1. Introduction

As modern society becomes more and more dependent on computers, a growing need to understand computers and their workings has developed. However, because of the breakneck speed that better, faster, and more efficient technology is being produced, many users are not able to keep up with the constant changes in computers. Too often these users never learn to use the latest hardware or software and keep falling further and further behind as technology rushes on. In hopes of minimizing this problem, the technology industry has developed a new field of research with the goal of better understanding the interactions between a computer and its user. Known as Human Computer Interaction (HCI), this new field focuses broadly on both humans and computers in hopes of simultaneously increasing computer usability and human productivity [1].

While the field of HCI has only recently developed into a serious venture for developers, the struggle to improve human/computer interactions has always existed. For example, previous HCI inventions, such as GUI's or the mouse, have revolutionized computing by making computers significantly easier to use [2]. Although today's HCI inventions are often designed with more specific implementations in mind, the basic concept is the same - by making interactions with the computer easier, a task is subsequently made easier for the user. By examining the history of computer use, common problems faced by users, and new technological developments to better HCI, this paper aims to propose methods to improve interactions between humans and computers with the ultimate conclusion that educating the user, not more HCI technology, is the only permanent solution to the problem of computer usability.

2. A Brief History of Computer Usability

The first computers, having no operating system and no easy method for fast data entry, were exceedingly difficult to use [3]. They cost millions of dollars and could only be operated by one user at a time. The subsequent creation of the first operating system ultimately led to the use of batch processing in which computers could queue multiple jobs, allowing more efficient use of the machine. Later, terminal based machines allowed multiple users to simultaneously use the same machine [3]. However, at this time, the costs of computers and programmers were high, so programs generally were written in the fashion most convenient for the computer or the programmer, often at the expense of the user [3].

The 1970's witnessed the development of a revolutionary creation from Xerox's Palo Alto Research Center [2] that is still the mainstay of many of today's operating systems and programs, the graphical user interface (GUI). GUI's graphical interface facilitated the move away from text-and-keyboard operating systems [2], for which users had to memorize a multitude of commands and often gave only terse and limited responses [3]. An example of such a pre-GUI operating system is the DOS operating system, which contains cryptic commands such as "cd", "dir", or "md" that would baffle the inexperienced user. By reducing the requirement for such technical knowledge, GUI's made computers dramatically easier to use.

GUI's were first made commercially available by Apple Computers on its Lisa personal computer. After Lisa's failure due to its high cost, Apple marketed the lower cost and greatly successful version called the Macintosh [3]. Apple success was, in part, due to its realization of the need to cater to the user instead of continuing the tradition of imposing the computer's limitations or the programmer's will upon the user [3]. Other companies soon followed Apple's lead and also began implementing GUI's into their software.

In addition to the GUI, several other inventions have dramatically improved computing by facilitating ease-of-use. Such inventions include object oriented programming, the mouse, the trackpoint and touchpad, and plug-and-play hardware [2, 4]. The mouse was, in part, the key to the GUI's success because it allowed users to easily access and manipulate the graphical interface [2]. The trackpoint and later the touch pad allowed for the same ease of use for laptop users. Plug-and-play is a relatively new invention that allows users to simply plug in and use new hardware without having to install software, making computer use easier for all users. These inventions, in addition to many others, all have a similar underlying characteristic. They all make computing easier for the user by saving the user time and energy or by requiring less computer knowledge.

The opportunities for continuing HCI research are endless. Computer use remains exceedingly difficult for many users and researchers are constantly coming up with new ways to address these problems. Several interesting research paths currently being pursued by researchers are virtual reality, speech recognition, and programming languages that resemble spoken language [5]. The development and implementation of such systems would even further increase ease-of-use and lead to better human computer interactions.

3. The Direction of Current Research

Historically, developers have not paid much attention to the ease-of-use of their programs [1]. Instead, because programs with more features tend to entice buyers, market forces have pushed developers to focus on producing products with more functionalities rather than product that are easy to use [6]. This ideology results in new computer technology that tends to be overly complex and harder to use.

To address this problem researchers are trying to find ways to minimize user interaction with the computer [6]. By minimizing user interaction and reserving more difficult or knowledge-intensive tasks for the computer, computer use can then simplified to the level of the average user. An example of such simplification is the system clock in many modern operating systems, which automatically changes time during daylight savings and requires no user intervention. As a result, the user does not need to know how to access or modify the clock.

Because even such simple tasks like changing the time requires more knowledge than some users possess, an automated system is beneficial to the user and saves both the time and energy that would be required to learn to operate the clock.

In cases when user interaction is required, companies such as Microsoft and Intel try to keep user interaction to a minimum and, thus, requiring less attention from the user and also allowing the user to multitask [4, 6]. Microsoft and Intel believe that by treating the user's time and attention as a limited resource, their software and hardware designers are then forced to design more efficient programs that minimize user interaction and increase automation while simultaneously preserving functionality [6].

4. The Problem of Human Computer Interaction

"We [computer developers] spend most of our time with people who scored over 700 on their math SATs, we know people involved in IPOs and stock options, and we work with folks who take computers apart for fun. We forget that the people within our industry are very different from the rest of the world. That's why going into the usability lab or a focus group seems like a trip into the twilight zone. It seems like those users are in the minority, visiting us from some twisted and slower universe. The reality is this: We are the overwhelming minority."

-Scott Berkun, Former Lead Program Manager, Microsoft Corporation [7]

While hardware and software developers tend to be a homogenous, educated, and tech-savvy group, computer users are often just the opposite. As a result, functions that developers see as beneficial and easy to use may be just the opposite for the average user. Furthermore, great diversity exists among users. Users may be left-brained or right-brained, genius or illiterate, or hard of sight or hard of hearing, making the development of software that caters to the individual needs of every user nearly impossible [8].

However, several widespread concerns are shared by most computer users, such as computer security and loss of data. These are practical concerns that allow the user to protect his or her own personal security, such as protecting passwords or credit card numbers [9]. Ultimately, many of these shared concerns stem from a lack of user knowledge about computers. For example, a simple firewall can prevent the large majority of unauthorized access into one's computer. However, many users simply do not understand firewalls well enough to use them effectively block malicious access.

To make up for the user's inability to use more advanced computer functions, developers tend to create new software or hardware that requires less user interaction and knowledge to perform the same tasks [8]. This ideology is flawed because it often results in increasingly less knowledgeable users who they never truly learn how to perform the given tasks. An example of this may be found in the system clock example given earlier. Because of automation, the user does not need to know how to access or change the clock time. However, such technology is only a short-term solution. While this method does allow the user perform tasks that they previously may not have been able to do, but it does not improve the user's understanding of the process or general ability to use computers. In this case, the user does not learn how to set the clock. What would happen if the same user moved to another time zone? Because the computer does not automatically detect its location, it cannot automatically change the time for the user. Meanwhile, the user, who has previously relied on the automated time switch to take care of any changes, does not know how to manually set the clock.

Scenarios such as these are the downside of reducing user interaction. However, for the majority of users, this is a convenient trade-off. The ability to get things done without having to worry about how they work or needing to go through all the steps involved saves the user a significant amount of time and energy. Furthermore, occasions in which the user needs to perform operations manually tend to occur rarely enough that the some user may never have to learn them. For example, a use could go through his or her entire lifetime without needing to change the time zone for a computer. As a result, automation is able to reduce the workload on the human in most cases by taking care of commonly performed tasks, but at the ultimate cost of the user's understanding and knowledge of the computer's workings.

Computers should be made easier to use, but not at the expense of having less knowledgeable users. To address this problem, developers should also focus on methods of educating users on the operation of their computers. By educating users, they become more competent in computing, needing less and less automation in their software, and also able to use computers more efficiently.

5. Research Survey

To analyze common problems in computer usage, a survey of computer users was taken to find the major problems that users experience. The survey polled user concerns, complaints, and what changes the user would like to see in computers. Each question was accompanied by a list of choices and the user was asked to check all answers that applied. An additional space was left for the user to respond with his or her own answer if desired. The survey also asked the user's gender, length of computer experience, and self-rated computer knowledge.

5.1. Survey Design

Because this research paper is not geared towards demographics, questions regarding age, race, and location were not included in the survey. Instead, this paper focuses on users as a whole so that conclusions and proposals resulting from this study may be applicable on a broad scale. Furthermore, the researcher assumes that a user's level of computer knowledge, and not demographics, is the largest determining factor of the problems that the user encounters during computer use. As such, the survey divided responders into three categories of self-rated computer knowledge based on a general description of each. The three groups were named "basic", "moderate", and "extensive" on the survey and were chosen to prevent any major name bias. However, for the sake of this convenience, these groups will be referred to as beginner, intermediate, and advanced users, respectively. The descriptions that the user based his or her self-rating on appeared on the survey as follows:

> Basic – Can perform simple tasks such as word processing or web browsing
>
> Moderate – Can manage files storage, run virus checks, and/or perform simple programming
>
> Extensive – Can code complex programs or have an understanding of computer architecture.

The survey also asked for gender in hopes of discovering a significant relationship between gender and computing knowledge. In today's technology dependent society, computer knowledge is a great asset. Thus, if such a gender gap does exist, then efforts should immediately be put forth to eliminate any knowledge differential among the sexes in the interest of equality. As mentioned above, a more in depth demographic analysis of users was not performed because it is not the purpose of this paper.

The user was also asked to give the number of years that he or she has been using computers. While a correlation between length of use and computer knowledge would not be surprising, outside factors such as training or frequency of use are expected to heavily influence the user's knowledge. Therefore, the presence of a strong direct correlation between length of computer use and computer knowledge would indicate that such outside factors have little impact on users. Similarly, the absence of a strong correlation indicates that factors besides length of use play a role in a user's computing knowledge. The effects of these outside factors on users should be a significant factor in deciding what methods should be used to raise a user's computer knowledge. For example, if a strong correlation exists, then it would suggest that the best way to increase users' computer knowledge is to introduce them to computers early on because other factors, such as training or frequency of use, are less significant.

Finally, the questions asked in the survey were intentionally broad. The questions were intended to deter any program or function specific responses from the user and, instead, focus on broader issues of computers in general that are recurring and common problems. By finding the larger problems that computer users experience, solutions could then be proposed that have a more significant impact on computer use in general.

The survey was given in printed form and responders were asked to mark or write in their responses using a pen or pencil. This method was chosen over an electronic medium, such as distribution by e-mail or website, because the researcher felt that the use of computers to conduct a survey about computer use would introduced significant bias into the survey results. However, this choice, in additions to logistical issues, constrained the surveyed population to Duke University students, faculty, and staff. Although no record of respondent age was kept, the surveyed population was predominately college-aged students. Therefore, there is an admitted and significant lack of diversity in the respondents to this survey. However, in the interest of an unbiased survey, the relatively homogenous survey population was unavoidable. The results of the survey should be examined with these facts in mind.

The surveyed was administered on a voluntary basis at various locations throughout the Duke University campus. Students were approached at randomly primarily in dining locations, namely the Great Hall, the Bryan Center, the Blue Express, and the Marketplace. Faculty and staff were also approached randomly and asked to fill out the survey when they were seen around campus or in their offices. Attempts were made to obtain responses from various locations around campus in order to maximize diversity in the small survey population.

5.2 Survey Results

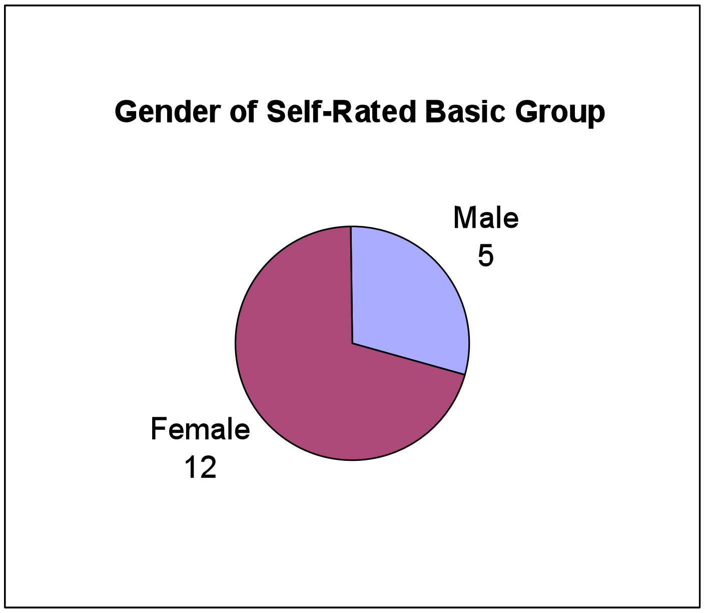

Figure 1.1

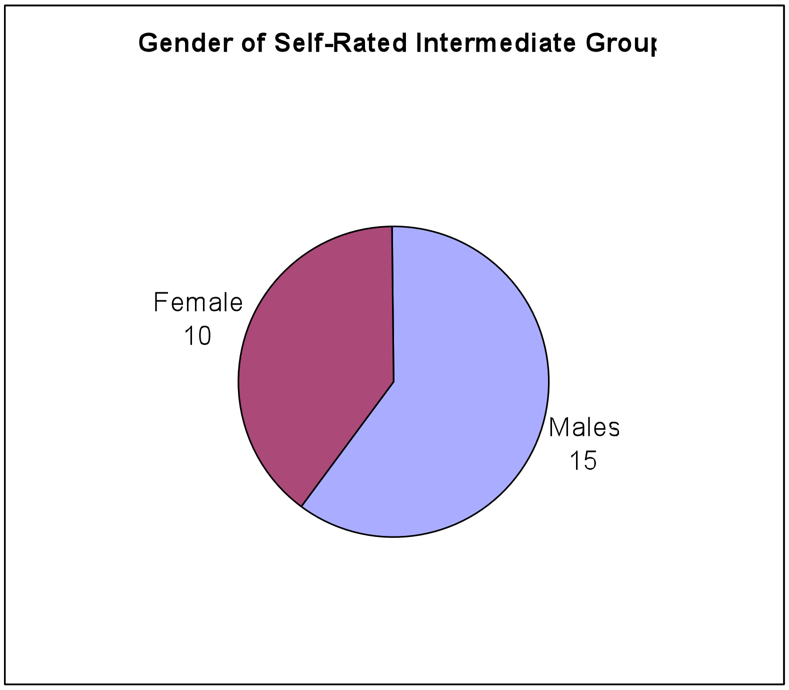

Figure 1.2

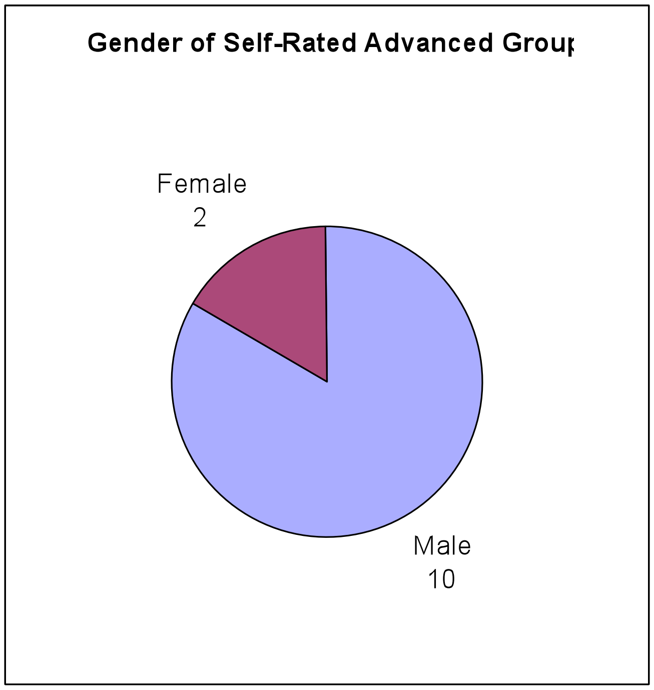

Figure 1.3

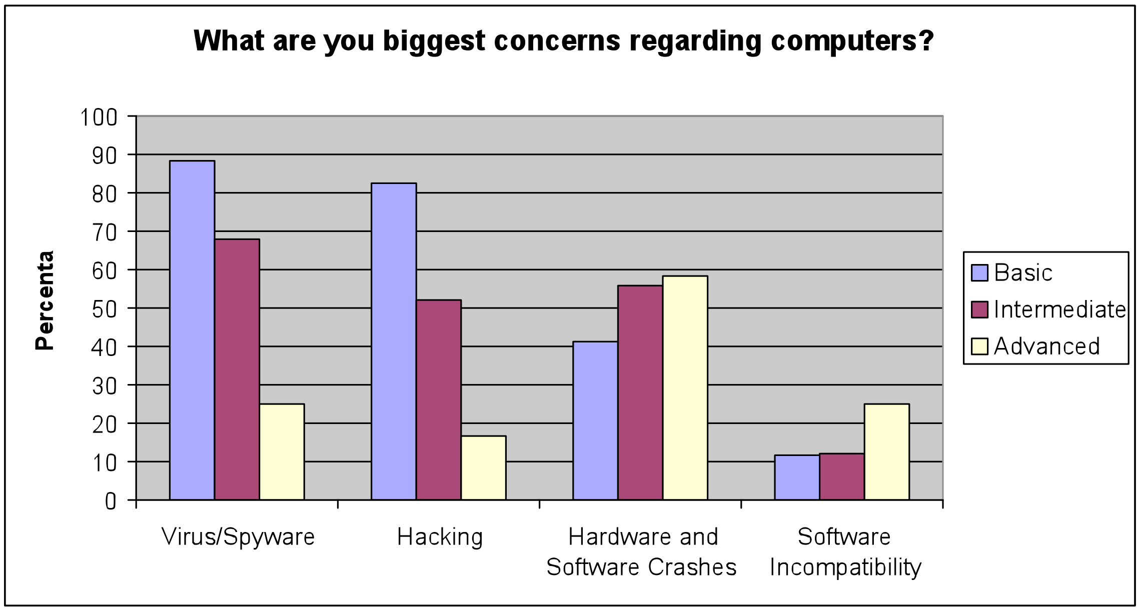

Figure 2.1

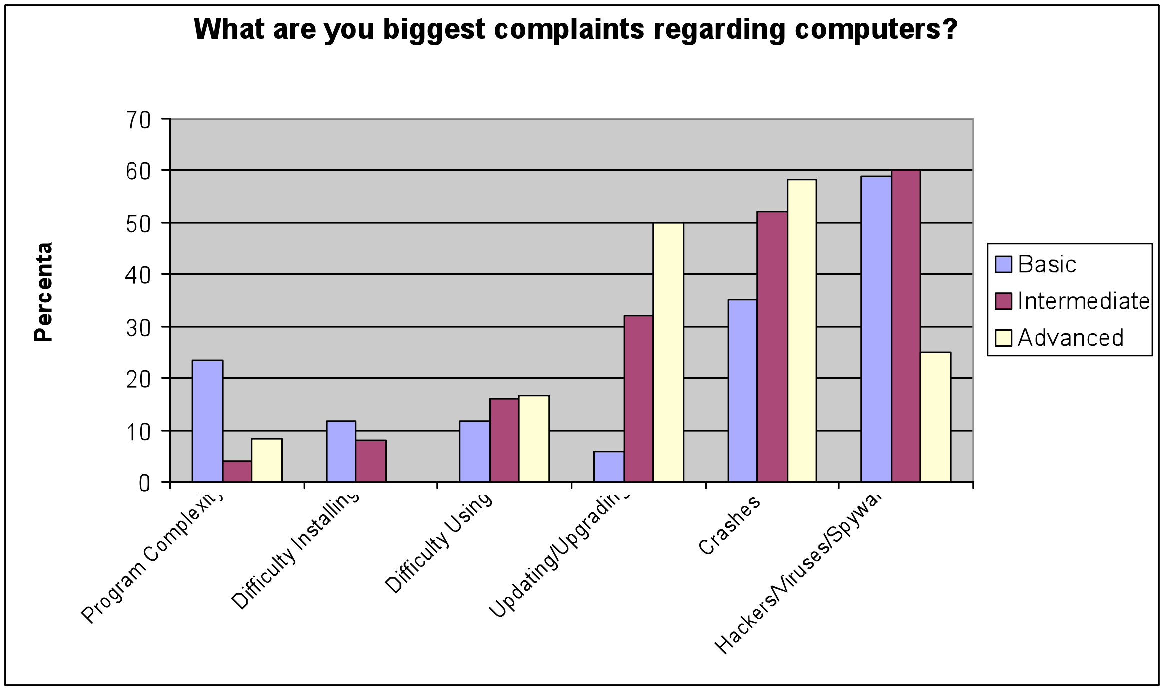

Figure 2.2

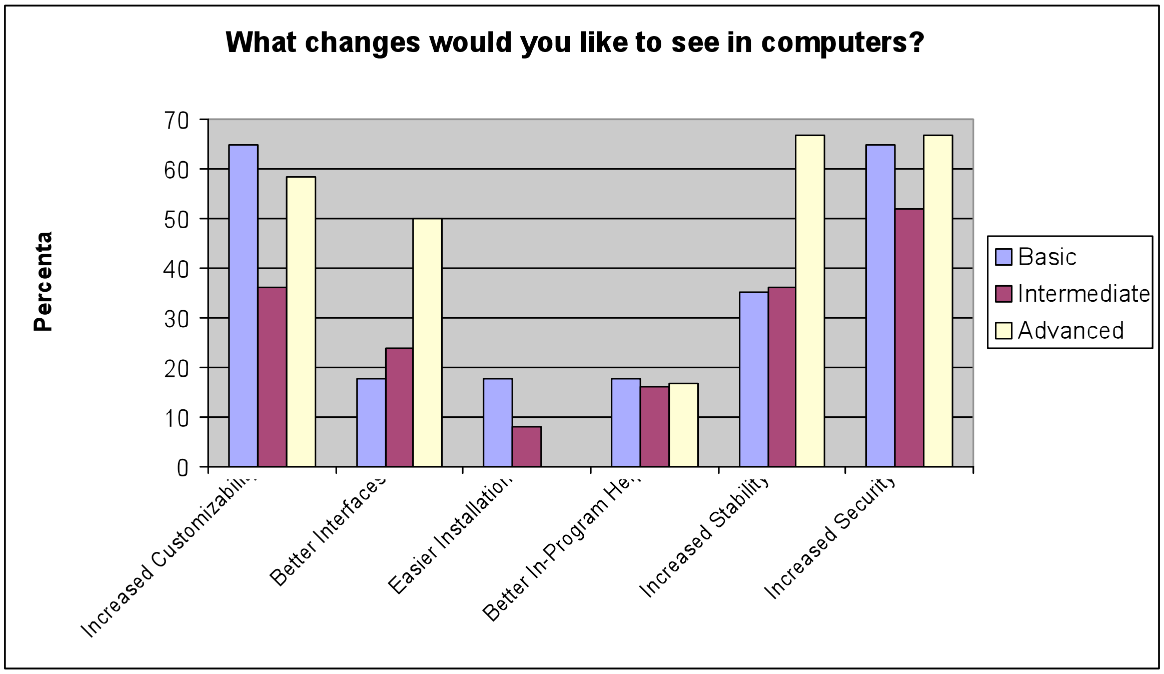

Figure 2.3

The preceding graphs summarize the survey results. The genders of the responders are given in the pie graphs and are listed by group. The bar graphs are represented in the percentage of positive responses per groups. The results are expressed as percentages so that they could be standardized and compared. Furthermore, no correlation was observed between user self-rated knowledge and length of computer usage, suggesting that length-of-use alone is not the determining factor of computer knowledge.

The pie graphs showing gender exhibit a trend that suggests a relationship between gender and computer skills. The basic-knowledge user group (Figure 1.1) showed a larger percentage of females then males. The intermediate group (Figure 1.2) showed a somewhat higher percentage of males than females and the advanced group (Figure 1.3) showed a significantly larger percentage of males over females. This trend suggests that females tend to be less versed in computer than males.

Responses to the questions show several significant differences in the concerns, complaints, and desired changes of the different levels of users. In terms of concerns, basic and intermediate users were primarily concerned about viruses and spyware, hacking, and computer crashes. Advanced users were largely concerned only by crashes. (Figure 2.1) The chief complaints of basic users were the frequency of hacking and viruses, the frequency of crashes, and the high complexity of software. Intermediate users also voiced complaints about the frequency of hacking and viruses and the frequency of crashes, but also complained about the constant need to update or upgrade products. Advanced users complained about the frequency of crashes and the constant need to upgrade or update. (Figure 2.2) When asked about changes that they would like to see, basic and intermediate users replied that they were most interested in increased customizability, stability, and security. Advanced users had the same complaints with the addition better user interfaces. (Figure 2.3)

5.4 Survey Analysis

When analyzing the responses to each question by group, an interesting trend appears to arise. The complaints and concerns of beginner users tended to be problems that can be addressed by the user. For example, the fear of viruses could be countered by installing anti-virus software. On the other hand, advanced users tended to voice concerns about problems that originate from issues associated with development of hardware or software. An example in this survey is the need for constant updating or upgrading. The need for updates or upgrades to fix problems or to increase functionality is determined by the developer and is not something users can control. Furthermore, some of the complaints, such as software instability, could be the result of both poor usage and poor programming. This could explain why both beginner users and advanced users share concerns about program instability. The trend of lower level users voicing complaints about problems that could be easily resolved by more knowledgeable computer use suggests that increasing user knowledge could address many of the problems that inexperienced users face.

6. Proposed Solutions

The problems of computer usage are not ones that are easily solved. They stem from a multitude of sources and some simply have no easy remedy. One of the major issues is the knowledge gap between software designers and those who use their software. Developers cannot be expected to think like inexperienced users, and users similarly cannot be expected to think like the programmers who write the software [7]. Thus, the problem of designing hardware and software that is easily used by all levels of users is a difficult one. While there is no single, simple solution, there are several methods that could potentially be used to make computers use easier.

Currently, many useful methods for easing computer use exist. These methods may be as simple as including a help menu or may be as complex as implementing software that can recognize voice commands. However, while these methods have allowed users to continue to operate software even as newer version grows more and more complex, they are all ultimately designed only to allow the user to keep up with technology, rather than learn the underlying skills [1]. Instead of producing programs with a multitude of functionalities coupled with methods for allowing users to easily interact with the programs, a better alternative is to create programs that are able to teach users to the needed skills to learn and master the program's functions.

As a simple analogy to this idea, modern software could be compared to a multi-tool that can perform hundreds of different functions. In the hands of an experienced user, such a powerful tool could craft a work of art. However, an inexperienced user does not know how to use most of the tool's functions and, thus, he or she does not have any use for the functions. To these users, modern software, full of shortcuts to ease its use, can be seen as a machine that will make a piece of art at the push of a button. While this is convenient, the user ultimately does not know anything about the working of the machine or the product it made. In the case that one were to break, then the user would not know how to fix it.

This paper proposes solutions to human computer interaction problems based largely on this ideology. Rather than propose new, innovative ways to make interaction with the computer easier, this paper focuses on steps that can be taken at the software design level to educate the user on the workings of a program or computer.

6.1 First-Time Use Setups

An initial setup option that allows an user to select his or her own level of use is a possible method for giving inexperienced users extra help without hassling more experienced users. Such an option could be presented to the user before installing a program and could be used to help novice users through the installation and setup process. For example, if the user states that he or she is inexperienced, the program could automatically install into the Program Files folder and install all default components and settings. This would make the installation of a program as simple as possible and would prevent the inexperienced user from unknowingly overwrite files or other potentially deleterious mistakes. Furthermore, if the user wished to learn more about the installation, a walkthrough explaining each step could be similarly helpful. On the other hand, experienced users could be presented with additional installation options that were not presented to the inexperienced user. This method would simplify and explain installations for inexperienced users while offering all options to the experienced user.

The user's selected level of use could also be applied while the program is running. If a low level of use is set, extra in-program instructions or help could be given aid to the unknowledgeable user. For example, the program could present an introductory menu to walk a new user through the program's setup or functionalities. Be leading the inexperienced user step by step through the installation and operation of the program, any guesswork is taken out of learning and using the program. On the other hand, experienced users could immediately be presented with menus to setup advanced program options or personal settings. Thus, the program is customized so that users are presented with options that, depending on their skill level, are more relevant to them. Experienced users would not have to read through information that they already know and inexperienced users would not have to setup options that they know nothing about.

6.2 In-Program Assistance

Increased assistance while using programs could similarly increase the usability of computers. An excellent example of such in-program assistance is Microsoft Word's Office Assistant. The Office Assistant allows for quick access to frequently asked questions. It also detects when the user is trying to perform certain tasks, such as making quotes, and offers pointers and information related to the task. Such additions that provide quick answers to problems allow the user to adjust to and learn the program more quickly. Once a user's knowledge has advanced beyond the need for the Office Assistance, it may simply be turned off.

Perhaps the most simple and effective of such in-program assistance is the help bubble. The help bubbles referred to in this paper are accessible from directly in a program and do not require a separate application or window. For example, in many programs, when the mouse is held still over a button or icon, a name or description pops up. This simple tool allows the user to easily identify buttons and icons whose functions may otherwise be unknown. A widespread implementation of such easily accessible descriptions would significantly increase usability for first time users by quickly and easily revealing how different aspects of the program work.

Along the lines of help bubbles and menus is the need for clearer and more detailed help files and error messages. Although less so than the past, help files and error messages still often tend to be technical and difficult for even the typical computer user to understand [7]. Furthermore, they are often also terse and undescriptive. For example, many software crashes in Microsoft Windows prompt the following unhelpful error message: "The Program Has Performed an Illegal Operation and Will Be Shut Down. If the Problem Persists, Contact the Program Vendor". The error is then followed by a cryptic description often full of computer terminology and hexadecimal addresses [10, 11] that are incomprehensible to all but computer experts.

The obvious remedy to this problem is to create help files or error messages that are more comprehensible to the average user. One solution to this problem is to write two versions of help files or error messages: one for the typical user and one for the more advanced user. For example, a "stack overflow" error could be coupled a second message that read "Program X used up all of its allotted memory and was ended to prevent it from crashing other programs". Because it is a problem with the program's code, this particular new message does not directly help the inexperienced user fix the problem. However, it does describe the problem in a way that allows the user to understand the problem. In other cases, if the user understands the problem, then he or she can prevent it from occurring again. An example is low virtual memory warnings, which may appear in terse messages, such as this message that occurs in Microsoft games: "Your system is low on virtual memory" [12]. The message does not offer any explanation or solutions to the problem and ultimately is of little aid to the user. On the other hand, if the user were told that the computer's virtual memory is low because too many programs are running, then he or she can easily fix the problem and even prevent it from occurring.

Although some of the solutions proposed in this section are already in existence, they are far from reaching their full potential. In-program tools, such as help bubbles, vastly increase the usability of programs by offering quick references to otherwise vague icons, buttons, messages, etc. Other tools, such as error messages, tend to be vague or technical and can easily be improved. Ultimately, developers should take advantage of these tools by using them not only to inform users, but also as a tool to educate users about the workings of their computers.

6.3. Open Source

Open source software has great potential for increasing the usability of programs. The inherent customizability of open software allows it to be easily modified to create different levels of use, which are created by patching functionalities into programs. For example, a new user may only require a program with basic functions. Experienced users, on the other hand, may upgrade the same program by patching in advanced functionalities. Thus, open software can cater to the needs of both experienced and inexperienced users simply by allowing the user to add functionalities as needed. In comparison, proprietary software is often designed from the beginning to incorporate as many functionalities as possible [13]. This makes the program relatively advanced and may make learning to use the program more difficult for inexperienced users.

The use of open software patches also has the advantage of allowing users to seamlessly progress to higher levels of use. Because new functions may be patched in one at a time, a user can add functionalities as he or she develops the need for them. Thus, as a user begins to use programs for increasingly complex tasks, he or she can patch in new, more advanced functions. This allows the user to add in more functions as he or she masters old ones. As a result, the level of complexity and functionality in a program rises as the user's knowledge of the program rises.

7. Benefits of Better Human Computer Interactions

An increase in the usability of computers would have a significant impact on today's technology dependent society. Computers are now incorporated into nearly all aspects of our lives, from the cars that we drive, to the registers at supermarkets, to personal entertainment, and even to communications. Easing computer use would have a positive effect on all of these aspects.

From a business perspective, increased ease-of-use results in increased productivity because the user is able to use programs more knowledgeably and with a higher degree of efficiency. Furthermore, programs that incorporate tools designed to help users, such as the walkthroughs or help bubbles described earlier, could allow users to learn to use new programs on their own. This would then translate into less time and resources needed to train users to use a program.

Open source software offers the unique ability to customize software to the level of the user. It has the benefits of minimizing the complexity of a program by eliminating unneeded functions and also of maintaining a high degree of upward mobility by allowing users to patch in new functions. Thus, users do not need to expend the extra resources required to learn functions that they may never use, as may occur with proprietary software.

8. Conclusions

Technology has become an integral aspect of today's society and computers are at the forefront of the rapid technical growth that we are experiencing today. Computers are everywhere: changing traffic lights, playing music from a CD, or even making our coffee in the morning. Although much of this technology is automated, at some time, a user must interact with the computer and such interactions require knowledge about the computer. However, as computer technology races forward at incredible speeds, users are often uneducated about new technology. What good is widespread and powerful technology if no one knows how to use it?

Education is the key to improving human computer interactions. With technology's rapid growth, the only way to truly keep up with the changes is to learn how the new technology works. Knowledge of a computer's workings allows the user to operate it more efficiently and skillfully. Methods to make existing technology easier to use and more user-friendly allow users to operate increasingly complex programs while their knowledge about computers remains stagnant. This is ultimately a short-term solution. When problems arise, the uneducated user is out of luck. Only through education can users become competent in computer use.

As the popular proverb says: "Give a man a fish; you have fed him for today. Teach a man to fish; and you have fed him for a lifetime." Providing users with shortcuts to perform functions that they otherwise would not be able to perform is a convenient short-term solution, but it does not solve the problems of human computer interaction. Instead, focus must be placed on educating users so that they can understand and take advantage of the computers that are continually revolutionizing the world that we live in.

Research Survey

Gender:___________ Age:______________

1) How would you describe your knowledge of computers in general (check one):

Descriptions:
Basic – Can perform simple tasks such as word processing or web browsing
Moderate – Can manage files storage, run virus checks, and/or perform simple programming
Extensive – Can code complex programs or have an understanding of computer architecture.

Basic_____ Moderate______ Extensive______

2) How long have you been using computers for? __________________________

3) What are you biggest concerns regarding computers (check all that apply):

____ Viruses/Spyware
____ Hacking
____ Hardware crashes (ei. Computer crashes)
____ Software crashes (ei. Internet Explorer crashes)
____ Software incompatibility (two programs do not work together)
Other: ____
____
____

4) What are your biggest complaints about computers (check all that apply):

____ Programs are too complex
____ Programs are too hard to install
____ Don't know how to use programs
____ Need for constant updating or upgrading
____ Fear of crashes
____ Fear of hackers, viruses, or spyware
Other: ____
____
____

5) What changes would you like to see in computers (check all that apply):

____ More customizability of programs
____ Better interfaces (ei. buttons can be moved around or better appearance)
____ Programs made easier to use/install
____ In-program instructions on how to use programs
____ More stable programs that don't crash as often
____ Better security against hacking, viruses, or spyware
Other: ____
____
____

Works Cited

1. Lorenzi, L., *HCI*. 2003.
2. *GUI*. 2001.
3. *Ease of Use*. 2002.
4. IntelCorp., *Ease of Use Initiative*. 2005.
5. Bundy, A.C., Phil, *Ease of Use*. 1997.
6. Berkun, S., *Importance of Simplicity*, in *The Human Factor*. 1999.
7. Berkun, S., *Why Great Technologies Don't Make Great Products*. 2000.
8. IBM, *What is User Experience Design?* in *IBM Ease of Use*.
9. Kinkus, J.F., *Science and Technology Resources on the Internet*. 2002.
10. *Deciphering Ten Dumb PC Error Messages.* 2005.
11. Joffe, D., *Microsoft Crash Gallery*. 2004.
12. Microsoft, *You receive the "Your system is low on virtual memory" error message when you start or play a Microsoft game*. 2005.
13. Anderson, R., *Why Information Security is Hard*.