Universidade Católica Portuguesa

Católica Lisbon School of Business and Economics

"The impact of Software as a Service in Software Piracy"

Has the change in the distribution and sale of software provided not only an accurate answer against software piracy but also an increase in consumer-value?

Advisor: Professor Nicoletta Corrocher

Discussant: Professor Lars Jeppesen

Dissertation submitted in partial fulfillment of requirements for the degree of MSc in Business Administration, at the Universidade Católica Portuguesa, December, 2012.

Master of Science Thesis of: João, Fonseca George 1609221

2012-2013

Table of Contents

Preface

Writing a thesis is more than a simple assignment but the work that represents all that one has learned throughout his academic years and, as such, requires a full time commitment as well as big amounts of will and patience. It is important to add that these conditions are hard to achieve in isolation, so, after fulfilling this task, one has to acknowledge that the support given by some people during the grueling exercise of writing this dissertation played a pivotal role in the overall outcome. As such, I would like to specially thank the following people:

- **Professor Nicoletta Corrocher**, for her wise advice and hands on approach throughout this dissertation, even knowing that the time was not the best for her;
- **Professor Lars Jeppesen**, for showing me a different side of software as well as for being an inspiration;
- My parents, **Ana Margarida Fonseca George and João Fonseca George** for all the expertise that they have lent me on software, for making the time to debate some of the issues of my dissertation with me and providing me with the tools needed to complete my academic education;
- To **Charlotte Manojlovic**, for all the support she has given me and for showing me that one can find a light even in the darkest of places;
- To **António Von Amann and the Lokomotiv Lisboa** football club, for teaching me that without heart there can be no glory;
- To **Benedita Quintanilha** , for being there for me for as long as I can remember.
- To **Beata Tomasiak**, for showing me that one has the responsibility of aiming as high as possible.

0. Abstract and Literature Review

This dissertation aims to answer the question “Has the change in the distribution and sale of software provided not only an accurate answer against software piracy but also an increase in consumer-value?” and it supports its argument not only on the current literature available on the software industry and on software piracy but also on the personal research carried out through an extensive survey of different people in different age groups and from different countries. To aid the analysis, strategic frameworks such as PEST and Five Forces of Porter were used, giving the reader a special landscape view of the industry’s characteristics to better understand how the changes in software distribution will impact consumer demand and satisfaction.

It is important to note also that, during the literature review stage, it was understood that authors disagree on several important issues regarding software such as its classification, its origin and whether or not it constitutes patentable subject matter. This particular path led me to question the actions of software companies and whether there was a need or not to patent software. Through this road, I was led to the issue of piracy and to the action that companies have taken with changing the way software was produced and sold to better fit the current landscape of the industry.

1. Introduction

Throughout the last century, with the evolution of computers and the increasing affiliation of human beings with computerized machines, software has become an increasingly important part of people's lives. As such, the industry of software has grown and evolved with the different software programs becoming increasingly more complex in order to cater to the ever growing needs of the market, forcing changes in the way software is perceived, produced, sold and distributed along the way. In fact, the software industry has evolved from being a complementary industry, one that was born on the back of the hardware industry from the evolution of computers into more complex yet attainable machines with estimates pointing to a 303 Billion Dollar[1] industry value from both direct and complementary business. In spite of this being an overwhelming number, industry pundits expect this number to grow in the future as new software arises to meet consumer demand.

In the time interval in which this evolutive process took place, software undertook several changes in its distribution form, going from coded tape to individual CD-ROMs, then to Product Key licenses, and, more recently, software programs started being distributed online as part of a new model of sales called Software as a Service[2].

This new innovative way of selling software is being adopted by more and more companies not only as a way to provide the consumer with added value to their product but also as a response to market externalities. The truth is that, much like in any other industry, the growth of the software industry did not happen without it facing some industry threatening issues. One of these issues, perhaps one of the most important ones, has been software piracy. As a general rule, piracy has been a very difficult topic to deal with for companies in several industries such as the film making industry, the music industry and, of course, the software industry as more and more methods of corruption and duplication of software have risen, making piracy somewhat of a normal practice. Using this expression to describe piracy may be viewed as an

[1] US dollars.
[2] Shortened to SaaS.

overreaction or as an exaggeration of the reality in question, however, by thinking about it one arrives to the conclusion that what was once considered a hideous crime against the effort of people and companies has now become a common practice which people normally view as being harmless. Regardless of such considerations, the fact is that the improvement in communication conditions given by the internet has fostered the rise of global piracy rates to uncontrollable levels in the last 15 years, affecting all industries and causing massive losses for the companies in them. In the particular case of the software industry, software piracy has cost software companies more than 63 million dollars[3] in lost sales[4] with this value constituting, obviously, a very worrying figure[5].

Because of this rise, software companies were forced, year after year, to find new solutions for protecting and, later on, distributing their software so as to stand a chance against these perpetrators and their illegally distributed copies of software. Of course, when discussing this issue one must remember that piracy is not an independent phenomenon in itself but is rather induced by certain factors such as the easiness to copy or, more importantly, the predisposition of the user of a given product to copy said product. With the evolution of the internet and the globalization effect that such emergence has entailed, software piracy has become a very simple activity to practice while engaging in piracy actions has become a while accepted action. Estimates point out that the number of people engaging in such acts has grown year after year and software companies have had a tough time fighting piracy as, to do so, they relied not only on themselves and the minds of their consumers but also on the action of third parties such as governmental authorities.

Today, software companies are looking at the adoption of the Software as a Service model as a much needed aid towards fighting piracy. It is now widely believed in the industry that these companies can reduce piracy by adopting software as a service as they would be reducing the consumers' incentives to take part in piracy acts.

[3] Estimates the Business Software Alliance in their 2012 Global Piracy Study.
[4] Sales that did not occur because the consumer was able to obtain an illegal copy of the software.
[5] See Figure 1.

In line with this, several software companies have started redesigning their distribution and sales operations in order to cater to the needs of selling software as a service. As a consequence of this change, we now see software companies selling their products through online platforms that they have devised, rather than distributing such programs using a CD-ROM or any other physical item. Concepts such as “cloud computing” and, more specifically, “software as a service” have fueled a change that resulting in companies now providing their products directly online, making it easier for the consumer to purchase such products as well as, in most cases, demanding costumer registration in such platforms so as to provide him with updates on the product and other complementary services. This represents a nearing between the customer and the company that did not exist before and may prove to be an important ally in the fight against counterfeited software.

Now, understanding the problem that the software industry faces when looking at software piracy and realizing that there has been a shifting trend in distribution methods regarding the companies in this industry, this paper ultimately aims to study how well this change in the distribution model of the industry has fared at combating software piracy and whether or not it brought more value to the end user of software products. In order to be able to formulate an answer for the underlying question presented before, this dissertation will follow a strict methodology which consists in understanding and underlining the characteristics of the industry, so that we can better understand the result of the solution; knowing the mind of people engaging in piracy acts, in order to understand if this change is capable of meeting their demands and get them to drop piracy; and finally, learning the current global situation and trends of software piracy.

Before we can actually begin the study and to not run the risk of losing the audience when referring to given terms, there are some definitions of concepts used in this dissertation as well as some general guidelines that should be clarified.

A) What is Software?

As we will be talking about the software industry in the entire dissertation, we must begin, of course, by defining the term “Software”. The denomination “Software” is said to have been firstly used in 1958 by the late American statistician John Wilder Tukey in an article he wrote in the American Mathematical Monthly journal.

In spite of the existence of several definitions, “Software” is widely accepted as being the term given to a certain array of mathematical instructions and computer data that are read by a computer system which will then produce a given output according to that array of information. Commonly speaking, software represents a group of instructions that are meant to be read by a computer processor which will, in turn, change the action of one or more hardware components of the computer machine according to those instructions. As such and, in a simplistic matter, one could say that Software is the language spoken by computer processors.

B) How will we distinguish between the different software programs?

As the previously written definition hints, the field of software is a very complex one. There are different software programs can perform very different functions and this leaves room for a lot of different ways of classifying software. This, along with the fact that the term software and the industry that was built around it are relatively recent, has created some controversy over the different types of software that exist in the industry.

In this particular dissertation we will, however, consider the approach that distinguishes different groups of software programs based on their function and on their accessibility.

C) What types of software are there according to function?

When classifying software programs regarding the function, several categories have emerged however, most authors agree on the existence of two major software categories: System Software and Application Software.

In a simplistic definition, System Software is what we call the group of different software programs whose function is to allow the different hardware components of a computer system to work all together, thus, helping the computer system to run. Good examples of System Software Programs are operating systems like Microsoft Windows, for instance, file managers or network control devices.

As for Application Software, it is the denomination given to a group of software programs that are used to fulfill certain tasks which are not exclusively vital in order to run the computer machine but, on the other hand, satisfy certain needs of the user. Examples of Application Software are utility programs such as Microsoft Office, which comprises a word processor, a spreadsheet manager, a database manager along with other programs or business software programs.

It is also important to note that when classifying software according to function, two other groups are normally mentioned by most authors: Malware and Programming Software. The first of these two groups, Malware, refers to all the software programs that constitute malicious software, that is, that are meant to threaten the security of a computer machine in any way regardless of doing so with the intent of harming the computer, causing financial loss to the user or making the machine perform a function which it was not asked to do by the user such as view an advert for a given product; while the second, Programming Software, is the name given to the group of software programs that either allow a user to write new computer programs or assist them in such tasks. Although I understand that both these two types of software are very different from other application software programs, I share the belief of some authors that both malware and programming software are created to “fulfill certain tasks which are not exclusively vital in order to run the computer machine but, on the other hand, satisfy certain needs of the user” and as such, when classifying software according to function, they should be grouped, in my opinion, as Application Software. As such, all Malware and Programming Software will be treated as Application Software throughout this dissertation.

D) What types of software are there according to accessibility?

Besides being classified in regards to their function, software programs are also grouped usually according to their Accessibility. The term “Accessibility” refers to the way in which a software program can be accessed by users, which is, of course, regulated by the framework of economic and legal barriers that the legal owners of the software choose to place upon it.

Essentially, according to the accessibility of software, I shall refer to four groups of software programs throughout this dissertation.

The first group is called Proprietary Software and it groups to the most commonly known software programs as it corresponds to software whose legal rights are exclusive property of the copyright holder. This means that the product is only distributed with the permission of the copyright holder and users can normally obtain it for a given fee.

Secondly, we have the group of software programs that are normally called Freeware. Freeware corresponds to programs that can be obtained for free on the internet as the copyright holder of those programs chose to waive his economic rights, not posing any economic barriers to users that want to obtain the product. It is however important to flag that the copyright owner still maintains his legal rights in this type of software, meaning that users cannot tamper with the product or profit from his work. Usually this is done so that an unknown software developer can get some reputation/bragging rights or when an amateur programmer develops software as a hobby and wants to be reviewed on his work.

Thirdly, I will consider the group of Custom Software. Custom Software represents all software programs that are developed for exclusive use of a given entity, may it be a person or a company. It is, thus, especially personalized to fit the needs of that entity and such entity is licensed to use it. This entity may not, however, resell the software[6] as it is customized to the company and the owner of the software maintains his economic rights over it.

Lastly, we have Open Source Software. Open source software is said to be so when, after a company has made the source code of a proprietary software (or part of it) available, independent users have changed the code leading to the

[6] As a general rule but this depends, of course, on the terms of the licensing agreement the licensor and the licensee.

emergence of a new software program. It is important to note that this type of software has become widely known as the cases of successful open source software cases now constitute a flag for anti-software patent movements according to which profits of intellectual property should flow to society rather than to a group of individuals.

E) What is Innovation?

Another important definition that we should lay down before beginning is the one we will accept for the concept of “Innovation”. Innovation can be defined as the appliance of society wise accepted ideas that facilitate the fulfillment of specific market needs by allowing the generation of more effective products, services or processes relatively to the existing standards at the time of the introduction of said ideas, creating a new level of performance which was unable to be reached beforehand. In this dissertation, the term “Software Innovation” will be used often so it is also important to say that, as much as this dissertation is concerned, software innovation is an innovation that manifests itself in the software industry.

2. The Evolution of the Software Industry

Now, having defined these base terms and having showed how I see the different types of software present in the industry, I feel that it is vital for our goal of understanding how the beforehand mentioned changes in distribution have impacted the role of piracy in the industry to briefly go over the history of the software industry. The reason for this is that by going over the past of the industry we will not only study and gather its most important characteristics but also understand how the view towards software programs has changed over time.

However, before we embark on the tale of the evolution and growth of the software industry I must point out a few important guidelines. First of all, as author and computer scientist Martin Campbell-Kelly notes in most of his work[7], it is extremely difficult to pinpoint the start of the software because, at the time, software as conceptually viewed as an extension of the technological advances that were taking place in the hardware industry, that is, as a new component of the technological landscape that complemented the pre-existing hardware units, allowing users the possibility of controlling and coordinating several hardware systems to amount to a specific goal. Secondly, and building up on the previous point, it is important to understand that the growth of the software industry has gone hand in hand with the growth of the computer industry as these two industries began as one[8]. Finally, I have to stress out that the main topic of this dissertation is not the history of the software industry and, as such, the brief historical summary of the industry that I will present here comes from a study of what has been documented about the matter as well as the application of the concepts that I have learned during my studies which allow me to understand the written works of other authors in my own way.

In order to make it easier to understand and to relate to our final goal as well as to describe it better, I have divided the software industry into four periods. In the first period I will talk about the emergence of the software industry (how software came to be, what was it used for and how was it perceived at the time), while in the second period I will show how the industry

[7] Mainly in "The history of the history of software" and "From Airline Reservations to Sonic the Hedgehog: A History of the Software Industry (History of Computing)".
[8] See next paragraph.

developed itself and how one of the most important companies in this industry emerged standing out from the rest. The third period is dedicated to the time interval between 1983 and 1995 in which the software industry matured and, finally, in the fourth part, the modern software industry will be discussed.

i. Software Contracting and the origins of the Software Industry

In spite of being, in our time, a relatively independent industry, the software industry took its first steps on the shoulders of the hardware industry, and so, to understand the beginning of the software industry we must go back to the beginning of the hardware industry. At the time, when the software industry began without knowing so, computers were more connected to science than to the business front that we see today. As a result, the first computers were massive big iron machines that would occupy large land masses and were used mainly for the empirical investigations of certain innovator scientists. In fact, while what became known as the modern computer appeared during the Second Great War, the beginning concept of such machine came from the mind of innovator mathematician Charles Babbage, in 1834[9], while attempting to devise a machine that would automatically solve mathematical equations. Today we know, of course, that Babbage's idea never came to light during his lifetime as the technology he needed to build his machine was not available, however, thanks to key technological advancements[10], the first modern computers were able to come to life. This happened when, in 1944, IBM[11], unveiled the Mark 1 and when in 1946 the ENIAC, which is known for being the "first electronic digital computer"[12] in the United States, was completed and ready for use. Again, the uses for these computers were mainly scientific[13].

[9] In "Hard Drive: Bill Gates and the Making of the Microsoft Empire" by James Wallace and Jim Erickson.
[10] These technological advancements included the punch card contribution that was given by statistician Herman Hollerith.
[11] Through the work of Howard Aiken.
[12] In "Hard Drive: Bill Gates and the Making of the Microsoft Empire" by James Wallace and Jim Erickson.
[13] The Mark 1 had the particular feature of being able to multiply two 23 digit number in a total time of up to five seconds and the ENIAC was originally built to be able to elaborate artillery firing tables by anticipating the trajectory of big gun shells but it was later also utilized by the physicists responsible for building the world's first neutron bomb

These early uses were usually government funded jobs, one of which was the development United States' first air defense system[14].

In all these machines, software was viewed by both the consumers and the manufacturers, which at the time were the likes of IBM, RCA, General Electric, Honeywell, Burroughs, NCR, Sperry Univac and the Control Data Corporation[15], as a part of the hardware that they were buying and selling, respectively. It was only by the end of the 1950's that computer manufacturers first realized that there was a place for a software industry in the market. Around this time, because these computers were so huge and complicated to operate machines, often requiring an intermediate, scientists and engineers craved for a smaller machine, one that they could operate themselves[16]. Knowing this, the hardware manufacturing companies began to realize that there was a much bigger market for computers as they could also sell them for commercial purposes and, as such, they began developing less specialized computers so that they could cater to more global purposes. For this, these companies started writing their own software programs; however, since every company used its own parts and developed its own programs in the language that only their own processors would understand, there was an emergence of very specialized software applications that would only run in a given machine. This represented a road-block for both the consumers and the manufacturers because if the computer machines were being produced with this one-machine-only software, the cost of switching to another machine was high, which in turn reflected on the production side as every new machine would bring with it a new software.

It was in this moment that the producing companies became aware that if they could make the instructions that composed their software programs less specific to a given computer, that is, if software was not produced exclusively to serve one machine at the time, then the software of such machines would be produced by the consumers themselves and not by the manufacturers. For the consumers, this would work out as they would be able to program their own software products tailored to the needs of their companies and it would also be benefic for the hardware producers as the ability to create software programs

[14] As was SAGE.
[15] IBM dominated the market. The other seven companies were known as the "seven dwarfs".
[16] The invention of the semiconductor prompted this mentality as it allowed smaller machines.

would provide added value to their machines, prompting their sales. It was in this moment that, in my opinion, the software industry started to dissociate itself from the hardware industry[17].

To follow up on this notion, the hardware companies started to concentrate on laying the foundations that would allow different people to build their own programs, by which I mean programming languages. Since the logic was that the companies buying the hardware would be the ones creating their own software programs, there was a gradual evolution in programming as the languages that the computer machines began to incorporate and read became simpler for the consumer to understand and deploy. Soon after, more advanced and widely accepted languages begun to appear such as FORTRAN[18] or COBOL[19].The introduction of these higher level programming languages facilitated the creation of software as it allowed users to design software programs more easily.

The decrease in difficulty of programming given by the new languages prompted users' ability to design even more specific software programs which, in turn, fed the need for the generation and incorporation of even higher quality languages. With this in mind, in 1964, understanding that in spite of being widely accepted at the time, FORTRAN and COBOL were still somewhat difficult to master, Dartmouth College professors John George Kemeny and Thomas Eugene Kurtz took programming one step further when they invented BASIC. Far from being just another programming language, BASIC was a relatively simple to learn language that helped ease the communication between the hardware and the user.

Around this time and keeping up with the development of software programming languages, the first software companies began to emerge. Their activity was to, by using the languages provided by different hardware providers, program software programs for different companies. The activity was well received at the time as it saved companies the time and the cost of having to employ in-house programmers to write their software. Of course, as more

[17] It would be, thus, in this moment that I would place the birth of the software industry.
[18] FORTRAN was developed by IBM in 1956 to be used for more scientific purposes and it stood for "Formula Translator".
[19] COBOL was more business oriented and appeared in 1960. The name stood for "Common Business Oriented Language".

universal languages appeared, these software programming companies started to make more and more money from the sales of their created software programs, leading to an industry growth.

In parallel with these advances in the software industry, in 1960, another landmark in the software and hardware industries was reached with the introduction of the IBM 1401[20]. Sold at the time with a new state of the art software programming language named RPG, this was the first machine to cater to the medium size user's needs as the new programming language was easier to use. The other main advantage of the IBM 1401 was also sold at a more affordable price. More than the evolution of programming, the IBM 1401 represented the computerization of medium sized companies which, in turn, would be translated in increasing hardware and software sales as a new market was being created. As a result of the introduction and success of this machine, as well as the language that it entailed, it is estimated that by 1965, the US market for "programming services, software products, and professional services" was worth 200 million dollars in annual revenues[21].

But if these numbers, considering the time, already represented an exciting prospect of the underlying potential of the software industry, there would soon be a game changer in the market. The year 1965 is a very important year in this topic because it divides the old from the new software and hardware industries. It was in this year that the Digital Equipment Corporation, taking advantage of IBM's decision not to move into the market for personal computers[22], introduced to the world the PDP-8, the world's first minicomputer. This new addition to the hardware world and the concept of the minicomputer represented a change in the field as this computer was affordable (it was priced at 18500$), practical,

[20] By the hardware producing company IBM.

[21] In "The U.S. Software Industry: An Analysis and Interpretive History",page 14, by W.Edward Steinmueller.

[22] IBM's decision not to move into the market of personal computers has been, throughout time, perceived as a surprising decision. No one really knows why IBM, who at the time dominated the computer industry, chose to not become involved in this market although there are rumors that it did so because it was afraid that the governmental authority for competition practices would make the company divide itself into smaller companies, as had been done with telecommunications giant Bell a few years earlier.

innovative[23] and generally easier to operate than the computers that had come before.

With the minicomputer market appearing, the software industry grew in the late 1960s. It is estimated[24] that from the introduction of the first minicomputer in 1965 until 1970, the total value of the annual revenues for programming services, software products and professional services in the United States jumped from 200 million to 2.5 billion dollars[25]. There are mainly two reasons that explain this growth: for starters, the introduction of the minicomputer opened the doors of the "digital world" to more people as it was more practical and affordable than any computer that had been before. Companies could now own or rent several computers and assign different tasks to them; secondly, as it introduced its1964 model, the IBM/System 360, IBM made a controversial decision by deciding to separate its hardware and software products which, in turn, allowed independent software firms to start competing directly with IBM for software products which meant that software firms could now develop higher end programs.

ii. Development of the Software Industry: The Personal Computer as a platform for software applications

In the previous section we have seen that by the end of the 1960's, the software industry was starting to break away from the hardware industry and was turning into an industry of its own, grossing some already interesting numbers for the time. Of course, today we know that these numbers were nothing compared to the annual revenue income that most companies in the software industry would enjoy in the coming decades.

Also in the last section, we have seen that the hardware industry was developing itself under the trend that computers would become smaller and would reach more and more people but, in 1970, computers were still something that was strictly for some companies to have. Yes, computers had become a smaller and more affordable item with the appearance of the minicomputers, however, they were still too pricey complicated to reach most of

[23] The PDP-8 was the first computer to feature a keyboard marking a change in user-machine communication, which was done before via punch cards.
[24] According to the Computer and Business Equipment Manufacturers Association
[25] See Figure 2.

the population and, as a result, only medium sized companies and bigger could afford to have their own computers. In fact, most software programmers would normally have to rent computer time in timesharing centers in order to work on their programs.

In spite of this we must note that the existing software companies were indeed making money at the time, but these consisted of small companies with about a dozen employees each that wrote small software programs for different companies. This, as we know, was about to change as a revolution would take place in the following years which led to computers developing into machines that more people could afford and use.

In 1971, a Santa Clara based company came up with a revolutionary product named “Intel 8008”. The Intel 8008 was the world’s first microprocessor, allowing all of a computer’s central processing unit to be compressed into a tiny computer chip. This meant that by incorporating the chip, computer machines could now be built much smaller which would enable their developers to reduce their prices. Realizing this and in an attempt to save his struggling company, Ed Roberts, the owner of a failing company named MITS[26], had the idea to commercialize a computer kit using this new technology. The people who purchased the kit would be able to assemble it at home. The Altair, as this microcomputer was called relied heavily on the Intel 8008’s successor, the Intel 8080 which came out in 1974 and was faster than its predecessor. In spite of this, as Intel had developed the chip to be used as support for a computer and not as a computer itself, no one had programmed the necessary software to allow the communication between the computer and the user. In fact, the only way to communicate with the Intel 8080 chip in the Altair was by using the machine language of the chip which could only be done resorting to the switches on the front panel of the machine.

As this process would never be accepted by the potential users[27], Ed Roberts decided that the company needed to develop software programmed under the high-level language BASIC so that users could communicate with the computer in an easier way. The problem was that, as said before, since Intel

[26] MITS for Model Instrumentation and Telemetry Systems, was a company that previously competed in the calculator manufacturing industry, but was pushed out of the market in the early 1970 by Texas Instruments following a pricing war.

[27] The process of flipping switches on the front panel of the machine was incredibly long and painful.

never expected its chip to be used as a minicomputer, there were doubts within the company that it would be possible for Ed Roberts' MITS or anyone else for that matter to develop this BASIC interface, that is, a program that would instruct the Altair to load and accept the BASIC language. Luckily, the Intel experts were wrong as this was accomplished by two kids from Washington who would arguably become two of the greatest minds in software history. In 1975, Harvard University student Bill Gates and his childhood friend Paul Allen[28] handed MITS what would be later known as the Microsoft Basic, the much needed BASIC interface for the Altair. They achieved this, astonishingly enough, resorting to a manual on the Intel 8080 and a PDP-10[29] that Bill Gates found in the Aiken Computer Center at Harvard University[30]. It was from this partnership that one of the main players in the software industry, Microsoft[31], was born in the summer of 1975.

Having the BASIC issue fixed, Microsoft signed a royalty based contract with MITS in which MITS would sell Microsoft Basic exclusively as the only interface for their Altair computer kit and Microsoft would be paid royalties on the sales[32]. Microsoft could negotiate with other companies to supply the software to such companies but MITS could veto any of these deals if they felt it was appropriate. This deal was to hold for ten years.

Soon[33] after, the Altair computer kit was launched at the price of 397$[34] and MITS began filling out orders from all over the United States. It was a modern day miracle that every computer fan could own a computer, something that had been unachievable for most people, for the low price of 397$, however, as the Altair was difficult to assemble and as it constituted a "cult" item, meant for computer buffs, clubs started being formed not only discuss and assemble Altair

[28] Bill Gates and Paul Allen were the founders of Microsoft.
[29] DEC continued the success of the PDP-8 with more models of the same line in the years that followed. The PDP-10 was one of those more advanced models.
[30] It is said in some literature that fellow student Monte Davidoff also helped in the elaboration of the Altair BASIC by providing Gates and Allen with the a subprogram that manipulated numbers.
[31] Microsoft are still big today as they produce among other things the Operating System Microsoft Windows.
[32] Although there were some other details in this complex agreement. There was a 180,000$ cap on the royalties that Microsoft could receive and MITS had to promote Microsoft BASIC to the best of their ability.
[33] It wasn't immediately launched afterwards as some bugs needed to be corrected on the code.
[34] For comparative terms, it is important to remember that the level of computer prices was much higher at the time, the PDP-8 had been priced at 18500$.

computer kits but also to gather people together just to talk about computers in general. If before there were people forming software companies to produce software for other companies, now there were people forming clubs to assemble their own personal computers and through those produce applications for other computers and for themselves. The Altair was, in several ways, the beginning of the computer (and of the software) revolution.

The start of many of these clubs was indeed encouraged by MITS. To promote the Altair, MITS organized a travelling road show in which representatives of the company would go to different towns and show the completely assembled Altair[35] and in most of these shows, MITS employees would urge people to start computer clubs in their towns as they felt that this would get even more people interested in the Altair.

Interestingly enough for our end topic, piracy and software are so connected that it was through these clubs, which were the backbone of the software revolution as they helped spread the Altair and, hence, attracted more people to being in contact with software through attracting them to be in contact in a computer, that the first recorded software piracy case actually took place. In March of 1975, there was a meeting held in a garage in Menlo Park, California which aimed to form another one of these computer clubs based in that area. More than thirty people showed up for the meeting and established what would become arguably the most important computer club in all of them, under the name "Homebrew"[36]. Homebrew became so well-known because, in June 1975, in one of these travelling road shows that was taking place in Palo Alto, some of the members of this club stole some tape containing the Microsoft BASIC code and within a month the club started copying and distributing the software among its members and then among the members of other clubs. These people were, thus, obtaining Microsoft's software without paying MITS for it, which would not result in royalty payments for Microsoft.

The important thing to realize is that these "hackers" as they were called did not do this necessarily to harm the Microsoft corporation but because, not only was MITS making it difficult for them to buy Microsoft BASIC on its own, as it

[35] In truth it was actually a better version of the Altair, one that the users could never obtain by assembling their kits as it included hardware components that were not in the kits.

[36] Among the people who showed up was a young Steve Wozniak, who would end up building his own personal computer shortly after, the Apple I, making his mark on the computer industry forever.

was making them buy the Altair Kit which a lot of people felt did not have the hardware necessary to match the possibilities of the software, but also because these people were fans, they just wanted to use the software that was difficult to get a hold of at the time[37].

This was indeed the start of the software revolution. For the first time there was a company who produced a largely accepted, industry setting software. But this revolution faced a problem which was, as mentioned before, the fact that MITS held the power to veto any Microsoft deal in which it saw fit to do so. At the time, other model computers were being launched such as PET or the Apple II and several of the companies behind such products were approaching Microsoft to license the use of Microsoft BASIC to them. MITS vetoed most of these deals as they felt like it was helping their own competition and, as such; they were blocking Microsoft from spreading their software and from growing as company. After arbitration[38], Microsoft was declared the sole owner of the Microsoft BASIC software and was freed from their contract with MITS becoming available to negotiate with other companies. This was an important mark in the software industry because the soon to be software giant was no longer restrained.

In 1975 Microsoft made around 160,000$ in annual revenue and in 1977 it more than tripled that figure[39]. Once freed from the contract, Microsoft started supplying different versions of their software[40] for a lot of other companies that were launching their own microcomputers[41] such as Apple or Tandy. These microcomputers already came assembled being largely adopted by users and because of this, by the end of 1978; Microsoft's annual revenues were more than one million dollars.[42]

The microcomputer market had exploded given the attention that more and more people gave to the Altair kit and, by 1980, there were about 200 different microcomputer brands. The majority on these computer relied on the Intel 8080

[37] Bill Gates would later start a controversial battle with the general public by taking active action against these piracy acts. He actually drafted a letter where he called such people "thieves" and had it run on a major publication.
[38] It was stipulated in the contract between MITS and Microsoft that all disputes would have to be delt with through arbitration.
[39] See Figure 3.
[40] For every microprocessor there was a different version of the software.
[41] "Microcomputers" was the name given to these new small computers like the Altair.
[42] See Figure 3.

chip to be the microprocessor of the computer but the problem was that Intel had created this chip envisioning it to be the microprocessor of a smaller machine such as a traffic light controller[43]. With this, Intel began developing a series of successors to the 8080, which were especially designed to integrate a computer. These new processors featured a move from 8 bit to 16 bit architecture, which would amount to faster and more powerful microprocessors. To build a computer with such a microprocessor represented a new market opportunity so, in the early eighties, the hardware industry would take another leap forward with the introduction of the personal computer[44] and this leap would fuel one of the most important events in the history of the software industry.

The software architecture of a computer has three layers: there's the bottom layer, which is a software in machine language that executes simple but crucial tasks required by the microprocessor named operating system; there's a middle layer composed by languages that establish a connection point between the user and the machine, such as Microsoft Basic; and there's a top level in which are the applications developed in the given language of the computer. At the time, in 1979, there were a few industry standards in each of these layers like Digital Research's CP/M, which as a standard for operating systems.

As it was referenced before, in those days, Microsoft was in the language business and their Microsoft BASIC was an industry standard for the second layer but Microsoft realized that what made people buy their languages[45]were the applications that you could build on top of this layer. Realizing this and after seeing interesting applications in the National Computer Conference in New York in that year such as Visicalc[46] and WordStar[47], Microsoft decided to establish an application developing division.

At the time, however, Microsoft was still getting around the problem posed by the operating system. The languages Microsoft supplied were developed for a given operating system (mostly for the CP/M) so, every time a manufacturer decided to change their operating system, as did Apple in the Apple II

[43] This was what the chip was originally developed for.
[44] Or PC.
[45] Microsoft also supplied FORTRAN, COBOL and Pascal for the different microprocessors.
[46] A spreadsheet program that executed mathematical and financial operations.
[47] A word processor.

computer[48], Microsoft would have to develop a new language loading application. In retrospect, it made no sense that the first and the second layers were separated and supplied by two different suppliers. Luckily the solution fell right into Microsoft's lap.

The successor of the Intel 8080 was the Intel 8086 which was, as noted before, faster and more powerful. At that moment, IBM realized that it could enter the market it had missed out on a few years earlier if it was able to build a personal computer with this microprocessor. IBM decided to take advantage of this opportunity and it began building a microcomputer around a variation of the Intel 8086, the Intel 8088[49]. Of course, back then, the players in the software industry were still developing their software to work with the Intel 8080 and, as such, there was no software for this new IBM computer. In fact, there was no way that any of those companies could know about this computer as IBM carefully kept the project under wraps. For this "personal computer" as it was dubbed, IBM approached Microsoft for developing the machine's language software and Gary Kindall's Digital Research[50] to supply these machines with their CP/M operating system[51]. Astonishingly enough, the IBM team was pressed for time and, after a bad initial meeting[52] with Digital Research, IBM turned to Microsoft and asked them to develop a new operating system instead. This was unheard of at the time and Microsoft did not know if they could do it.

Fortunately for them, in September 1980, Tim Paterson, a young programmer from Seattle with no ties to Microsoft or knowledge about the IBM personal computer project was able to develop an operating system of his own for the Intel 8086 chip. He did this because he had the idea to develop Computer Processing Unit boards around this chip and was tired of waiting for Digital Research to come up with the software for this model. Microsoft knew that, because the Intel 8086 targeted by Paterson had many common features

[48] Apple always developed their own operating system for their computers.
[49] The difference between the two microprocessors was that the Intel 8088 was a less powerful but more recent version of the Intel 8086. IBM's engineers chose to do this as they thought they did not need such a powerful microprocessor. Still, the Intel 8088 was built with 16 bit architecture and was, therefore, much faster than the Intel 8080.
[50] Digital Research was the manufacturer of the CP/M operating system.
[51] The CP/M operating system that Digital Research created was the industry standard for operating systems at the time.
[52] It is rumored that Gary Kindall refused to make the time to meet IBM officials, sending his wife and a lawyer to do so. The latter pair refused to sign a confidentiality agreement which led to the falling out of the agreement.

with the Intel 8088 that IBM were using, Paterson's software could be adjusted. They quickly bought the rights to the software and put the programmer under contract[53]. Ultimately, Microsoft ended up supplying the IBM PC with an operating system as well as with the FORTRAN, COBOL, BASIC and Pascal languages.

Everything was set and the IBM PC was unveiled in August of 1981. Priced at 1585$, the IBM PC even came with a few applications[54]. The microcomputer that IBM had devised was a big success as it was faster and more powerful than the microcomputers it was competing against. It was quickly adopted by a large number of people as sales went through the roof and this made Microsoft's MS-DOS an industry standard for operating systems as it was the software that came with the IBM PC and, therefore, it got the most exposure.

This marked another evolutionary step in the software industry as the applications that were beginning to appear at the time and whose value was already perceived would, regardless of the language that they were built in, operate on top of Microsoft's MS-DOS operating system. Also, and perhaps even more important than realizing this step, this event marked the beginning of the growth of Microsoft that from 1983 until 1995 climbed a few steps until it ultimately became the market leader in the industry of software.

iii. The Maturing of the Software Industry: Establishment of Microsoft as the dominant software platform

Following the release of the IBM PC, the software industry completed a change that had been brewing for years. Back in its origins, the industry was a contracting industry in which a company or government entity would hire someone to design a program which would only be used by the contractor and pay a large fee for it. Then, in the sixties, following the release of the IBM System/360, it was now a corporate software industry as companies would develop software to fit a certain process and the companies that possessed a compatible technological and business structure would buy that software,

[53] He would help Microsoft develop their MS-DOS Software which was used by the personal computer.
[54] The applications that the IBM PC came with included the acclaimed spreadsheet calculator VisiCalc, a word processor called Easy Writer and the game Adventure.

paying the developers a much lower fee than in the previous case. With the release of the IBM PC, the first personal computer, the software industry became a mass market industry where companies would build programs to be used by several users, receiving the least unitary fees of the three cases but making the highest revenue in total sales.

The concept of "platform"[55] has been ever present in the hardware industry. Once Intel started manufacturing its microprocessors, all hardware and software was developed around these, making Intel the platform for the hardware industry[56]. In the software industry, between 1983 and 1995 there was an increase in the level of concentration of the market, which led to Microsoft having its software MS-DOS and Microsoft Windows emerge as the platform[57] for the industry. In fact, Microsoft's market share increased from 20% to 50% between these twelve years with the continued spread of MS-DOS and later with the establishment of the highly acclaimed Windows software[58].

This conquering of the market that Microsoft enjoyed in this time interval can be attributed to several factors. Firstly, Microsoft enjoyed first movers' advantage over most software. At the beginning of this interval, the two big players in the personal computer market were IBM and Apple. While IBM relied completely on Microsoft for the software of their IBM PC, Apple's software was produced in-house which, allied with bad pricing decisions[59] by Apple on its first personal computers, caused users to pick the first over the latter. This led to more and more applications being developed to be compatible with the MS-DOS software.

Secondly, there is the fact of one of the main characteristics of the industry being the need for high R&D investments. This means that a company that has more funds to fund research will be more likely to achieve success. This need for R&D funds came into play when the software companies all identified a key

[55] Annabelle Gawer & Michael A. Cusumano define the modern high-tech platform as "an evolving system made of interdependent pieces that can each be innovated upon" in their book "Platform Leadership: How Intel, Microsoft, and Cisco Drive Industry Innovation".
[56] In "Platform Leadership: How Intel, Microsoft, and Cisco Drive Industry Innovation" by Annabelle Gawer & Michael A. Cusumano.
[57] Following the concept of "Platform", we can understand that MS-DOS and later Microsoft Windows became the platform for the industry of software because all the other applications were being designed to fit this software.
[58] In "Hard Drive: Bill Gates and the Making of the Microsoft Empire" by James Wallace & Jim Erickson.
[59] Apple priced its first personal computers, Lisa in 1983(16,995$) and the Macintosh in 1984(2,500) too highly.

flaw in the personal computer which was the non-ability to load more than one application at the same time. Of course, some applications would provide functions beyond their core (like Lotus 1-2-3, spreadsheet software, providing a word processor) but the best solution would be to be able to incorporate the concept of "windowing", that is, to be able to run applications in windows allowing users to run more than one application at the same time. Realizing this, Microsoft and its direct competitors[60] began developing new versions of their software to be able to allow this. Oddly enough, in 1984 when most of these software programs came out, they were met with disappoint results as the personal computers' microprocessors were not fast enough for such a function, causing these software programs to be very slow. Given this failure, most of these companies who were actually one product companies had a hard time carrying on investing in this new concept of software and so only Microsoft kept trying to introduce this change as it was the only one who had the finances to do so. Turns out that "windowing" would become a very important innovation, and Microsoft reaped the benefits of its persistence.

Thirdly, another contributing factor was the understanding that Microsoft had about the software industry relying heavily on the presence of network effects[61]. This was shown by Microsoft's attempt to increase user integration by developing applications meant to complement their software. The main move by Microsoft in this front was the launch of Office which was a bundle of applications at a price that was lower than if the consumer had to buy these applications separately from more renowned application producers. This led to companies that had industry standard applications like Lotus 1-2-3[62] or Word Perfect[63] to not be able to compete and lose their markets to Microsoft's similar applications.

As a result of these factors, as said before, by 1995, Microsoft was the most dominant company in the software industry as their MS-DOS and Windows software had become adopted by more and more hardware firms. Also at this

[60] At the time, Microsoft's main competitors were Digital Research and Softech.
[61] A network effect is the phenomenon of the influencing of a product or service's value by the usage of more people. In the event of a network effect, the rule is that the value of the service will increase as more users adopt it.
[62] Lotus 1-2-3 was the standard spreadsheet calculating program of the time.
[63] Word Perfect was the dominant word processing application at the time.

time, the software market became highly concentrated, with Microsoft having a 50% market share while the six companies below it had market share percentages between 10 and 2 percent[64]. This concentration is a reflection of Microsoft's practice of beating software companies for their public and then absorbing their markets with resource to a strong network effect.

There were, however, companies that were able to keep their own crowd and survive the rise of Microsoft but, to do this, these companies had to adopt certain strategies that avoided a clash with Microsoft, keeping them at a safe distance.

One of these strategies was to develop complementary software to Microsoft's, that is, compatible software that would increase the value of Microsoft's software with its existence. One good example of this is Symantec. Founded by Peter Norton in 1982, Peter Norton Computing produced MS-DOS compatible software that recovered deleted files. By 1987, Norton Utilities, as the software was named, reached 10 Million in annual sales. Since then, this company has moved on to becoming a market leader in the production of anti-virus software. The problem with this strategy is that should Microsoft decide to develop an application with such capabilities, as it did to other compatible applications, the network effect alone will cause the market to be absorbed by Microsoft. Norton was safe as it was uneconomical for Microsoft to invest in developing anti-virus and utility software. Novell, another company that used this strategy was not so lucky. This company produced Netware, a networking system that integrated separate computers in corporate networks and lost their market when Microsoft incorporated this ability in their Windows software.

Another strategy that was followed by such companies was to enter niche markets where Microsoft had no presence. Autodesk followed this strategy by developing CAD[65] as did Corel and, most notably, Adobe Systems. Founded by two former Xerox employees, this last company achieved success by producing printing software; however, they faced a challenge when Microsoft and Apple developed their own systems at the end of the 1990s.

[64] Martin Campbell-Kelly in "From Airline Reservations to Sonic the Hedgehog: A History of The Software Industry", Chapter 8, page 259.
[65] Computer aided design.

In order to escape certain doom, Adobe had to adopt a different strategy, which was to diversify its activity, so, Adobe began making acquisitions of other software companies, migrating into desktop publishing and electronic document distribution.

We can, thus, see that only companies who adopted diversification or complementing strategies to the platform of the industry[66] were able to survive and keep their own markets. This should, in itself, illustrate not only the power of Microsoft in this time frame but also, the acceptance of Microsoft's products as the dominant platforms of the industry.

iv. Modern Software and changes in distribution of the product

Studying the evolution of the software industry and of its players, we become able to understand some important characteristics of this industry. We can see from the past that the software industry tends to rely heavily on network effects and that the commodity of software is also a very important factor. In all the cases of companies discussed so far in this thesis, the successful companies were always able to take advantage of network effects and to lock in their consumers.

With such reliance on network effects, the software industry is a platform industry, that is, one software program becomes the "platform" and is widely used causing other software companies to develop their software to be compatible with the first. We can see this in today's software for personal computers as most anti-virus software, design software, business software and entertainment software are developed to work with the platform, Microsoft Windows. The same is true for software with smart phones as so far, Apple has the network effect of their own applications such as ITunes working in favor of their "Iphone" sales.

The industry has changed a lot from 1995 up to today. This change is mostly attributed to the facilitation and evolution of communication methods, most notably, the Internet. The Internet allowed the emergence of important

[66] The platform was Microsoft Windows.

software industry concepts such as Cloud Computing and Software as a Service[67] which changed the way software is perceived, used and sold. Most software is no longer distributed through the use of material means such as floppy disks, CD-ROMS or DVD-ROMS but is rather distributed through online platforms and this trend is expected to keep growing.

To understand this new form of distribution, these concepts should be clarified. The concept of "Cloud Computing", it can mean different things to different people but the general definition is that it refers to the possibility of storing, managing and processing data through servers hosted on the internet rather than on a single computer[68]. It has been acclaimed by companies because it cuts their costs with all these operations.

Cloud computing has allowed the emergence of other concepts such as "Software as a Service", "Platform as a Service", "Hardware as a Service" and "Database as a Service". Particularly interesting for us is the concept of "Software as a Service" as it refers to a new model of software distribution that is being adopted more and more in modern times by software companies[69]. Basically, this model allows the user to remotely use an application. The way this works is, the software is hosted in a remote service provider and offered as a service to customers, which are able to access it through the internet, having also a variety of complementary services such as updates on the software or backup hosting space available for them. This model has the appeal of providing a stronger method for the protection of software against piracy, also enabling a decrease in prices because of the amount of money that companies save in manufacturing and distribution costs. At the same time, this model is meant to appeal to consumers because, besides the above mentioned price decrease, it allows customers to pay for the usage of the program, not making them buy a program to use it only a few times. Also, this model tends to increase the value of the product for users as the software is always automatically updated and bug free.

The implementation of the "Software as a Service" and "Cloud Computing" models in the distribution network and in the array services of the major

[67] Software as a Service is often shortened to "Saas".
[68] "Cloud Computing: A Practical Approach" by Anthony T. Velte, Toby J. Velte and Robert Elsenpeter.
[69] "Cloud Computing: A Practical Approach" by Anthony T. Velte, Toby J. Velte and Robert Elsenpeter.

software companies has reportedly been successful both in the sales front and in the piracy front in spite of being still in its early days. This has to do with the need for companies in the software industry to lock in their consumers, which this model fulfils as it increases the degree of proximity between manufacturer and consumer. In fact, several big companies in the software industry have implemented “Software as a Service” causing this concept to be now embedded in some of the most important areas of software such as: for office applications, Microsoft offers this model in the distribution of their “Microsoft Office” bundle of applications; Google offers it in their e-mail service “Gmail” as well as in “google apps for smartphones”; in design, Adobe offers this model in some of their creative suites; in antivirus software, market leader Symantec offers this model by protecting and storing information rather than protecting the hardware where the information is stored(as it was done until recently). Another important case that can be mentioned here is the case of Apple, whose online Apple Store has proven itself to be a success among users, hosting Iphone applications and offering them as a service to be directly used with the smartphone. Of course, this is a slightly different case as Apple does this to increase the value of its own hardware but it represents a case of success nevertheless.

The truth is that most companies are switching to this distribution model since, as it was described above, it is better for producers and consumers alike. In spite of the business advantages of such changes, some companies have integrated these concepts into their distribution networks in order to fight the recurring threat of software piracy, a parasite that has been draining this industry since the day that Homebrew members started distributing MS-BASIC among themselves and others.

3. The Software Industry: A Strategic Analysis

Having already described the software industry and the issue of piracy in detail, I will now proceed to do a strategic analysis so that the main characteristics that shape the industry's behavior can become absolutely clear. To carry out this analysis, I shall use two frameworks: a PEST analysis, which is a framework for environmental or industry analysis that goes over the Political, Economical, Social and Technological landscape of a given industry; a Five Forces of Porter Analysis which is the most widely accepted industry analysis framework.

While using each of these frameworks to analyze the industry, I will distinguish the current scenario from the past to underline the change in the industry as well as evidence the different roles and constraints of its actors.

Framework 1: Five Forces of Porter:

Bargaining Power of Buyers – Medium/High: In spite of the fact that most business and everyday procedures are done with the use of software, which means that there is no actual alternative to software, between most different software companies, buyers have a lot of bargaining power. It is so because, as it stands, there are basically no switching costs preventing a consumer from changing software. Only the existence of the strong network effects that characterize the software industry can constitute a minor switching cost but it is still a very weak barrier. Before software was mass sold, when it was personalized, there were high switching costs but not anymore.

Bargaining Power of Suppliers - Low: Given that, for most software, buyers have low switching costs then, most suppliers have very low bargaining power over their consumers. There are, of course, some software programs, mostly platforms like Microsoft Windows that have more bargaining power than others like Anti-Virus software; however, the biggest weapon that suppliers can use to keep their consumers from switching is to constitute synergies, that is, to make

their software complementary of other software in order to form a barrier to switching[70].

Threat of Substitutes – Moderate/High: We live in a digital world so there is no point considering alternatives to software. We can, however, consider the product as being legally purchased software and the substitute to be illegally purchase software. As such and for the reasons that will be described in detail in section 4, there is not a significant enough motive for software consumers to choose to purchase legally; hence, the threat of substitutes should be high. However, since the pirating of software requires the developing of a product, that is, the substitute always trails the product, there will always have to be people buying the product so this threat is somewhat diminished.

Threat of New Entrants - Medium: After going through the history of the software industry we are led to take some conclusions about it. First of all, this is an industry where barriers are mixed as if manufacturing costs are low and R&D costs can be supported, it is usually an industry composed by several "winner take all" markets which are highly dependent on swinging network effects. That said, the barriers that a new company would face when entering the industry would be the high R&D costs of developing software (for most cases), and the stern opposition of a market dominated by much bigger companies, however, since what the "make or break" factor of the industry are the "network effects" which are very difficult to block, the possibility of entering the market exists and should be considered.

Industry Rivalry - High: The rivalry in the software industry is high as it is in most industries characterized by a "winner-take-all" mentality. Different companies exist in different segments but they do so because they are able have their products become complementary to products of dominant companies. Also, the existence of the network effects that characterize the industry and normally lead to a "winner-take-all" situation increase the rivalry in the industry. As such, the software industry is a very competitive industry.

[70] Microsoft Windows is the perfect example of this. It constitutes a common platform and most software are developed to work exclusively with it, making buyers choose them over other interfaces.

Framework 2: PEST Analysis

Political Factors: In its early beginnings, the software industry was closely tied with governmental intervention as the only clients of its few companies were normally governmental authorities. Today, governmental authorities are more tied with protecting the companies in the industry then driving the industry economically as they did before. As the companies in the industry normally require a high level of protection and regulation (rights for its products, licenses to distribute, growth control), this industry is has a very strong political regulatory component, however, since the market is free, as the driving economies of the industry are free economies, political forces do not determine the fate of the industry.

Overall, although this industry is highly connected with a political component, one cannot say that political factors drive the industry.

Economical Factors: Like all industries, one would expect the software industry to be closely connected with economic trends, with its companies reaching their best numbers when the global economy was in its best period, in spite of the fact that people and companies always need software to perform their daily duties. The fact is that, software companies tend to perform worse in the years of economical crisis and this happens because, in such years, there is a ridiculous increase of company piracy levels accompanied by a decrease of monitoring of such act by the governments. Economically it makes sense that, in a year in which companies are performing worse than expected, governments are looser with enforcing regulations; otherwise they would worsen the situation of such companies as they would have to fine them constantly.

As the business software sales constitute the bigger part of the industry's sales, one can say that the software industry is closely tied with the performance of the economy.

Social Factors: With the existence of network effects that make or break most software programs in the industry, this industry is highly dependent on social factors such as trends.

Technological Factors: The software industry is, obviously by definition, a technology intensive industry. Moreover, history has shown us that technological advances in hardware normally lead towards a need for the development of new software for said devices and, hence, in an industry that is so connected to innovation, technological factors are very important. Essentially, technological advances will lead to the development of more software[71].

In conclusion, the two frameworks show us a very clear landscape of the software industry. This is a very competitive industry because of its "winner take all" mentality and fueled by the network effects that it presents, making it a rule that a company must enter the market hard and lock in its consumers to preserve its market position and ultimately triumph over the competition. The frameworks also show us that political factors are very important as they constitute a regulatory base for the industry and that technological factors are vital as technological advances in hardware normally lead to innovation in the software industry. Finally, it is also now understood that the software industry is closely tied to the economy as its company piracy rates and, consequently, its sales tend to be severely altered with the economical situation of a given country.

[71] A great example of this are the applications for smartphones.

4. Software Piracy

After having gone through the history and the characteristics of the software industry as well as understanding the reasons behind the decision to change their distribution model by the companies in it, we are now closer to the conclusion of this dissertation. The main question that this dissertation tries to answer is: "How has the change in distribution of product by the companies in the software industry impacted the issue of software piracy?", however, to be able to answer this question we must first go over the issue of software piracy.

The issue of piracy is not, by any means, a new one. It has been around for almost three decades and it has always been a negative force on the software industry and its companies not just because of the money that it costs in lost sales[72] but also because of the utter disrespect for the intellectual property behind such software programs. In spite of this not being a new problem, the evolution of the internet has brought new ways for "software pirates" from all over the world to gather and communicate, resulting in a major improvement of software piracy schemes. This, in turn, has urged software companies to, now more than ever, stand up and stop piracy.

But what exactly constitutes software piracy? Well, despite the lack of knowledge or recognition of piracy by some people, software piracy exists when someone uses or distributes a given software without the express authorization of the creators of such software and without paying them an agreed fee for such use or royalties for distribution.

According to the Business Software Alliance, a global organization that who aims to advance free and open world trade for legitimate software, currently more than 1 out of every 3 software copies installed is pirated[73]. In fact, according to their "2011 BSA Global Software Piracy Study", the rate for worldwide software piracy last year[74] was 42% which is an astonishingly high number[75]. Also, according to this report, we can break down this number into different world areas. The biggest piracy areas in the world are Central and Eastern Europe with 68%, Latin America with 65%, the Asia Pacific with 60%

[72] See Figure 4.
[73] This information was taken from "BSA Global Piracy Study 2011" available in www.bsa.org .
[74] 2011.
[75] See Figure 4.

and the Middle East and African region with the same percentage. Inversely, the regions where software piracy happens at the lowest rate are North America with 21% and Western Europe with 33%.

Having seen these numbers it becomes increasingly obvious that there is a large difference in software piracy rates between developing countries, emerging economies and developed countries[76]. The report also states this difference by showing that the average software piracy rate in emerging economies is 68% and thus, is much higher than the average rate for mature economies which is 24%. These numbers suggest that software pirates in these emerging economies resort up to almost four more times the amount of illegal software than the mature market software pirates. Still, it is also important to note that these percentages are relative to the size of the software market in each country and so, while the North America has the lowest relative software piracy rate; it also represents one of the areas where more money from software sales not concluded due to piracy is lost with over 10 Billion Dollars. Therefore, under the corporate light, the 21% piracy rate in North America is a more worrying issue than the 68% rate in Central and Eastern Europe as the North American software market represents the biggest software market in the world, currently[77] worth nearly 45 Billion Dollars[78].

In terms of individual countries the trend is the same since, because of the size of its software market, the United States of America have the lowest piracy rate worldwide but are also the country where more sales are lost to piracy with over 9 billion dollars lost sales last year. Other valuable piracy markets are China with 8 Billion Dollars lost in sales and Russia with over 3 Billion[79].

These numbers account for all piracy, both corporate and individual, so faced with such grim projections, software companies are finally being forced to recognize that these piracy acts have gone by for far too long without being met with severe action.

Perhaps, however, the most astonishing thing is not the software piracy rates itself but the way in which these pirates actually feel about Intellectual

[76] See Figure 1.
[77] By saying currently, I mean what the market is actually worth in legal sales.
[78] The number is 44,749 billion dollars and it is obtained by the sum of the legal sales of software in the US (41,664 B$) and Canada (3,085 B$).
[79] See Figure 5.

Property rights and about the need to reward people for their work. Research conducted by BSA actually suggests that of the people that declared to take part in software piracy, 71% actually feel strongly about intellectual property rights and feel that innovators should be paid for their work[80]. This constitutes a surprise outcome as, in my view, since these people are willingly taking part in a crime, the expected result would be that they felt strongly against intellectual property rights, thinking that the profits from software innovation should flow to the society instead.

One reason for this inconsistency of opinion is that most of these pirates do not actually see themselves as committing a crime when pirating software programs. This may have to do with the fact that software programs are mostly protected legally by copyright law which is not as strong of a protection as a software patent would be and, although some software patents have been awarded in recent years, the notion that a software program constitutes patentable subject matter is still pretty much up for discussion. The lack of strong legal protection for software producers may explain this difference in criteria of crimes committed, that is, it would explain why is it that if a man walks inside an electronics store and steals a television set it is universally ruled as a crime but when the same happens with a software program it is not.

To better understand the reasons behind this inconsistency of opinion as well as to answer the research question of these dissertation, that is, to be able to conclude whether the change in software distribution methods has had or will have an impact on software piracy, we must first understand the reasons for which people engage in these acts.

Why do people pirate software?

There are many reasons why people pirate software programs; however, my research has led me to a few main topics[81]:

- **Lack of legal risk:**

This is perhaps the heart of the question. When asked about why they pirated software most people admitted doing so because there were really no

[80] See Figure 6.
[81] See Figures 7 and 8.

downsides to it, in fact, they were absolutely sure that if they installed a software program in that very moment without having paid for it, nothing whatsoever would happen to them.

People do not seem to see the risks of software piracy in spite of the existence of legal matter on the topic. Software programs are covered under the World Trade Organization's scope of protection by the TRIPS[82] agreement. Signed at the end of the Uruguay Round of Negotiations in 1994, this treaty recognizes software as an array of mathematical formulas that displayed in a certain way instruct one or more hardware devices to perform a certain way and, hence, places software programs under the protection of copyright law[83]. It is, however, important to observe that copyright law is not the strongest form of intellectual property protection and it no longer seems to fit an industry that is as nearly as technological as the hardware industry, in which products are protected by patents. For this reason, more and more software companies began applying for software patents although, the lack of consensus as to whether software constitutes patentable subject matter is slowing down the grant of such protection for software programs.

Regarding software, people often break its copyright law protection. This happens because, with such a low degree of intellectual property protection, the software companies do not have the time or the resources to pursue legal action against every single person that pirates software and because the governmental organizations of most countries are, while continuously acknowledging the need to combat software piracy and to uphold anti-piracy laws, turning a blind eye to the situation.

It must be, however, noted that the United States of America have recently been debating the possibility of having tighter laws to combat not just software piracy but every type of online piracy. This country and some other like the United Kingdom have taken action against websites that feature illegal content in an attempt to strike a killing blow in the way software pirates organize themselves but, although it disabled some of the structures that people used to download illegal content, others were able to escape this action by switching

[82] Meaning the agreement on trade related aspects of intellectual property rights.
[83] Copyright law protects the expression of ideas, rather than the ideas themselves.

their domain to another country, falling outside jurisdiction and minimizing the effect of such actions.

These spontaneous actions are of course insufficient as they are uncoordinated, allowing pirates to find new ways to engage in these acts. Also, the lack of success of said operations in minimizing piracy leads us to believe that a solution for this problem should come from the software companies on the technological front rather than from the worldwide governments on the legal front because, although the existence of legislation is required and useful to protect intellectual property from theft, it is not that useful at protecting companies from the illegal use of their programs by individual users.

- **Price:**

Another much cited reason[84] for pirating software was the price. Software prices have decreased since the beginning of the software industry, when selling software was making a special software program for one company and getting handsomely paid for it down to today's mass scale software selling. Software prices are, as a matter of fact, at an all time low, but still, there is a growing belief in people's minds that software is still too expensive and that it has not kept up with the advancement of times.

A study of software and hardware prices of the past can offer a shed of light on the matter. The reason why people see the decrease in software prices as insufficient has to do with the fact that, since the price of hardware has decreased much more, people tend to think that software prices should be much lower by now, that is, while in absolute terms the difference of prices is lower, in relative terms it has actually increased.

With this, software companies are being perceived as making too much money given the product that they are selling. If we compare the cost sheets of a hardware and software company we can understand that, although the improvements in manufacturing and distribution processes led to a decrease in cost for the hardware industry, this mattered little for the software industry as these companies never had high manufacturing costs but rather always had huge research and development costs which are still there today. In fact, if

[84] The most cited reason with 34% of the answers in fact.

anything, with all the new hardware devices that are emerging these days such as PDA's and Smartphones, the research and development bill of most software companies has actually increased as they now have to adapt their software to these new platforms.

Also, we have seen in our strategic analysis[85] of the industry that software sales are tightly connected with economic performance so, it is no wonder that in the middle of the financial crises that the world faces, most users and companies cite the price of software as a reason for using pirated software.

Given all this, it is understandable that, when presented with an alternative that will provide them a similar level of quality at no cost to them, people choose to pirate rather than to buy the software.

- **Evolution of piracy schemes:**

One more vital reason why people choose to be involved in piracy acts and why software piracy has risen to a new high was the evolution of piracy schemes. If we recall the Homebrew episode we can understand that the software pirates of today is very different from the software pirates of that time because their motives for getting involved in such illegal acts differ. While the software pirate of the late 70s was so because it was the only way that he could satisfy his need for more knowledge, as the quality of distribution of software was poor like it was shown with the Altair, today's software pirate does it on the basis of convenience, because he can. This difference in motivation has, of course, to do with a change in conditions that software pirates have to spread and practice piracy acts.

In the 1970s, when members of the Homebrew computer club stole bits of tape containing BASIC code for Altair and began to distribute it among themselves and then other similar clubs[86], practicing software piracy was a very tough activity. It was so because, if though copying the software code was not exactly challenging (although still required a bigger effort than to do it today), distributing the software constituted an arduous task as there was no communication method that could keep software pirates in contact and that

[85] See Part 3.
[86] In "Hard Drive: Bill Gates and the Making of the Microsoft Empire" by James Wallace and Jim Erickson.

would allow them to receive the pirated software in a reasonable time frame. For this, even though there were software pirates, it was much more comfortable and easy at the time to just buy the software if you could afford it.

What exists today is a rather different situation than the one mentioned above hand. With the evolution of and wide spread of the World Wide Web, the problem that hackers had of not being able to distribute the illegal software in a fast and efficient way vanished as they now can just place a pirated software online and have it downloaded by any amount of people all over the world. Also, the internet has also provided these pirates with the ability to communicate with other pirates in real time all around the world which led to emergence of these "pirate platforms".

The Internet has allowed hackers to get organized, which in turn has led to better conditions for pirating through advancements on pirating schemes. As such, a person can now access one of these pirate websites, search for the software they want and just download it from the comfort of their homes. Seen as though until the adoption of the Software as a Service model most software was sold physically in packages containing CD-ROMS, it became more convenient for people to just download the software illegally free of charge than to have the trouble of going to the store and do it legally. This allied to the lack of a proper reaction by the software companies and to the lack of ability/willingness of local governments to control the problem, less to the massification of such communities which, in turn, led to the increase in piracy levels.

Besides this, it is also important to note that the fact that these pirates were able to build organized and widely known structures to spread their illegal material while taking practically no consequences for their actions made these structures more accepted by the general public. Suddenly, everyone was downloading pirated software, it became almost as if only non-intelligent people bought software, the others had a better way of acquiring it since it had basically no cost[87] and was delivered in a more convenient way without having to move from their chairs. One can understand that this kind of networking effect that made piracy "not as bad" in the eyes of most people could have also

[87] If you do not take into account the cost of the internet.

played a pivotal role in adoption of such techniques by a large amount of people.

- **Intellectual Property Protection Opposition:**

The conducted study also revealed that, although a very minor one, another cause for the adoption of software piracy by a group of people has to do with opposing ideas to the concept of intellectual property protection.

Despite the notion that an innovator should get rewarded for his work being an universally accepted topic, which was of course vital for the development of any kind of intellectual property laws, there is also an idea shared by a growing minority which argues that the benefits of the innovation should flow to society rather than to specific companies. Although it would be expected that these views would always clash, the most accepted idea among this minority is that innovators should indeed be rewarded for their innovative thinking, but as things stand they are reaping to big of a reward and neglecting the part that should be mostly flowing to society.

This is not, by any means, a new view. After the “Homebrew” piracy episode in 1975 Bill Gates, one of the owners of the breached software, sent a letter to be printed in magazines calling software pirates “thieves” for stealing his software. The response that came from software pirates was that Bill Gates and Paul Allen themselves had developed their BASIC software for the Altair on a PDP computer belonging to Harvard University, which they used without this entity’s knowledge. Moreover, history has shown that innovation in the software industry is a cumulative phenomenon, that is, new software usually emerges from old software. With this in mind, the people that believe in this do not really feel as though it is “fair” for them to pay so much for software, leaving the question of whether the decreasing of the price of software would reduce piracy, taking us back to another previously discussed point.

5. Answering the Question

Now that we have gathered some knowledge about the forces that govern the software industry as well as approached the issue of piracy and its reasons, we can now make a final conclusion about the impact that the changes in product distribution had on software piracy.

Essentially, there are two ways in which we can realistically measure this impact. The first of them is by looking at the variation of piracy numbers in recent years. By doing so, the idea is to understand whether the time of adoption of software as a service coincides with a significant decrease of piracy rates in its key markets.

With this, if we look at the software piracy studies provided by the Business Software Alliance to get a landscape view of the variation of piracy rates between 2003 and 2011, we can actually see that although the piracy numbers have actually increased in this interval, they have become relatively stable in the past 2 years at 42% global piracy rate. Since the sale of software through clouds started to be applied in large scale in 2010, is it actually possible to state that this stabilizing of the rate reflects the effective action of this measure against software piracy?

To take such a conclusion would be more of a leap of faith than a faithful reproduction of the facts as we would be making some false assumptions. For starters, knowing that the percentage of sales done via cloud is currently of about 8%, it would be premature to attribute any changes in world piracy to this change in distribution method. Secondly, we cannot assume that in a period of time that is marked by the tightening of piracy laws and a swift change of the worlds' biggest piracy zones, the stabilizing of piracy rates can be exclusively attributed to this change in distribution as there are too many elements in play. The fact that we have witnessed that the decrease of software piracy the zones where it was historically lowest, North America and Western Europe, has been accompanied by an increase in software piracy in the Latin America, Central Eastern Europe and Asia Pacific zones illustrates this same uncertainty. It is not exactly possible at this point to determine whether the so-called developed countries are being able to bring their piracy rates down because it is where the sale of software through cloud computing has been implemented the most

aggressively or because the anti-piracy legal framework of these countries is tighter and better enforced than in developing economies.

Perhaps the best example of the uncertainty of the data at this point would be the case of Japan. In spite of being in the Asia Pacific zone, which constitutes the most valuable market for pirated software with a reported 20 million dollars in lost sales per year, Japan is currently the third lowest piracy rated country with 21% of pirated software. The astonishing fact that a country can perform so differently from the region it is inserted in can either be explained by the fact that this country has a tighter legal anti piracy framework around it than the other countries in the zone or that its higher technological conditions allowed a better implementation of the beforehand distribution methods.

The conclusion that we can actually take from looking at the historical data on piracy is that users in emerging economies pirate more than users in developed countries and that this is true for both individual users and companies. This, of course, aids nothing to solving our previous debacle as over the possibility of being explained by either of the two conditions mentioned before, this difference may also be explained by the fact that software piracy, as we have seen before, is closely tied to the economical side meaning that companies in less resourceful countries would be more prone to pirate software in order to mitigate their own cost structure.

Considering all this, it becomes relatively clear that using historical data to make conclusions about the effect of the change in distribution of software on software piracy would be biased, reckless and borderline wrong.

The other way in which we can make conclusions about the impact that this change has had or may have in the future on reducing software piracy is by exploiting the strategic side of the matter. This would be done by understanding how this change responds to the main reasons for software piracy seen before while considering what we have learned about the functioning of the industry. It, hence, becomes more a question of “will the change help reduce software piracy” rather than “has the change helped reduce software piracy” which would make more sense, given the already mentioned fact that the penetration of this new distribution level is still in a very early stage.

If we look back to the reasons why consumers choose to be involved in software piracy acts described before, we can see that there are 4 main reasons which are lack of legal risk, price, opposition to intellectual property and evolution of piracy schemes.

Regarding the first and third reasons, the change in distribution methods constitutes a step up in the defense against software piracy by companies. Often companies have relied on governments to pass and enforce laws that would prevent software pirates from engaging in such practices, but it has become clear with the widespread of the internet during the last 12 years that these governments are unable to do so, whether because justice is not the same everywhere leading to different law enforcing standards in different countries (anti piracy law does not mean the same in Bangladesh that it does in the US) or simply because it is an action from a third party rather than from the company itself, hence, it is normally pursued with less haste. Given this, the change in distribution represents an internal action against software piracy by the companies in the industry. By providing software as a service and not through material goods, the companies of this industry are now changing the topology of the field because, as in this model the company simply provides the software through a limited time period and through the internet, it retains control of the assets (the software itself), stopping pirates from distributing counterfeited products as they no longer even hold a single copy of the product themselves. What happened until now was that a hacker would obtain a legal physical copy of a software program and from that copy he would reproduce other pirated copies however, not being able to obtain a physical copy will stop the whole process dead leading to a decrease in piracy.

This course of action is very similar to what is happening in the music industry. The uncontrolled growth of software piracy that has occurred in the last 10 years is identical to what the one verified in the music industry over the last 15. Like in our case, Windows Media Player provided users with the possibility of extracting the files from music CDs, allowing people to share them online if needed. It was obviously a matter of time until P2P networks were organized and before anyone could see it, album sales were dropping in all zones and genres as people were practicing large-scale musical piracy. This was obviously a problem for many years with record labels trying to pursue

legal action against software pirates and coming up to the same previously cited governmental inefficiencies until the actors of the musical industry realized that the problem was that people were pirating software because they could do so. Upon realizing this, it was a small step until mass music playing platforms such as Spotify rose up and started effectively contributing to the reduction of musical piracy. Now, with such platforms, what is happening in the musical industry is that song files are no longer provided but rather the service of reproducing the selected songs is. So now, users sign up and choose any song they want to listen to and the platform will play it for them, allowing them to save storage space, to have a more convenient experience and, most importantly, as they have to sign up to such platforms, they are no longer engaging in piracy acts. Moreover, the convenience of the whole process to the user allows a network effect to take place which in turn will work against piracy.

The main notion we can take about these two reasons for software piracy (the opposition to intellectual property and the lack of legal risk) is that software pirates engage in such activities mainly because they can, it is more convenient for them to do so and it presents no risk. As such, the only way to effectively reduce or stop software piracy is to, of course, to take control of the assets and provide software as a service and not as a product. While a product can be counterfeited and reproduced as it has been happening, by making the software a tool and not the end product software companies will be stopping this reproduction and, therefore, reducing piracy rates.

Most of the interviewed subjects cited the high price of software as a reason for committing software piracy and so, any solution that software companies may try to implement against software piracy will naturally have to deal with this problem. When going in detail about this point, we went over the fact that software prices are at an all time low but what we must understand is that the notion of high price does not necessarily mean that per say, but rather that the value of the product is inferior to the asking price. This happens because users are not provided with cheaper alternative, pirated software, which not only costs them virtually nothing but is also more conveniently delivered to them in their own homes. Let’s face it, no one is going to, in the long run, go to a store or order online and pay a high price for software that most people are getting home for free. The solution of selling software as a service, however, changes

that. By stripping pirates of the tools to pirate software, software companies are increasing the value of their products because they are taking away the free solution that users had while matching the convenience level. But there is more to the price-value change that this solution brings to the software industry as, providing software as a service and not as a product represents a nearing between company and consumer. We have seen before that the main costs that the companies in the software industry face are research and development fees and with this new model, companies are brought closer to their consumers being now able to extract direct data from them, which in turn should lead to a decrease in R&D costs. In this, once again, the software industry is like the music industry since both industries have high R&D costs. With the approximation that the internet brought between the labels and the users, music industry companies were able to reduce their R&D costs as they could see if an album with be a hit by the online reaction, not having to produce 5 albums to obtain 1 hit album(as they previously did).

This change in value should change the perception on the price, however, in true price terms, by offering software as a service; companies will be able to save a lot in manufacturing costs. Granted that, as we know, manufacturing costs are not high in this industry but the absence of such should decrease the price of the final product. Since the software industry, as seen before, is incredibly connected to the economic situation of the surrounding environment, fact that can be illustrated by the higher piracy rates on emerging countries versus the low ones on developed ones, a reduction in price would theoretically result in more sales.

The other key aspect that could potentially make this solution a success, which we have briefly addressed before, is the convenience level that this solution provides the customer with. The fact that the software companies will become, with this solution, closer to the consumers, being able to offer different packages to different consumers and to deliver the service (product) in a fast and convenient way leads us to think that it will be widely accepted by the consumers once it finally penetrates the market. Furthermore, one most understand that the aspect of convenience is a very important one because the software industry is characterized, as seen before, by networking effects and the convenience that a software is able to provide the consumer with is normally

what triggers said effect as we have seen countless times throughout this dissertation.

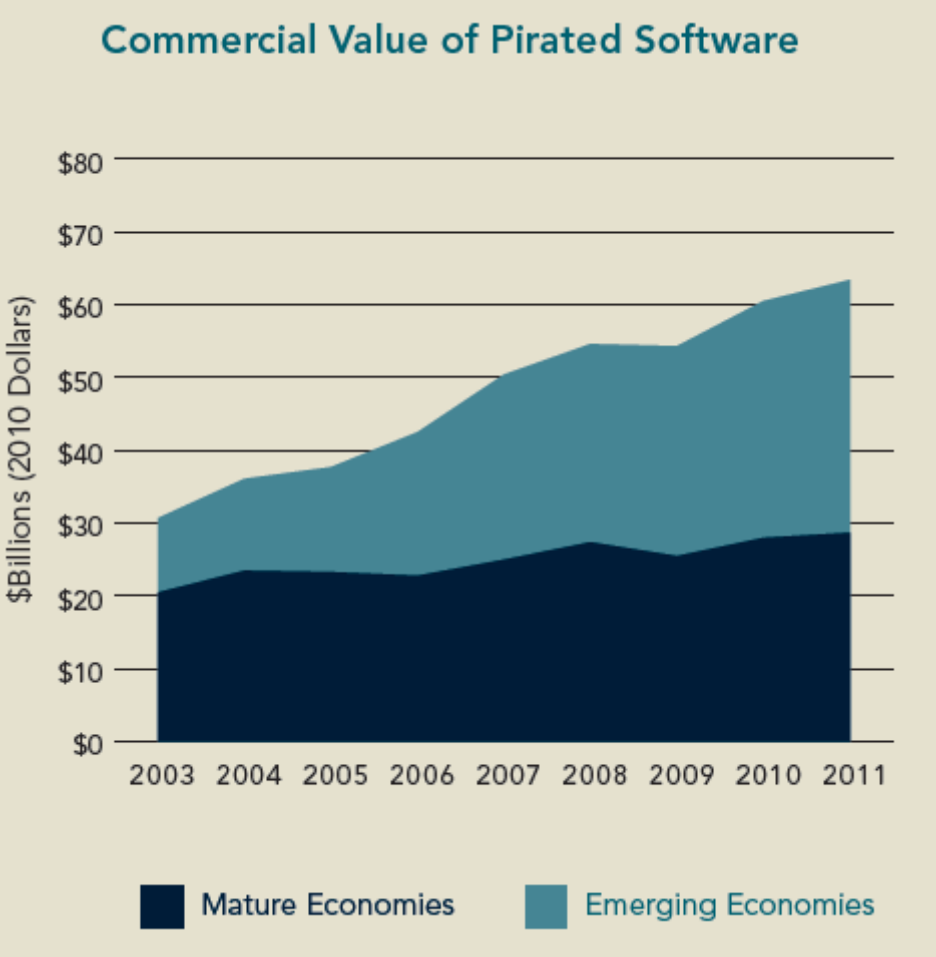
This last point is tied with the evolution of piracy schemes, which was one of the reasons presented by consumers for engaging in piracy acts. As attested to before, piracy platforms soared because of the network effect that they were able to create. It would be wrong to pirate software but once everyone was doing it, it seemed more like a sign of intelligence then theft. With this, what this solution is doing is disabling hackers' ability to pirate software and substituting the void left by the absence of such material with software as a service, pushing the network effect to swing against piracy rather than for it. It has been historically known that a motivated hacker can find a way to pirate almost everything, but even if such crime takes place with software as a service (although it would be much more difficult then with software as a product) the increase in value of software resultant of this change is, in itself, probably going to discourage people to use such pirate material. In the last few years, the consumers have complained about different reasons which sum up in an idea that the convenience level and the value of the product are simply not enough to discourage people from using pirated software however, by providing it as a service, software companies are able to increase both these points which should, in turn, lead to a decrease in piracy.

In conclusion, answering the question that names this dissertation is a complex task. If asked if the change in software distribution to a software as a service model has impacted software piracy in the last few years I would, looking at the low penetration levels of the model and at the piracy rates, be forced to say that no it has not, not in a significant way at least however, if asked whether I think that in a not so distant future, this change will impact software piracy I would, given the increase of value, convenience level and the difficulty to pirate software as a service, since the companies will now hold the product at all times, have to say that my understanding is that this change is bound to reduce software piracy and it constitutes the fierce reaction that this industry has been craving for at least the past 12 years. Of course one must, as a final note, understand that a skeptic person might with all fairness argue that if this solution represents such an increase of value of the product to the consumer, then why are the number of sales of software as a service so low at

this point? Well, the thing is that this is a revolutionary way of selling something that was sold in a different way, traditionally as a product, for a very long time, more than half of the industry's lifetime and, thus, to implement such a change some time is required. Software as a service is now, as we speak, waiting for the consumers and even the manufacturers to realize its benefits. Once that happens, more and more software will be sold as a service and, consequently, less and less as a product which will, ultimately, trigger the network effect that will on one hand bring the users to adopt this new model and on the other hand corner software piracy by stripping it of its tools of operation.

Appendix

Figure 1: The increase in commercial value of pirated software between 2003 and 2011:



Source: BSA Global Piracy Study 2011 (2011).

Figure 2: The evolution of revenues in the software industry:

Year	Processing Services	Software Products	Professional Services	Total
1965	n/a	n/a	n/a	200
1970	1,200	500	800	2,500
1975	3,300	1,000	2,200	6,500
1980	10,800	2,850	4,350	18,000
1981	11,550	3,950	5,500	21,000
1982	12,650	4,900	5,950	23,500
1983	14,400	6,900	6,900	28,200
1984	17,150	10,000	8,100	35,250
1985	19,310	12,120	9,270	40,700
1986	20,750	14,150	10,100	45,000
1987	23,600	18,500	11,750	53,850
1988	26,900	27,850	13,300	68,050

Source: Juliussen and Juliussen (1990).

Figure 3 – Microsoft financial statistics:

Year	Revenues($)	Annual Growth
1975	160,000	-
1976	22,000	38%
1977	382,000	1636%
1978	1,356,000	255%
1979	2,390,000	76%
1980	8,000,000	235%
1981	16,000,000	100%
1982	24,000,000	53%
1983	50,000,000	104%
1984	97,000,000	95%
1985	140,000,000	44%
1986	198,000,000	41%
1987	346,000,000	75%
1988	591,000,000	71%
1989	804,000,000	36%
1990	1,186,000,000	48%
1991	1,847,000,000	56%
1992	2,777,000,000	50%
1993	3,786,000,000	36%
1994	4,714,000,000	25%
1995	6,075,000,000	29%

Source: "From Airline Reservations to Sonic the Hedgehog: A History of the Software Industry", Chapter 8, page 233 – Martin Campbell-Kelly.

Figure 4 – World Piracy Statistics:

	2003	***2005***	***2007***	***2010***	***2011***	***Current Commercial value of pirated software(M$)***
Global Piracy Rating	*36%*	*35%*	*38%*	*42%*	*42%*	*63456*
North America	*23%*	*22%*	*21%*	*21%*	*21%*	*10958*
Western Europe	*36%*	*35%*	*33%*	*33%*	*33%*	*13449*
Asia Pacific	*53%*	*54%*	*59%*	*60%*	*60%*	*20998*
Middle East/Africa	*56%*	*57%*	*60%*	*58%*	*60%*	*4159*
Latin America	*63%*	*68%*	*65%*	*64%*	*65%*	*7459*
Central Eastern Europe	*71%*	*69%*	*68%*	*64%*	*68%*	*6133*
1st Most Pirate	*China*	*Vietnam*	*Armenia*	*Georgia*	*Zimbabwe*	
%P1	*92%*	*90%*	*93%*	*93%*	*92%*	
2nd Most Pirate	*Vietnam*	*Zimbabwe*	*Bangladesh*	*Zimbabwe*	*Georgia*	
%P2	*92%*	*90%*	*92%*	*91%*	*91%*	
3rd Most Pirate	*Ukraine*	*Indonesia*	*Azerbaijan*	*Bangladesh*	*Moldova*	
%P3	*91%*	*87%*	*92%*	*90%*	*90%*	
1st Least Pirate	*United States*	*United States*	*United States*	*United States*	*United States*	
%p1	*22%*	*21%*	*20%*	*20%*	*19%*	
2nd Least Pirate	*New Zealand*	*New Zealand*	*Luxembourg*	*Japan*	*Luxembourg*	
%p2	*23%*	*23%*	*21%*	*20%*	*20%*	
3rd Least Pirate	*Denmark*	*Austria*	*New Zealand*	*Luxembourg*	*Japan*	
%p3	*26%*	*26%*	*22%*	*20%*	*21%*	

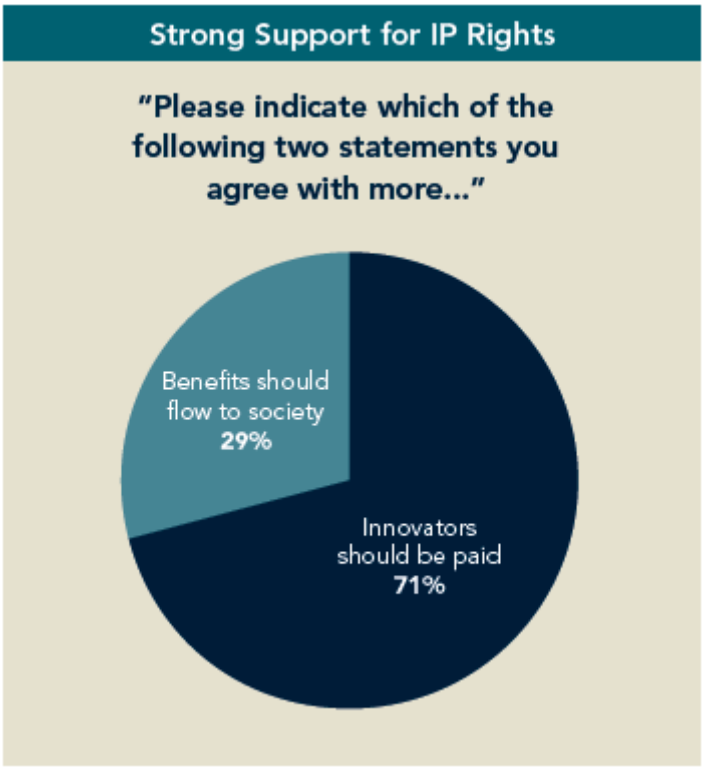
Source: Business Software Alliance.

Figure 5 – Top 20 Economies in Commercial Value of Pirated PC Software, 2011:

Country	Pirated Value(M$)	Legal Sales(M$)	Market Potential(M$)	Piracy Rate
US	9773	41664	51437	19%
China	8902	2659	11561	77%
Russia	3227	1895	5122	63%
India	2930	1721	4651	63%
Brazil	2848	2526	5374	53%
France	2754	4689	7443	37%
Germany	2265	6447	8712	26%
Italy	1945	2107	4052	48%
UK	1943	5530	7473	26%
Japan	1875	7054	8929	21%
Indonesia	1467	239	1706	86%
Mexico	1249	942	2191	57%
Spain	1216	1548	2764	44%
Canada	1141	3085	4226	27%
Thailand	852	331	1183	72%
South Korea	815	1223	2038	40%
Australia	763	2554	3317	23%
Venezuela	668	91	759	88%
Malaysia	657	538	1195	55%
Argentina	657	295	952	69%

Source: BSA Global Piracy Study 2011 (2011).

Figure 6 – How do users feel about IP Rights:



Source: BSA Global Piracy Study 2011 (2011).

Figure 7 – Questionnaire to expose the reasons why people find themselves engaged in piracy acts:

How old are you?

☐ 10-18

☐ 19-35

☐ 16-50

☐ 50-65

In which country were you born?

Specify:_____

Have you ever been involved in software piracy acts (used paid software without paying for it)?

☐ Yes

☐ No

If engaged in such acts, tick the box (es) of the software program(s) in which these piracy acts were committed?

☐ Utility Software(Norton, Kaspersky, McAfee)

☐ Operating System Software(Windows, MAC OS)

☐ Design Software(CAD, Adobe)

☐ Entertainment Software(Video Games)

☐ Business Software(PHC)

☐ Software developing software

☐ Website developing software

☐ Other – Which?_____

If engaged in such acts, what was (were) the reason(s) that led you to take part in them?

☐ Price

☐ Convenience

☐ Quality

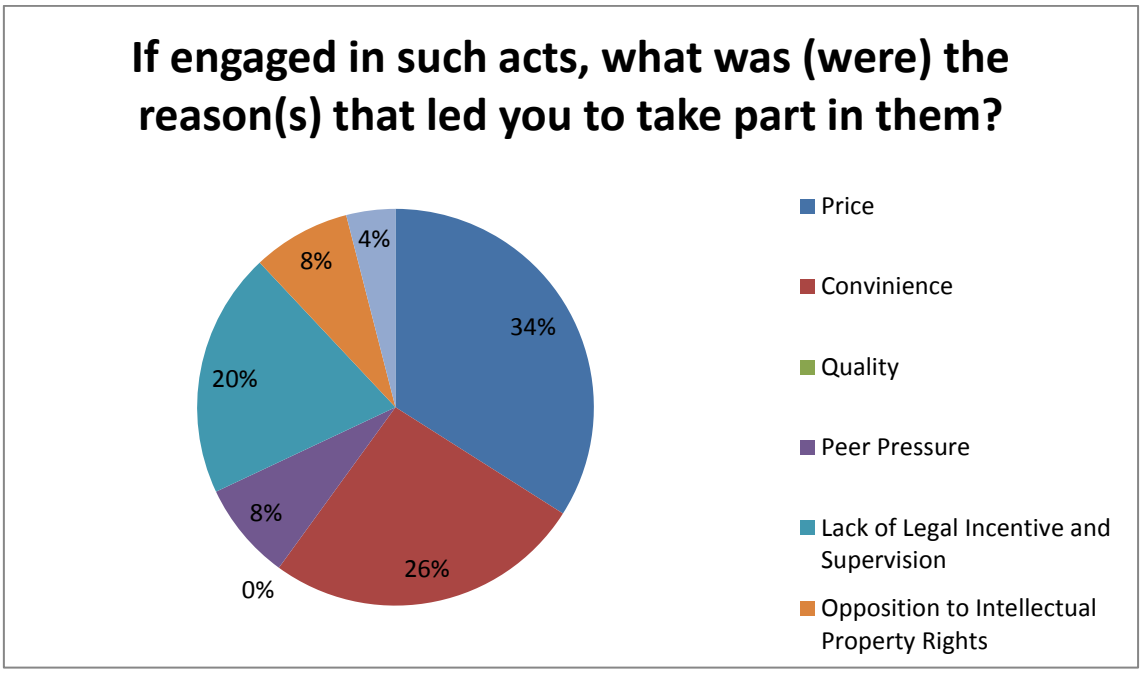
☐ Peer Pressure

☐ Lack of legal incentive and supervision

☐ Opposition to Intellectual Property Rights

☐ Because I don't see the harm, it's a victimless crime.

Figure 8 – Questionnaire results on what led people to take part in piracy acts?



Bibliography

Books:

Velte, A Velte, T & Elsenpeter, R 2010, *Cloud Computing: A Practical Approach*, McGraw-Hill.

Campbell-Kelly, M 2003, *From Airline Reservations to Sonic the Hedgehog: A History of the Software Industry*, The MIT Press, Cambridge, Massachusetts.

Wallace, J & Erickson, J 1992, *Hard Drive: Bill Gates and the Making of the Microsoft Empire*, John Wiley & Sons,Inc, New York.

Gawer, A & A. Cusumano, M 2002, *Platform Leadership: How Intel, Microsoft, and Cisco Drive Industry Innovation*, Harvard Business School Press, Boston, Massachusetts.

Chesbrough, H Vanhaverbeke, W & West, J 2006, *Open Innovation: Researching a New Paradigm*, Oxford University Press, Oxford. Chapters 1 & 5.

Journal Articles:

A. Cusumano, M 2010, 'Technology Strategy and Management: The Evolution of Platform Thinking', *Communications of the Acm*, January 2010.

MacCormack, A Verganti, R & Iansiti, M 2001, 'Developing Products on “Internet Time”: The Anatomy of a Flexible Development Process', *Management Science*, January 2001.

Mann, R 2005, 'Do Patents Facilitate Financing in the Software Industry', *Texas Law Review*, March 2005.

Teece, D Pisano, G & Shuen, A 1997, 'Dynamic Capabilities and Strategic Management', *Strategic Management Journal*, August 1997.

Majumdar, S & Vora, D 2010, 'Industry Structure Characteristics and International Entrepreneurship in India’s Software Industry', *Jornal of Entrepreneurship*, September 2010.

Cockburn, I & MacGarvie, M 2009, 'Patents, Thickets and the Financing of Early-Stage Firms: Evidence from the Software Industry', *Journal of Economics & Management Strategy*, Fall 2009.

Working Papers:

Campbell-Kelly, M . *The History of the History of Software*. Working Paper.*Warwick University, UK*.

Steinmueller, W (1995). The US Software Industry: An Analysis and Interpretative History. Working Paper. University of Limburg, The Netherlands.

Bontis, Dr. N & Chung, H (1995). The Evolution of Software Pricing: A Framework for Box Licenses to Application Service Provider Models. Working Paper. DeGroote School of Business, McMaster University.

Etro, F (2005). Innovation and Patents for the Software Industry: An Empirical Analysis of the EU and US Cases. Working Paper. University of Milan, Italy.

Stuart, T . Interorganizational alliances and the performance of firms: A study of growth and innovation rates in a high-technology industry. Working Paper. University of Chicago, United States.

Rentocchini, F (2008). Sources and characteristics of software patents in the European Union: some empirical considerations. Working Paper.

Iansiti, M (2009). Principles that Matter: Sustaining Software Innovation from the Client to the Web. Working Paper. Harvard Business School, United States.

Websites:

Smith, M 2012, 'The evolution of software distribution', *Channel Pro*, Accessed 20th July 2012, Source: from http://www.channelpro.co.uk/opinion/6735/evolution-software-distribution

'Software Industry Facts and Figures', *Business Software Alliance,* 2008, Accessed: 30th August 2012, Source: from http://www.bsa.org/country/public%20policy/~/media/files/policy/security/general/sw_factsfigures.ashx

Oak, M 'Major Types of Software', *Buzzle*, Accessed 24th August 2012, Source: from http://www.buzzle.com/articles/major-types-of-software.html

Prestipino, P 2011, 'Profit from Shifts in Software Distribution', *Website Magazine*, Accessed 15th August 2012, Source: from http://www.websitemagazine.com/content/blogs/posts/archive/2011/11/10/profit-from-shifts-in-software-distribution.aspx

Rouse, M 2010, 'Software as a Service (Saas)', *Search Cloud Computing*, Accessed 15th August 2012, Source: from http://searchcloudcomputing.techtarget.com/definition/Software-as-a-Service

Piracy Studies:

Business Software Alliance & IDC (2004). First Annual BSA And IDC Global Software Piracy Study.

Business Software Alliance & IDC (2005). Second Annual BSA And IDC Global Software Piracy Study.

Business Software Alliance & IDC (2006). Third Annual BSA And IDC Global Software Piracy Study.

Business Software Alliance & IDC (2007). Fourth Annual BSA And IDC Global Software Piracy Study.

Business Software Alliance & IDC (2008). Fifth Annual BSA And IDC Global Software Piracy Study.

Business Software Alliance & IDC (2009). Sixth Annual BSA And IDC Global Software Piracy Study.

Business Software Alliance (2010). Seventh Annual BSA Global Software Piracy Study.

Business Software Alliance (2011). Eight Annual BSA Global Software Piracy Study.

Business Software Alliance (2012). Shadow Market 2011 BSA Global Software Piracy Study Ninth Edition.