstra raised the price to $249 on the grounds that "the more you raise the price, the more you sold."[12]

The IBM personal computer had not yet emerged as an industry standard, and the normal industry strategy—followed by Microsoft and others—was therefore to target the same application program on all the top selling machines. Software Arts produced versions of VisiCalc for Tandy, IBM, HP, DEC, Sony, and TI computers during the period 1980–1982. An improved edition (VisiCalc Advanced Version) was also developed for the Apple computer in early 1981. By the end of 1982, some 475,000 copies of VisiCalc had been sold, and the monthly run rate was 14,000 copies.[13]

In August 1981, IBM introduced its first personal computer, and during the next three years it came to dominate corporate sales. A version of VisiCalc was provided for the IBM PC from the first day of its introduction and it sold very well. But in January 1983, a new spreadsheet with many more features, Lotus 1-2-3, provided the first real competition for VisiCalc.

The rise of Lotus 1-2-3

The Lotus Development Corporation was founded by the developer and entrepreneur Mitch Kapor. Its first product, the 1-2-3 spreadsheet, was developed by Jonathan Sachs, with Kapor's architectural oversight. Like VisiCalc, 1-2-3 was no accident: Kapor and Sachs were both seasoned computer professionals well placed to exploit the emerging IBM PC.

Kapor had, thus far, led a checkered career.[14] His exceptional mathematical ability took him to Yale University at the age of 16, but he took several years to mature socially, drifting in and out of transcendental meditation, a period as a disc jockey, and taking a master's course in psychological counseling. He enrolled for an MBA at MIT's Sloan School of Management in 1978. In July 1978, he bought an Apple II computer and it was a turning point in his life. He left the Sloan School one term short of graduation in 1980 to leap into the burgeoning personal computer industry.

While still at the Sloan School, Kapor had developed two complementary software programs for VisiCalc that were published by Personal Software in April 1981. These ''add-on'' programs, VisiTrend and VisiPlot, respectively performed statistical analysis on a spreadsheet and represented a spreadsheet graphically. The concept of complementary add-on products was to prove a very important aspect of Lotus 1-2-3's development in the mid-1980s.

VisiTrend and VisiPlot sold very well, and Kapor derived a generous royalty rate of 33 percent—about two or three times the going rate. By 1982 he had accrued royalties of $600,000 and sold the perpetual rights to VisiCorp for $1.2 million. After putting money aside for taxes and an amount sufficient to guarantee his future financial security, he decided to use $300,000 to start a software venture, initially called Micro Finance Systems. He obtained further funds from the Sevin Rosen Venture Fund. Ben Rosen, a founder of Sevin Rosen, had been an early evangelist of spreadsheets and was familiar with Kapor's work. Kapor found he was pushing at an open door. He obtained $1 million first-round funding, and the name of the firm was changed to the Lotus Development Corporation.

Kapor decided that Lotus Development would compete in the spreadsheet market, with which he was personally familiar. Although Kapor specified the 1-2-3 program, he left development to Jonathan Sachs. Sachs, after graduating in math from MIT in 1970, had spent several years as a systems programmer at MIT and in the local minicomputer industry.[15] In 1980 he formed a contract programming partnership, and one of his assignments was to write a VisiCalc clone for a Data General minicomputer. In the summer of 1981 Kapor learned of Sachs' spreadsheet experience, and invited him to work for Lotus.

At this time the big idea in personal computer software was product integration, all-in-one software packages that would serve all of a user's needs.[16] Up to that point, users had had to use discrete programs for each of the four main information processing task areas—word processing, spreadsheets, maintaining simple databases, and ''graphics'' (a catch-all for drawing packages and the visual presentation of data). There were market leaders such as WordStar, VisiCalc, dBase II, and Harvard Graphics in each of these niches. The need to switch between several different programs was a major impediment to ease of use. For example, to incorporate an analysis of information stored in a database in a printed report, a user would likely need to use all four key application programs and manage the tricky business of transferring data between them. The long-term solution to this problem would be graphical user interface (GUI)-based operating systems such as Microsoft Windows or IBM's OS/2, but personal computers were not yet powerful enough to support a multitasking operating system. Although Lotus 1-2-3 was positioned to compete in the market for integrated software, its genius was that it got the balance between interactivity and integration exactly right, at exactly the right moment in time. A year earlier or later, the niche that 1-2-3 occupied might not have existed.

Initially, the plan was that Lotus 1-2-3 would incorporate three major applications: a spreadsheet, graphical presentation (a legacy of Kapor's VisiTrend and VisiPlot), and a word processor. Sachs was working on the word processing function of the program when a powerful new integrated program, Context MBA, arrived on the scene. Sachs evaluated Context MBA and obtained two key insights. First, the product ran very slowly, and second, the database feature was more useful in practice than word processing, which Sachs was in any case finding difficult to implement. So database became the ''3'' in 1-2-3. Informed by Context MBA's sluggishness, he made 1-2-3 as responsive as possible. For example, he considered writing it in the high-level language C instead of assembly language, but

found the result was several times worse in speed and size. He also decided to write 1-2-3's display directly to the screen, bypassing the normal screen writing protocols. This affected long-term compatibility but gave 1-2-3 a superior performance that won many sales. It was a crucial understanding that users valued responsiveness above almost any other feature.

One of the most important innovations in 1-2-3 was a macro facility. A macro enabled a set of keystrokes to be stored in a spreadsheet cell. Activating the macro then caused the keystrokes to be replayed. This simple idea enabled users to automate repetitive tasks. Sachs would have preferred to use a more traditional programming method (of the kind that Visual Basic for Applications now represents), but the macro scheme did have two great advantages. First, it was easy for users because it required almost no learning. Second, it was easy to implement because it required little more than a tweak of the basic spreadsheet engine. Macros became a very important part of the 1-2-3 culture: macros were featured in articles in magazines and textbooks, and a few small firms provided macro templates. The potential disadvantage of macros, of which Sachs was fully aware at the time, was that it would never be possible for the menu structure to be changed because of the need to maintain backward compatibility of macros. In fact, this was perhaps a blessing in disguise, because stability of the user interface turned out to be of critical importance to locking in users. Most famously, the leading word processor WordStar changed its user interface in 1985, and users, faced with the necessity of learning to use a new program, defected to other products in large numbers.[17]

A great deal of the 1-2-3 usability development effort addressed the need to provide user support in the absence of a traditional in-house IT support unit. The larger capacity of the IBM PC enabled an online help system to be implemented. Sachs also developed an infrastructure that could run a tutorial system which gave new users a tour of the system and took them through worked examples. Like VisiCalc, 1-2-3 included a 500-page user manual that was superbly written, although textbook-like and intimidating. The way one actually learned about spreadsheets was by word of mouth and from enthusiasts inside companies.

As a second mover, Lotus benefited from the existing market created by VisiCalc and well-established distribution channels. Launched at Comdex in October 1982 with a retail price of $495, 1-2-3 took $900,000 in advance orders. Kapor was an instinctive promoter and used a great deal of Lotus Development's start-up funds to create a publicity splash reported to have cost $2.5 million.[18] Lotus 1-2-3 was released in January 1983. At the same time, he hired Chris Morgan, former editor in chief of *Byte*, as director of communications. One outcome of Morgan's appointment was *Lotus* magazine, launched at a cost of $1.5 million in May 1985.[19]

Another scheme to diffuse 1-2-3 was the promotion of user groups. In 1985 Lotus introduced a "development kit" for user groups, and groups were soon established in most US cities. At user-group meetings individuals would share applications and trade techniques and macros. The process was similar to early user groups such as SHARE, but the membership extended far beyond computer professionals.[20] Lotus also invested in the training of dealers. Graduates of the Lotus training course could become "authorized dealers." Authorized dealers valued 1-2-3 not only because of its generous markup (about 100 percent), but also because it could lead to sales of fully equipped $5,000 IBM PCs, and to a lucrative sideline in training users. By mid-1985, there were two dozen approved training programs for Lotus 1-2-3 around the nation.

Lotus 1-2-3 was phenomenally successful. By mid-1985—after just 18 months—it had sold 850,000 copies and captured 60 percent of the spreadsheet market. Lotus Development had become one of the fastest-growing firms in history. With 900 employees, cash reserves of $70 million, and 1985 revenues of $226 million, it was rivaled only by mainframe software vendor Computer Associates ($228 million revenues); Microsoft was only half the size of Lotus Development.

Lotus Land: Add-ons and add-ins

Lotus Development's profits derived entirely from 1-2-3. This was a program of perhaps 20,000 instructions, the creative endeavor of a few months by its single author Jonathan Sachs, and which would be very easy to clone.

Hence, after managing the operational difficulties of extremely rapid firm growth, the most critical problem for Lotus Development was finding successor products, and eventually a stream of products, to reduce its dependency on 1-2-3. This it never succeeded in doing. In October 1984, Kapor assumed responsibility for product strategy, and passed

operational control to Jim Manzi who became the long-term president and CEO. Kapor took on the role of "nurturing new products,"[21] and used Lotus Development's $70 million reserves to create a flurry of ill-judged products, only one or two of which were profitable, and none remotely rivaled 1-2-3. Then, as now, no one understood the chemistry that went into creating a category-killer product.

The most important new product was Symphony, an integrated program intended to replace Lotus 1-2-3 on the grounds that if 1-2-3 was good, then 1-2-3-4-5 would be better.[22] Lotus also developed an integrated package, Jazz, for the Macintosh computer (see later). Such integrated products were a holy grail of the industry, but the market rejected them all—from VisiCorp's VisiOn to Ashton-Tate's Framework. Sluggishness, user reluctance to switch products, and complexity were all contributory factors. However, buoyed by the income stream from 1-2-3, Lotus Development could afford many product failures.

It was only after the lukewarm response to Symphony that the code base of 1-2-3 was dusted off and "release 2" shipped in September 1985. Although release 2 had some marginal improvements in terms of functional performance, the user experience and interface were essentially identical. This turned out to be fortuitous because 1-2-3 now acted as a stable platform that could support many complementary products. These products constituted a software ecosystem that defined the evolutionary direction of the spreadsheet.

Complementary products—usually called add-ons or add-ins in the 1980s and sometimes called plug-ins today—are a relatively unstudied aspect of the software industry. After-markets for complementary products developed very early in the personal computer software industry. Kapor's VisiTrend and VisiPlot were typical examples. These early products were not directly integrated with VisiCalc, but were separate "add-on" programs that extracted data from a VisiCalc file and performed statistical analysis or visual presentation. Vendors of major products such as Ashton-Tate's dBase and Autodesk's AutoCad actively encouraged developers to create complementary products. Typically, these products were application templates that customized the host program for a particular market niche. Lotus 1-2-3 likewise had templates for particular industries, such as manufacturing or tax preparation. Templates consisted of prewritten spreadsheets into which users plugged data values.[23]

Figure 2. Funk Software's Sideways, c. 1986. (Courtesy of Paul Funk and Juniper Networks.)

In the case of 1-2-3, however, complementary products served not only narrow application niches, but also enhanced the product for large market segments by augmenting the functional behavior of the system. Easily the most successful early add-on product of this type was Funk's Sideways, launched in 1985, which enabled a spreadsheet to be printed sideways on a tractor-feed printer of the period (see Figure 2).[24] This deceptively simple program enabled users to avoid the use of paste and scissors when dealing with large spreadsheets. By 1988 Sideways had sold some 500,000 copies, which meant that at least one in 10 Lotus 1-2-3 users had acquired a copy.[25]

Funk Software, founded by Paul Funk in 1982, was located on Third Street in Cambridge, Massachusetts, just a few blocks from Lotus Development's headquarters on First Street. This geography hints at a regional development aspect to the software industry,

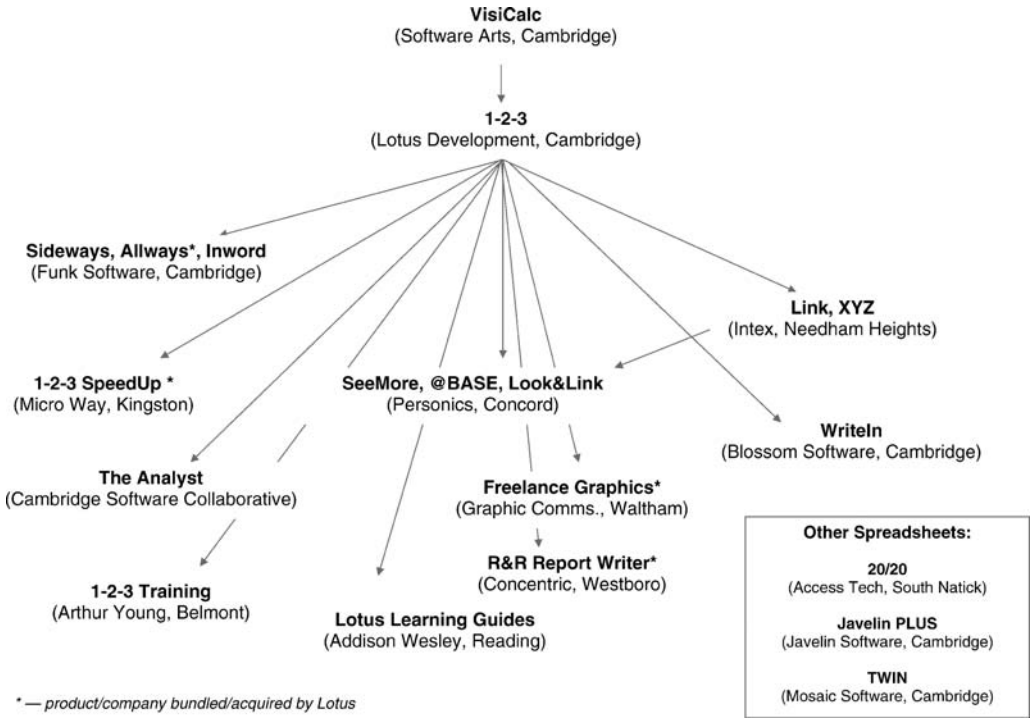


Figure 3. Regional geography of the Massachusetts spreadsheet industry.

which has been little studied. The early development of spreadsheets was very much a Cambridge phenomenon. Figure 3—which is illustrative rather than exhaustive—shows the cluster of Massachusetts firms in the 1980s spreadsheet industry. Thus, VisiCalc gave rise to 1-2-3, and 1-2-3 in its turn encouraged clones (such as Mosaic Software's "The Twin"—see later). Sometimes there was a direct connection with one of the pioneer firms but in other cases spreadsheets were simply in-the-air, and they were not difficult to imitate. Lotus 1-2-3 also encouraged a constellation of add-on makers.

Several of the add-on firms had direct or indirect links to Lotus. For example, Blossom Software—the creator of a word processing add-in, WriteIn—was founded by two former Lotus employees, a marketing executive, Jim Kinlan, and a principal software engineer, Eric Shultz.[26] When Blossom Software folded, Kinlan moved on to Funk Software. Funk Software made use of informal links with developers inside Lotus to gain the necessary knowledge to turn Sideways from being an add-on to becoming a fully integrated "add-in."

Initially, Lotus was ambivalent or even hostile to add-ons and add-ins, seeing them as parasitic. However, the company gradually came to appreciate that complementary products had the effect of extending the base product in myriad ways that no one firm could achieve. They also made users less likely to switch to competing spreadsheets for which their stock of complementary products would likely no longer function. In early 1985, Lotus created a Development Services Department, which provided telephone support for add-in developers and established a CompuServe special interest group.[27]

One of the most important enhancements of release 2 was to systematize and expose the programming hooks by which developers could integrate their add-ins (the term *application program interface*, API, was not yet used). An add-in development kit was released in spring 1986, which included a set of software tools and a roadmap of 1-2-3. It cost a highly subsidized $150. Lotus Development also published the file formats for all of its products.[28] In March 1986, the company organized its first add-in developer conference at the Hyatt Regency in Cambridge, which was attended by 400 would-be developers. Two months later the company announced a free-of-charge Registered Developer Program "to recognize and support independent after-market developers."[29]

From that point on the development of add-ins no longer required inside knowledge

from Lotus, and the number of products and firms developing them simply exploded. Jeff Tarter, the publisher of the *SoftLetter* industry newsletter, called this happy community ''Lotus Land.'' By December 1987, *Lotus* magazine claimed that over 1,000 add-in products had been created by 500 firms.[30]

The vast majority of add-in products failed in the market. But in the shaping of the spreadsheet they all played a role in a kind of Darwinian evolution—a primordial swamp in which the market selected out the most useful add-ins, and allowed the others to die. The most successful and persistent add-in categories are listed in Table 2. *Cell annotators* enabled explanatory comments to be attached to a cell. *Workbooks* enabled 1-2-3 to open up several spreadsheets at the same time. *Auditing* packages facilitated spreadsheet debugging by revealing errors such as circular definitions. *Advanced charting* packages enabled the production of high-resolution charts with color and 3-D effects. *Publishing* add-ins enabled the production of type-set quality reports that could take advantage of the new laser printers. *Zoom* add-ins enabled a spreadsheet to be viewed with different degrees of magnification. All of the products listed in Table 2 sold many thousands of copies.

By the end of 1986, Lotus had a very definite vision of 1-2-3 as a technological system—a platform consisting of the spreadsheet ''engine'' that could support ''an entire system of products'' (see Figure 4).

Table 2. Prominent 1-2-3 add-ins.

Category	Leading Products
Cell annotators	SmartNotes (Personics)
	NoteWorthy (Funk Software)
Workbooks	Look&Link (Personics)
	X-Y-Z (Intex Solutions)
Auditing	The Analyst (Cambridge Software Collaborative)
	The Auditor (Consumers Software)
Advanced charting	GraphWriter (Lotus)
	Freelance (Graphic Communications)
	Chart (Microsoft)
Publishing	Allways (Funk Software)
	Impress (Aleph 2)
Zoom utilities	SeeMore (Personics)
	WideView (Spies Laboratories)

Almost as important as the add-in products that succeeded, the categories that failed served to define the features that users did not value. For example, there were several word processing add-ins, including Blossom Software's WriteIn, all of which were market failures. It turned out that users familiar with WordStar or WordPerfect did not value a rudimentary word processor integrated into a spreadsheet. Again, several relational database products were produced by leading firms

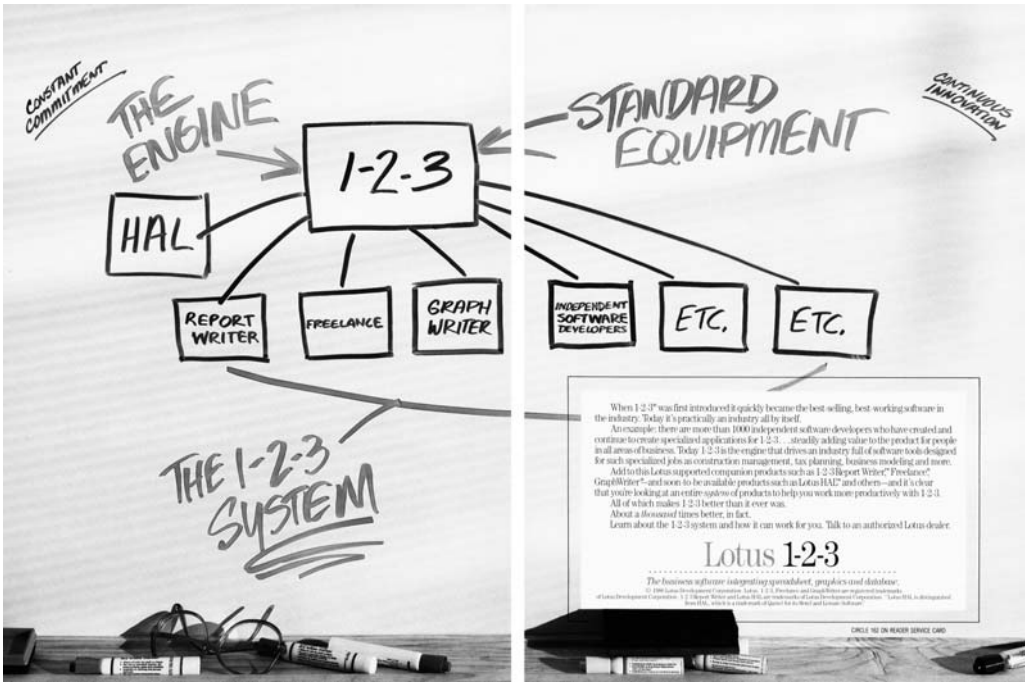


Figure 4. Lotus 1-2-3 as a technological system. (Courtesy of IBM.)

Figure 5. Lotus 1-2-3 clones. (Courtesy of Eleanor Robson.)

including Informix, Oracle, and Computer Associates, and all of them were market failures.

Lotus also developed or acquired add-in products. The most interesting, and the biggest failure, was a product launched in 1987 called HAL (named for the intelligent computer in the movie *2001: A Space Odyssey*). HAL was developed by a Californian startup, GNP Development Corporation, which Lotus acquired in January 1986. HAL enabled a user to specify cells and commands in English instead of symbolic formulas. For example, the user could write "MOVE THIS ROW TO ROW 25." Lotus commissioned extensive usability studies in which respondents were enthusiastic about the product.[31] However, it was a resounding market failure and was allowed to wither on the vine.

Usability and the law

In 1987, following 1-2-3's meteoric success, a number of clone products came on the market. Three of these—Paperback Software's VP Planner, Mosaic Software's The Twin, and Borland's Quattro (see Figure 5)—were such a threat to Lotus Development that it filed lawsuits for intellectual property infringement. The *Lotus vs. Borland* dispute went all the way up to the US Supreme Court and was the most important software case up to that date.[32] User interfaces and menu structures were at the heart of all of these proceedings.

Since about 1965, the US courts had held that software could be copyrighted, both in source and object form. In theory, copyright offered two types of protection. First, it made illegal the unauthorized copying of a software product (later called "piracy"). Second, it prevented the direct copying of a software maker's code by a competitor. In practice, however, copyright offered little protection against either piracy or imitation.[33]

In the case of piracy, the fact that copying was illegal did not prevent many users from making and using illicit copies of commercial software products. Several technological means were devised to prevent such copying. Lotus Development adopted the most widely used technique of ensuring its disks could not be duplicated on an ordinary PC, and requiring that the master disk containing the program be inserted in the disk drive for the program to function (even if the program itself was stored on a hard disk drive). This form of software protection was a public relations fiasco, because it impugned the honesty of legitimate users and also made it impossible to use a legitimate copy of the program in the not uncommon event that the master disk became unreadable. Lotus Development abandoned copy protection mechanisms in release 2. In effect, a degree of piracy became built into the cost structure of PC software (although business software was in any case much less vulnerable to piracy than consumer software and video games).

For Lotus Development, the source code of 1-2-3 was its "crown jewels" and it was a closely held trade secret, notwithstanding the provisions of copyright law. Indeed, the APIs published for third-party developers can be viewed as a mechanism for avoiding the otherwise necessary disclosure of source code.

In order to clone 1-2-3, imitators such as Paperback Software, Mosaic Software, and Borland "reverse engineered" rather than copied the program. This process involved observing the behaviors of the program and writing a new program with identical behaviors. Until software patents got the green light in the early 1980s, software makers were powerless against most forms of imitation by reverse engineering. Incidentally, Dan Bricklin had taken advice about patenting VisiCalc in 1978, but the advice that his attorney gave him was that an application for a patent was unlikely to be successful.[34] In retrospect this was not good advice—although this judgment of hindsight has to be tempered by the fact that the software patent environment was far from certain at that time, and, of course, no one could have predicted that VisiCalc would become one of the killer applications that set the personal computer train rolling. Had

VisiCalc been patented, one can speculate that the history of the software industry might have unfolded considerably differently. At all events, it was completely legitimate to write and market spreadsheet programs, even though they were closely modeled on VisiCalc and replicated its algorithms. Indeed, this was how 1-2-3 had itself been created.

One of the important advantages of a first-mover in any software category was the uniqueness and complexity of its user interface. Having, with some difficulty, mastered one product, users were unlikely to defect to a competing product, even if it was cheaper and better. The three 1-2-3 clones, VP Planner, The Twin, and Quattro all adopted the tactic of making their user interfaces compatible with 1-2-3 in order to make product switching easier. Indeed, VP Planner—which sold for $99 compared to 1-2-3's $495—was keystroke-for-keystroke compatible with 1-2-3. Paperback Software was the creation of Adam Osborne (also the founder of Osborne Computer), and his strategy was to develop software packages in the major categories for sale in regular bookstores, at much lower prices than through conventional computer retailers.

In January 1987, Lotus Development filed suit against Paperback Software and the case was heard in the District Court of Boston. Lotus Development's legal team made use of a somewhat obscure body of law relating to the ''look and feel'' of commercial products. There had been a few previous cases, although this was the first in software. For example, in 1970 in *Roth Greetings Cards vs. United Card Co.*, Roth had successfully won a suit against an imitator that published ''substantially similar'' greetings cards. There had also been cases involving knock-off TV quiz show formats.

Lotus Development argued that, although VP Planner was not visually identical to 1-2-3, it had copied its menu structures. In its defense, Paperback Software asserted that VP Planner had originally been developed with its own distinctive menus, but by the time it came to be released, 1-2-3 had such a lock on the market that VP Planner had to be compatible to induce users to switch products. The Court decided in Lotus Development's favor, and VP Planner was withdrawn from the market. In light of this decision, Mosaic Software also pulled The Twin from the market before its case came to Court.

This was a controversial decision because it implied that software products would have to have perversely different user interfaces to avoid look-and-feel suits, whereas usability was enhanced by having a consistent user interface across applications. It was analogized that if this principle were applied to automobiles, for example, then safety would be compromised and the rental car industry would be unworkable.

The suit against Borland had a different outcome. Borland's Quattro spreadsheet came with two user interfaces from which the user could select. The primary interface was entirely distinct from 1-2-3, both in its visual appearance and its menu structure. The second interface, however, though visually distinctive, used the 1-2-3 menu structure, so that 1-2-3 users could easily switch into the new product. Lotus Development won its case, but this was reversed on appeal by Borland. Lotus Development took the case to the Supreme Court where the reversal was upheld in 1996. During the appeal process, Quattro had remained on the market and by 1990 had a 20 percent market share compared with 1-2-3's 50 percent.[35]

As is so often the case with legal decisions, however, by the time *Lotus Development vs. Borland* was settled in 1996, the industry had already moved on. With the rise of the GUI in the Macintosh and Windows-based PCs, a common user interface became standard industry practice.

The rise of Excel

In 1980 Microsoft began an ''electronic paper'' project—later called Multiplan—to compete with VisiCalc and its most successful clone, SuperCalc.[36] Like those products, Microsoft's strategy was to develop a program that could be deployed on many makes of computer with minimal reprogramming.

Multiplan was written by Charles Simonyi to an original specification produced by a consultant, Paul Heckel, whose innovations included the R1C1-type notation in preference to the ''battleship'' (A1-type) notation used in VisiCalc. Simonyi's background included several years at Xerox PARC where he had worked on user interfaces; Multiplan incorporated some of these ideas, and they would eventually feed into Excel.

Multiplan was released in the fall of 1982 for MS-DOS, CP/M, and Apple machines; it received excellent reviews, including an *InfoWorld* software-of-the-year award. It was, however, outclassed by the superior speed and capabilities of Lotus 1-2-3, which was released a few months later and quickly dominated the

category. Microsoft made several further releases of Multiplan, up to version 3 in 1987, but the program never rivaled 1-2-3 for IBM-compatible PCs. Multiplan did, however, do well in some niches where 1-2-3 did not compete, particularly in some European markets and on the Apple Macintosh.

The Macintosh was designed from the ground up as a machine with outstanding ease of use, drawing extensively from GUI developments at Xerox PARC. In the summer of 1981 Microsoft made an agreement with Apple to develop Multiplan and various other programs for the new machine. Multiplan was released alongside the Macintosh in January 1984 and it dominated the spreadsheet market for that machine. Excel grew out of that experience.

Although successful on the Macintosh, Microsoft was much more interested in the far bigger IBM-compatible PC market. In 1984, Lotus Development secured more revenue from 1-2-3 on IBM PCs than Microsoft did from its entire range of software. Microsoft began development of an IBM PC spreadsheet (later to be called Excel) in fall 1983. Rather than compete head-on with 1-2-3, Microsoft decided to initially develop the spreadsheet for the Macintosh platform (convinced, also, that the future lay with GUIs). Excel was launched for the Macintosh in May 1985, shortly before Lotus announced its integrated package Jazz. Like other integrated packages, Jazz was a market failure, and then it was too late for Lotus to recover its position.[37] Microsoft had consciously avoided an integrated package because such software was generally considered difficult to learn and none had succeeded up to that date. Excel thus filled a spreadsheet vacuum on the Macintosh that might otherwise have been filled by 1-2-3. During 1986, Excel sold 160,000 copies compared with Jazz's 10,000.[38]

Excel benefited from being developed in the relatively sheltered Macintosh market and for a platform on which usability features (such as print preview and zoom) were relatively easy to implement. Such features would not have been supportable on a conventional IBM-compatible PC. By 1986, however, with the arrival of the 386 processor, the prospects of running more processor-intensive programs were improving. In October 1987, Microsoft launched Windows Version 2.0 with Excel as a complementary application (although Excel would also run without Windows if users preferred).

Besides benefiting from the experience of developing GUI-based applications for the Macintosh, Excel was also heavily informed by user research. Microsoft worked with five companies (including Boeing and Arthur Andersen) which were heavy 1-2-3 users and agreed to act as test sites for Excel. These users highlighted some important usability issues. For example, file compatibility with 1-2-3 was viewed as essential to facilitate product switching. It was also essential that Excel should be able to interpret 1-2-3 macros without modification.

Excel 2.0 was aimed at high-end machines using a 286 or better processor. Microsoft did not expect to outsell 1-2-3 in the short term, but rather was consciously positioning its more user-friendly software for a time when a standard IBM-compatible PC would support Mac-like interfaces:

> We are heading into a technology transition, and over the long term we have a major opportunity. ... We think this is a case where a competitor's strength is also its greatest weakness. In the case of Lotus, the strength is in an interface that users are very used to. But the world is moving towards a graphical interface. And Lotus is hoping that people won't want to switch.[39]

The press reception of Excel was enthusiastic. Like the launch of the Macintosh in 1984, it was unmistakably a glimpse of the future. Microsoft spent ''more than any other product in its history'' on publicity, including lavish ''Soul of the New Machines'' advertisements, in apparent homage to Tracy Kidder's best seller. A promotional video with very high production values captured perfectly the environment into which Excel would fit.[40] Excel 2.0 quickly achieved a respectable and profitable 10 percent market share. This was a notable achievement, given Lotus 1-2-3's market dominance.

Excel opened up a huge usability gap with Lotus 1-2-3. Although 1-2-3 was in its second release, the user experience was essentially the same as with the original program of January 1983. A third release had been under development since 1986, but instead of evolving the existing code base, Lotus had decided to redevelop the entire product, and it was now mired in a classic software engineering crisis.[41]

Although in the long term, Excel's GUI represented a feature gap that would have to be bridged by Lotus 1-2-3, this was not a critical factor in 1987. Existing 1-2-3 users were very familiar with the 1-2-3 interface, and most

machines were not in any case sufficiently powerful to run Excel. Much more important was the fact that Excel could effortlessly produce typeset-quality reports, using the laser printers that had just come onto the market.

In the words of a Lotus historian, it was a case of "add-ins to the rescue."[42] In 1988 Funk Software introduced a publishing add-in, Allways, which enabled the production of laser-printed reports of comparable quality to Excel. Funk's Allways was a life-saver for Lotus Development. The company's developers had been focused on power features for release 3, and were taken by surprise by the popularity of the spreadsheet publishing tools in Excel 2.0. In mid-1988 Lotus cut a deal with Funk Software to bundle Allways with each of the 100,000 spreadsheet programs sold each month.

Lotus 1-2-3 release 3 was finally shipped in June 1989, but it was so loaded with features that it "could barely run on the 286-based machines that were still standard in many businesses."[43] In order to generate sales of 1-2-3 on low-power machines, during 1989 to 1991 Lotus released two upgrades of release 2. For all of its sluggishness, release 3 contained relatively few usability enhancements, and some catching up was made in a new release in September 1990. Even then, there were no integrated laser-printing facilities, and so the new release came bundled with an Allways-like add-in, Impress, which Lotus had acquired from Aleph 2, a firm that it had recently taken over. The several incremental releases of 1-2-3 and bundled add-ins created considerable market confusion.

Lotus Development might well have sustained its market dominance had it not bet on the OS/2 operating system. Lotus Development had sophisticated and rational product planning processes, and it had released 1-2-3/G to run under OS/2 in early 1990. However, in May 1990 Microsoft shipped Windows 3.0, which received overwhelming market acceptance. Microsoft's Excel 3.0 (a modest enhancement of the 1987 2.0 release) filled the spreadsheet vacuum for the new platform. Lotus eventually released 1-2-3 for Windows in August 1991, but it was a lackluster product that was poorly tuned to the Windows platform. Lotus 1-2-3 remained the dominant spreadsheet on DOS-based PCs for several years, but during the first half of the 1990s, as Windows began to replace DOS, its market share slipped from 70 percent to 20 percent, while Excel's grew from 10 percent to 70 percent.[44]

Conclusion

To read the contemporary press, the problems of Lotus Development at the beginning of the 1990s make the company appear confused and unfocused—a rabbit caught in the headlights of the GUI juggernaut. Certainly the flurry of incremental upgrades gave the impression of band-aids being applied to a product that was badly damaged. Another interpretation, however, is that Lotus was a firm caught in the dilemma of a technological discontinuity—a phenomenon that has affected many technological industries, from radial tires to disk drives.[45]

A technological discontinuity occurs when a new development results in a superior technology that is fundamentally incompatible with the existing technology. An incumbent in the industry is then faced with the almost insuperable problem of migrating its product to the new technology while also protecting its existing customer base during the period of transition. Preventing the mass defection of customers to alternative suppliers of the improved technology is exceedingly difficult. In the case of Lotus 1-2-3, this defection was partially stemmed by gradual evolution of the product through the numerous releases and upgrades in the late 1980s. By contrast, new products in the market, such as Excel, had no customer base to protect, and therefore had no need to dilute or to bridge the technological discontinuity.

To induce consumers to adopt Excel, it was necessary for Microsoft to reduce users' switching costs by protecting their investments in Lotus 1-2-3. These investments included the learning of and proficiency in using 1-2-3, the users' existing spreadsheets, and the ability to share files with other 1-2-3 users. Over time—often several years of constant use—1-2-3 users became extremely proficient and had internalized the menus and keyboard shortcuts. To ease their transition to Excel, Microsoft provided printed materials and online tutorials. The Excel program also intercepted attempts to use 1-2-3 shortcuts and supplied the Excel equivalent. To preserve users' existing spreadsheet files, Excel could both read and write in the 1-2-3 file format. Of course, spreadsheets that used features unique to Excel could not be stored as 1-2-3 files, so as users began to use the full range of Excel facilities they were gradually weaned off the 1-2-3 format. The ability to read and write 1-2-3 files also preserved a degree of interoperability between Excel and 1-2-3, so that the two worlds could share files.

Again, as more and more users switched to Excel, the 1-2-3 file format fell into disuse.

Of course, variations of these strategies had been used by all of Lotus 1-2-3's competitors over the years, but none succeeded the way that Excel did. One reason was perhaps Microsoft's mastery of creating and using proprietary standards to its advantage—an issue that came up time and again in Microsoft's engagements with the Department of Justice and the European competition authorities. However, it is not clear that Microsoft competed more aggressively than other would-be 1-2-3 killers, such as Borland; for example, Excel's use of 1-2-3 menus was much less imitative than Borland's Quattro, and its use of compatible file formats was no different. More likely, the demise of 1-2-3 resulted from the paradigm shift occasioned by the mass adoption of the Windows operating system, for which Lotus Development was unprepared. Quite simply, Lotus Development bet on the wrong operating system. Microsoft bet on its own Windows operating system and became the winner. It was a crucial junction in the history of the software industry. In 1987, when Lotus made its OS/2 decision, it had annual sales of $400 million, which were marginally greater than Microsoft; both had somewhat more than 2,000 employees. In the expanding market for personal computers in the early 1990s, Lotus continued to grow, achieving $1 billion annual sales and 6,000 employees by 1995, when it was acquired by IBM. But by then, Microsoft had achieved $6 billion sales and had over 17,000 employees.

Excel 3 was not, of course, the end of spreadsheet evolution. Since its launch in 1990, the program has gone through many upgrades, each release more feature-rich than its predecessor—to the point where many users have found the product's complexity bewildering.[46] Microsoft also innovated by bundling Excel with Word and other programs as Microsoft Office, leading to its dominance not just of spreadsheet programs but all office applications.[47] Excel's principal competitors, Lotus 1-2-3 and Quattro, survived, but only just—by 1997 their market shares in terms of revenues were well below 10 percent.[48] Quattro, now owned by Novell, is a component of WordPerfect Office; 1-2-3 is now a component of IBM's SmartSuite. In 2001 an open-source office productivity suite offered a free, high-quality alternative to Microsoft Office, although the latter is still reported to have a 95 percent market share,[49] suggesting that free software is not sufficient to overcome switching costs and the uncertainties of a novel source of supply. Much more recently, Web-based software has entered the spreadsheet domain, notably Google Spreadsheets.[50] Web services and software-as-a-service may prove to be a paradigm shift as significant as the transition to GUI-based desktop operating systems in the early 1990s.[51] If so, we may see a new chapter opening in the spreadsheet story.

Acknowledgments

My thanks to three anonymous reviewers and Burt Grad for their valuable comments on an earlier draft of the article. I would also like to thank Dan Bricklin and Bob Frankston, Jonathan Sachs, Paul Funk, Jim Kinlan, and Scott Tucker for interviews and research materials.

References and notes

1. See, for example: N. Oudshoorn and T. Pinch, *How Users Matter: The Co-construction of Users and Technology*, MIT Press, 2003.
2. S.J. Douglas, *Inventing American Broadcasting, 1899–1922*, Johns Hopkins Univ. Press, 1987.
3. M. Campbell-Kelly and W. Aspray, *Computer: A History of the Information Machine,* 2nd. ed., Westview Press, 2004, pp. 211-213.
4. S. Smulyan, *Selling Radio: The Commercialization of American Broadcasting*, Smithsonian Institution Press, 1994.
5. J.C. Herz, *Joystick Nation: How Videogames Gobbled Our Money, Won Our Hearts, and Rewired Our Minds*, Little, Brown and Co., 1997.
6. D. Bricklin and B. Frankston, interview by M. Campbell-Kelly, 7 May 2004, p. 23.
7. Contrast with R.X. Cringely, *Accidental Empires*, Addison-Wesley, 1992.
8. Unless otherwise stated, the following discussion of VisiCalc is based on my interview with Bricklin and Frankston (see Ref. 6).
9. Ibid., p. 13.
10. Quoted in R.P. Rumelt, *VisiCorp 1978–1984 (Revised)*, Graduate School of Management, Univ. of California, Los Angeles (UCLA), 2003, p. 2.
11. D. Fylstra, interview by T. Haigh, 7 May 2004.
12. D. Bricklin and B. Frankston, interview by M. Campbell-Kelly, 7 May 2004, p. 26.
13. J. Tarter, ''Who's Hot in Business Software,'' *SoftLetter*, 11 Feb. 1983, p. 2.
14. M. Kapor, interview by W. Aspray, 7 May 2004.
15. J. Sachs, interview by M. Campbell-Kelly, 7 May 2004.
16. ''New Software Takes the Drudgery Out,'' *Business Week*, 13 Dec. 1982, pp. 72-76.

17. M. Campbell-Kelly, *From Airline Reservations to Sonic the Hedgehog: A History of the Software Industry*, MIT Press, 2003, p. 255.
18. P. Petre, ''The Man Who Keeps the Bloom on Lotus,'' *Fortune*, 10 June 1985, pp. 92-94, 96, 98, 100.
19. Aimed at users, consultants, and add-on makers (see later), *Lotus* magazine is a superb historical source for the development of the spreadsheet industry.
20. A. Akera, ''Voluntarism and the Fruits of Collaboration: The IBM User's Group Share,'' *Technology and Culture*, vol. 42, no. 4, 2001, pp. 710-736.
21. B. Buell, ''Coming of Age at Lotus: Software's Child Prodigy Grows Up,'' *Business Week*, 25 Feb. 1985, pp. 54-55.
22. J. Sachs, interview by M. Campbell-Kelly, 7 May 2004, p. 15.
23. D.W. Carroll, ''Lotus Enhancements Blossom Forth,'' *Software News*, Mar. 1985, pp. 66-71.
24. P. Funk, interview by M. Campbell-Kelly, 12 May 2003.
25. K.H. Hammonds, ''The Little Software Maker that Lotus Loves,'' *Business Week*, 7 Nov. 1988, p. 168.
26. J. Kinlan and S. Tucker, interview by M. Campbell-Kelly, 12 May 2003.
27. J. Tarter, ''Report from Lotus-Land,'' *SoftLetter*, 21 Mar. 1985, pp. 1-3.
28. Lotus, *Lotus File Formats for 1-2-3 Symphony & Jazz: File Structure Descriptions for Developers*, Addison-Wesley, 1986.
29. ''New Programs for Third-Party Developers,'' *Lotus*, May 1986, p. 7.
30. C. Brown, ''Your Add-in Holiday Shopping List,'' *Lotus*, Dec. 1987, pp. 74-81.
31. J. Potwora, ''1-2-3 Utilities: A Variety of Ways to Enhance Your Spreadsheets,'' *Ingram Journal*, Apr. 1998, pp. 13, 24.
32. L.D. Graham, *Legal Battles that Shaped the Computer Industry*, Quorum Books, 1999, pp. 56-68.
33. E.E. Keet, *Preventing Piracy: A Business Guide to Software Protection*, Addison-Wesley, 1985.
34. M. Campbell-Kelly, ''Not All Bad: An Historical Perspective on Software Patents,'' *Michigan Telecommunications and Technology Law Rev.*, vol. 11, no. 2, 2005, pp. 191-248.
35. S.J. Liebowitz and S.E. Margolis, *Winners, Losers & Microsoft: Competing and Antitrust in High Technology*, The Independent Inst., Oakland, Calif., 1999, p. 175.
36. For comprehensive histories of Multiplan and Excel, see D. Ichbiah and S.L. Knepper, *The Making of Microsoft*, Prima Publishing, 1991, pp. 98-119 and 197-210.
37. 1-2-3 was eventually released for the Macintosh in 1990.
38. Ichbiah and Knepper, *Making of Microsoft*, p. 169.
39. Ibid., p. 205.
40. Microsoft Corp., ''Soul of the New Machines,'' VHS Video, 1987.
41. K.H. Hammonds, ''The Spreadsheet that Nearly Wore Lotus Out,'' *Business Week*, 3 July 1989, pp. 50-52.
42. K.R. Conatser, ''1-2-3 Through the Years,'' *Lotus*, June 1992, pp. 38-45.
43. Ibid., p. 42.
44. Liebowitz and Margolis, *Winners, Losers & Microsoft*, p. 175.
45. C.M. Christensen, *The Innovator's Dilemma*, Harper Business, 2000.
46. See, for example, W. Leonhard, L. Hudspeth, and T.J. Lee, *Excel 97 Annoyances*, O'Reilly, 1997.
47. Campbell-Kelly, *From Airline Reservations to Sonic the Hedgehog*, pp. 258-259.
48. Liebowitz and Margolis, *Winners, Losers & Microsoft*, p. 176.
49. ''OpenOffice.org,'' *Wikipedia*, http://en.wikipedia.org/wiki/Openoffice#Market_share.
50. J. Markoff, ''Google Takes Aim at Excel,'' *New York Times*, 6 June 2006.
51. M. Campbell-Kelly and D.D. Garcia-Swartz, ''The Rise, Fall and Resurrection of Software-as-a-Service: Historical Perspectives on the Computer Utility and Software for Lease on a Network,'' *The Internet and American Business*, W. Aspray and P. Ceruzzi, eds., MIT Press, forthcoming.

Martin Campbell-Kelly is a professor in the Department of Computer Science at the University of Warwick, where he specializes in the history of computing. His publications include *Computer: A History of the Information Machine* (2nd ed., Westview Press, 2004), co-authored with William Aspray, *From Airline Reservations to Sonic the Hedgehog: A History of the Software Industry* (MIT Press, 2003), and *ICL: A Business and Technical History* (Oxford University Press, 1989). He is editor of the *Collected Works of Charles Babbage* (Pickering & Chatto, 1989).

Campbell-Kelly can be contacted at m.campbell-kelly@warwick.ac.uk.