

**Debra Vogt**

**From:** Bill Gates  
**To:** Pam Edstrom; Marty Taucher; Steve Ballmer; Jeff Raikes  
**Cc:** Mike Maples; Mike Murray; Bill Gates; Bill Neukom; Jonathan Lazarus; Pete Higgins; Paul Maritz; Nathan Myhrvold  
**Subject:** Microsoft and the Chinese Wall  
**Date:** Saturday, December 04, 1993 4:08PM

The Great Wall of China is so large it is the only man made item viewable by astronauts in space. Our Chinese wall never existed and yet is constantly cited in press article after press article. What is the truth about this Chinese wall?

[This is another piece relative to PR brainstorming that may go on in the future. These pieces are to stimulate thought and to help you position the company. Perhaps someday something written will come out of them.]

Many times I have been asked about a Chinese Wall. I don't know how this started. I know in dozens of cases I have said there is none and there shouldn't be one but there has never been an article that explained why. I believe the idea of a Chinese wall came up in investment banks making a market in a security and providing arms length analysis at the same time they are involved in an offering or a merger. This creates conflict because the investment banker is supposed to be independent while having a stake in a transactions success. In order to deal with this some separation was created. I don't know much about this.

When Microsoft is working on the design of a new piece of systems software it needs massive amounts of input from developers. What was wrong with the last version? What needs to be sped up? What features would they like to see included in the next version? This exchange of information is critical to the success of the operating system. Many of the new elements in the systems software are designed jointly or solely by applications groups. The boundary between applications and systems is constantly changing with more and more work being taken over in an abstract way by the system to simplify the creation of complex applications, allow for tighter sharing and integration and to allow abstractions that allow hardware to improve without forcing software redesign. User interface, printer drivers and data exchange were not part of the operating system in the DOS era. Applications developers with their own user interfaces were reluctant to switch to a standard but the benefit to users in learning and to the industry as a whole as users bought more applications per PC were incredible. Likewise application developers who had invested massively in printer driver libraries (including Microsoft's word processing group and Wordperfect) did not want to see Windows replace these drivers since it made those capabilities available to all developers. Over time, however, these groups recognized the inevitability of system advances and moved on to provide features in other areas. Today the boundary is advancing in messaging, rich data storage and networking and it will cause as much change as the ones we have seen in the past. In order to make these changes as smooth as possible an increased dialog between Microsoft and application developers is critical.

When Microsoft began promoting Windows in 1982 it spent a lot of time trying to convince the leading developers to do Windows work. The basic question was commitment to graphical interface. The primary leaders in software at the time, Lotus, Wordperfect and Ashton Tate did not show a strong interest in graphics on any platform - Macintosh, Windows or OS/2. Actually Lotus made a contractual commitment to port 123 under Mitch Kapers leadership but this was withdrawn after his departure. It was very risky for Microsoft to focus its development on graphical implementations. Our allocation of resources meant that DOS Word would lose out to DOS Wordperfect and DOS Multiplan would lose out to DOS 123. The effort to redesign for graphical interface is fairly hard. Moving from one graphical platform to another is not. This is why Microsoft was the first of these companies to have a spreadsheet and the first to have a word processor on all 3 of the graphical platforms. I think it is fair to say that Windows would have taken many more years to succeed if it wasn't for the risk and commitment that Microsoft's applications group made to these platforms. Certainly the Macintosh would have not been able to penetrate business the way it did without the presence of products like Word and Excel and the marketing behind those products.

Some people like to suggest that developers were confused about whether OS/2 or Windows would succeed. Microsoft certainly didn't have a crystal ball on this. We shipped Excel well before 1-2-3 shipped. We shipped Word way before anyone except Describe shipped a Word processor. Other than IBM we lost more money on Os/2 than anyone else in both systems and applications. Fortunately it is not hard to retarget a program written for one graphical platform to another and a number of libraries were available to make the task very straightforward.

Plaintiff's Exhibit

7703

Comes V. Microsoft

Page 112

EXHIBIT

Meyers  
24

MX 5064157  
CONFIDENTIAL

MS 5035892  
CONFIDENTIAL

FL AG 0101912  
CONFIDENTIAL

Microsoft's Windows applications have been very successful. Some surveys suggest we have as much as 28% of the total Windows software sales. However if we compare the % of software sales made on a platform by the owner of a systems software platform as a % of all sales of software on that platform I believe Microsoft lags most other platform creators. Specifically I am saying that IBM sold a higher percentage of the software onto MVS and DEC onto VMS and Apple onto Macintosh (excluding Microsoft) and SUN onto Solaris than Microsoft has sold on DOS/Windows. People who really want to understand how different platforms can be should examine Nintendo, 3DO and Sega. They not only have leading shares of software on their machine but they have a high degree of control and a royalty structure to boot.

The definition of a platform includes a lot of things. For example data exchange standards and user interface standards are crucial in allowing users to take advantage of the system. However the systems software does not enforce these approaches - any sort of appearance and data format can be presented by applications. Another important factor is that systems software is sold with a number of utilities and applications. These utilities and applications are often not as rich as those sold separately but they are important to the user experience.

So back to the Chinese Wall. Does Microsoft systems allow Microsoft applications to provide input on the next version of the system? Absolutely. Microsoft is the largest seller of applications on the Windows platform. Windows version 2 would have been an absolute failure without Excel in two ways. First, it was the feedback from Excel that allowed the system to be design properly and second it was Excel marketing that made people aware of Windows in a meaningful way. Does Microsoft systems give information to Microsoft applications about new versions? Absolutely. It is critical that this information is taken into account in determining what our plans should be.

There is no Chinese wall. Information is encouraged to flow in both directions. It happens on a formal basis and an informal basis everyday. I have no idea why these articles keep being printed throwing up this imaginary wall just to knock it down or suggest it has holes in it. We would be crazy to suppress this important dialog. Sometimes the dialog is confusing because our systems specifications and schedules change from time to time. Most ISVs choose to wait until a product is in widespread beta testing before they start to take advantage of its features. ISVs don't want to restrict their products to only sell to new system software users. They want to sell to the installed base and to users who don't choose the new version. The strong degree of compatibility Microsoft generally provides allows ISVs to focus on previous system software without giving up the ability to run on new systems.

An example of this would be our latest release of DOS, MS-DOS 6. At this point no programs except utility programs have chosen to release versions that run only with DOS 6 despite the extra testing required and the millions of upgrades that Microsoft has offered. Likewise with Windows, some Windows applications require Windows for Workgroups which is quickly becoming our fastest selling version but most simply require 3.1 or even the version before that 3.0. With our most advanced version of NT even now that the system is shipping it is a challenge to convince developers to take advantage of the new features. Microsoft published the information on NT over 2 years ago and spent millions of dollars providing very inexpensive development kits and support in order to encourage developers to focus on NT. Today there are over 250 shipping NT applications but that is a small percentage of the over 5000 Windows applications.

Going back to the Windows experience - when would developers have had to wake up and listen to our speeches about graphical interface in order to avoid it hurting their market share? They had the chance as early as 1992 before we began our development of applications. Assuming it takes around 18 months to develop a major new application they could have ignored being serious about Windows until at least 1988 without it being a major problem for them. Fortune magazine had its cover article in 1984 with Strat Sherman explaining that our strategy was Windows (quote). In other words people could have ignored Windows for over 5 years during which time they were being given every scrap of information and begged to pay attention. I have no idea why the transition to graphical interface proved to be so difficult for several of these vendors but it wasn't because of a lack of effort by Microsoft. We were totally open about our commitment to do applications on Windows and given the incredible profits they had during this time they should have at least had some insurance against Microsoft being right about GUI.

Information about new software is valuable but the critical decision for software companies is strategy. Imagine what would have happened if Improv had been integrated into a quality version of Windows 123 and shipped first on Windows? Imagine if LOTUS had done 123 for the Macintosh instead of JAZZ? Imagine what would have happened to Microsoft's applications group if I had been wrong about the volume NEXT would achieve and they had sold millions of units? Strategy is scary stuff but software CEOs are paid to make these decisions on a regular basis.

As long as I am striking down canards let me close with 2 others.

Does Microsoft get an advantage in applications markets by being the creator and risk taker with Windows? Yes we do. Our reputation has been enhanced by the success of Windows. We have a list of customer names we get from Windows. In a number of ways Windows benefits Microsoft. Our applications groups would respond that it is an incredible overhead for them to be forced to support every new system feature when the feature is not fully mature and they end up figuring it out and debugging things that would not be their normal priority. When the applications groups invent various user interface techniques or DDE we often take these and put them in the system making them easily available to all of their competitors. They would also suggest that competitors paint us into a Windows only corner because it is our product despite our strong support for Macintosh and UNIX environments. However on balance the benefits outweigh the negatives and ISVs should take this fully into account when they decide to develop on our platforms. Our platforms have been the leading platforms since 1982 and our way of handling distributing information effectively to developers has improved every year since then. I am sure they will continue to improve in the future.

Does Microsoft rely on hidden system calls to get an advantage for our applications? We do not. The success of our products like Word, Excel, and Office is based on the innovation of the respective development groups. One way to appreciate this is to look at our success of the Macintosh. Apple has had us in court and discriminated in providing developer information to us. However our greatest market shares are not in the very competitive Windows environment but rather on the Macintosh. The same features that we implement on Windows we implement on the Macintosh despite the lack of a perfect relationship with Apple. A developer can look at the way we call Windows with very straightforward utilities and despite some sloppiness in using somewhat different entry points that we cleared up long ago no serious observer has ever suggested that anything we do gives us an advantage.

Microsoft spends tens of millions a year getting information out to developers about our systems. We do this for good business reasons. Any book store with computer books will have dozens of books from many publishers telling anyone who is interested how to exploit our platforms. Even before we release products we use our Open Process to get developer input and development conferences to share our plans well in advance.

The software business is a challenging one. Picking platforms, understanding how they will incorporate new functionality that was handled by applications in the past, deciding not only what system software version but what hardware to require or exploit, competing with established firms like Microsoft that have technical and marketing strength - all of these things challenge a software CEO. However I feel confident that no industry will generate as many opportunities and success stories in the years ahead and there is no chinese wall.

*MX 5004158, 1*  
**CONFIDENTIAL**