4

Dropouts and Evangelists

So we went to Atari and said, "Hey, we've got this amazing thing, even built with some of your parts, and what do you think about funding us? Or we'll give it to you. We just want to do it. Pay our salary, we'll come work for you." And they said, "No." So then we went to Hewlett-Packard, and they said, "Hey, we don't need you. You haven't got through college yet."

—Steve Jobs, founder of Apple Computer Inc., on attempts to get Atari and H-P interested in his personal computer.[1]

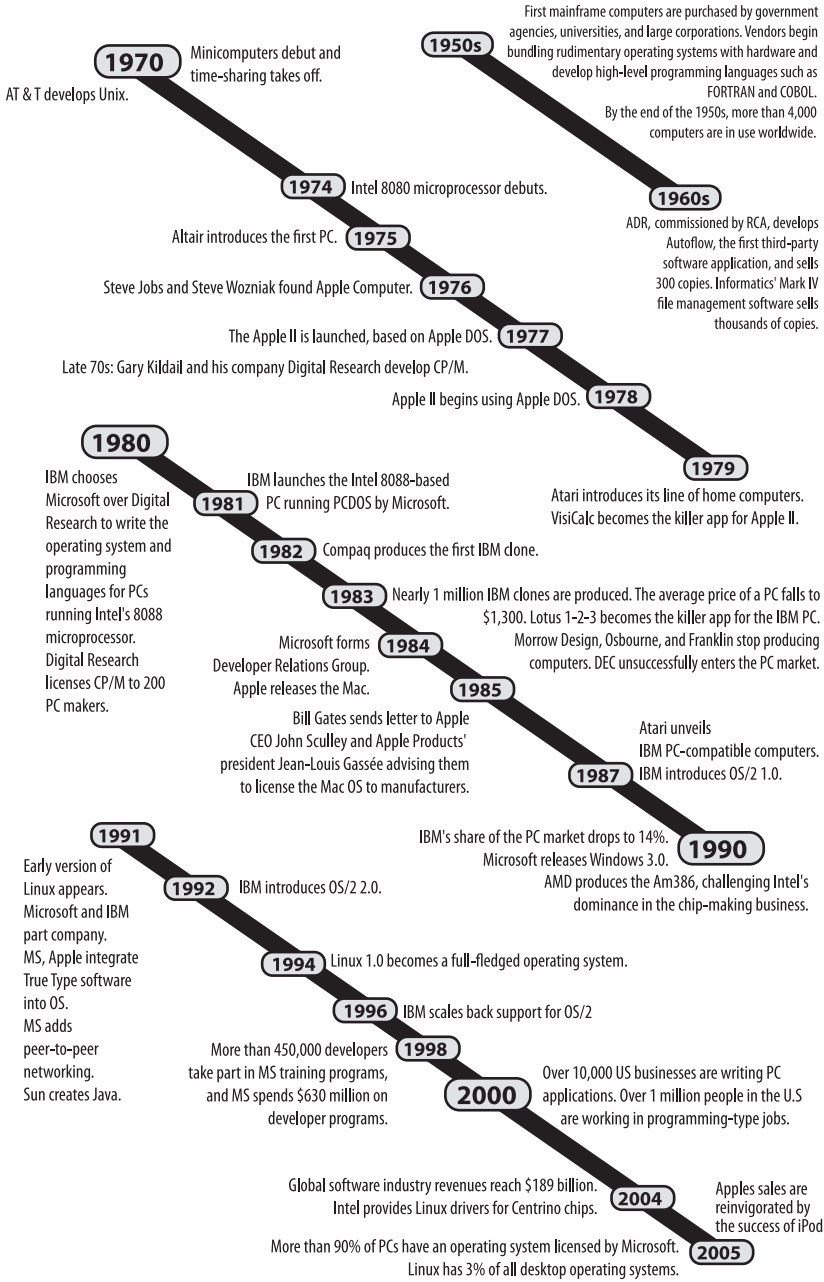
INSIDE THIS CHAPTER

- The history of PC software platforms
- The role of multisided strategies in promoting growth and profits
- Hardware integration and its effect on the growth of the Apple and Microsoft platforms

In the first years of the computer industry, every computer was on its own island. In the early 1950s, a few large corporations, government agencies, and universities bought mainframe computers from a few large companies such as Sperry Rand. They didn't get much beyond the hardware. They got a few manuals and the basic software they needed to run programs written in assembly language. They didn't even get an operating system. Each computer's owner needed a team of in-house programmers who, perhaps with some technical help from Sperry, would write applications customized for that organization and that computer. Buying a new computer, even from the same

1. Blech, Benjamin, *Taking Stock: A Spiritual Guide to Rising Above Life's Financial Ups and Downs* (New York: AMACOM, 2003).

Chapter 4: Personal Computer Timeline



manufacturer, often meant laboriously rewriting those applications almost from scratch.

The isolation of computer centers began changing at the end of the 1950s, when there were around 4,000 computers in use worldwide. Computer vendors began bundling rudimentary operating systems with their hardware. And the development of high-level programming languages such as FORTRAN and COBOL made programming simpler and made it easier, though hardly simple, to move programs from one machine to another. Computer owners could also start calling out for help. Two computer analysts who worked in the aerospace industry, for example, started one of the first programmers-for-hire companies—Computer Sciences Corporation—in 1959. Others followed suit. At first these software companies focused on helping companies write specialized software, from compilers to applications, for their expensive mainframe computers.

It didn't take much longer, though, for computer entrepreneurs to realize that there could be a market for general-purpose software that many companies would find useful. In the early 1960s, RCA, a computer manufacturer, commissioned Applied Data Research (ADR) to develop software that would automate the flow-charting of programs so as to facilitate debugging new applications and updating old ones. RCA had planned to give away the ADR product to help sell its computers, but it ultimately decided not to do this. Other computer manufacturers showed no interest in bundling ADR's software with their machines either. So ADR decided to try something new: it marketed its product, christened Autoflow, directly to computer users. It was hardly a mass market success by today's standards. But by 1968 ADR had sold about 300 copies of AutoFlow for operating systems from RCA, IBM, and Honeywell.

Others followed in ADR's footsteps. Informatics was one of the most influential. Its Mark IV file management software was, for computers, selling like hotcakes in the late 1960s. It sold thousands of copies of Mark IV, which ran on IBM's System/360 computer, for $30,000 each. That amounted to over $100 million in sales from late 1967 to the early 1980s. The "packaged" software industry was born. IBM gave this new arrival a significant boost in 1970 when the largest producer of

mainframes began charging for all its software products (except its operating systems) rather than including them at no charge with its computers.[2]

Fast-forward to today's personal computer industry. The changes in industry structure are dramatic. Millions of people buy computers from dozens of manufacturers. Most computer users have never written a program of any sort. More than 90 percent of today's PCs have an operating system licensed from Microsoft, which plays a major role in the industry, even though it doesn't make computers.[3] Microsoft and many other firms sell a wide range of applications that can be run on most new computers.

Indeed, the software industry has become enormous: in 2003, the global software industry had revenues of $178 billion for packaged software. More than 10,000 businesses specialized in writing applications—Independent Software Vendors (ISVs), to use the industry jargon—in the United States alone.[4] More than one million people worked in programming-type occupations in the United States at the turn of the twenty-first century.[5] And, increasingly, large quantities of programming work are outsourced to software factories in India.

In addition, a large number of other firms produce monitors, printers, mice, and other peripheral equipment that can be used with virtually any PC. Those few isolated individuals struggling to make computers useful in the early 1950s might have been able to imagine much more powerful machines than those available then. But they would almost

2. Martin Campbell-Kelly, *From Airline Reservations to Sonic the Hedgehog: A History of the Software Industry* (Cambridge, Mass.: MIT Press, 2003), pp. 36, 101, 103–118.

3. Al Gillen and Dan Kusnetzky, "Worldwide Client and Server Operating Environments 2004–2008 Forecast: Microsoft Consolidates Its Grip" (IDC report no. 32452), December 2004.

4. Richard W. Heiman and Anthony C. Picardi, "Worldwide Software 2004–2008 Forecast Summary" (IDC report no. 31785), August 2004.

5. In the 2000 Census, there were 521,105 full-time year-round workers in the Computer Programmers title and 595,965 Computer Software Engineers, for a total of 1,117,070 people doing programming-related jobs. In addition, there were 554,720 Computer Scientists and System Analysts, which if included would bring the total to 1,671,790. http://www.census.gov/hhes/income/earnings/call2usboth.html.

certainly have been unable even to dream of today's rich and lively PC ecosystem, and they would never have been able to imagine the key role that software in general and operating systems in particular play in that ecosystem. The chronicle of this great structural transformation is mainly about the emergence of popular PC software platforms that sit between the hardware and applications. It is a tale, at the human level, driven by entrepreneurs who dropped out of college to pursue dreams that came true, and, more important for our purposes, of evangelists who worked at popularizing software platforms and thereby helped stoke the indirect network effects that propelled the PC revolution.

The Apple and Microsoft Software Platforms

Innovation was already shaking the stodgy mainframe computer industry by the mid-1970s. Computer power was coming to the masses—sort of. Companies such as Digital Equipment Corporation (DEC) were making minicomputers that were far less expensive than IBM's mainframes and that more businesses could use for more applications. And schools, too: Bill Gates learned how to program on his high school's DEC PDP computer. The time-sharing business was taking off: companies rented access to powerful computers to businesses with remote terminals. And companies such as Wang had developed specialized computers for office work. Innovations were occurring in operating systems as well, as we saw in Chapter 2. AT&T had developed Unix, which, in its several somewhat incompatible variants, became a powerful operating system for many of the new minicomputers and workstations.[6]

Minicomputer makers, however, still largely followed the highly integrated model pioneered by mainframe makers. They provided hardware, operating systems, and some applications—though they often charged separately for the apps. The biggest challenges to the industry's traditional structure and way of doing business were under way but almost invisible. Few noticed as the foundations were being laid for the PC revolution. The ensuing story has been told often, so we will just sketch some of the highlights.

6. Campbell-Kelly, *From Airline Reservations to Sonic the Hedgehog*, pp. 143–144, 159.

The Intel 8080 microprocessor, which debuted in 1974, made it possible to produce cheap electronic devices for a variety of purposes. The first PC was the Altair 8800, which became available in 1975. It came as a kit that hobbyists could use to build their own computers with 8080 chips. Like the earliest mainframes, the Altair came without an operating system. Bill Gates famously dropped out of Harvard to work with his childhood friend Paul Allen on a program that would allow users to compile and run BASIC programs on the Altair.[7]

Two years later Steve Jobs and Steve Wozniak (dropouts from Reed and Berkeley, respectively) took the next step. They sold the Apple II as a product bundled with a keyboard, a monitor, and a 6502 microprocessor from MOS Technology. It also came with a tape drive and Apple's version of BASIC. Apple later shipped a floppy disk drive for the Apple II that included a disk operating system called DOS or Apple DOS. Before that, BASIC was used to run programs.

Commodore Business Machines' Commodore PET and Radio Shack's TRS-80 were two of the more popular contemporaries of the Apple II. Like the Apple II, these machines came with a BASIC interpreter that functioned as the software platform. The TRS-80 also came with TRSDOS, a disk operating system, and a floppy disk drive.

None of these machines came with any applications to speak of. Many applications soon became available, though, especially for the Apple II. Most were programs that people shared freely.

By the early 1980s, hundreds of new computer companies were selling machines based on 8-bit processors. Then IBM appeared with its Intel 8088-based PC in 1981. It came with a version of Microsoft BASIC, and most purchasers also bought a new operating system produced by Microsoft called MS-DOS (more on this later).[8]

Early Software Platforms

After Bill Gates and Paul Allen developed their BASIC compiler for the Altair 8800, they went on to develop BASIC programming languages and tools for other early PCs. BASIC, which had been developed in 1963 at

7. Ibid., pp. 202–204.

8. http://inventors.about.com/library/weekly/aa033099.htm.

Dartmouth College as a teaching tool, became a software platform for these machines. Many PC owners, particularly hobbyists, used BASIC to write their own programs. Often users could copy BASIC source code from magazines and books such as *BASIC Computer Games*. BASIC applications could use the commands in BASIC (such as those controlling printing) to perform various tasks so programmers didn't have to write assembly language code themselves to perform those tasks. These commands thus played something like the role of the APIs we discussed in the last chapter.

BASIC and other programming languages were Microsoft's core business through the late 1970s. They accounted for about 30 percent of its revenue in 1984, shortly before the firm went public in 1986.[9] While BASIC was important during these early days, most prospective users weren't programmers, and BASIC never became for any manufacturer what Apple quickly acquired—a killer app.

Microsoft's pricing of BASIC departed from industry practice. Other software companies had translated various programming languages for specific operating systems used for mainframes and workstations. They had generally licensed the code to computer vendors for a substantial flat fee. Microsoft, though, charged PC makers a royalty for each copy they distributed—$30 a copy in the case of the MITS Altair 8800.

This approach worked well for both buyer and seller. Especially for cash-poor computer startups, it reduced their upfront costs. It also reduced their risks: if they didn't do well, they didn't have to pay much. They could also easily pass on the per-copy royalty cost to their customers. Per-copy charges also helped Microsoft capitalize on its investment in programming languages in the face of great uncertainty as to which computer makers would succeed. A flat fee would have earned less from the top sellers and would have discouraged other makers from even trying. Microsoft retained this basic pricing model when it went into the operating system business.

Another software platform seemed very promising during the late 1970s. Gary Kindall developed an operating system for the Intel 8080

9. Andrew Pollack, "Lotus Is the Spoiler at Microsoft's Party," *New York Times*, September 9, 1985.

chip called CP/M. He initially sold copies to hobbyists by mail for $75 each. He also gave one of the new computer makers, IMSAI, a blanket license for $25,000 in 1977. Kindall's company, Digital Research, wrote versions of CP/M for other new startups.

CP/M was important in the early days for two reasons. First, it relieved PC startups of the cost of designing their own operating systems, thereby reducing barriers to entry into the PC market. By 1980, Digital Research had licensed versions of CP/M to some 200 PC makers.[10] Second, CP/M to some extent provided a cross-platform environment for third-party application developers. Even though CP/M applications were not perfectly portable between computers from different manufacturers, the widespread use of CP/M significantly reduced the burden of writing applications for multiple otherwise incompatible computers.

From the beginning, Jobs and Wozniak decided that Apple should develop its own proprietary operating system. They followed the same model as mainframe and minicomputer companies. At first this seemed to be an enormous competitive advantage. A killer application for the Apple II, the VisiCalc electronic spreadsheet, appeared in 1979 and helped turn the Apple II into a highly successful computer platform shortly after its introduction.

VisiCalc didn't run at first on the competing CP/M software platforms. And while the CP/M machines were popular, there was no killer application for them in this period. Moreover, had one appeared, it is unlikely that it would have lighted a fire under any one of the manufacturers of CP/M computers—the flip side of low barriers to entry is generally low ability to sustain the profits needed to recoup investments. But a CP/M killer app might have given at least a short-term boost to the fortunes of Digital Research.

Many other companies followed Apple's highly integrated model, including Tandy, Commodore, Texas Instruments, Coleco, Atari, Timex, and Sinclair. Like the CP/M-based computers, they now appear mainly in trivia quizzes for computer buffs.

10. Campbell-Kelly, *From Airline Reservations to Sonic the Hedgehog*, pp. 205–206.

Atari became a household name thanks to its popular VCS game console and, to a lesser extent, its arcade games. However, the company was making home computers as early as 1979. After producing several 8-bit machines, Atari released the ST line of computers in the mid-1980s with the slogan "Power without the Price." These computers compared favorably with IBM PCs, Apple Macintoshes, and Commodore Amigas in terms of performance per dollar. They also included a MIDI port that made them popular with musicians.

Microsoft, IBM, and the Birth of a New Platform

In 1980, mighty IBM was in the uncomfortable position of playing catch-up in the PC market; Apple, Commodore, and Atari, among others, were already well established. Contrary to its usual practice of doing almost everything itself, IBM decided to speed development by securing partners to make much of the necessary hardware and systems software. The company offered Microsoft a contract to produce programming languages, its specialty at the time, for its new PC. Microsoft didn't have the time or interest to write an operating system and thought CP/M was their and IBM's best bet to meet their deadlines. According to Bob O'Rear, who led the IBM technical efforts at Microsoft, "[O]ur first shot at IBM was to get them to pick up CP/M from DRI and Bill helped set up a meeting."[11] When DR failed to come to terms quickly with Big Blue, IBM came back to Microsoft; Microsoft realized its programming language deal required an operating system and agreed to do it.[12] According to O'Rear,

> The [operating system] we thought fit the best for a personal computer was CP/M. It was small, it was targeted at the right audience, it was something we could build on. We had a lot of faith in DR. [But] that didn't work, so we folded MS-DOS into the technical proposal and submitted that and IBM went for it. And then they also went for a huge list of modifications that had to be done to 86-DOS.

IBM was in a hurry for its new operating system. Microsoft bought a rudimentary operating system for the Intel 8086 from neighboring Seattle Computer Products to get a quick start. Seattle Computer Products had been waiting for a version of CP/M for a computer they had

11. O'Rear interview notes from MS-DOS encyclopedia project. Ray Duncan, ed., *The MS-DOS Encyclopedia* (Redmond, Wash.: Microsoft Press, 1988).

12. Campbell-Kelly, *From Airline Reservations to Sonic the Hedgehog*, pp. 206–207.

built. Frustrated by delays, they had one of their employees, Tim Patterson, write a "quick and dirty operating system" (dubbed Q-DOS) for it. Seattle Computer Products didn't want to be in the software business. Patterson joined Microsoft to help lead the effort to turn his Q-DOS into something that would meet IBM's specifications.

Microsoft's programmers then wrestled with a multitude of bugs and complexities to produce a finished operating system that was more efficient and included numerous enhancements. Among other things, it offered increased hardware independence, improved disk space allocation and management, and greater ease of use for users with less technical know-how. Microsoft turned over the completed version of PC-DOS modified for the 8088 chip nine months after sealing its deal with IBM. The basic system consisted of roughly 4,000 lines of assembly language code that took up 12 kilobytes of memory. IBM was able to ship its PC with PC-DOS a year after its aggressive decision to take on Apple and the other startups.[13] (Under Microsoft's agreement it could also license DOS to others, and it did so under the name MS-DOS.)

The computer giant thought it had kept control of the platform it was developing. It had a royalty-free license for PC-DOS. It was IBM that shipped PC-DOS, not Microsoft. And IBM planned to make it possible for its hardware platform to work with several operating systems. It reached an agreement with Softech for the UCSD p-System. The USCD p-System was available when the new IBM PC was launched, but it ran very slowly. IBM also belatedly reached a deal with Digital Research to produce a version of CP/M for its new machine. CP/M-86 for the IBM PC appeared several months after the launch, but Digital Research decided to price it at $240, four times the $60 cost of PC-DOS.[14]

13. Duncan, *The MS-DOS Encyclopedia,* pp. 15–24; Daniel Ichbiah and Susan L. Knepper, *The Making of Microsoft: How Bill Gates and His Team Created the World's Most Successful Software Company* (New York: Prima Publishing, 1991), p. 85; Campbell-Kelly, *From Airline Reservations to Sonic the Hedgehog*, p. 207.

14. Michael A. Cusumano and Richard W. Selby, *Microsoft Secrets* (London: HarperCollins, 1995), p. 159; Campbell-Kelly, *From Airline Reservations to Sonic the Hedgehog*, pp. 239–240.

In retrospect, having multiple operating systems run on a hardware platform is a poor strategy. The idea, of course, was to ensure that the hardware, not the operating system, became the standard that defined the platform and determined its evolution. Indeed, IBM followed an important economic principle for traditional industries: all firms would like everyone else in the supply chain to be competitive. IBM didn't seem to recognize that this was far from a traditional industry.

If IBM's strategy *had* worked, and if several operating systems had been installed on substantial numbers of IBM PCs, what would have happened? Most likely, having multiple operating systems would have made the hardware platform less popular than having a single operating system. Applications are generally written for software platforms, not the underlying hardware. The more fragmented the installed base of operating systems, the less attractive it is to write an application for any one of them. Thus, operating system fragmentation would have reduced the number of compatible applications for each of them, reducing their attractiveness to end users and thus reducing the value of the underlying hardware platform.

As we noted in Chapter 2, that is in fact what happened with the UNIX operating system for minicomputers. Several versions were created, and applications weren't compatible across them. That fragmentation (often called "forking") stunted the growth of UNIX. As of 2006, Linux, a stepchild of UNIX, has managed to overtake UNIX in part because Linus Torvalds and the rest of the committee that manages Linux have worked very hard to prevent fragmentation (more on this later).

But, as we now know, IBM's multiple-OS strategy did not work.

In order to get to market quickly and hold down system cost, IBM decided to create an open hardware platform—one quite unlike the walled garden it had tended for years in mainframes. And, of course, it had outsourced operating systems to Microsoft and other firms over which it had limited control. It appears to have believed nonetheless—one can only conjecture at this point—that it could reap the lion's share of profits from this innovative computer platform through its brand name, its marketing muscle, and its intellectual property in the basic input-output system (BIOS) that starts the computer when it is turned

on. After all, it was selling the computers and should be able to charge a premium for them, as it had always done.

Things didn't work out that way. Microsoft had retained the rights to license MS-DOS—an exact replica of PC-DOS—to other computer manufacturers. It was keen to do so. At the same time, dozens of manufacturers started trying to clone the IBM PC. Their main stumbling block was the BIOS. Copyright law wouldn't allow them just to copy it. But nothing prevented them from reverse-engineering it. Like recreating a gourmet meal without the chef's secret recipe, this involved writing code for the BIOS by observing what the code did rather than what the code was.

Compaq produced the first truly legal IBM PC clone after reportedly spending $1 million to figure out the secrets of the BIOS. By 1983, IBM competitors had produced almost one million IBM PC clones. All ran MS-DOS, which was already the most popular operating system for PCs. And the price was right—an estimated $10 per computer at a time when the average PC went for about $1,300.[15] IBM tried to develop other proprietary technology to recapture control, but it had to give up the fight by the end of the 1980s. IBM's share of IBM-compatible PC sales tumbled to 14 percent by 1990.[16] It stopped making PCs altogether with the sale of its PC division to Lenovo in 2004.

The IBM/DOS-compatible PCs quickly killed off the many CP/M-compatible manufacturers. Between 1981 and 1986, Morrow Design, Osborne, and Franklin went out of business, and the rest failed not much later.[17] Companies like DEC tried to enter with CP/M-compatible machines during this period but had little success.

15. Campbell-Kelly, *From Airline Reservations to Sonic the Hedgehog*, pp. 207, 240–242; Dataquest, "Personal Computer Industry Service Worldwide Shipments and Forecast," tables 1.3.5 and 1.3.13.

16. Bruce Stephen and Mark Levitt, "Worldwide PC Market Review and Forecast 1990–1995" (IDC report no. 6077), December 1991, table 3.

17. Helen Grant, "Zenith High But Maker Goes Broke," *Australian Financial Review*, March 13, 1986; http://www.absoluteastronomy.com/encyclopedia/O/Os/Osborne_Computer_Corporation.htm; http://www.ti99ers.org/timeline/time1984.htm.

By the mid-1980s it seemed clear that the battle would be between two PC platforms, Apple computers and "IBM-compatible" computers. By the early 1990s, when IBM and Microsoft had competing operating systems for PC's based on Intel microprocessors, the phrase "IBM-compatible" was no longer in use. In the mid-1990s, IBM decisively lost the competition between these two operating systems. Since then, platform competition has been between Apple's Macintosh platform and the "Wintel" platform: Microsoft's Windows operating system running on computers based on Intel's microprocessors.

IBM's OS/2 versus Microsoft's Windows

In 1985, IBM and Microsoft agreed to develop a new operating system for the PC. IBM led the project and provided most of the resources.[18] The first version, OS/2 1.0, was released in 1987 but was intended mainly as a preview for developers. Among other things, it lacked a graphical user interface (GUI) and a comprehensive hardware support.

The relationship between Microsoft and IBM was always difficult, in part because of the very different styles of the two companies. For example, IBM measured programmers' contributions by the number of lines of code they wrote, which Microsoft thought encouraged the production of sloppy, inefficient code. The fact that Microsoft was developing Windows in parallel to OS/2 did not help the relationship. Moreover, IBM and Microsoft had different visions for OS/2. From the beginning, Microsoft urged IBM to base the OS/2 GUI on Windows APIs. However, IBM had different plans. It sought to create a single graphical interface across all of its platforms, from mainframes to PCs. Consequently it rejected Windows and included features in OS/2 that added little for PC users. OS/2 would also run only on the then most powerful PCs. In a joint statement in the late 1980s, IBM and Microsoft positioned OS/2 as the operating system of choice for powerful PCs, with Windows the alternative for lower-end machines, which constituted about 75 percent of the shipments at the time.[19]

18. At times, IBM devoted as many as 10,000 developers to the project, compared with Microsoft's 100. In 1986 IBM had roughly 150 times Microsoft's sales and 120 times its market capitalization. Microsoft Corporation 1986 Annual Report and Form 10-K; Fact Set Research Systems, CompuStat Database, 2001.

19. Maurice F Estabrooks, *Electronic Technology, Corporate Strategy, and World Transformation* (Westport, Conn.: Greenwood Publishing, 1995), p. 64.

(continued)

In 1990, Microsoft released Windows 3.0. Not only was Windows 3.0 very successful, it was also a viable option for the upper end of the market. Later that year, Microsoft and IBM parted company in one of the most famous corporate divorces. IBM got the project that would produce the next version of OS/2, while Microsoft got the research in progress on what would eventually become Windows NT.

IBM released OS/2 2.0 in 1992 with the slogan, "a better DOS than DOS, and a better Windows than Windows."[20] It was the first PC operating system to run on 32-bit microprocessors, which could support more ambitious applications, yet it was also able to run programs written for DOS and contemporary versions of Windows. OS/2 was backed by the IBM brand name and IBM's research capacity and marketing muscle. Moreover, the company seemed committed to making the platform work, advertising heavily (if sporadically) and regularly updating OS/2 through 1996. But version 2.0 and its successors never effectively challenged Windows. Why not?

Some argue that IBM didn't invest enough in developer support and evangelization. "The company stupidly reckoned that if you give developers a good operating system, coders will code for it," wrote the computer columnist John C. Dvorak.[21] IBM also charged substantial prices for developer tools. The incompatibility of Windows and OS/2 APIs also made it harder for developers to write simultaneously for OS/2 and Windows. Forced to choose, most chose Windows.

IBM also sent confusing signals to developers. OS/2 was only one of at least four operating systems for microcomputers under development at IBM.[22] IBM was also unable to explain its overall strategy to developers, prompting an editor to exclaim that "IBM's strategy is about as comprehensive [sic] as Balkan politics."[23] In 1996, IBM scaled back its OS/2 efforts, and in July 2005, IBM finally withdrew support for OS/2.

20. http://en.wikipedia.org/wiki/OS/2.

21. John C. Dvorak, "Obituary: O/S," *PC Magazine*, December 16, 2002 (http://www.pcmag.com/article2/0,4149,767456,00.asp).

22. Instead of OS/2, IBM used AIX, its version of Unix, for its workstations running its PowerPC processors. In 1991, Apple and IBM reportedly were jointly developing a new version of Unix, PowerOpen. In 1992, they formed a joint venture, Taligent, which started out with the goal of developing yet another new operating system, code-named Pink. Roy A. Allan, *A History of the Personal Computer: The People and the Technology* (Allan Publishing, 2001), p. 19.

23. Doug Barney, "Big Blue Pitches a Play-to-Play of Its OS Plan," *InfoWorld*, July 11, 1994, pp. 21–22.

Apple versus Microsoft

During the 1980s, Apple operated a two-sided platform. The company made its own hardware, which it sold with its own operating systems. It also branded its own peripheral equipment. Until 1998 it also refused to include industry-standard ports to facilitate connections to peripherals made by others.[24] Even today, Apple computers are designed in ways that discourage the use of third-party peripherals. Some of its models, for example, integrate a proprietary monitor, disk drives, and speakers in a single computer unit. Apple also wrote applications software for its operating systems. But early on, its managers understood the importance of building and sustaining a two-sided platform, attracting software applications from independent developers (including Microsoft) to add to the appeal of the Mac.

Microsoft went four-sided. Like Apple, it encouraged third-party development of applications for MS-DOS and subsequent operating systems while also writing applications software for its own operating systems. But it did not sell computers, only dabbled in peripheral equipment, and stayed out of the markets for big-ticket items such as monitors and printers. Instead, it encouraged computer and peripheral makers to make best use of its software platform.

The Microsoft platform was therefore more complex than the Apple platform: Microsoft had to harness the indirect network externalities between computer manufacturers, peripheral equipment makers, software developers, and, of course, computer users. That meant getting them all on the same platform—Microsoft's operating system for Intel-compatible computers—and generating positive indirect network effects between them. The multisided strategies we discussed in Chapter 3 were critical to its success.

24. "Hands On–Mac–Universal Solution," *Personal Computer World*, February 1, 1999; "Mac Ports, Past and Present," http://charm.cs.uiuc.edu/users/olawlor/ref/mac_ports/.

Apple's Blunder?

It is now commonplace to view Apple's choice of an integrated two-sided platform as an unpardonable strategic error, one that consigned erstwhile market leader Apple to a marginally viable niche in PCs. Indeed, to this day, some argue that sticking with an integrated hardware-software platform undermines the profitability of workstation and server computer makers ranging from IBM to Sun.

Bill Gates wrote to Apple's CEO John Sculley and Apple Products' President Jean-Louis Gassée in 1985, a year after Apple reported a $40 million loss as lower-cost IBM clones grabbed an ever-greater chunk of the PC market. He advised them to license the highly regarded Macintosh operating system to clone-makers, concluding as follows:

As the independent investment in a 'standard' architecture grows, so does the momentum for that architecture. The industry has reached the point where it is now impossible for Apple to create a standard out of their innovative technology without support from, and the resulting credibility of other personal computer manufacturers. Thus, Apple must open the Macintosh architecture to have the independent support required to gain momentum and establish a standard. (From a memo dated June 25, 1985, Quoted by permission from Microsoft.)

This wasn't just friendly advice. At the time, Microsoft earned about half of its revenue from applications for the Macintosh (including its hot Word word-processing package, which hadn't yet made a dent in the IBM-PC segment, as well as its Excel spreadsheet program, which accounted for 90 percent of Macintosh spreadsheet sales by September 1985) and only about 20 percent from MS-DOS.[25] It wasn't at all clear at the time that Microsoft would get a second home run after DOS. So Gates was covering his bets. Microsoft might have done extremely well as a leading application developer for a dominant Mac OS.

But hindsight has a way of making uncertain outcomes seem inevitable. A closer look suggests that integrating the hardware and software platforms tightly had significant advantages over letting a thousand hardware makers bloom. Apple was able to tailor its operating system software to its hardware during a period in which operating systems were rapidly growing more complex and hardware performance was rapidly improving. And since it controlled both the hardware and the operating system, it was possible to test the operating system with every possible hardware

25. Campbell-Kelly, *From Airline Reservations to Sonic the Hedgehog*, p. 253; Jonathan Chevreau, "Apple Hopes Macintosh Will Take Bite of Market," *The Globe and Mail*, January 23, 1984; "Lotus Is the Spoiler at Microsoft's Party," *San Francisco Chronicle*, September 1985; Owen Linzmeyer, *Apple Confidential* (San Francisco: No Starch Press, 1999), p. 134.

(continued)

combination before Apple computers were put on the market, something Microsoft could not possibly do. Apple had a powerful graphical user interface before Microsoft, and its systems have long been viewed as more stable.

What we know in hindsight is that Apple's share of the PC business plummeted, so that today it has only a 4 percent share of sales.[26] But unlike the many CP/M clone-makers it once faced, Apple is still around and quite well known. Apple either knows its own strengths or is stunningly stubborn—it has chosen the same vertically integrated hardware/software strategy for its latest hit product, the iPod, which we discuss further in Chapter 8.

Managing the Software Side

Both Apple and Microsoft have focused considerable efforts on persuading third-party producers to write applications for their software platforms. These efforts paid off and were critical to the success of these operating systems.

Killer Apps In the first decade or so of the PC industry, several computer platforms took off after the emergence of a killer app for them. VisiCalc, the first spreadsheet application for PCs, was a killer app for Apple. Dan Bricklin, its inventor, and his team wrote it in assembly language—the tedious process required in those days to get good performance—for the microprocessor used in the Apple II.

After a number of limited-distribution versions—we would call them demos, alpha, and beta versions today—the first "real" release came out in October 1979. An analyst report captured its significance: "VisiCalc could some day become the software tail that wags (and sells) the personal computer dog."[27] Although VisiCalc was quickly ported to other platforms, Apple had the early lead. And, most important, businesses realized that these tiny new computers were not toys; they really could provide important productivity tools for their workers.

26. "Worldwide Client and Server Operating Environments 2005–2009 Forecast: Modest Growth Ahead" (IDC report no. 34599), December 2005.

27. http://www.bricklin.com/history/saiproduct1.htm; http://www.bricklin.com/history/rosenletter.htm.

Spreadsheets continued to provide the spark needed for platforms to get off the ground over the next decade or so. Lotus 1-2-3 appeared in 1983, not long after the IBM PC had started building momentum, and it only ran on PC/MS-DOS. It was a major advance over VisiCalc because it combined a spreadsheet, a rudimentary database, and the ability to create graphs into one product. Microsoft's Excel turned out to be one of the hot apps—killer is perhaps too strong a term at this point—that finally got Windows off the ground in its third release.

Killer apps have also played important roles for the other software platforms we consider in later chapters. These applications helped set up the positive network effects that make platforms grow. More people got a computer system with a particular software platform. That encouraged more application developers to write more applications for that platform.[28]

Apple, though, had no role in the development of VisiCalc. Dan Bricklin's Web site on the history of the development of his product makes no mention of interactions with anyone from that company. Likewise, neither Microsoft nor IBM helped Lotus create a killer application for their software-hardware platform. IBM even declined the exclusive marketing rights to the Lotus 1-2-3 spreadsheet.[29] VisiCalc had already been ported to DOS, after all; what more did they need?

Nevertheless, it didn't take long for Apple, Microsoft, and others to recognize that applications were so important to the success of their platforms that they needed to nurture their development and not just sit back and hope they became available.

Evangelization The realization that independent software vendors were vital to their success led both Apple and Microsoft to mount aggressive, ongoing efforts to recruit independent software developers to their platforms. In part, these efforts took the form of old-fashioned "you must believe" marketing long practiced by tent revivalists and self-help gurus, and perfected by Apple's Guy Kawasaki. The author of books with titles such as *The Art of the Start*, *Rules for Revolutionaries*, *Selling the*

28. Campbell-Kelly, *From Airline Reservations to Sonic the Hedgehog*, p. 216.

29. Paul Carroll, *Big Blues: The Unmaking of IBM* (New York: Crown, 1993), pp. 77–78.

Dream, and *The Macintosh Way*, Kawasaki led the charge to give the Apple Macintosh cult status among both early computer users and software developers.[30] To hear Kawasaki tell it, the interdependence between the user and developer sides of the platform was not behind the original effort: "I was never told, 'OK, you go get XYZ to write software, and they in turn will get more customers to buy your software and to buy Macs.' That's what happened, but that was not the plan."[31]

Kawasaki's marketing innovations ranged from developer conferences, which were part technical presentations and part pep rallies, to EvangeList, an email newsletter sent to Apple devotees in the mid-1990s that was designed to counter worries that Apple would disappear as its market share dwindled. To this day, his success has set the tone for marketing to software developers. It was an important contribution to the development of the positive indirect network effects needed to grow these multisided platforms. We will see that "evangelism of the platform" has been critical to all of the successful software platforms we consider. Indeed, Google appointed Vinton Cerf, one of the intellectual founders of the Internet, to be its first chief evangelist in September 2005.

As a practical matter, though, Microsoft's formation of its Developer Relations Group (DRG) in 1984 probably had more impact on the way computer platforms evolved than Kawasaki's barnstorming. This team was charged with attracting independent developers to the then-unborn Windows platform: "Drive the success of Microsoft's strategic platforms by creating a critical mass of third-party applications" was its mission.[32] DRG has pursued this goal with a determination reflected in both the degree of long-term planning and the significant resources invested. Long before Microsoft introduces a new operating system, it solicits advice on the tools that developers will need to create applications to run on it. For example, it sent tentative specifications for Windows NT to developers in November 1990, asking for feedback three years before the operating system was released.

30. http://www.guykawasaki.com/about/index.shtml.

31. Ben McConnel and Jackie Huba, *Creating Customer Evangelists* (Chicago: Dearborn Trade Publishing, 2003), p. 13.

32. "Microsoft Developer Relations: Microsoft's Commitment to Third-Party Developer Success," Microsoft Corporation white paper (Redmond, Wash.: Microsoft Corp., 1998).

Many developers belong to the Microsoft Developer Network (MSDN). They are regularly sent information on how to create applications to run on Microsoft platforms: between 1993 and 1998, Microsoft shipped 100 million CDs with this sort of information. The company sponsors several series of conferences to keep the programming community both informed and involved. In 1998 alone, some 450,000 developers took part in various Microsoft training programs. All told, the company spent $630 million on its evangelism effort that year.[33]

Application Program Interfaces Evangelization is unlikely to succeed with application developers without a good product—in particular, software services made available through APIs. On the one hand, it is important to convince developers that the platform will attract many end users interested in their products: hence direct, visible advertising to end users and the pep rally aspects of evangelization. On the other hand, it is important to convince developers that they can write attractive programs to run on the platform relatively easily: hence efforts to reduce the costs of writing applications for the Apple and Windows platforms.

The platform owners make heavy investments in technical assistance to developers as part of this effort. All platform managers maintain the so-called developer networks. MSDN, with over 3 million members, is one of the biggest, while the Apple Developer Connection has about 500,000 members.[34] These networks are subscription-based, with annual charges ranging from a few hundred dollars to over $10,000, depending on the services provided. They offer access to news, technical documentation, developer forums, and online support. Members may receive discounts on select developer tools or conference fees. Although the open-source Linux platform has no formal platform manager, Linux developer forums are regarded as some of the best because all the members of the ecosystem participate, including IT managers, application developers, and platform developers.

33. *United States v. Microsoft*, Civil Action No. 90–1232, Testimony of Paul Maritz, January 20, 1999, § 140, § 136–152.

34. http://www.edn.com/blog/400000040/post/740000874.html; "Microsoft Announces Unprecedented Momentum for MSDN at 3 Million Members," *M2 Presswire*, March 7, 2000.

Platforms make another key investment in reducing developer costs: they constantly add and improve the software services provided through the APIs discussed in Chapter 2. For example, the Apple Mac OS X, introduced in 1999, has about 8,000 APIs that expose underlying software services. Some (called Cocoa) were designed to support new software applications and others (called Carbon) were designed to ease the transition from the Mac OS 9.[35]

These numbers by themselves don't say much, of course; it is the range of services these APIs offer to developers that is impressive. The original MS-DOS offered developers APIs for keyboard input, file operations, and time control, to name a few. In the late 1980s, Windows came with many more, including APIs that enabled developers to use memory in a much more sophisticated way, to take advantage of the GUI, and to use a mouse for input. Media functionality was added in the early 1990s, CD support in the mid-1990s and DVD APIs at the turn of the twenty-first century. Throughout the 1990s, operating systems added support for new networking technologies such as infrared, Bluetooth, and WiFi.

The end user does not see any of this. Instead, she sees the applications that are built on top of these APIs. The many media players on PCs these days, for instance, rely on the underlying operating system for the core media functionality, in addition to using the APIs that display the media player on the screen and let the user control it with a mouse. Instant messengers, Palm synchronization, and other applications use networking APIs, while games use the operating systems' 3D graphics support.

Managing the Hardware Side

Apple treats hardware from other suppliers in much the same way that a vertically integrated automobile company like Toyota treats parts and optional equipment made by others: it buys many components from independent suppliers. Over the years, for example, Apple's microprocessors have come largely from Motorola and IBM.[36] But as a major customer,

35. William Peterson, Jean Bozman, and Dan Kusnetzky, "Apple Announces New Operating System Strategy for the Mac" (IDCFlash no. 16257), May 1999; http://en.wikipedia.org/wiki/Mac_OS_X_history.

36. Stephen Shankland, "Apple to Ditch IBM, Switch to Intel Chips," *CNET News.com*, June 3, 2003 (http://news.com.com/2100-1006_3-5731398.html).

Apple has a strong say in their design and specifications. Apple's first commercial computer, the Apple II, was an open system and owes a large part of its success to the availability of many third-party hardware add-ons. However, Steve Jobs felt that a true PC should be an appliance like a TV that requires no interaction with the circuitry, no technical knowledge, and no assembly. He realized this vision in the original Macintosh, released in 1984. This machine had no expansion slots, no hard disk drive, and no standard ports. Its keyboard also lacked arrow keys, to force the user to use the mouse. After Jobs' departure, the Macintosh design was relaxed and expansion slots, standard ports, hard drives, and arrow keys all appeared on the models released in 1986.[37]

One of Apple's most successful branded products was the LaserWriter line of printers. Launched in 1985, they helped create what is now known as desktop publishing. As of 2005, Apple's high-end thin-screen displays are one its most popular branded peripherals. They, like other Apple products, including the iPod discussed in Chapter 8, are sold in Apple's chain of retail stores, among other places.

Microsoft, by contrast, has specialized in software from the outset—a "stick to your knitting" strategy. It has made only a few forays into hardware, such as the Microsoft mouse and its wireless keyboards. Microsoft makes hardware, however, mainly to help sell more software. Microsoft's SoftCard, introduced in 1980, for example, enabled Apple II computers to run CP/M applications, including Microsoft BASIC. Similarly, Microsoft introduced its mouse in 1983 to help spur sales of Microsoft Windows, which was in development at the time.[38]

These exceptions aside, Microsoft mainly relies on third parties to make the complementary hardware that helps sell PCs and thereby its operating systems. This may seem like a difference without a distinction. After all, Apple purchases many of the parts for its machines, and it has to make sure these suppliers provide technology that will help Apple sell its computers. But Microsoft, like other multisided platform firms, has structured a complex series of relationships with third parties to promote

37. http://lowendmac.com/history/1984dk.shtml; http://www.lowendmac.com/history/1986dk.shtml.

38. Paul Freiberger and Michael Swaine, *Fire in the Valley: The Making of the Personal Computer*, 2nd ed. (New York: McGraw-Hill, 2000), p. 329; Stephen Manes and Paul Andrews, *Gates* (New York: Simon & Schuster, 1994), p. 221.

the licensing of its operating system software. These relationships are managed partly through financial incentives and partly through developing the software platform in close cooperation with these third parties to promote their sales as well as Microsoft's sales.

That difference is best seen in the incorporation of CD-ROM drives into computers. Apple could just decide to do this, buy CD-ROM drives from third parties, and build them into its computers. Microsoft had to encourage the computer manufacturers in its ecosystem to install them. Those manufacturers didn't have much incentive to do this, however, when there wasn't much software that relied on CD-ROMs. Microsoft provided financial incentives to install CD-ROM drives and promised that its software platform would ensure the development of applications that used CD-ROMs. We return to this later.

The Microsoft-Intel partnership has been central to the hardware-software platform that is the basis for the PCs that most of us use. Intel had virtually no competition until 1990, when AMD, Intel's former second source supplier, released the Am386 chip. As of 2004, Intel had an 82 percent share of the global PC microprocessor business.[39] Intel and Microsoft have had to work closely to ensure that Microsoft operating systems get the computing power they need from Intel processors and that Intel's processors get the support they need from the software platform. Not surprisingly, the relationship between these two elephants has not been free of conflict. Each has sought more control over the Wintel platform, and with it, presumably, a larger share of the profits associated with the platform's spectacular success.

Both have sought to hedge their bets with other partners. Microsoft has long dealt with Intel's microprocessor rival, AMD. Intel, for its part, is reportedly underwriting efforts to develop applications for the Linux platform in China, India, and Brazil.

Microsoft's relationship with makers of branded PCs is simpler. Virtually all PCs are now sold with an operating system installed. Microsoft provides information to PC makers on how changes in the operating

39. AMD had already been making Intel compatible chips. It had a cross-license agreement with Intel until 1986, when Intel ended the contract. A lengthy legal battle between the two companies ensued. AMD, http://www.amd.com/us-en/Weblets/0,,7832_12670_12686,00.html; Shane Ran "Worldwide PC Processor 2004 Vendor Shares" (IDC report no. 33398), May 2005.

system will affect the optimal design of the hardware, and it solicits feedback during the development process. Manufacturers pay license fees to Microsoft, which they pass on, as they would any costs, to end users as part of the price of the box.

Microsoft offers some discounts on its licensing fees in return for computer makers doing certain things that improve the overall quality of the entire platform. For example, Microsoft provided a small discount to computer makers in 1996 to give them incentives to install USB ports on their computers. Microsoft benefited from these incentives: USB ports promoted the addition of various peripherals that Windows would support, and that made Windows a more valuable platform. Of course, the computer makers and peripheral manufacturers in aggregate benefited from additional sales. But none of them individually had the incentive to promote the inclusion of USB ports. Microsoft as the maestro of the multisided platform had both the incentive to subsidize the inclusion of USB ports and the ability to do so.

These sorts of financial incentives are only one aspect of the platform strategy to get customers on board. As we noted earlier, in the mid-1980s Microsoft pressed hard to accelerate the development of CD-ROM technology as a cornerstone of multimedia computers. It held annual developers conferences for interested parties, worked with major manufacturers to create an industry-standard format, and evangelized computer makers to package built-in CD-ROM drives with new machines. Beginning with Windows 95, Microsoft has included code to create a relatively seamless "plug-and-play" experience with thousands of peripheral devices.[40] As we will see throughout this book, this sort of platform management is hardly unique to Microsoft. Most software platforms engage in similar activities, if not always with Microsoft's drive and skill.

Platform Pricing and Hardware Integration

Pricing is key for getting customers on board a platform and harnessing network effects to increase its size, as we saw in Chapter 3. The PC

40. Randall E. Stross, *The Microsoft Way* (Reading, Mass.: Addison-Wesley, 1996), p. 65; http://searchwin2000.techtarget.com/sDefinition/0,,sid1_gci212799,00.html.

industry quickly settled on a particular *pricing structure*. Virtually all revenue and profit have come from end users, not from the businesses that have relied on the services provided by APIs to write applications. No commercial maker of PC software platforms—whether integrated into hardware, as was the case with Apple and Atari, or sold separately, as was the case with Microsoft, Digital Research, and IBM—has tried to make money from application developers. So the "end user pays/the developer gets a free ride" pricing structure has held firm for more than a quarter of a century over several significant shifts in the industry.

It took longer to settle on *pricing methods*. Several of the early operating system companies licensed their code to manufacturers for a flat fee and allowed the manufacturers to modify the source code for their machines. Microsoft took a different approach. It licensed the binary code on a per-machine basis. Neither computer makers nor end users could modify the software platform easily. Apple took yet another approach. It didn't license its operating system at all (with the exception of a short period in the late 1990s). Nor did it make the source code available for modification.

The different makers of operating systems also took very different approaches to the *price levels* they were charging. We already saw the stark contrast between Microsoft and CP/M for the early IBM PCs. Later, IBM initially priced OS/2 at $325, compared with Microsoft Windows 3.0 at $149. A more interesting although difficult comparison is between Microsoft and Apple, since Apple's operating system generally comes bundled with its hardware, with no separate price. However, there is a clue: the 1990 upgrade to Windows 3.0 was $50, about half the price ($99) of a 1991 upgrade to Apple's System 7.0. Another useful clue comes from a comparison between computers with similar hardware: in this same period the average price of an Apple PC was over $200 more than the average price of a similarly equipped and powerful Compaq PC sold with Microsoft operating systems.[41]

41. Campbell-Kelly, *From Airline Reservations to Sonic the Hedgehog*, p. 250; "Microsoft Corp.: Windows 3.0 Is Here," *Business Wire*, May 22, 1990; Ron Wolf, "Apple Begins Shipping Long-Awaited System 7.0 Operating System," *Austin American-Statesman*, May 13, 1991; Dataquest, "Personal Computers U.S. Vendor Segmentation: 1998," April 19, 1999.

Thus, it appears that Microsoft chose a low-price strategy relative both to other stand-alone operating system vendors and to sellers of integrated software-hardware platforms. This encouraged computer makers to sell more machines with Microsoft's operating systems installed. Competition among them forced hardware prices down further. That extended the pricing advantage of the DOS/Intel and later Windows/Intel computer platform.

We examine the determinants of these pricing choices—and why they differed dramatically from the choices made in the video game industry—in Chapter 10. In the next chapter we learn that the video game console platform took a "developer pays/console user gets a cheap ride" pricing strategy.

Bundling

Early PC operating systems did relatively little, just managing basic functions like the input and output of data and the loading and execution of applications. Operating systems were only as capable as the computers on which they ran and accordingly provided only a fraction of functionality of their modern counterparts. But as computer technology advanced, operating systems expanded their reach—often into areas previously served by applications software made by others. For example, all modern PC operating systems include code for applications as basic as arithmetic calculators and as advanced as automating connection to networks.

Competition in PC operating systems has served to accelerate this trend. For example, in 1991, both the Mac OS and Windows integrated TrueType software for manipulating font sizes. Likewise, Apple offered QuickTime, a collection of multimedia functionalities, as a free add-on for the Mac OS in 1991; Microsoft followed with Video for Windows in 1994. Both Apple and Microsoft added peer-to-peer networking features in 1991 and 1992, respectively. IBM scored a first by adding an Internet browser to OS/2 Warp in 1994. Microsoft, playing catch-up with IBM as well as the independent release of Netscape Navigator, offered a Windows browser in 1995, based largely on code licensed from Spyglass. Microsoft included Outlook Express email software with Windows 2000, and Apple

followed with its Mail email client in the first release of the Mac OS X, early in 2001. Both Windows XP (2001) and the "Jaguar" version of the Mac OS X (2002) included instant messaging applications.[42]

The Linux platform for desktop PCs does not have a single manager with competitive incentives to bundle applications. However, all the companies that package and support Linux include most of the features now bundled with Windows and the Mac OS X. Indeed, because many Linux applications are free, Linux distributors often include more. Novell's Desktop 9 distribution, for instance, includes an Office-like productivity suite as well as instant messaging software that is compatible with AOL, MSN, and Yahoo IM applications. Red Hat does the same.

The Platforms in Perspective

Microsoft came to dominate PC platforms in the 1990s by pricing low and by capitalizing on what we now see as the strategic errors of others in the 1980s. Network effects associated with DOS's head start may well have given the company a competitive advantage. But Microsoft was able to translate that advantage into success only by understanding what it took to nurture its multisided platform and acting decisively on that knowledge to bring application developers on board and keep them there.

After some very rough patches, Apple has managed to stabilize revenues from its tightly integrated hardware-software platform and may

42. http://en.wikipedia.org/wiki/System_7_(Macintosh); "Windows 3.1: What's New Is for the Users on Networks," *LAN Times*, April 6, 1992; http://www.macos.utah.edu/Documentation/MacOSXClasses/macosxone/macintosh.html; http://www.microsoft.com/windows/windowsmedia/press/dmtimeline.aspx; http://support.microsoft.com/default.aspx?scid=kb;EN-US;q126746; http://www.macos.utah.edu/Documentation/MacOSXClasses/macosxone/macintosh.html; http://channel9.msdn.com/ShowPost.aspx?PostID=10049; http://en.wikipedia.org/wiki/Spyglass; http://en.wikipedia.org/wiki/Spyglass; "Outlook Express," *Internet Magazine*, May 1, 2000; http://en.wikipedia.org/wiki/Mac_OS_X_v10.0; Joe Wilcox, "Apple to Unleash Jaguar OS Upgrade," *CNET News.com*, August 29, 2002 (http://news.com.com/Apple+to+unleash+Jaguar+OS+upgrade/2100-1001_3-955063.html); http://en.wikipedia.org/wiki/Windows_xp#Windows_XP_Starter_Edition.

yet be able to profit from its inherent strengths. By controlling both the hardware and the operating system, it has been able to produce an exceptionally benign computing environment that attracts nonbusiness users who are prepared to pay more for handsome design and superior stability. And it may find ways to leverage its great success in portable digital devices—specifically, the iPod—to the benefit of the Mac OS platform. Indeed, surveys of iPod users indicate that the "halo effect" from the iPod has given a very substantial boost to sales of computers based on the Mac OS X platform.[43]

Still, it would be folly to make predictions about the evolution of the PC platform competition in coming years with any confidence. For one thing, technological change—for example, greater penetration of broadband that made server-based platforms practical—could undermine today's PC platforms. Google looms large as of this writing. This advertising-supported search engine offers an extremely popular platform that seems to reside on what we call the Web but of course really resides on Google's vast array of Linux-based servers. Many take it for granted that this new firm, so different from Apple and Microsoft, could push these old warriors aside. We return to this in Chapter 12.

For another, regulation—or corporate response to the threat of greater regulation—could slow innovation in Windows the way it impaired innovation at IBM in the 1970s and 1980s. Microsoft is subject to regulation stemming from adverse antitrust decisions in both the United States and the European Union—economies that together account for 59 percent of the world's gross domestic product and at least 70 percent of Microsoft's sales.[44] The U.S. regulation expires in 2007, while the EU regulation is perpetual, although it is the subject of an ongoing legal appeal. Then again, Microsoft's rival of the moment, Google, is coming

43. Daniel Drew Turner, "Apple Could See Near Doubling of Market Share," *eWeek.com*, March 22, 2005 (http://www.eweek.com/article2/0,1759,1778538,00.asp).

44. http://en.wikipedia.org/wiki/List_of_countries_by_GDP_%28nominal%29; http://www.sec.gov/Archives/edgar/data/789019/000119312505174825/d10k.htm.

under scrutiny from many quarters, and that could slow any challenge it might make to Microsoft.

Yet another wild card, to which we now turn, is the Linux open-source platform, which has made great inroads in server software but is only now beginning to make a dent in platforms geared for client computers in business uses.

The New Challenger with No Owner: Linux

Windows' tens of millions of lines of code are a well-guarded trade secret. Until recently, only carefully screened outsiders—major software and hardware developers, along with government experts seeking to uncover security flaws in the code—ever got to see the proprietary code, and then only under strict conditions of secrecy. Apple used some publicly available code as a key building block for the Macintosh OS X platform, and the company made portions of Mac OS X available to the programming community, both as a gesture of goodwill and as an enticement to develop applications for the platform. But vital features of the operating system remain secret, including Apple's GUI and the code that makes the operating system compatible with earlier generations of Apple applications. Windows and the Mac OS, moreover, are both owned and carefully managed by companies that seek a return on their investments in these systems.

There is an alternative model, though. Open-source software, designed and maintained by volunteer programmers, has been successful in several areas[45]—something of a surprise in light of the worldwide success of market-driven incentives and the general failure of communal production. For example, the Apache Web Server is widely used on standard server computer platforms, as well as being distributed with major proprietary operating systems such as Sun's Solaris. And

45. Surprisingly at least to economists, who have long assumed that profit incentives were critical to the design of modern software. David Evans, "Is Free Software the Wave of the Future?" *Milken Institute Review* (4th Quarter, 2001); Josh Lerner and Jean Tirole, "The Scope of Open Source Licensing," NBER working paper, 2002.

many have argued in recent years that the most potent competitive threat faced by Windows comes not from Apple but from a PC operating system that is built entirely from open, publicly available source code and that, accordingly, nobody owns or manages for a profit: Linux. Linux has already secured a strong presence as an operating system for server computers—it had a 12 percent share of paid shipments (from firms like Red Hat) in 2004, along with a large but unknown number of free downloads, and is clearly a significant competitor for Microsoft, Sun, Novell, and other companies in that business.[46] The big open question is whether this un-owned, open-source alternative will evolve into a major multisided platform for desktop PCs that competes successfully with Windows and the Macintosh operating systems.

An early version of Linux, a rudimentary kernel, appeared in 1991 for use with Intel 386-compatible hardware. By 1994, with the release of Linux 1.0, it had evolved into a full-fledged operating system. Unlike the other multisided OS platforms, however, Linux had—and has—no corporate parent to guide its development or to evangelize about either the development of applications software or the development of device drivers to make it compatible with peripheral equipment. Instead, hundreds of open-source enthusiasts, loosely organized and uncompensated, have both directed and executed the work of enhancing the Linux platform. One can debate its merits, but one can't dispute that many sophisticated users choose it for a significant number of important tasks.

Over the years, Linux has made an important transition from an operating system created mainly by volunteers with little money to gain from its success to one that is supported by many employees of companies that do have money to gain. As Linux's popularity has grown, technology companies have started contributing code and evangelizing the operating system. As of 2006, large companies, including IBM, Computer Associates, and HP, contribute code to Linux development. IBM has modified Linux for use with its zSeries mainframes. Similarly, Intel is working to

46. "Worldwide Client and Server Operating Environments 2005–2009, Forecast: Modest Growth Ahead" (IDC report no. 34599), December 2005, table 2.

make its chips and Linux fully compatible, presumably in order to sell chips for "Lintel" equipment and to reduce its dependence on the Windows software platforms.

Customers who choose Linux today are different from those who choose proprietary operating systems for servers such as Sun's Solaris or Windows. Numerous information technology specialists in charge of corporate and government networks have embraced Linux. They can customize Linux to their own needs, since they can see and alter its source code. IT specialists are able to fix bugs in Linux without help or permission from the licensor. Desktop users, on the other hand, rarely benefit from access to the operating system's source code. Also, volunteer programmers don't have as much incentive as salaried employees to do the more mundane work necessary to make a desktop operating system easier to use—for example, writing the device drivers necessary for the thousands of peripherals available for PCs to be compatible with the operating system or making the user interface easy for novices to master. With the increasing popularity of Linux, however, some vendors of peripheral equipment are starting to fill the device driver void themselves, a process that is arguably made easier by easy access to the source code. But until enough desktop users demand such drivers for Linux, most of the burden of producing them will continue to fall on volunteers, making it more difficult to balance the sides of the Linux platform. The challenge of producing an easy-to-use graphical interface seems to have proven even more difficult.

To date, Linux has yet to dent the market for desktop operating systems, with only a 2 percent share of such systems delivered in 2004. Still, that's doing about as well as Apple, which has been around much longer.[47] If the un-owned Linux desktop platform does continue to make headway against the competition, it seems most likely that the advance will be led by large businesses that have long experience with Linux servers and that can provide internal support for free applications. Or it may come in rapidly emerging economies such as South Korea, where the government subsidizes Linux applications development, or China,

47. Ibid.

where mandatory use of Chinese-produced software by government agencies has motivated adoption of Linux by many local and national organizations.[48]

In any case, the success that Linux has attained has interesting implications for those contemplating developing software platforms. On the one hand, it provides an alternative production model that has obviously achieved some successes. Many companies are learning from the Linux experience. On the other hand, it is a bit scary to proprietary software firms and their backers. One wouldn't have thought that an un-owned platform that is free to all sides of the market could have taken almost 20 percent of the server business in competition with Microsoft, Novell, and Sun in about a decade.

Yet open source hasn't had any significant impact yet on video games, personal digital assistants, mobile telephones, or digital devices. We therefore won't see it mentioned much in what follows. We turn next to a hardware-software platform that looks almost identical to PCs but has evolved very differently—video game consoles.

INSIGHTS

- The PC software platform has changed the way computer power is delivered to businesses and consumers. Today it coordinates a nonintegrated and decentralized process in which separate firms deliver hardware, peripherals, applications, and software platforms.
- Killer applications were important for the early success of PC software platforms; many people bought systems because they could run a particular killer application, such as VisiCalc. That in turn stimulated more applications developers to write for the underlying software platform.
- Platform "evangelists" were also crucial to the success of PC platforms. They helped persuade independent developers to write applications

48. http://news.com.com/China+Local+software+for+local+people/2100-7344_3-5951629.html; http://linux.slashdot.org/linux/05/03/29/0322248.shtml?tid=163&tid=190&tid=106.

for the platform. Evangelism went hand-in-hand with the development of software services for developers that were made available through APIs.

- All commercial PC software platform vendors have adopted the "charge users/let developers free-ride" pricing structure in order to encourage software developers to write applications for their platform.
- Four key strategies helped Microsoft obtain the leading position in personal computers: (1) offering lower prices to users than its competitors; (2) intensely promoting API-based software services to developers; (3) promoting the development of peripherals, sometimes through direct subsidies, in order to increase the value of the Windows platform to developers and users; and (4) continually developing software services that provide value to developers directly and to end users indirectly.

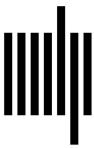
Invisible Engines

How Software Platforms Drive Innovation and Transform Industries

By: David S. Evans, Andrei Hagiu, Richard Schmalensee

Citation:

Invisible Engines: How Software Platforms Drive Innovation and Transform Industries
By: David S. Evans, Andrei Hagiu, Richard Schmalensee
DOI: 10.7551/mitpress/3959.001.0001
ISBN (electronic): 9780262272421
Publisher: The MIT Press
Published: 2008



The MIT Press

MIT Press books may be purchased at special quantity discounts for business or sales promotional use. For information, please email special_sales@mitpress.mit.edu or write to Special Sales Department, The MIT Press, 55 Hayward Street, Cambridge, MA 02142.

This book was set in Sabon by SNP Best-set Typesetter Ltd., Hong Kong.
Printed and bound in the United States of America.

Library of Congress Cataloging-in-Publication Data

Evans, David S. (David Sparks)
Invisible engines : how software platforms drive innovation and transform industries / David S. Evans, Andrei Hagiu, and Richard Schmalensee.
p. cm.
Includes bibliographical references and index.
ISBN 0-262-05085-4 (alk. paper)
1. Application program interfaces (Computer software). 2. Industries—Data processing. I. Hagiu, Andrei. II. Schmalensee, Richard. III. Title.
QA76.76.A63 E93 2006
005.3—dc22

2006046629

10 9 8 7 6 5 4 3 2 1