Harvard Business School
Case Selections (Reprint)

哈佛商学案例精选集

（英文影印版）

实务系列
From the Field

商用和家用软件业

INSIDE Business and Consumer Software

中国人民大学出版社

Harvard Business School
Case Selections (Reprint)

哈佛商学案例精选集
（英文影印版）

实务系列
From the Field

商用和家用软件业

INSIDE Business and Consumer Software

中国人民大学出版社

图书在版编目（CIP）数据

商用和家用软件业/A. 麦科马克等编写
北京：中国人民大学出版社，2002
（哈佛商学案例精选集. 实务系列）

ISBN 7-300-04173-6/F·1290
Ⅰ. 商…
Ⅱ. 麦…
Ⅲ. 软件-电子计算机工业-英文
Ⅳ. F407.67

中国版本图书馆 CIP 数据核字（2002）第 037937 号

哈佛商学案例精选集（英文影印版）
实务系列
Inside Business and Consumer Software
商用和家用软件业
A. MacCormack　　A. 麦科马克 等 编写

出版发行：中国人民大学出版社
（北京中关村大街 31 号　邮编 100080）
邮购部：62515351　门市部：62514148
总编室：62511242　出版部：62511239
本社网址：www.crup.com.cn
人大教研网：www.ttrnet.com
经　　销：新华书店
印　　刷：涿州市星河印刷厂

开本：890×1240 毫米 1/16　印张：9.75 插页 2
2002 年 9 月第 1 版　2002 年 9 月第 1 次印刷
字数：343 000

定价：29.00 元
（图书出现印装问题，本社负责调换）

From the Field

INSIDE BUSINESS AND CONSUMER SOFTWARE

Harvard Business School Publishing

SERIES INTRODUCTION

Welcome to this entry in the *From the Field* series of case collections from HBS Publishing. We have three main objectives for this series:

To enrich readers' understanding of business by presenting coherent collections of field-based research published by Harvard Business School. Understanding business involves much more than earnings reports and news headlines. It means understanding how managers perceive and analyze the complex challenges their companies face and the strategies and tactics they devise in response. For nearly a century HBS has been researching the world of managers from *inside* companies and delivering their stories to facilitate superior teaching and learning. You won't find easy answers or quick fixes in these cases, but you will discover balanced, detailed pictures of industries, markets, and technologies, and the intelligent professionals who – like you – are trying to cope with them.

To focus on the latest HBS work on tumultuous, fast-growth industries. Today's companies are fast-moving targets. We keep our *From the Field* collections current by concentrating only on newer case studies. And the series is focused on rapidly evolving industries with business practices that no one could have foreseen even a decade ago.

To guide readers to Web-based sources of information that can supply good supplemental information. Using the best Web resources you can find out "what happened next" in the companies that our case studies examine – and, if you dig in the right places, *why* it happened. We get you started by steering you to sites where you can learn more. As the *From the Field* series develops we will enrich and refine our Web guidance, partly based on suggestions we receive from readers such as you.

Harvard Business School Publishing has additional resources – including more case studies – on all the topics covered in the *From the Field* series. To research other products, to learn about other titles in this series, or to order additional copies of this reader, call 1-800-545-7685 or visit our Web site at www.hbsp.harvard.edu.

The Editors

INTRODUCTION TO THIS COLLECTION

The items in this collection have been chosen because they reveal important aspects of today's business and consumer software industry. Each item is preceded by a summary. Following each item is a set of "Questions and Ideas to Consider" that we hope will stimulate your thinking as well as drive you to the Internet to do your own research.

We open this collection with a case detailing the development of Microsoft's Office 2000 product, and follow with cases on two other large, well-known companies, Oracle and Adobe. Each of these cases depicts specific evolutionary challenges confronting well-established leaders in their respective software domains.

We move on to a case on Novell, which sounds a note of caution for companies seeking to take on powerful leaders like Microsoft.

Next are two cases about younger, less well-known companies who nevertheless have established leadership positions: i2 in supply-chain management and Kana in online communications software.

Our seventh and final case, on Harley-Davidson, represents a shift in perspective, presenting a well-known industrial company's systematic search for the right enterprise software solution.

Did You Know?

HBS Publishing has many other recent cases on business and consumer software companies (and their customers). Here is just a small sampling of cases published since 1999:

- Red Hat and the Linux Revolution 600-009
- Microsoft: Competing on Talent 300-001
- @Hoc: Leveraging Israeli Technology in the United States 800-264
- Trilogy Software (A) & (B) 699-034; 600-123
- Inktomi: Scaling the Internet 699-156
- Computer Associates: Securing the Internet 799-087

To get information on these and other HBS Publishing materials, visit our Web site: www.hbsp.harvard.edu. In the search field, enter "software." You might be surprised by how much you'll find!

RESEARCHING COMPANIES ON THE WEB

As you read the cases in this collection we are sure you will want to conduct research using a variety of Internet sources. Obviously, it always makes sense to visit the Web sites of companies profiled in these cases, since that is often the handiest way to gather basic information about current lines of business, marketing campaigns, and recent financial performance. But there is a wealth of information available on other sites, too. Below we list a number of Web sites that provide information about public companies, much of which is available free of charge.

Business-information sites we've come to like:

- Hoovers.com for basic company profiles, including lists of key subsidiaries, executives, and competitors.
- The "News and Media" section of hotbot.com, a regularly updated archive of items from many news sources.
- Kompass.com for information on foreign companies.
- For information on and discussions of technology companies, magazines run some of the most useful sites, including redherring.com and thestandard.com.
- Quicken.com, Smartmoney.com, Dowjones.com, and the "Business and Finance" section of Yahoo.com, for clear, readable presentations of key financial performance data and access to useful screening tools.
- CBS Marketwatch.com or – by paid subscription – wsj.com, for breaking financial news.

A final note about currency: At certain points we will tell you what we found at particular Web sites while we were putting this collection together. We apologize for any out-of-date directions and "dead links" you may find, but such is the transitory nature of certain information on the Web.

CONTENTS

MICROSOFT OFFICE 2000

(A. MacCormack / #9-600-097 / 18 p)

Summary

Describes the history of Microsoft's Office product suite. Discusses evolution of the Office 2000 project. Set at the end of the project when Steven Sinofsky, Office vice president, must decide upon the direction for the next version of Office, as well as make changes to the process.

Harvard Business School

9-600-097
Rev. June 20, 2000

Microsoft Office 2000

VIDEO CLIP 1: OFFICE 2000 RELEASE PARTY[1]

Covered in champagne from head to foot, Steven Sinofsky, vice president, Microsoft Office, squeezed his eyes tightly shut as he sailed head-first into the fountain behind Building 17, headquarters of the Office development team. As he braced for impact with the freezing water, he couldn't help but reflect on how far the team had come. At the start of the Office 2000 project, many people had questioned the future of Microsoft's star product in a world increasingly dominated by the rise of the Internet. Now, two years later, it looked like Microsoft had once again beaten the odds and developed another winner, at least if the early customer feedback proved representative of the broader market.

Yet, as he continued his downward arc towards the foaming ferment, his thoughts turned to the next release. If anything, the environment for Office had become even more uncertain than it was at the start of Office 2000. And while customer feedback from the Office Advisory Council had been positive, some of the early press reviews claimed the product was boring and still "bloated" with irrelevant functionality. Adding to the cloudy picture, Sinofsky had been under pressure to make changes to the development process. Office 2000 had shipped eight months later than originally planned. Some developers felt the team had never been so unproductive, given the huge overhead involved in coordinating such a complex project. How should he address these complaints?

His thoughts were rudely interrupted as he entered the icy blue abyss and sank.

The History of Office

Microsoft Corporation: Early Beginnings

Microsoft Corporation was founded in 1975 by Bill Gates and Paul Allen, with a mission to create software for the personal computer (PC) that empowered and enriched people in the

[1] Video clips cited throughout the text refer to interviews available on HBS Video 600-502.

Professor Alan MacCormack and Research Associate Kerry Herman prepared this case as the basis for class discussion rather than to illustrate either effective or ineffective handling of an administrative situation. *This paper version of Microsoft Office 2000 (Multimedia) is intended to be used in conjunction with video interviews, found on HBS Video 600-502.*

workplace, at school, and at home.[2] Microsoft's big break came in 1980 when IBM approached the company to develop the operating system for its personal computer. The result was MS-DOS (Microsoft Disk Operating System), a copy of which accompanied virtually every IBM-compatible personal computer sold. DOS gave Microsoft a large, solid customer base on which to grow, and soon thereafter the company added software applications such as word processors and spreadsheets to its product line. From 1980 to 1989, Microsoft's annual sales increased from less than $1 million to over $800 million. The company's employment grew from 45 people to more than 4,000. By 1990, Microsoft was the largest personal computer software company in the world, with applications accounting for over half its revenues. It had achieved a dominant position in the market for Macintosh software, and was a solid second in most applications for the PC.

The Evolution of the Development Organization

The early days at Microsoft were characterized by an approach to development described by Jon DeVaan, vice president in charge of Office 2000, as the "development herd."

> There was one central team which was in charge of developing all our applications. Once a project team finished developing the latest release of Word, they'd jump ship and start working on Excel. And then on to PowerPoint, and so on. So everyone worked on everything. It was fun for a while, but eventually we realized that we needed more focus if we wanted to produce leading-edge products.

During this early period, the development process at Microsoft was very informal, with little emphasis on keeping to schedules or following a prescribed software development methodology. Stories about legendary developers abounded:

> There was one guy who could type 80 words a minute. That's pretty impressive, but what's really impressive was that he actually wrote code at that speed. He'd write a 10,000-line application in two days, then if it didn't work, he'd throw it out and start again from scratch. He'd go through this process two or three times, until he ended up with a working program. Not only did it take him less time to do this than if he had sat down and tried to think everything out in advance, but the program that resulted was better too. Because he had implemented the same program several times before, he knew how to avoid all the pitfalls, so his code became very clean.

Another of Microsoft's early developers described his own views about software design:

> I design user interfaces to please an audience of one. I write them for me. If I'm happy, I know some cool people will like it. Designing user interfaces by committee does not work very well; they need to be coherent. As for schedule, I'm not interested in schedules; did any one care when *War and Peace* came out?

Such an approach, however, often led to software that, though technically excellent, was difficult to use. In addition, relying on superstar programmers was not seen as a viable long-term strategy. As one general manager described at the time:

---

[2] This section draws extensively from "Microsoft Corporation: Office Business Unit," HBS case No. 691-033, rev. May 31, 1994 and "Microsoft Office: Finding the Suite Spot," HBS case No. 699-046, rev. February 10, 1999. A timeline of Microsoft's history can be found at http://www.microsoft.com/MSCorp/Museum/timelines/microsoft/timeline.asp?theDir=Microsoft

> There are a lot of problems with relying on individual superstars: they are in very short supply; someone has to maintain and update the software they've written (which doesn't interest them) and often other people have difficulty understanding their code; sometimes they don't understand what the market wants; and finally, if you try to put several of them on the same project, you get real problems with design decisions—"too many cooks spoil the broth."

The first move to gain more control and specialization in its development efforts happened in August 1988, with the formation of the applications division. Within this division, a number of specialized business units were created, each focused on a specific application. For example, the Office Business Unit developed and marketed all of Microsoft's business word processors (PC Word, MacWord, and Word for Windows) and the Excel Business Unit engineered and marketed Microsoft's spreadsheet products. This arrangement allowed teams to focus on the specific needs of only one application, and the different competitors each faced in the market. Each business unit contained all the functions necessary to develop, test, and launch its products, and provided marketing support to the field-based sales team. An *esprit de corps* began to develop in each unit, along with a healthy rivalry between business unit teams.

Alongside these changes, Microsoft began to formalize a new role in the project team—that of program manager. The role of the program manager was to integrate and coordinate the efforts of everyone involved in the project, and also to take responsibility for the vision and specification of the product. Program managers shared the lead in a project with several other positions, including: software design engineers (SDEs), software test engineers (STEs), product managers, software localization engineers, and user-assistance. The development manager, who managed the software design engineers, was responsible for overseeing the development effort, including handing out programming task assignments, scheduling, and coordinating the development effort. There was usually a single SDE who served as the technical lead and had final say in all technical decisions, code reviews, and programming standards. Product managers handled all the marketing issues such as competitive analysis, positioning, packaging, and advertising. The on-line and print-based leads handled the user-assistance functions and the localization lead oversaw the customization of the program for various international markets. These people were supposed to work as a group, with no one person having total authority.

The Genesis of Office

The concept of an office suite of products was developed by the team working on Macintosh applications. They had noticed that customers purchasing both Word and Excel had to complete these transactions separately. At the time, customer research showed that most people used multiple applications, thus was born the idea of better integrating the features and user-interface of these applications so that the product more closely reflected customer usage patterns. The resulting product, MS Office for Macintosh, was released in 1989, and included Microsoft Word, Microsoft Excel, the newly released Microsoft PowerPoint, and MS Mail. The suite retailed for $895, customers responded favorably, and the overall applications business grew.

During 1991, Microsoft began making plans for an Office version targeted at Windows users. While the market for applications was still dominated by MS-DOS products, Windows 3.0 was beginning to gain momentum. Windows promised to give the same ease of use to PC users that Macintosh users had enjoyed for many years. But without a set of applications to run on the system, it was unlikely Windows would make much progress. Microsoft wanted to be the first to introduce a suite of products to run on its new flagship operating system.

The plan for the first version of Office for Windows was to integrate the current generation of products that were being shipped by the individual application groups. Creating the integrated office suite was handled by a small group within the applications division—the Application

Interoperability Group (AIG). This group, a handful of program managers with some development and testing resources, created the Microsoft Office Manager, a small floating window that allowed users to easily launch each application and attempted to generate some consistency in user-interface elements like the toolbar. Because these features were implemented differently in each application, AIG had to negotiate with developers in each group to bring together the integrated product.

The first version of Office, Office 3.0 was finally released in late 1992; it included the second release of Word for Windows (2.0). While the product was viewed internally as a success, industry reviews for the product tended to see it as a suite of "second-place" applications; this was especially true when compared to the MS-DOS competitors that were deeply entrenched in each individual application. As a result, sales of Office 3.0 accounted for only a small fraction of the group's revenue.

Microsoft Office 4.x

The planning cycle for the next release of Office was short—all that needed to be done was to decide a schedule upon which each application group would deliver its next version. Given that each of Office 3.0's individual applications had been completed at different times, the focus for the next version of each had already been determined. The Word team focused on "ease-of-use," reacting to feedback from the Word for Windows 2.0 project and the WordPerfect features Microsoft customers wanted; the Excel team focused on the display and analysis of complex information, targeting the sophisticated user who needed to employ custom-programmable solutions; and the PowerPoint team (based off-site in California) focused on a new technology called Object Linking and Embedding (OLE), which enabled one application to embed data and files into another.

Despite the apparent simplicity of these plans, however, development of the individual applications that comprised Office 4.0 encountered many problems. Word 6.0 shipped six months late, and Excel 5.0, which started significantly later than the Word team, shipped three months late. When the time came to launch Office 4.0, only the new version of Word was available. Microsoft decided to release the product with old versions of Excel and PowerPoint. Customers who bought Office 4.0 were given coupons allowing them to upgrade when new versions of these applications became available. This was not a popular decision, and customers expressed disappointment with the release. Even when the complete version of Office—Office 4.2—was released in April 1994, it still seemed buggy and unstable. The product did not fully achieve the targets the team had set for it until the third maintenance version was released (Office 4.2c) in October 1994.

Changes for Office 95/97

Office 4.2 became an incredible success in the market, with sales of the product for the first time outpacing those of the individual applications. It was clear, however, that the process of development needed attention, given the bitter conflicts that had emerged between the application teams during development as they wrestled with user-interface consistency. These had been especially apparent between the Word and Excel groups, where two infamous debates flared. In one instance, the height of toolbars became a harshly contested battle: one team felt that toolbars should be 15 pixels high, the other team was committed to 16; Office released with two different visuals. The second conflict was over cascading menus: the Excel team claimed these were a good user experience, the Word team was adamantly opposed; so Office shipped with different menu organizations.

Another factor adding to the call for change in organizing Office development was the emergence of a strong competitor—SmartSuite from Lotus Corporation. SmartSuite had been introduced in 1992, to a disappointing reception. The revised release, however, combined Lotus's new word processor, Ami-Pro (acquired from Samna Corporation); the Windows version of its best selling Lotus 1-2-3 spreadsheet; and the Freelance Graphics presentations program. In contrast to

Microsoft's vision for Office, long targeted at individual productivity, Lotus touted SmartSuite as a workgroup suite and began to define a new product category called workgroup software.

As a response to these challenges, Microsoft made two major changes to the Office organization. First of all, it decided that all the applications contained in Office would have to follow the same development timeline, significantly increasing the coordination that would be needed across units. Secondly, it introduced a new unit within the applications division—the Office Products Unit (OPU)—staffed by "volunteers" from the applications units. The purpose of this unit was to "oversee the shared customer experience." Ownership of the shared application internals (memory management, for example), shared graphics and drawing tools, shared user interface (such as command bars—the bars seen at the top of each application—and the Office Assistant), along with ownership of the overall development process, was given to OPU, which would create shared code to be included in each application. Integrating this code would be the job of "Feature Integration Teams," located within each business unit. OPU was allocated about twice the number of developers working within each of the application teams.

The first challenge for the OPU team was to create a version of Office which would ship relatively soon after the new version of Windows due out in mid-1995. This was a difficult task, given that the new version of Windows incorporated a shift from a 16-bit to a 32-bit architecture, in order to take advantage of Intel's latest microprocessor. In order to meet this deadline, it was decided that the product would focus mainly on re-architecting the products so they performed optimally with the new version of Windows, avoiding major feature changes. This required only a small team of developers, hence the bulk of development resources was applied to the next major release for Office—Office 97—scheduled to ship in June 1996.

Office 95 shipped on the same day as Windows 95. The success of the product was bolstered by the fact that none of the competing suites had delivered 32-bit applications by that time. Office 97, however, became a long drawn out affair, with much tension between the OPU and the application teams. Although these teams had shared code, they had not done so willingly or in an optimal fashion. After the product shipped in November 1996, the subsequent postmortem was quite intense and bitter. Comments like these below were commonplace:

> I felt like we were all working to our own objectives. Nobody had any idea what the overall product was supposed to do. There was one vision for Word, one for Excel, one for PowerPoint, etc., but none of these seemed to bear any relation to each other. Sure, there was a document, but that was just created using the cut and paste function. What we needed to know was "What is the vision for Office?" Without this overall direction, each of the application teams just continued making decisions based on their own perspectives.

> Writing shared features in the OPU seemed like a good idea. We'd write the code, test it, and hand it to the integration teams in each application unit. But then later, when we wanted to make some changes or improvements to this code, we needed them to be responsive. And quite frankly, they had their own agendas. In one unit, they'd give it a low priority relative to other tasks, so we'd have to wait. In another, they'd be able to do three out of five changes we wanted. And in another, it might be three again, but a different three. It drove me crazy. Every time I went to a meeting, I felt like the unwanted uncle who comes to Thanksgiving dinner.

The Office 2000 Project

As the Office 2000 team geared up to start in the fall of 1996, Microsoft had registered its twenty-first successive year of growth since its inception. The success of the Office business had helped propel sales to a record $5.94B the previous year. Building on this momentum, Office 2000 was set to be the most comprehensive productivity suite Microsoft had ever assembled, encompassing 10 primary applications, all of which would ship on the same day (see **Exhibit 1**).

The Market Context

Reviews of Office 97 in the mainstream press had been mixed. In one of the more pointed critiques, Stephen Manes of *The New York Times* wrote about Office as "an upgraded leviathan" about to set sail.[3] The main target for his humor was the large footprint (i.e., space) the product occupied on a computer's hard drive—for a typical installation, more than 130 megabytes (though this was only a slight increase from the 100MB required for Office 95). No one customer was likely to use more than a fraction of the functionality that Office contained, hence many dismissed the new release as "bloatware." The problem for Microsoft, however, was that each type of customer typically used a different fraction of the functionality (e.g. the *Times* reviewer applauded an upgrade to the character counter—a feature which few users other than journalists had cause to celebrate). Avoiding a large footprint while at the same time satisfying a diverse range of customers appeared to be an impossible proposition.

The most substantial challenge to Office's dominant position, however, was not to come from the size of its footprint. It was to come from the rapid emergence of a technology which had the potential to revolutionize the way software applications were delivered to the customer—the Internet. By late 1996, as the Office 97 project was drawing to a close, the euphoria surrounding the Internet was at its peak. The "browser wars," a popular name for the battle between Netscape and Microsoft to develop the leading Internet-browsing application, in full swing. And surprisingly, many industry commentators were voicing public doubt about Microsoft's ability to compete effectively in this new arena:

> To consider the decline of Microsoft today seems almost perverse. After all, the company has never done better...Yet IBM also looked good when the PC was born in 1981. The Internet amounts to another such revolution...It will turn Microsoft's operating system advantages into so much surplus baggage, slowing it as nimble newcomers sprint ahead....[4]

According to analysts, the most pointed threat to Office arising from the Internet was based upon the promise of a new platform-independent programming language called "Java." Developed by Sun Microsystems, Java allowed programmers to develop software which would run on many different types of hardware (as opposed to writing software for each type individually). Using this technology, it was possible (at least in theory) to develop programs which could be sent to a user's PC only when they were needed, then disposed of once the need had passed. Storing such programs (called Java "applets") centrally rather than on each individual PC had the potential to make the management of software deployment and maintenance within large corporations much easier and cheaper. The difficulty in deploying and managing PCs equipped with productivity software was coming to represent the Achilles' heel of the Windows and Office solution.

---

[3] Stephen Manes, "Personal Computers; An Upgraded Leviathan Sets Sail," *The New York Times*,14 January 1997, C5.

[4] "A Survey of the Software Industry," *The Economist*, May 25, 1996.