User Experience Research

User Interface Friction Research

Research Report

Pfeiffer Consulting

01001011

User Interface Friction Research

© 2005-2006 Pfeiffer Consulting
33, rue Guy Moquet • 92240 Malakoff • France
Tel.: (33) 1 49 65 05 74 • Fax: (33) 1 42 53 58 03
e-Mail: research@pfeifferreport.com

Pfeiffer Consulting
01001011

Table of Contents

About this Report

1 Introduction

1.1 About This Research

Introduction

In 2005 and early 2006, Pfeiffer Consulting conducted an extensive research project collecting information about Macintosh and Windows operating systems. During the research interviews, that included users of both platforms, many Macintosh users stated that they found their computer "more fluid", more productive, easier to use. They were, however most often at a loss when they were asked to quantify their perceptions.

User perception transcends functional analysis

These recurring statements were intriguing: **from a purely functional perspective, both operating systems have become increasingly similar,** and even in terms of user interface, the basic concepts and user interface paradigms used by Windows and Macintosh are almost identical. This discrepancy between user perception and technical features led us to have a closer look at user interface differences, usability, and productivity. **During this research, we realized that the terms and concepts we use to analyze technology have remained surprisingly simplistic given the importance digital tools and devices play in our life.**

1.2 Web usability vs. device user experience

Since the advent of the Internet, there have been extensive usability and user interface studies, and our first reaction was examine the abundant literature that is available around this subject.

There are significant differences between usability of web-sites and an efficient computer user interface

We quickly realized, however, that available research did not cover our interrogations: most user interface and ergonomics studies deal with creating a good user experience for web-sites, and do not cover usability of computers or digital devices.

But usability does not mean the same thing for a web-site or for a photo-editing program. Using an iPod has little to do with purchasing a book on Amazon.com. The differences between these two kinds of user interfaces are very significant: first of all, **web-sites generally need to be conceived primarily for a unique, "first-time" user experience,** not for being used on a repetitive basis. Second, **the internet is generally about interacting with data, not about actively creating or manipulating documents or manipulating a device.**

Over time, the lines between the internet and local computing devices is likely to blur increasingly, **but as far as activities and user experience go, significant differences will remain.**

1.3 Analysis and Efficiency Measures

These considerations were the starting point for the research and analysis presented here. **In addition, Pfeiffer Consulting conducted extensive productivity and user interface efficiency measures to quantify some of the concepts discussed in these pages.** Productivity benchmarking has been a mainstay of Pfeiffer Consulting's research activities fort years, and we have noticed over and over that seemingly unimportant differences in efficiency can add up to significant productivity gains if an operation is frequently repeated.

1.4 Structure of the Report[1]

This report is structured in four distinct sections:

"Platform Differences, Ease of Use and Productivity" (page 10) provides some consideration of user interfaces and their relationship to user experience and productivity.

"Understanding User Interface Friction" (page 24) presents and discusses the concept of user interface friction.

"Measuring User Interface Friction" (page 33) presents selected results of the productivity and user interface efficiency measures conducted by Pfeiffer Consulting in order to quantify User Interface Friction.

"Hardware-Related User Interface Friction" (page 41) shows another series of benchmark results, documenting user interface friction imposed by differences in hardware management, in this case, problems linked to the mouse.

1.Parts of the information in this report are published separately in the research report 'Macintosh/Windows: Cost and Productivity Analysis", © Pfeiffer Consulting 2005-2006.

1.5 Please send Feedback!

It is our conviction that understanding user interface friction is essential for moving ahead in technology development. We believe that once we better understand the impact of the concepts outlined here, it will contribute to creating better, easier to use and more user friendly products, be they application software, operating systems or digital devices such as digital cameras, music players, among others.

Understanding, defining and quantifying aspects of user experience and productivity have been an ongoing effort for us. This current report is a starting point for us: **do not hesitate to send us feedback at research@pfeifferreport.com**

This report is made available for download free of charge. Please contact Pfeiffer Consulting if you would be interested in reproducing parts of this report.

1.6 About Pfeiffer Consulting

Pfeiffer Consulting is a Paris-based, international research and consulting operation specializing in technology and media. Pfeiffer Consulting's mission is to provide unique high-level, international market intelligence, user experience analysis and strategic consulting for both content and technology providers.

Pfeiffer Consulting is the publisher the *Pfeiffer Report on Emerging Trends and Technologies*, an online resource on trends in the technology and content industry, as well as numerous specialized studies and reports.

Pfeiffer Consulting has developed **a comprehensive methodology for task-based productivity and efficiency benchmarking that provides reliable productivity and UIF data** on application software, operating systems and digital devices. Please contact us at research@pfeifferreport.com to find out how we can help you with your projects.

For more information on Pfeiffer Consulting's reports and services, please visit: http://pfeifferconsulting.com

Platform Differences, Ease of Use and Productivity

1 Platform DIfferences: Introduction

1.1 Comparing Operating Systems

Discussions comparing Windows and Macintosh platforms usually focus on one big question: which is the better computer? Despite the fact that they have been going on for fifteen or twenty years, these debates have lost none of the emotional charge they usually carry, and **it is surprising to which extent our collective perception and understanding of the issues involved has failed to evolve.**

Even today, personal platform preference often outweighs non-partisan analysis of the issues: in the course of this research a majority of respondents had a clear platform preference, and only few participants were truly "platform-agnostic". To make matters worse, proponents of both Windows and Macintosh platforms accuse the other of being biased and even "religious" about the issues.

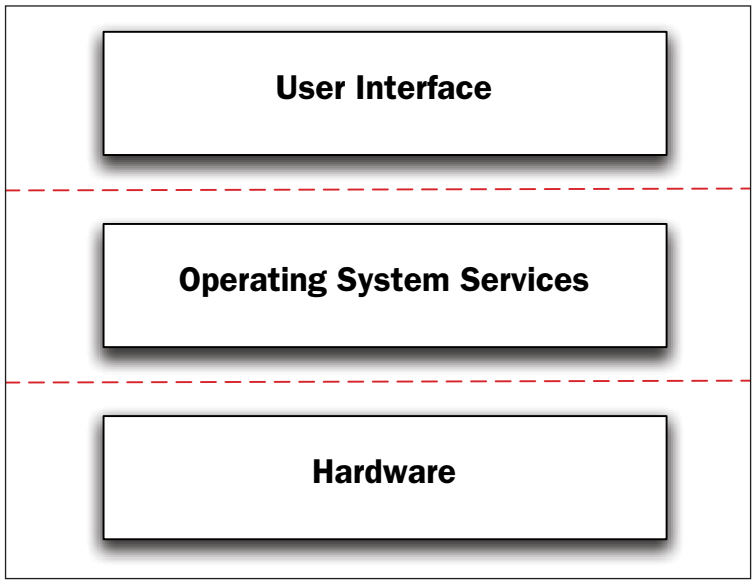
1.2 Spotting the Difference

Analysis of the computing platforms needs to go beyond purely technical, functional considerations.

Looking at these issues more closely, one thing becomes immediately apparent: **common perceptions and methods of analysis have remained very limited, too limited, in fact, to properly assess the underlying differences between the two computing environments in question.**

Even a very simplistic analysis of a computer environment (Figure , "The Computing Platform," on page 12) shows that we need a clear distinction between hardware, system software, and user interface in order to understand the differences. Even more important, **only a clear grasp of the exact nature of these differences will allow us to see if and what impact they can have on efficiency, productivity and overall user experience.**

The Computing Platform
Comparing computing platforms such as Macintosh and Windows requires a clear distinction between differences relating to hardware architecture, operating system functionality, and user interface.



These consideration are particularly important when considering the different levels of technology involvement in a larger company.

Today, many of the perceived differences, especially at the user level, **have much more to do with the user interface than with actual operating system functionality,** although both overlap. On the following pages we will analyze in some detail some of these differences, and their impact on productivity.

1.3 What's in a User Interface?

The user interface itself is a much more complex beast than one might imagine, and only a structured approach will allow in-depth analysis and understanding of user interface differences and their potential impact on the user. (See "Aspects of the User Interface" on page 13 for details.)

A coherent, well-structured user interface can make the difference between a good an a bad product. Badly structured user interfaces, on the other hand can significantly slow down the user. Two programs that both use pull-down menus offering the same options can vary in efficiency just by the way in which the menus are organized.

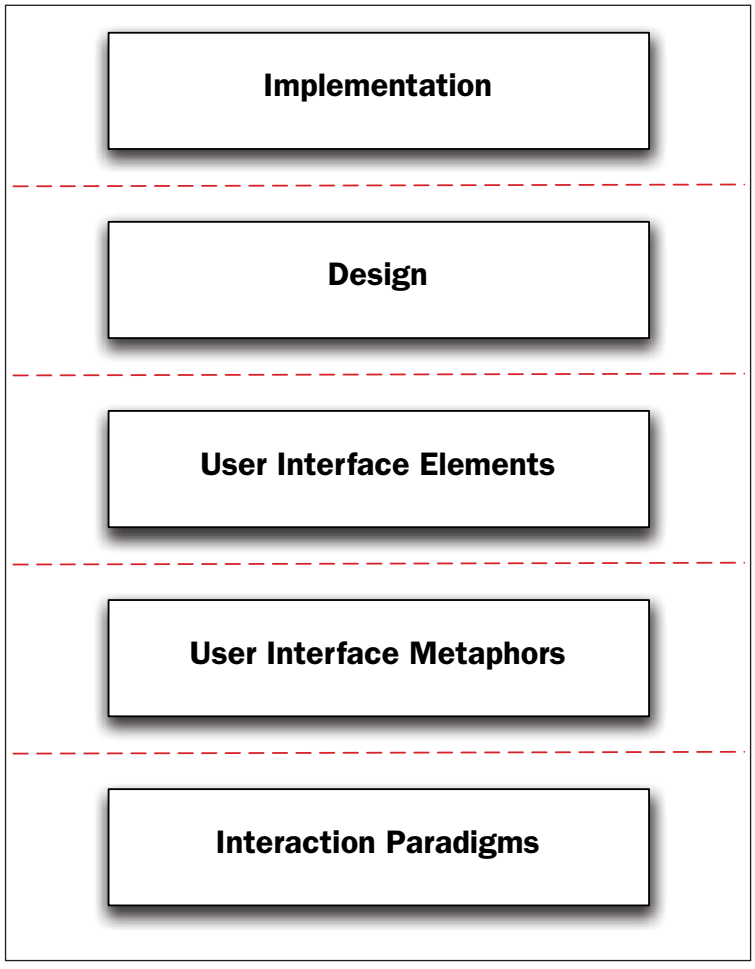
Coherence is an essential aspect of a good user interface: imposing stringent user interface conventions on developers was an important aspect of the ease of use of the original Macintosh, ensuring that by learning the basic user interface conventions once, a user would be able to find his way around new application programs without having to learn a completely new user interface.

Aspects of the User Interface

The user interface is a much more complex entity than one might imagine, and only a structured approach to the constituting elements can yield a coherent user interface.

Likewise, analyzing technologies, operating systems and application programs in terms of user interface requires taking these different aspects in account.

Mac OS X and Windows, for instance, are very close in terms of the underlying user interface levels, yet reveal significant differences concerning higher levels of the user interface structure.



2 Why Is It Important to Understand Platform Differences?

2.1 Functionality vs. Ease of Use

In earlier days of computing, comparing two technologies was usually done exclusively on the basis of functionality provided: the more features, the better the program or the computer.

As the computers and digital devices move into the mainstream, however, it is increasingly clear the this functionality-based approach is not sufficient to properly assess technology. Usability, coherence of user interface, design have become as important in software and computer hardware as they are in consumer electronics.

Microsoft Word, for instance, offers so many features, that it tends to become overwhelming, becoming less efficient for baseline word processing tasks than less feature-rich programs (or, for that matter, than older versions of Word itself.)

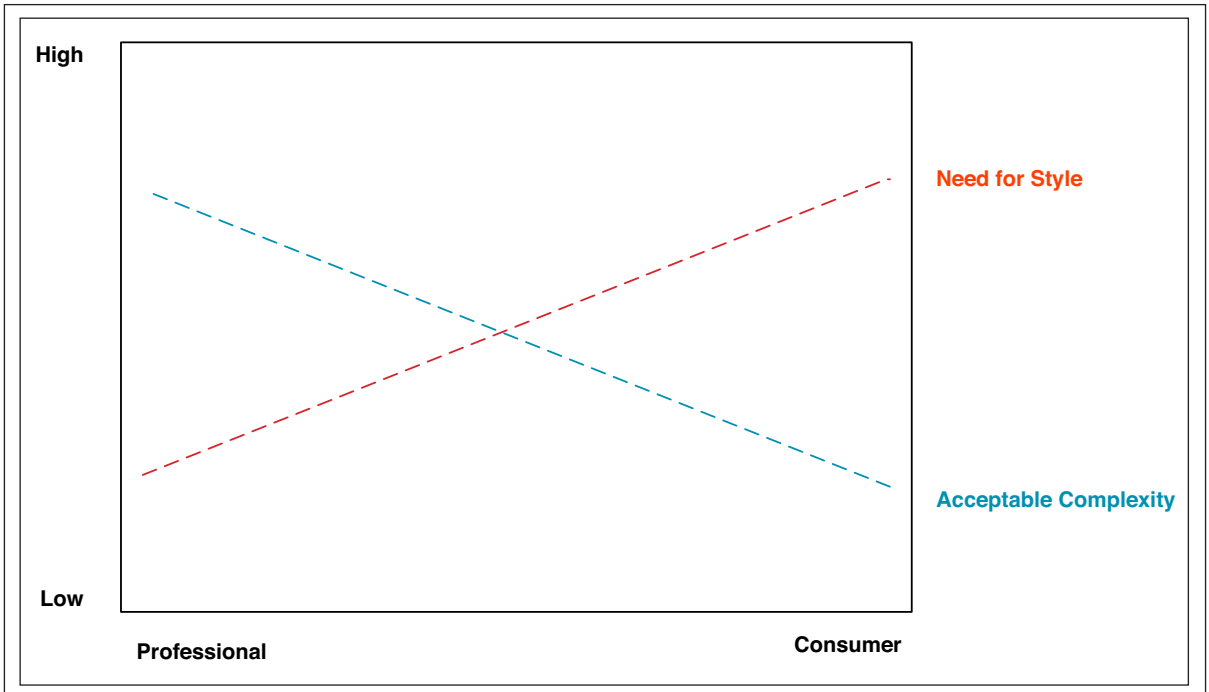
Technology is only as good as the use we make of it. Functionality is only a quality if users adopt it.

Any technology is only as good as the part of it that is actually used. While in earlier days of computing, a large number of users may have had technical curiosity, enjoying the discovery and experimentation when a new release of a program or operating system arrived, **average computer users today care little about cutting-edge functionality, and are mainly focused on getting their job done as fast as possible.** Functionality is only a quality if it is immediately useful; unused features can slow the user down.

Functionality, in other words, is less important then the ease of *accessing* it: Microsoft Office 12, due in 2006, will innovate mainly in terms of user interface and provide relatively few new features that are immediately useful to the average user.

This an emerging trend in the technology business: **the main challenge for a software publisher today is not so much to invent new functionality for an existing program, but to motivate users to experiment with and adopt new features.**

The most important aspect of a program or technology is not so much **what is theoretically possible, but which features are easy to understand and to use.** The phenomenal success of Apple's iPod is a very good illustration of the importance of ease of use over functionality. This applies not only in traditional consumer electronics, but increasingly also to digital technology and computing in general.



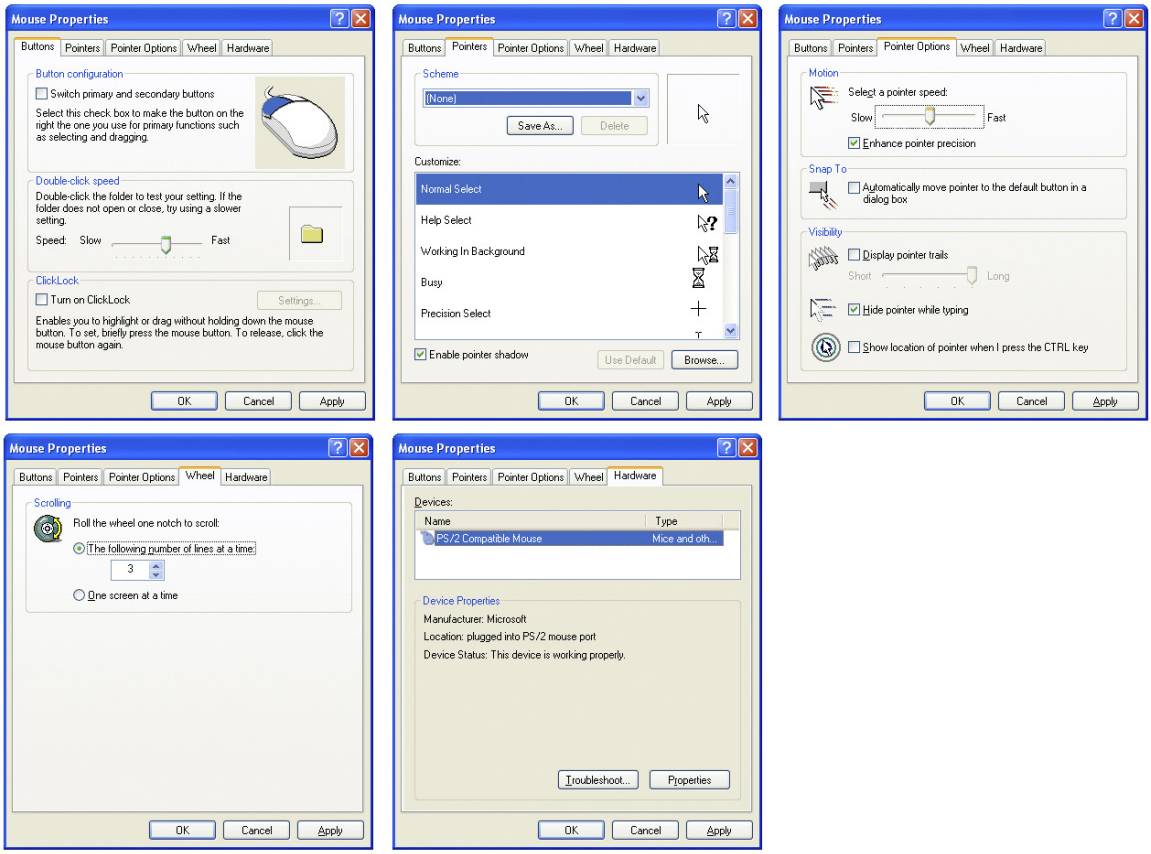
User Interface Requirements Relative to Market Positioning
As technology moves into the mainstream, ease of use and style become not only essential for the commercial success of a product, but also for the actual use of the technology. The iPod was such a huge success not only because of slick design, but ***because its ease of use that effectively introduced millions of consumers to a technology they would not have used had it been more complex or less desirable.***

2.2 Feature-Driven User Interface Design

Microsoft is feature-driven, while Apple has a strong focus on ease of use and elegance.

The approach to user interface design is one of the core differences between Microsoft and Apple when it comes to technology-development. These differences are directly reflected in their products, and in the audience they spontaneously appeal to: **Microsoft is a very functionality-driven company**, generally preferring to add as much granularity to its tools end technologies as possible. **Apple, on the other hand, focuses primarily on ease of use and elegance**, often choosing simplicity over complex feature-sets.

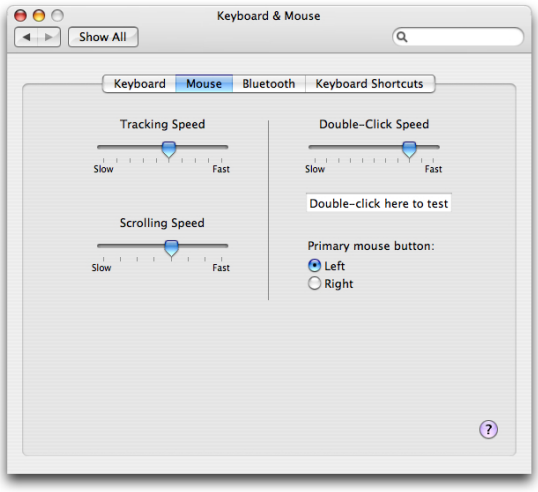
The positive side of Microsoft's approach is feature richness, which appeals to IT professionals who have a solid grasp of the technologies (See "Levels of Competence" on page 18 for details.) **The negative aspect of this approach can be feature overload (FO) that can be over-**



Different Approaches to Operating System Functionality

Windows XP often exposes much more granular settings to the user, while Mac OS X tends to focus on essential parameters: in this example (system settings for mouse operation), Windows XP offers the user five different control panel tabs with a wide range of options (top), while Mac OS X provides 5 essential settings in a single control panel.

Power users who appreciate subtle customization of their system settings are likely to prefer the Windows approach, while casual users can be overwhelmed by the amount of different options.



whelming or intimidating for less experienced users. Feature overload can result in a slowdown when a user has to tackle infrequently used options.

See "Different Approaches to Operating System Functionality" on page 16.

Apple's approach to operating system user interface, on the other hand, is more targeted towards the non-technical user. Mac OS X offers a smaller degree of granular settings for system management. This can be frustrating for power users who enjoy customizing their computer, but makes it easier for casual users to understand and manage their computer. **In general, simpler, less convoluted user interfaces tend to encourage user experimentation, and favor casual learning more than complex ones.**

3 User Interface: Understanding the Differences

3.1 What Do We Need to Look At?

Comparing Macintosh and Windows from a user interface perspective requires a structured approach.

There is no good or bad user interface per se. We use best what we use most, and familiarity with a tool is critical. However, understanding the sometimes subtle differences between computing environments is important to properly assess their impact on productivity.

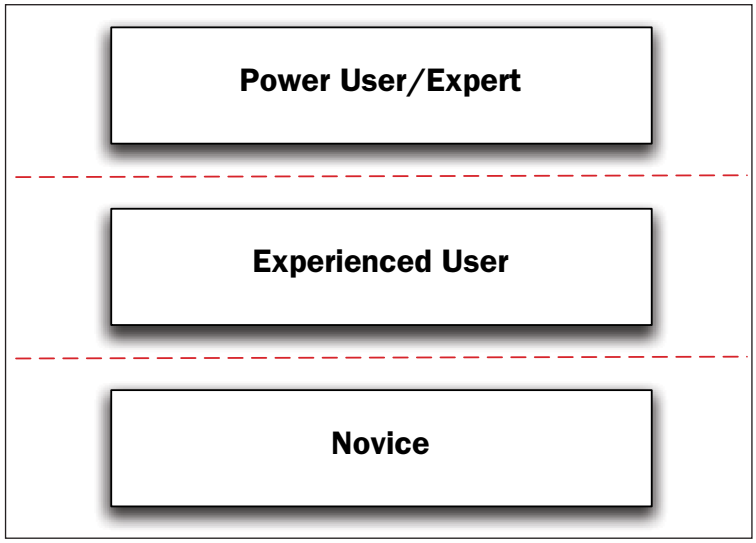
The most important notion when analyzing user interface is the user itself. **How experienced is the person using the computer and manipulating the operating system? This is particularly important when managing large groups of computer users.** What level of competence is required for manipulating a program or operating system? (See "Levels of Competence" on page 18 for details.) Should a novice be able to access a functionality, or is it intended for "Power Users"? Should the user be able to *understand* what he or she is doing? With respect to these questions, Macintosh and Windows differ considerably, thus appealing to different types of users.

Perception of the computer environment is further complicated by the fact that competence-levels of users can vary depending on the program or environment they are using: thus, **a page layout operator can have highly sophisticated understanding of the functionalities offered by the core application for his job, yet have only limited understanding of the operating system options.**

Levels of Competence

Taking in account the level of competence required for using a program or operating system option is essential when designing a user interface.

Functionality only intended for experts or experienced users should not be exposed to novices, since it will contribute to confusion. (See "User Interface: Levels of Perception and Understanding" on page 19 for details.)

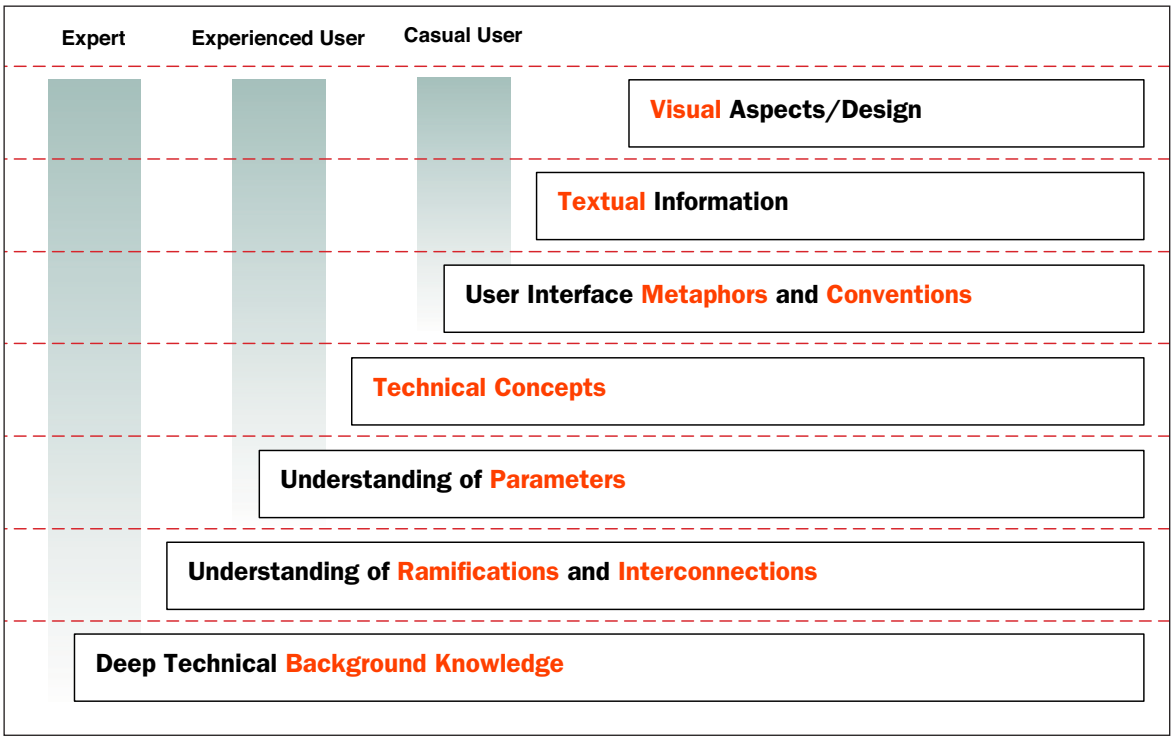


From not-knowing to guessing to understanding

Usage patterns are equally important. We do not know all aspects of a program or operating system equally well: some we may use all the time, others only occasionally, while some we probably ignore. **The quality of a user interface environment will depend on the ease at which we can move from not knowing to guessing to understanding with a minimal amount of learning effort.**

3.2 The Importance of Casual Learning

Casual learning, the capacity to pick up the sense of a program option without formal training, is becoming increasingly important in modern technology environments; it depends in a large part on the coherence of the user interface.

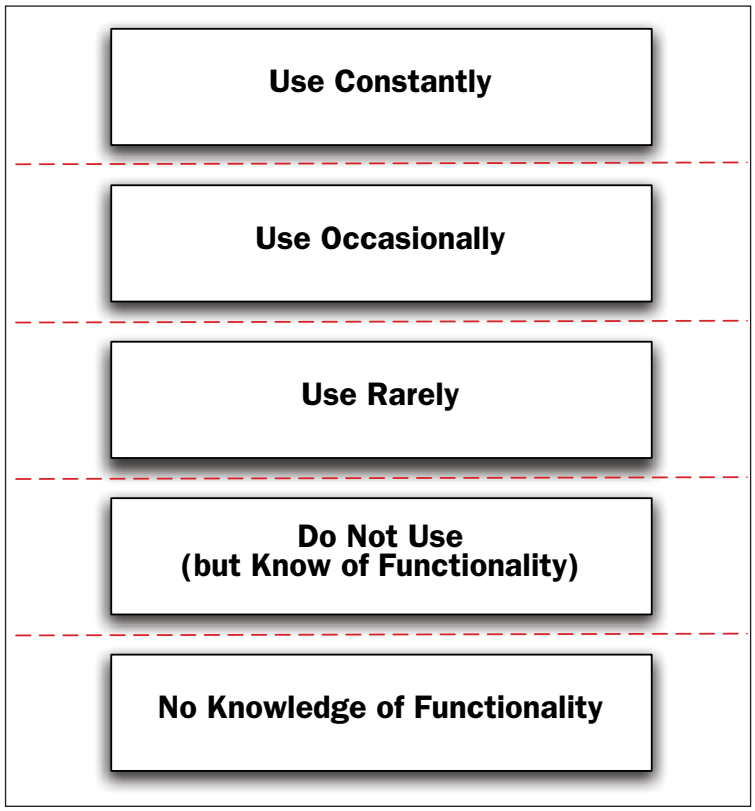


User Interface: Levels of Perception and Understanding
Understanding the target audience, and the levels of perception and technology-literacy is essential for designing a successful product. This means that one has to very consciously use different types of user interface elements according to the expected proficiency of the majority of users.

Patterns of Use

Our understanding of the technology we use is directly linked to the frequency and nature of our exposure to it. This in turn needs to be taken in account in the creation of user interface elements, as well as in the analysis of existing systems and technologies.

Coherence of the user interface is one of the most important factors for a streamlined, efficient workflow.

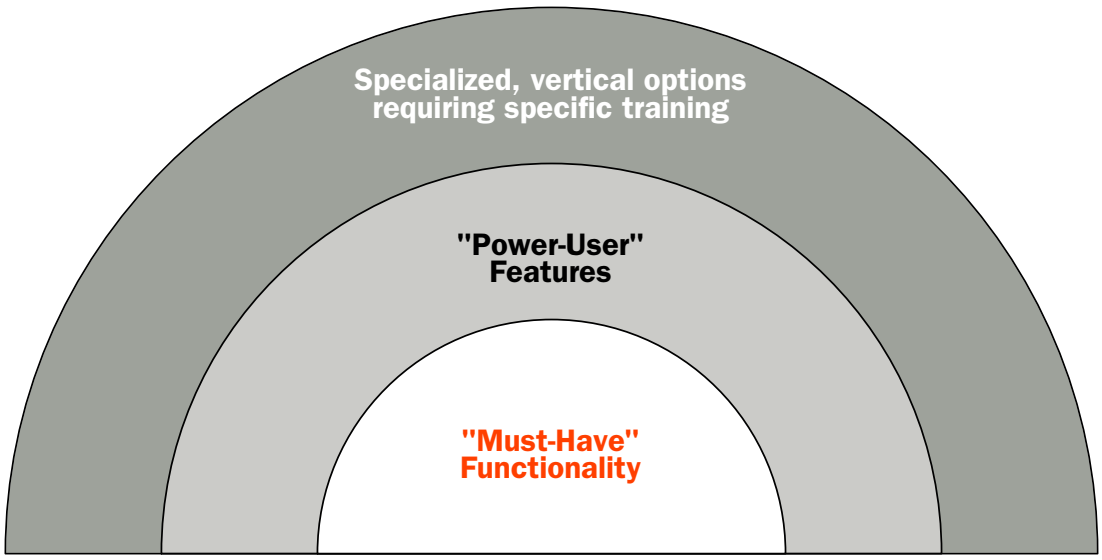


Another very important aspect to consider in user interface design is the level of knowledge of the user when presented with an information. Does he know what he is doing, or does he simply guess? How do we structure the available options so that they do not induce concept confusion or feature overload? **Given the richness of today's technology, it is very hard not to succumb to feature-creep.**

3.3 Why Its's Always 12:00

See "Understanding User Adoption of Features" on page 21 for details

Most users acquire a given technology—computer, device, software application—for a very small set of "must-have" features, and rarely go beyond the few tasks that were initially acquired. Only the relatively small portion of technically minded users will take pleasure in learning new or more obscure features, even if they could be useful for a wide range of users: how many users never learned how to program the date on their VCR? Or, in more modern terms, how many Windows users know how to add programs to the Start menu?



Understanding User Adoption of Features
Motivating users to learn new features has become one of the biggest challenges for technology providers. A lot of this has to do with good user interface design: a product needs to encourage the user to experiment, to move beyond the "must-have" feature set into more refined functionality. The biggest mistake would be to assume that users are eager to learn new features.

Motivating the acquisition of new features is one of the biggest challenges software developers face: even if a feature would be desirable, it will not be used unless there is practically no learning effort involved.

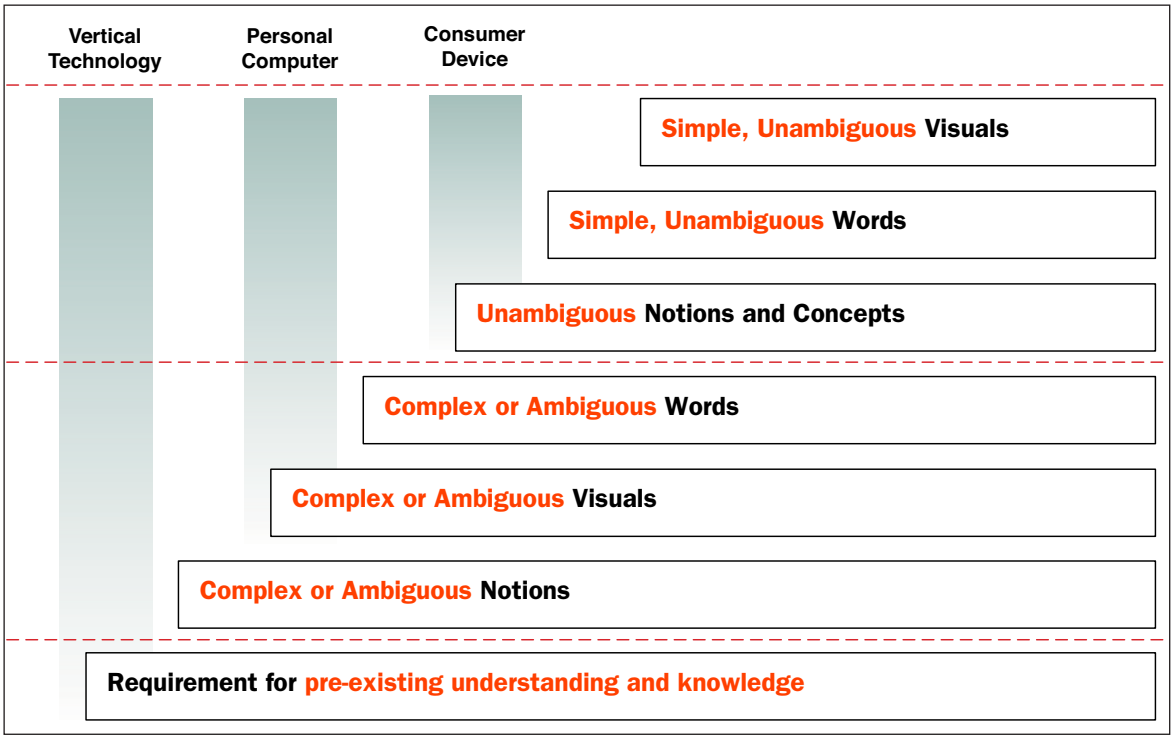
As an example, one of the most popular innovations in Mac OS X, **Widgets (small programs that can be called up with a single keystroke) owe a large part of their success to the simplicity of accessing them.** Had invoking Widgets involved more steps than just simply pressing a function key, it is a fair bet that far fewer people would use them.

4 What Is Ease of Use?

4.1 Understanding Simplicity

Ease of use is a complex notion

Ease of use and simplicity are popular notions in the technology industry: every new product, it seems, is aiming at being simple. Technology that is so compellingly easy to use that consumers are lining up to adopt it (like iPod) is the holy grail of electronics and computer manufacturers alike. This is particularly true in an aging technology business where novelty alone is rarely enough to trigger adoption of a new product. **Ease of use and an elegant, intuitive user interfaces have become as essential as good manufacturing to make a product a success.** Yet simplicity and ease of use are complex issues.



Understanding Ease of Use
Ease of use is a complex notion. A device or technology intended for the consumer market needs to be as unambiguous as possible, while complexity can increase with the expected technology literacy of the potential user. In any case, it is essential to understand perception in order to create a truly easy to use product—and this is much harder to do than it may initially appear.

Producing simple, easy to use hardware and software is far more complex than it may initially seem. **Simplicity is a matter of perception, and only if one understands the perception of the user can one hope to make things simpler.**

This also implies **understanding the notions and concepts a potential user will be familiar with**. That, in turn, depends on the technology literacy of a person.

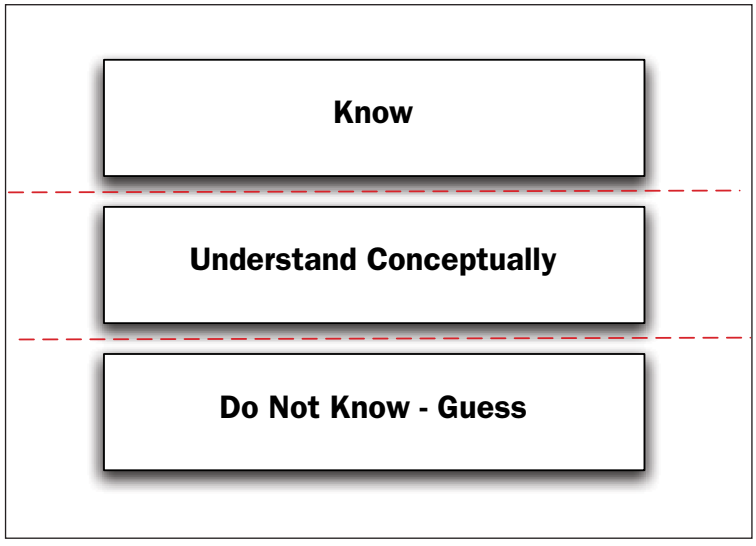
4.2 Everything Counts

Ease of use and simplicity are a form of perfection that can be very compelling

From a user's perspective, simplicity is far more pervasively compelling than individual features. Coherence in the approach to common tasks breeds a familiarity users recognize and appreciate. **In terms of ease of use, every aspect of the user interface is important.** Making an operation simpler means reducing the number of steps necessary, not adding information. Reducing the number of steps not only makes it easier to reach a desired result, **it also reduces the confusion that can result from an abundance of steps and options.**

Beyond the comfort-level simplicity brings, reducing the number of steps necessary pays off in terms of productivity. Every click, every mouse movement counts in terms of overall throughput: **Productivity measures show that the most impressive productivity gains can be obtained by reducing the number of necessary steps on very frequently repeated operations**

Levels of Technology Literacy
Designing easy to use products requires deep understanding of the technology literacy that can be expected from the average user of a planned product. What pre-existing knowledge can we expect? What are the concepts that a user can handle intuitively? What functions can he or she guess easily, even if there is no pre-established knowledge? Only if these aspects of design are properly integrated can a product be truly easy to use for a targeted audience.



Understanding User Interface Friction

1 What Is User Interface Friction?

1.1 The Limits of Technology Analysis

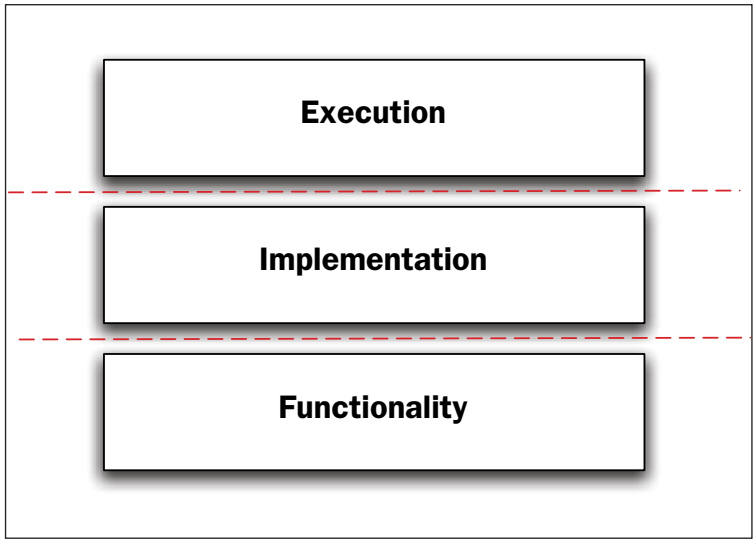
It is not because two technologies offer the same functionality that they are identical.

In the (frequently heated) discussions between Macintosh and Windows zealots, a recurring argument is that both operating systems have become almost identical. Yet the Macintosh users interviewed for this research project insist that the Apple's computer is easier to use, more fluid and more productive.

There is more than anecdotal evidence for this perception; indeed, **it was so recurring during the research interviews that we decided to look into the matter more deeply, by defining and conducting specific user interface productivity measures.**

Our analysis itself forced us to adopt a more nuanced way of analyzing technology and user interface. Since computing technology is relatively

Aspects of User Interface
It is very common to confuse different aspects of the operating system: functionality is important, but it is not the only aspect that users will perceive. In many cases, bad implementation can significantly limit powerful functionality. On the other hand, excellent implementation and execution can make up for limited functionality in the perception of the average user.



young, it is not surprising that the analysis and appreciation has so far remained relatively superficial, dwelling almost exclusively on functionality.

In an analogy to the automotive industry, one could say we are looking at the horse-power of the motor, but neglecting the quality of the transmission or the manufacturing of the chassis. This means that **in terms of user interface and operating system analysis, we need to take in account not only functionality, but also implementation and execution if we want to get a complete picture.**

1.2 The Concept of User Interface Friction (UIF)

User interface friction is a concept coined and defined by Pfeiffer Consulting to describe and quantify the differences in fluidity and reactivity that exist between operating systems and programs.

Pfeiffer Consulting defined the concept of User Interface Friction **to describe and quantify the perceived differences in efficiency and user experience between operating systems, application programs and digital devices.** User interface friction is the resistance imposed upon a user-guided process through the way the user interface reacts. It has nothing to do with functionality: **we use the term User Interface Friction to define the difference in fluidity and productivity that can be observed when performing the same operation on different computer systems, programs or devices.**

User interface friction is inherent in any modern, menu-driven computer system, and depends on a number of aspects, ranging from the speed at which the computer displays a menu or sub-menu, to the efficiency of the mouse.

Just like the smoothness of the paper or the ink-flow of a pen can impact the speed of handwriting, User Interface Friction affects practically any procedure where the user interacts through the user interface with the computer system.

Major Points: User Interface Friction

- User interface friction is **the resistance that the implementation and execution of a user interface feature imposes on the user of a program, device, or operating system.**
- User Interface Friction is not related to the functionality of an application program, and does not depend on the processing power of a computer, although these can be mitigating factors.
- **The same application program, functioning on two different operating systems can yield different overall UIF** because of differences in the way the user interface responds.
- **User interface friction can result in significant productivity loss** when it occurs on frequently repeated operations.

2 Key Friction Points

2.1 Menu Latency

See "Measuring User Interface Friction" on page 33 for details.

Some friction points are obvious: **having to wait, even a little bit, for a menu or submenu to be displayed is one of the main factors that can result in measurable productivity loss.**

Examples:

One of the best examples of menu latency is the Start-menu in Windows. Selecting an option from this menu or one of its sub-menus is affected by a slight lag, that occurs each time, slowing down the user.

2.2 Mouse Operation

See "Hardware-Related User Interface Friction" on page 41 for details.

One frequently overlooked aspect of computer efficiency is mouse operation, particularly in operations that require precision. Mouse operation can also slow down selecting sub-menus.

Examples:

Problems linked to mouse operation can be very annoying for the user and result in significant slowdown. Beyond the tracking problems a mechanical mouse can produce, driver issues linked to the operating system can reduce the productivity of the operator as well as his efficiency in precision tasks.

2.3 User Interface Ambiguity

Ease of use and simplicity are directly linked to unambiguous, clear options. **Ambiguous icons, convoluted or unclear explanations can create hesitation and slow-down.** A well-thought-out, clearly presented hierarchy of options, on the other hand can reduce User Interface Friction and increase productivity an user efficiency.

Examples:

One of the most common examples of user interface ambiguity has nothing to with computers, but occurs in elevators. Who hasn't experienced the annoying moment when one tries to open elevator doors for an approaching passenger - but can't decide which button to press? The personal computer is full of examples like this one.

2.4 File Navigation

Navigating the file system can be a significant slowdown factor in operating systems. **The file-and-folder paradigm used in modern operating systems can become cumbersome as the number of files and folders increases**. Users comments collected for this research confirm that this is a widely perceived issue.

Interestingly, file navigation is not only an issue limited to operating systems, but can be a problem with digital devices as well: navigating several thousand songs in a MP3 player can be cumbersome, for instance.

Examples:

All operating systems that use nested directories or folders to structure data files have this problem. Increasingly, operating system developers implement aids to reduce some of the time spent navigating: Shortcuts (called Aliases on the Macintosh) have been used for years. A more recent example is the column-view popular with Macintosh users.

2.5 Option Overload

Presenting the user with an abundance of available, non-structured options contributes to option overload: **Having too many items to choose from induces hesitation and slows the user down.** While it is difficult to precisely measure the impact of option overload, it is obvious that selecting the right option out a selection of three possible choices will be faster than selecting one out of ten. This can be aggravated by ambiguous descriptions or icons.

Examples:

Option overload is one of the most common UIF factors in modern computing. While many different options and possibilities can be very exciting for a power user, their impact on an average user can be devastating, even in common software applications.

One such example is Microsoft Word: it is not unusual for users to give up using a desired option they know exists in the software, just because they can't remember how to access it.

2.6 Concept Confusion

Concept confusion is another friction point the is increasingly problematic as software, operating systems and digital devices become more powerful and feature-rich.

The icon-based user interface was a ground-breaking invention and has contributed significantly to ease of use of computers. **Nevertheless, the visual user interface can become overwhelming and slow the user down, as the concepts it attempts to convey become more abstract.** Icons usually work well with actions that have an obvious counterpart in physical reality but can slow down the user when their meaning is ambiguous. **Words can be equally confusing if their meaning has not been clearly established.**

Examples

Concept confusion is very frequent with casual users confronted with system options: many settings in system control panels will baffle an uninitiated user. **Confusion about print-settings, or network access, for instance, are among the most common help-desk issues in the enterprise,** because many users just cannot handle them on their own, or try and get lost.

Major Points: Key Friction Points

- User Interface Friction **depends on a variety of factors that often occur in parallel** and can result in significant slowdown of the user.
- Some key friction points such as menu latency **depend on the way an operating system or a device execute common operations** such as displaying a menu or navigating the file system.
- Incoherent or badly conceived user interface elements can significantly contribute to user interface friction: **user interface ambiguity**, **option overload** and **concept confusio**n are mostly dependent on inefficient user interface design decisions.

3 Analyzing User Interface Friction

3.1 Why Is it Important to Understand User Interface Friction?

Understanding User Interface Friction can help in assessing the technology, or when choosing one software solution over another.

Understanding User Interface Friction, what it means and how it can impact the overall productivity of a user and the throughput of a workflow is important, **because it is a usually overlooked factor in the overall technology analysis, yet its long-term impact can be very important.**

While the impact of User Interface Friction will vary according to the work situation, **the ripple effect over time can be significant in terms of overall productivity.** Pfeiffer Consulting strongly recommends integrating UIF analysis when assessing competing technologies. Good examples are the comparison of operating systems for production purposes, or the evaluation of competing application software packages or systems. For more information on conducting User Interface Friction analysis, please contact Pfeiffer Consulting at research@pfeifferreport.com.

3.2 Operating Systems vs. Application Software

While some core friction points (such as menu latency) are identical between the operating system and application software, other aspects will vary. Part of this is linked to user experience: an experienced Photoshop or QuarkXPress user can be comparatively inexperienced in terms of operating system functionality and operation.

See "Functionality vs. Ease of Use" on page 14 for details.

Some of the most significant friction points in modern application software are option overload and concept confusion: **in the race to provide new features to make subsequent releases more interesting, software developers often make programs intimidating or unwieldy.**

3.3 Mitigating Factors

Ease of use and casual learning can alleviate User Interface Friction. A user can be annoyed by not finding the option he was looking for immediately. Nevertheless, if in the process of searching he understands the logic, it can make him faster in the future. This can make up for the initial frustration and reduce the overall slowdown.

3.4 User Interface Friction and User Expertise

User Interface Friction is not identical for the average computer user and the expert. A casual user will be more resilient to the perception of slowdown, because he does not have a lot of experience working with a computer. **An expert user will notice little slow-down factors more, but will also be more likely to seek out ways of speeding up his work**, such as keyboard shortcuts for commonly used options.

Professional users, even if they are not computer experts, are the ones who can feel the slowdown the most strongly, since their work is about performing similar tasks over and over again, and they instinctively know that every second lost counts in their efficiency

3.5 Absence of Friction Can Be a Killer Feature

Just as User Interface Friction can reduce efficiency and create frustration, absence of friction can contribute to make a feature attractive. The widely popular Widgets functionality included in Mac OS X 10.4 (small programs that can be invoked very rapidly) has become such a success in large part because it is immediately available: invoking and hiding takes a single keystroke, and does not disturb whatever the user is working on. It is highly likely that far fewer users would be attracted by this feature if it needed to be accesses in a more time-consuming fashion.

3.6 Understanding User Experience

Understanding the different aspects of User Interface Friction is essential for building best-of-breed products. Ease-of-use is a very elusive concept, and **only if one masters and quantifies every aspects of User Interface Friction is it possible to create truly compelling products.**

This is particularly important in the realm of digital devices: in many companies, hardware development and user interface are two distinct domains; frequently, elegant hardware will suffer from an inefficient user interface. **Combining first-rate industrial design with an elegant and efficient user interface is the hallmark of a killer product.**

4 Can User Interface Friction Be Measured?

4.1 A Question of Methodology

Measuring User Interface Friction is mostly a question of choosing the right methodology: The impact of aspects such as menu latency, user interface efficiency or file navigation can be measured through sophisticated task-based benchmark methods such as the **Pfeiffer Productivity Benchmarking Methodology**.

Other aspects such as user interface ambiguity, concept confusion, option overload can be quantified through behavioral studies. Pinpointing specific individual aspects can be achieved by fine-tuning research procedures and methodologies, depending on the products that need to be compared.

Please contact Pfeiffer Consulting at research@pfeifferreport.com if you have any questions about our benchmarking and market research services.

Major Points: Analyzing User Interface Friction

- Understanding the different aspects of User Interface Friction **is essential for creating best-of-breed software and devices.**
- Absence of User Interface Friction can be a great perceived benefit of a product, and **contributes to creating a compelling user experience.**
- **User Interface Friction can be measured** through sophisticated task-based productivity benchmarks, as well as behavioral studies. Selecting specific methodologies will depend on the products or devices that need to be compared.

Measuring User Interface Friction

User Interface Efficiency Measures

Benchmark: Cut/Paste a Paragraph of Text

Focus: User Interface Friction

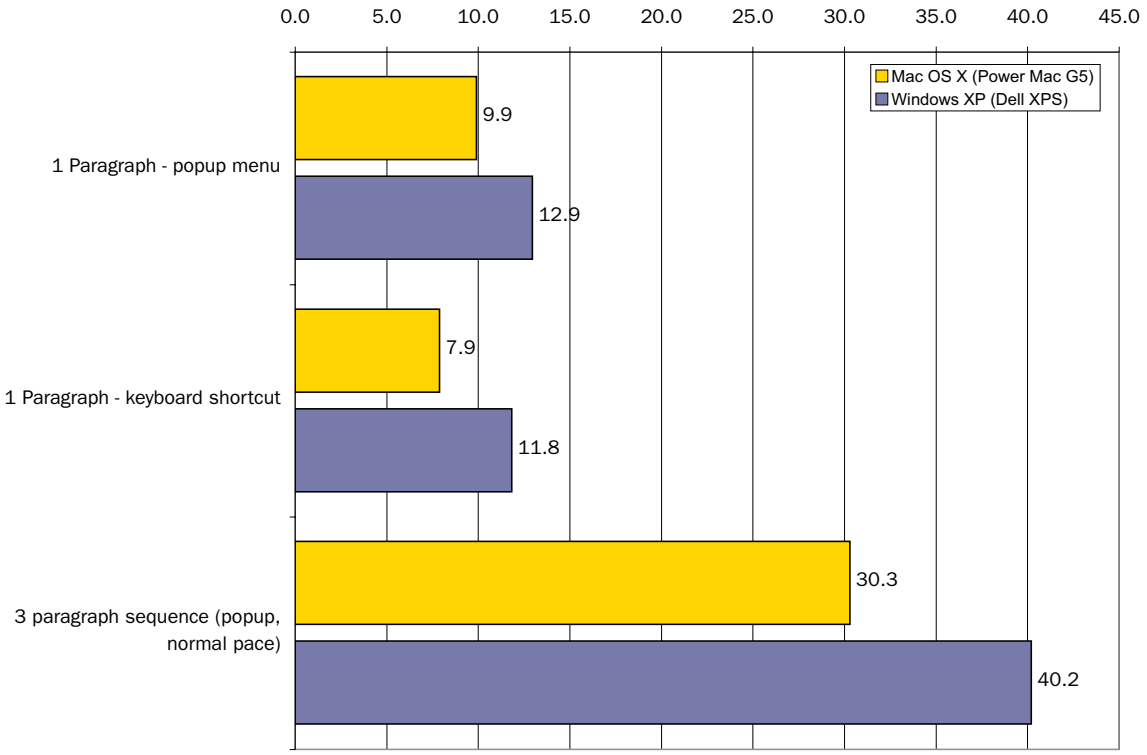


Chart 1 : Cut/Paste a Paragraph of Text

Type of Benchmark:

Task-based productivity measure

Time in seconds. Shorter is better.
(Figures shown are the average of 3 individual benchmarks)

Reference: 2

Notes:

This benchmark measured the time necessary to move text from one part of a short document to another. The test compared the use of popup-menus (right-click), keyboard shortcuts, as well as a three-paragraph sequence using right-click. Benchmarks were conducted using Microsoft Word.

Windows shows a slight inertia in pop-up menu performance. Also, **the key position (Command-X on the Macintosh vs. Control-X on Windows) favors faster execution**, since on the Macintosh, the command key is situated right next to the x-key.

User Interface Efficiency Measures

Benchmark: Launch 3 Applications in Succession

Focus: User Interface Friction

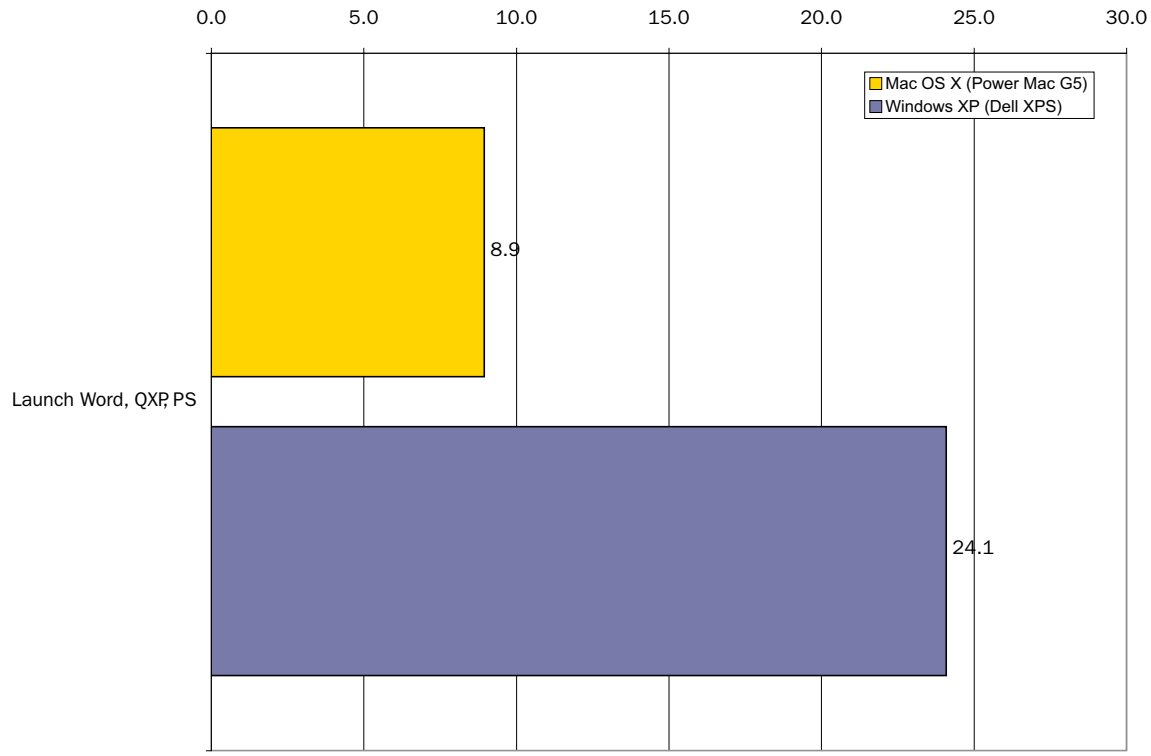


Chart 2 : Launch 3 Applications in Succession

Type of Benchmark:

Task-based productivity measure

Time in seconds. Shorter is better.
(Figures shown are the average of 3 individual benchmarks)

Reference: 3

Notes:

Launching Word, QuarkXPress and Photoshop in succession took almost three times as long on the Windows computer than on the Macintosh. **The reason for this discrepancy lies in the Windows Start-menu, which is far less efficient for launching applications than the dock on Mac OS X.**

The way the Start-menu forces the user to navigate hierarchical menus to access a specific program is an excellent illustration of User Interface Friction.

User Interface Efficiency Measures

Benchmark: Menu Access

Focus: User Interface Friction

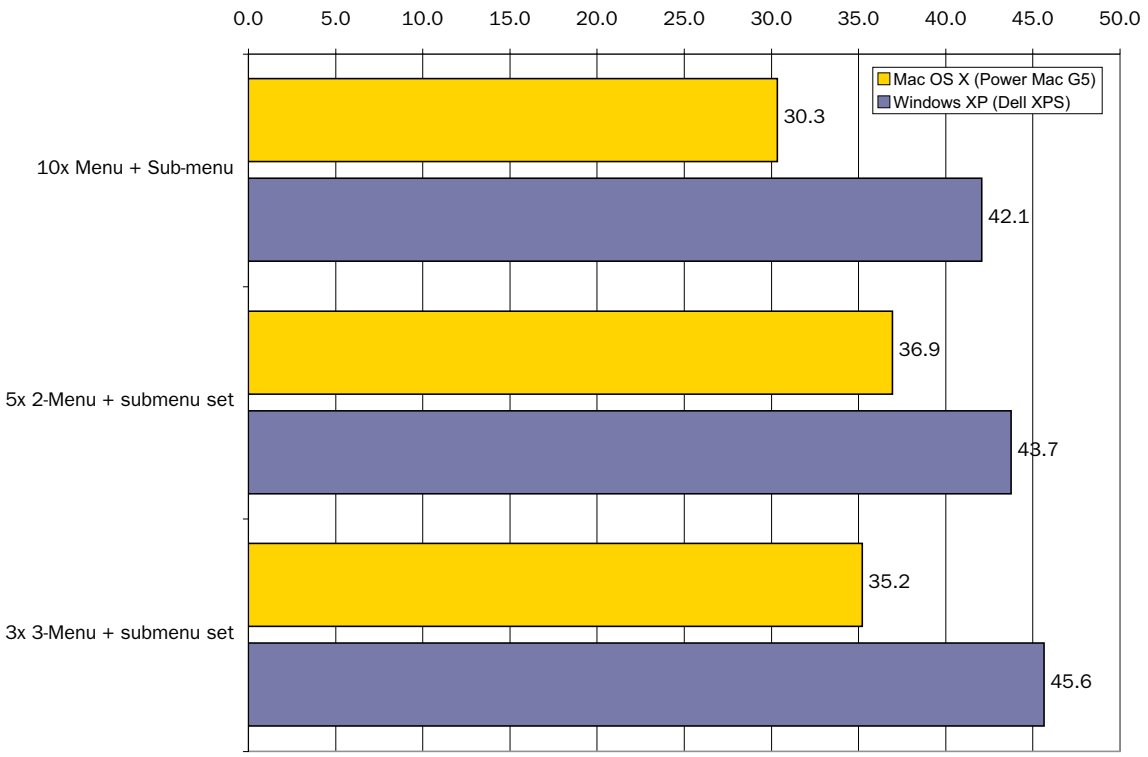


Chart 3 : Menu Access

Type of Benchmark:

Task-based productivity measure

Time in seconds. Shorter is better.
(Figures shown are the average of 3 individual benchmarks)

Reference: 6

Notes:

This test measures the time necessary to select a menu command situated on a first-level submenu. The different measures represented in the chart correspond to: top, selecting the same menu article ten times in succession; center, selecting two different commands 5 times; bottom, selecting a sequence of 3 different menu commands three times in a row.

The core aim of these repetitions is to provide reliable productivity comparisons covering different usage scenarios.

User Interface Efficiency Measures

Benchmark: Menu Access (Average)

Focus: User Interface Friction

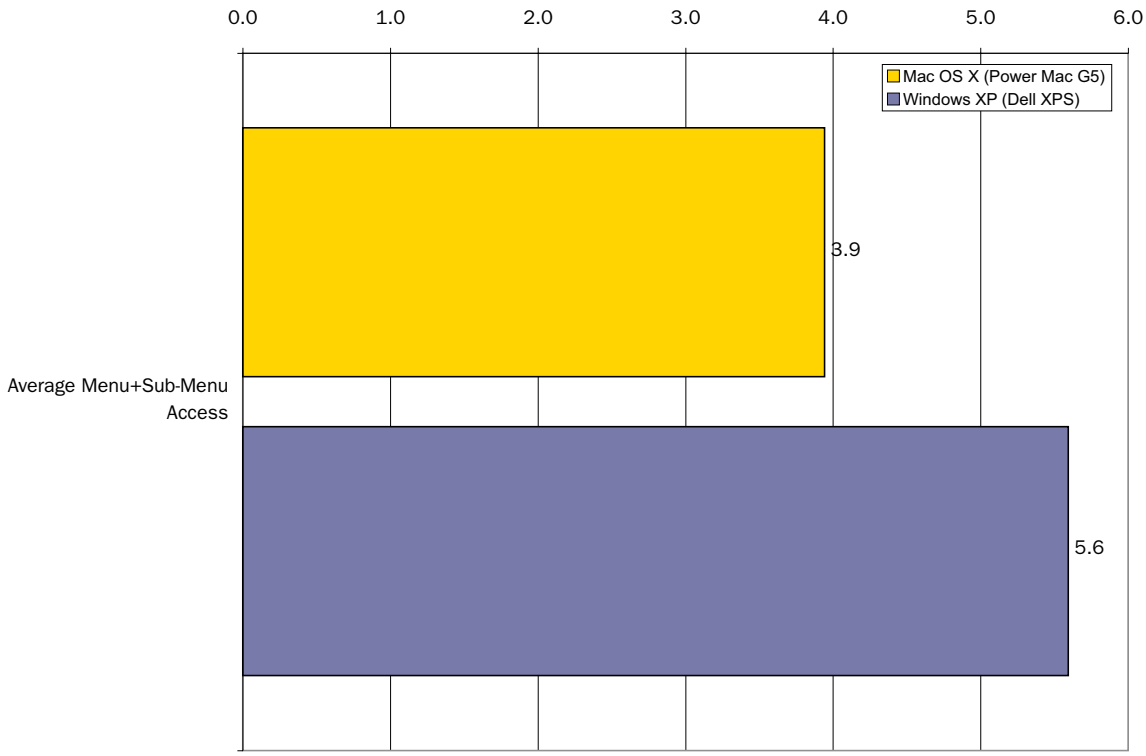


Chart 4 : Menu Access (Average)

Type of Benchmark:

Task-based productivity measure

Time in seconds. Shorter is better.
(Figures shown are the average of 3 individual benchmarks)

Reference: 7

Notes:

Based on the productivity measures in the previous chart, the average time necessary to select a menu command on a first-level sub-menu takes almost 30% longer on a Windows computer than on the Macintosh: this is a clear illustration of the user interface friction due to greater menu latency.

User Interface Efficiency Measures

Benchmark: Excel: Suite of Common Editing Operations

Focus: User Interface Friction

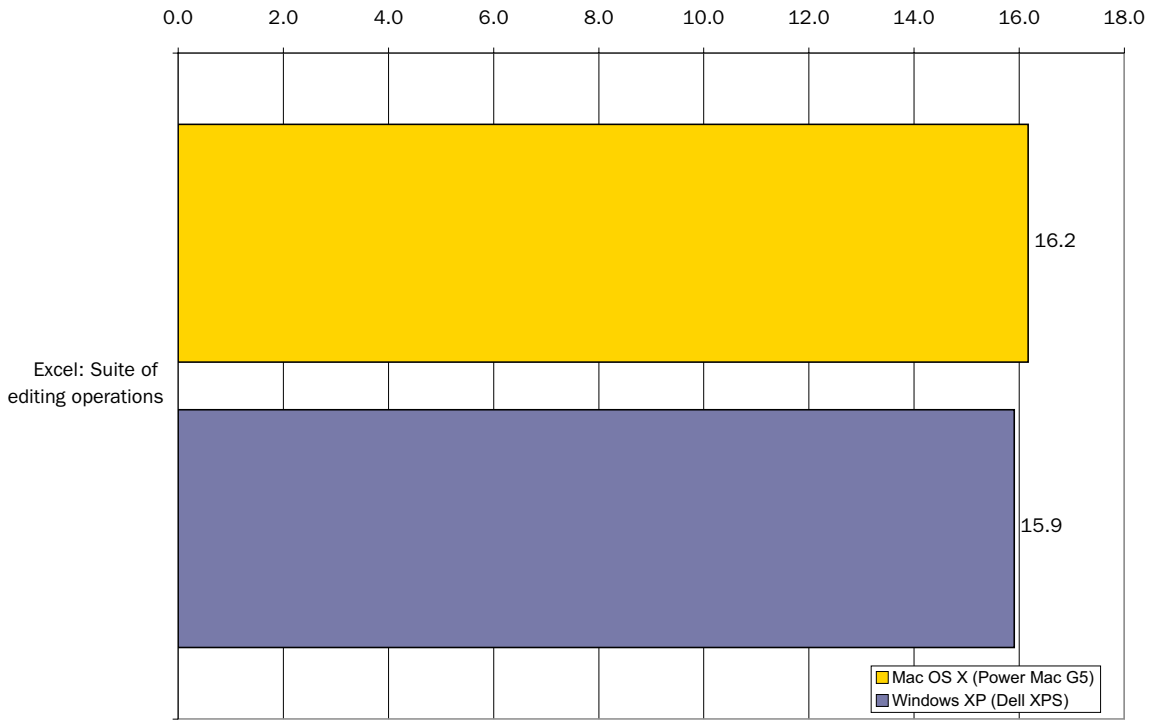


Chart 5 : Excel: Suite of Common Editing Operations

Type of Benchmark:

Task-based productivity measure

Time in seconds. Shorter is better.
(Figures shown are the average of 3 individual benchmarks)

Reference: 5

Notes:

While almost all applications tested here show higher latency in menu operations on Windows than on the Macintosh, Excel performs slightly better on Windows. **This indicates that there are ways of speeding up menu-performance on Windows, but that they are not widely used.**

User Interface Efficiency Measures

Benchmark: Print Simple Document

Focus: User Interface Friction

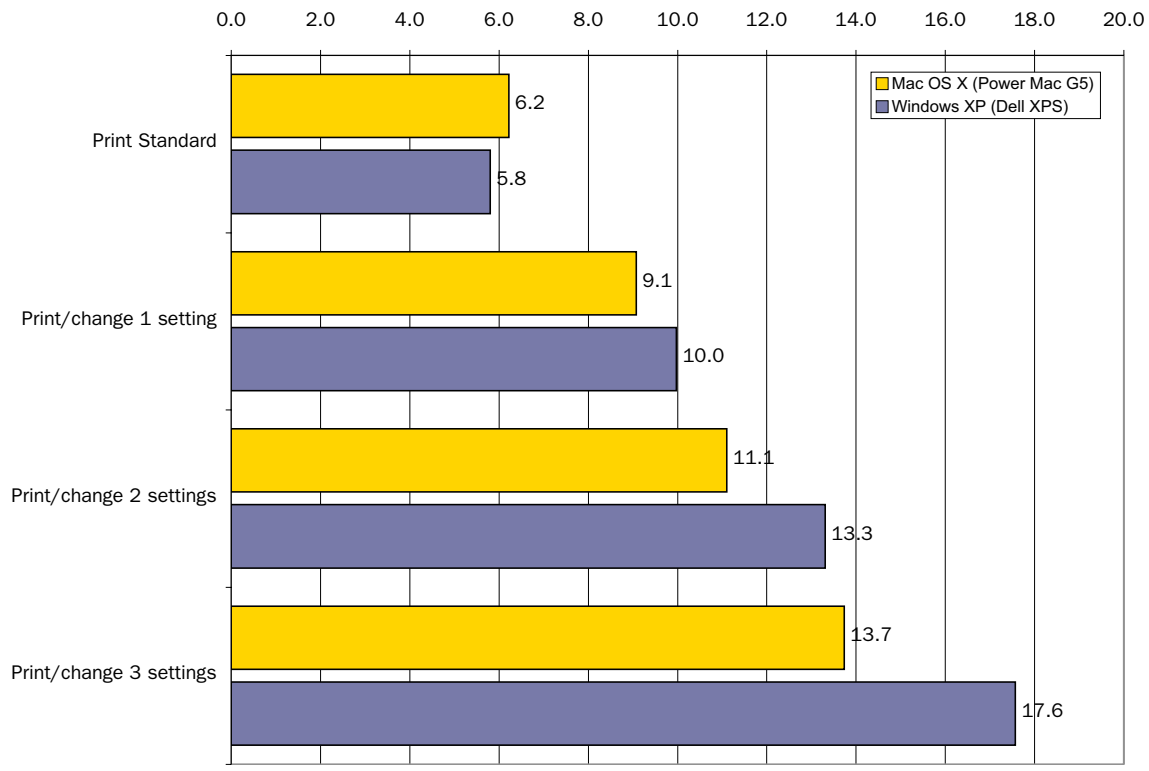


Chart 6 : Print Simple Document

Type of Benchmark:

Task-based productivity measure

Time in seconds. Shorter is better.
(Figures shown are the average of 3 individual benchmarks)

Reference: 8

Notes:

This test measured the time necessary to print a document (from selection of the Print… command to clicking the "Print" button in the dialog box.)

Individual tests were conducted changing one, two or three different settings in the dialog box.

The same printer driver software was used on both platforms. **Differences in efficiency are mostly due to higher menu latency on Windows computers.**

User Interface Efficiency Measures

Benchmark: Average of Common User Interface Operations

Focus: User Interface Friction

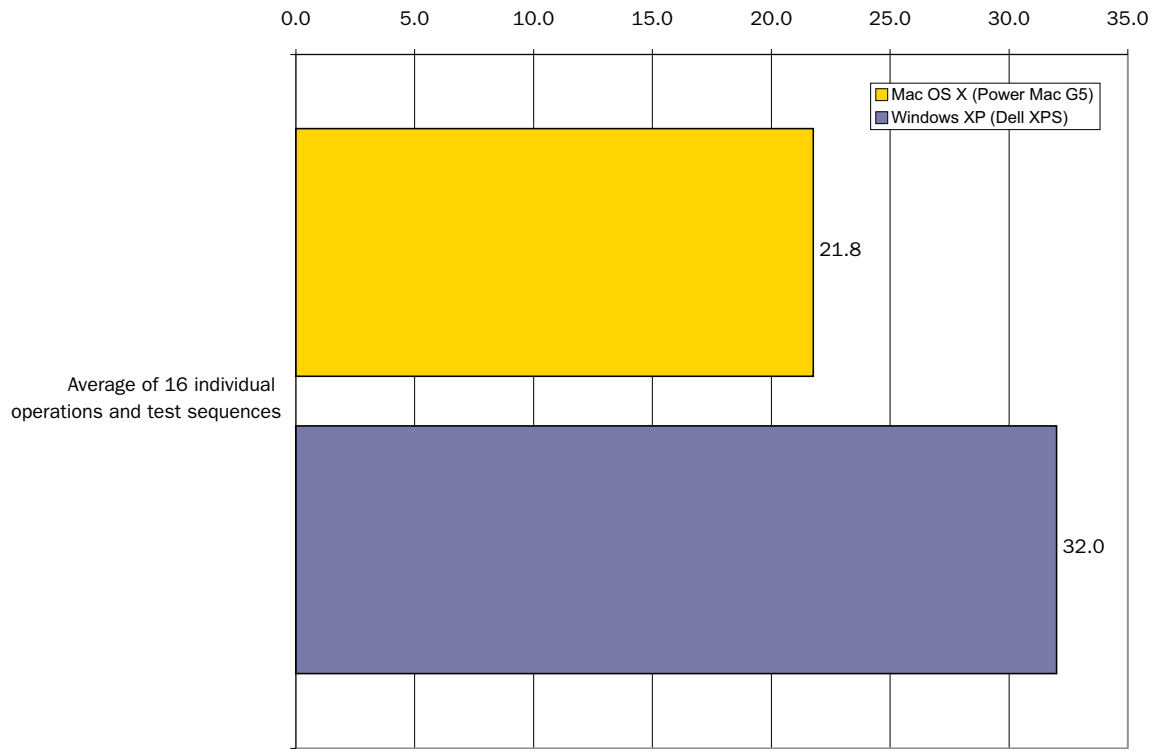


Chart 7 : Average of Common User Interface Operations

Type of Benchmark:

Task-based productivity measure

Time in seconds. Shorter is better.
(Figures shown are the average of 3 individual benchmarks)

Reference: 12

Notes:

Calculating the average of all user interface efficiency measures underlines that the User Interface Friction of Windows computers is significantly higher than on the Macintosh.

This difference in operating system efficiency is corroborated by user comments collected for this research project.

Hardware-Related User Interface Friction

Mouse Precision Productivity Measures

Benchmark: Mouse Precision Measures: Average of All Tests

Focus: Hardware-Related Productivity Measures

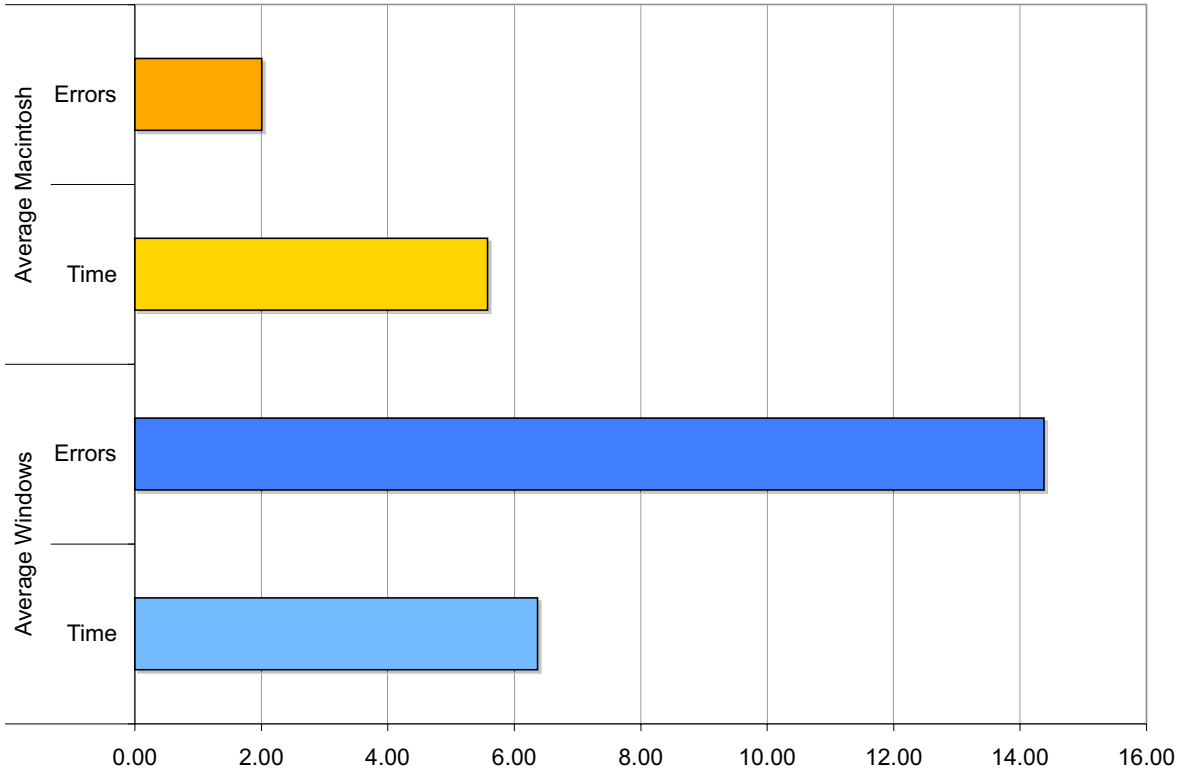


Chart 8 : Mouse Precision Measures: Average of All Tests

Type of Benchmark:

Task-based productivity measure

Time in seconds. Shorter is better.
(Figures shown are the average of 3 individual benchmarks)

Reference: MP1

Notes:

This set of benchmarks measured the impact of mouse performance on productivity in precision tasks. The benchmark uses a Photoshop-based test consisting in re-drawing rows of simple shapes with single-pixel precision. Each series of tests comprises 8 rows of ten squares. The complete test project comprised 6 complete series by platform; 3 sets were performed by a Macintosh user; 3 by an experienced Windows user.

This chart shows the average values for a total of 18 complete test series.

Mouse Precision Productivity Measures

Benchmark: Mouse Precision Measures: Windows User

Focus: Hardware-Related Productivity Measures

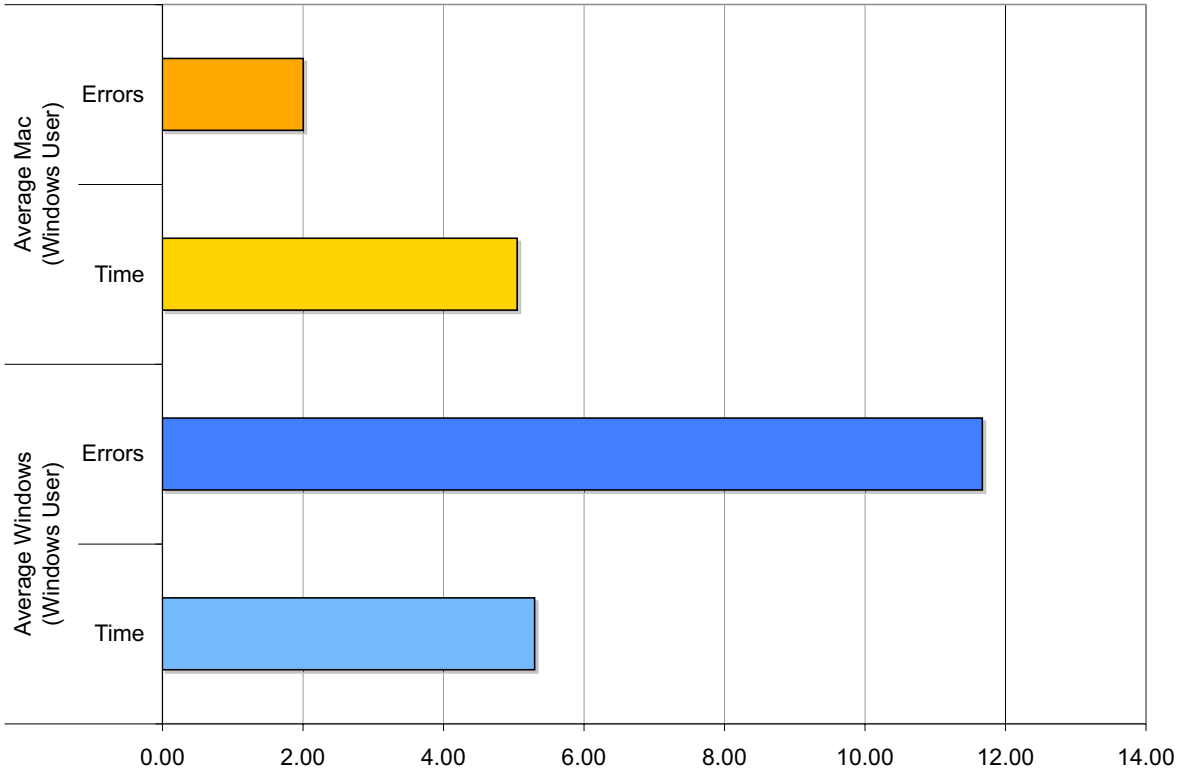


Chart 9 : Mouse Precision Measures: Windows User

Type of Benchmark:

Task-based productivity measure

Time in seconds. Shorter is better.
(Figures shown are the average of 3 individual benchmarks)

Reference: MP2

Notes:

This chart presents the average results for all the test series performed by the Windows user.

This chart underlines the fact that the mouse precision issue is a problem that has little if anything to do with the expertise of a user with a given platform: **the time-to-error ration is practically identical, whether the tests are performed by the Macintosh user or the Windows user.** (Windows users are usually not aware of the problem until they execute this test, and are often shocked by the difference between Macintosh in Windows in this respect.)

Mouse Precision Productivity Measures

Benchmark: Mouse Precision Measures: Macintosh User

Focus: Hardware-Related Productivity Measures

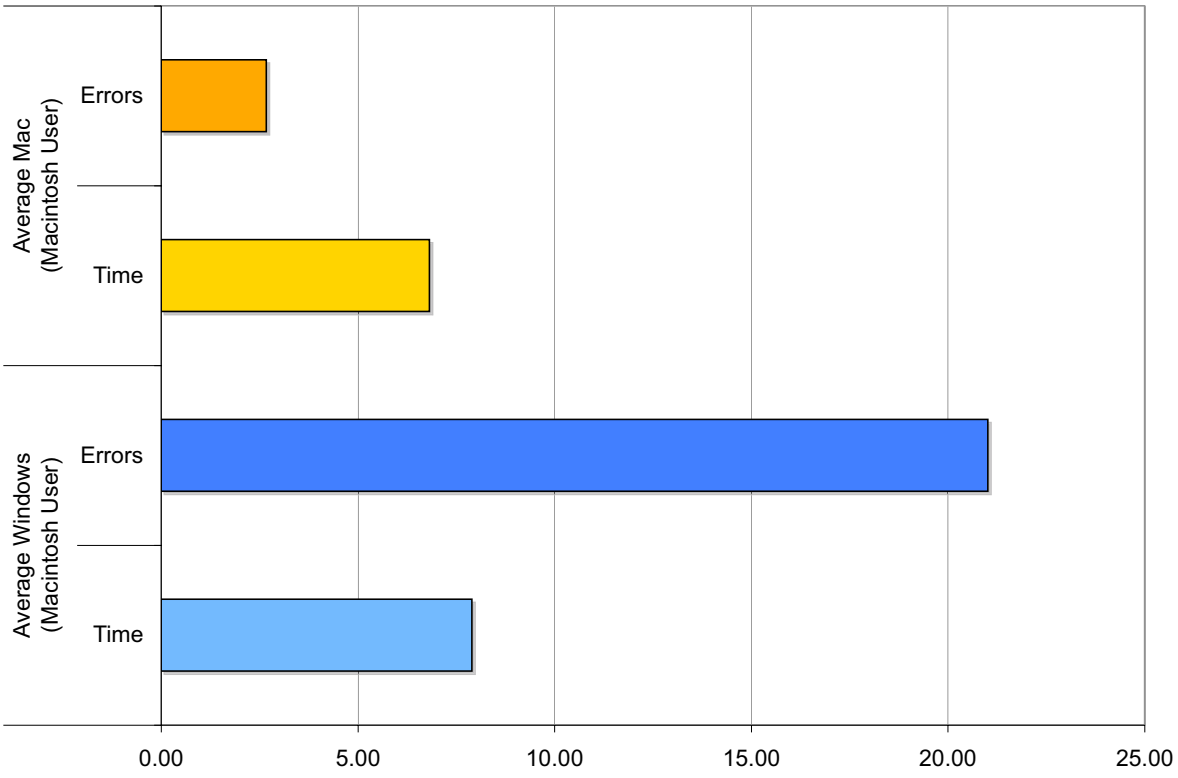


Chart 10 : Mouse Precision Measures: Macintosh User

Type of Benchmark:

Task-based productivity measure

Time in seconds. Shorter is better.
(Figures shown are the average of 3 individual benchmarks)

Reference: MP3

Notes:

This chart presents the average results for all the test series performed by the Macintosh user.

Mouse Precision Productivity Measures

Benchmark: Mouse Precision Measures: Average by OS Version

Focus: Hardware-Related Productivity Measures

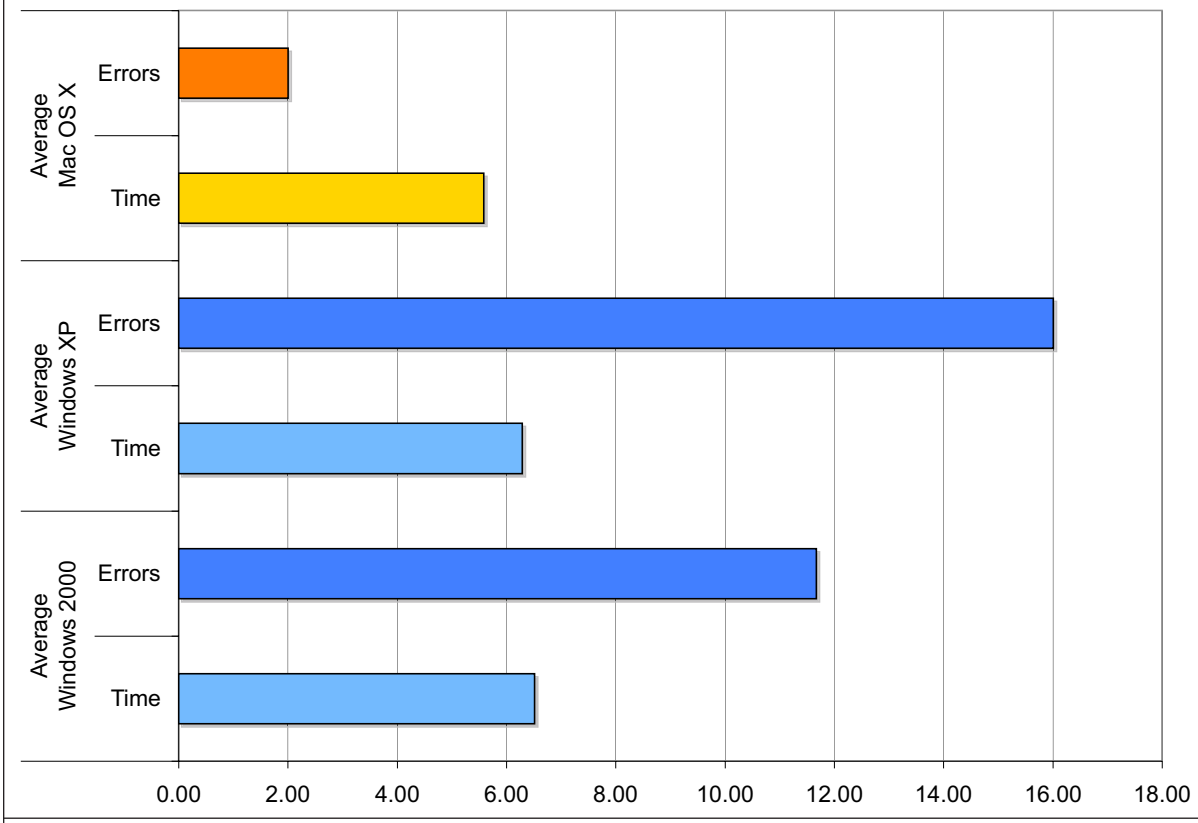


Chart 11 : Mouse Precision Measures: Average by OS Version

Type of Benchmark:

Task-based productivity measure

Time in seconds. Shorter is better.
(Figures shown are the average of 3 individual benchmarks)

Reference: MP4

Notes:

This chart presents the average results for all the test series, grouped by operating system version.