

## Chapter 1

# History of Open Source

### 1.1 Introduction

Open source software (OSS) is not a new concept. Although the term open source was coined just a few years ago, the concepts behind this particular way of developing software have existed since a long time. One of the pioneers in this field, Richard Stallman, remembers how software was developed several decades ago with these words:

When I started working at the MIT Artificial Intelligence Lab in 1971, I became part of a software-sharing community that had existed for many years. Sharing of software was not limited to our particular community; it is as old as computers, just as sharing of recipes is as old as cooking. But we did it more than most.

This was not only the case at universities. In the 1960s, IBM and others sold their first large-scale commercial computers with free software. “Free” meant that the source code was free and available and, therefore, the software could be improved and modified. In other words, hardware was the product being sold, not software. By the mid-1970s this began to change. Software became a commercial proprietary product that could not be redistributed or modified. As a result of this change software could no longer be freely shared by programmers as it had been in previous times. Stallman was not happy with this evolution and wanted to take software development back to its former status by creating a new

software sharing community. For this reason he left MIT. Several years later, he explained his choice to leave his post at MIT:

In January 1984 I quit my job at MIT and began writing GNU software. Leaving MIT was necessary so that MIT would not be able to interfere with distributing GNU as free software. If I had remained on the staff, MIT could have claimed to own the work, and could have imposed their own distribution terms, or even turned the work into a proprietary software package. I had no intention of doing a large amount of work only to see it become useless for its intended purpose: creating a new software-sharing community.<sup>1</sup>

Stallman would probably have just been a visionary had others not latched on to his ideas obtaining unexpected results. This was the case, for example, of Linus Torvalds and his creation Linux. Torvalds himself was surprised by the success of Linux:

Linux today has millions of users, thousands of developers, and a growing market. It is used in embedded systems; it is used to control robotic devices; it has flown on the space shuttle. I'd like to say that I knew this would happen, that it's all part of the plan for world domination. But honestly this has all taken me a bit by surprise. I was much more aware of the transition from one Linux user to one hundred Linux users than the transition from one hundred to one million users.<sup>2</sup>

What is even more noteworthy is that software developers and hackers were not the only ones interested in this new phenomenon. So-called *free software* also caught the attention of companies and institutions. The most surprising reaction was the fact that Microsoft saw free software, or what came to be defined as open source software, as a threat to its own business. In 1998, Microsoft carried out a careful study of open source software that was reported in a series of originally confidential documents that the open source community baptized the "Halloween Documents". In these documents Microsoft concluded the following:

In recent years, corresponding to the growth of Internet, OSS projects have acquired the depth & complexity traditionally associated with commercial projects such as Operating Systems and mission critical servers. Consequently, OSS poses a direct, short-term revenue and platform threat to Microsoft - particularly in the server space. Additionally, the intrinsic parallelism and free idea exchange in OSS has benefits that are not replicable with our current licensing model and therefore present a long term developer mindshare threat.

The Linux OS is the highest visibility product of the Open Source Software process. Linux represents a best-of-breed UNIX, that is trusted in mission critical applications, and - due to its open source code - has a long term credibility which exceeds many other competitive OS's. Linux poses a significant near-term revenue threat to Windows NT Server in the commodity file, print and network services businesses.<sup>3</sup>

These documents made it clear that by 1998 the open source community had become more than just a large community of hackers developing software, but rather was beginning to pose a threat to one of the most important software company in the world.

Our journey into the world of open source software will begin with the history of the open source phenomenon. In this chapter we will take a look at the main historical events that mark steps in the development of the community and its products. The history of OSS can be divided into four main stages: the age of the pioneers, the beginnings of the open source movement, the diffusion stage and the institutionalization stage. Specific events define and characterize each stage and show how the attention given to this type of software and the importance it took on grew in each stage.

As I write this book, on the one hand OSS is becoming more and more diffused and on the other hand it is under the attack of lawsuits. Companies that use or support OSS are being brought to courts by those who want to defend the ways of developing and distributing proprietary software. The aim of these court cases is to reduce the momentum OSS has achieved in recent years.

## **1.2 The Age of Pioneers**

In the sixties and seventies the knowledge and tools regarding information technologies and software development belonged to a small community of researchers. Software was developed at universities in close contact with the research laboratories of a few large companies. The software that was developed was distributed freely without any limitations, respecting academic traditions of sharing knowledge. In other words, software was considered, as was scientific research, to be a public good. Free access to research results was an essential characteristic of software that could not be renounced.

These researchers were able to collaborate thanks to a very important tool: Arpanet, the antecedent to Internet. This network linked the various communities developing software and facilitated the sharing of programs and knowledge. Sharing and exchange were what kept technological innovation in motion.

New software and hardware were developed at a very fast speed. However, fast development and the lack of any form of standardization meant that software products and computers were often incompatible. However, since these new software products needed to be tried out and used in very different operational settings, researchers had to deal with the boring and tiresome task of translating large amounts of software. Manual translation was the only way to achieve the best possible compatibility with each single system. In order to avoid wasting time translating, specific projects were undertaken to develop software that could be compatible with other software or hardware. The laboratories at Bell (AT&T) developed some software products that could easily be used on different hardware platforms.

The efforts to develop products that were universal and independent of the system being used involved developing programming languages as well. Programming languages are the basic tool used to write the codes that computers must interpret to execute any given task. In 1972, Dennis Ritchie developed a programming language called "C". C was a multi-platform language, or, in other words, it could be used regardless of the specific hardware or software. This language quickly became one of the

most widely used instruments for developing software products and continues to enjoy a large user community today.

From 1969 to 1974, Ken Thompson and his team of researchers at Bell Labs Computing Research Department developed the first version of the Unix operating system. An operating system is “[t]he foundation software of a computer system, responsible for controlling and launching the installed applications and computer peripherals.”<sup>4</sup> Unix was one of the first products to be developed using the C programming language and, as a consequence, was the first operating system to be developed with the aim of being usable regardless of the type of machine/computer it would be used on.

The source code of Unix was freely distributed during the seventies. The source code is the series of instructions written in a language that is easy for programmers to use, e.g. Unix’s source code is written in C. Human-readable source code must then be translated by a compiler or assembler into the computer-readable object code in order to execute the instructions. By distributing Unix with the source code, Bell Labs effectively opened up the development of Unix to a wider community of researchers and programmers. In 1975, Ken Thompson spent a year as a visiting professor at the University of California-Berkeley. He took Unix with him and the University became involved in the development and debugging of Unix. The Berkeley Software Distribution (BSD) developed versions of Unix in parallel with Bell Labs versions of Unix. The first version of BSDUnix was released in 1979. This was covered by what may be considered one of the first open licenses guaranteeing use, modification and redistribution of the source code.

A fast, diffused and low-cost means of communication was still needed to allow researchers in various universities and laboratories to collaborate on software development projects. In 1973, Vinton Cerf and Bob Kahn, two researchers working with DARPA (The Defense Advanced Research Projects Agency of the US Department of Defense) created the foundations for what would become the Internet: the communication protocol TCP/IP (Transmission Control Protocol/Internet Protocol). The new protocol made it possible to connect the different computers in the network in a faster and more reliable way.

In 1979, Eric Allman, a student at the University of California Berkeley, developed Sendmail, a program for sending electronic mail over the Arpanet network. This program, which was free of any intellectual property constraints, quickly became the *de facto* standard for Arpanet and is still widely used for the Internet network.

Up to this point in our story, software was still considered to be a mere support tool for hardware, or, in other words, a tool needed to use and, above all, spread the use of computers. Therefore, only hardware had a commercial value whereas software was basically a freely distributed knowledge product. In the late 1970s, Unix was the only portable, machine-independent, affordable operating system. AT&T began to recognize the commercial value of this software product. In 1984, a provision of the Antitrust Division of the US Department of Justice forced AT&T to separate its software products from other products. AT&T decided to focus on producing, distributing and selling Unix as a commercial product. From this point on Unix became protected by property rights and was no longer available for free to development communities outside the company. Not even the academic research departments that had participated in the development of the Unix operating system were able to easily obtain new versions of the product.

In the early to mid eighties, the widespread diffusion of personal computers eventually lead to the transformation of software for personal computers from free products into commercial products. Since PC users usually did not have programming skills and depended on software to use computers, they were willing to pay for software, giving it a commercial value.

### **1.3 The Beginnings of the Open Source Movement**

The most significant consequence of the birth of a market for software proved to be the migration of the best programmers from academic research centres to the R&D laboratories of private companies. For example, at the beginning of the eighties, the company Symbolics hired almost all of MIT's Artificial Intelligence Lab's programmers. This

move marked the growth of a software market characterized by the protection of intellectual property. The researchers who stayed on at universities had to negotiate these property rights with companies in order to obtain executable copies of some important products. Many researchers who had, up to then, directly contributed to developing these products felt frustrated by these changes. Furthermore, commercializing software meant that it couldn't be freely distributed and for academic researchers this was considered to be an obstacle to scientific research and innovation. For there to be a free exchange of information and cooperation, there had to be free access to source codes. In academic development communities the fundamental values of freedom and collaboration had not changed. Companies, on the other hand, considered any form of software sharing to be on the same level as pirating software. Outsiders were not allowed to make any changes to software protected by copyright even if the change might actually have improved the product.

Frustration with the destruction of the academic development community lead Richard Stallman, then a researcher at MIT, to try to rebuild the community. Stallman wanted to change the socio-economic and legal system that was limiting software sharing. Therefore, his first aim was to oppose the software industry and the rules it applied to defend intellectual property. To do this, in 1984 Stallman started the GNU project ("GNU's not Unix") to develop a free-access operating system based on Unix with applications and development tools as well. To support the GNU project, the Free Software Foundation (FSF) was founded in 1985. The main aim of the FSF was to redevelop a series of products based on the concept of free software. For the FSF, the word "free" meant "freedom", i.e. the possibility to use programs, modify source codes and then distribute the modified versions. The word "free" did not, therefore, mean "free of charge" but rather "free access". Stallman himself clearly explained this concept: "The term free software has nothing to do with price. It is about freedom". In other words, the aim of the FSF was to protect free software from being appropriated by others and used for commercial ends.

The first problem that the Free Software Foundation had to face was that, at the time, there were no software licenses that could actually

protect free access and free use. The aim of licenses is to describe in detail what can and cannot be done by software programmers and users. In 1988, the FSF published the first version of the General Public License (GPL) which introduced the concept of “copyleft”. The concept of copyleft opposes the concept of copyright. Rather than protecting a product’s property rights, copyleft protects the freedom to copy, distribute, and change a product. As Stallman explained:

The goal of GNU was to give users freedom, not just to be popular. So we needed to use distribution terms that would prevent GNU software from being turned into proprietary software. The method we use is called “copyleft.” Copyleft uses copyright law, but flips it over to serve the opposite of its usual purpose: instead of a means of privatizing software, it becomes a means of keeping software free.<sup>5</sup>

And the preamble to the GPL states:

...[t]he licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users.<sup>6</sup>

The General Public License had a strong influence on the development community because it laid down clear rules protecting the freedom of projects developed by the Free Software Foundation. Furthermore, the license proved that there was an alternative model for software development which strongly opposed the commercial view of companies. A community of voluntary software developers, the so-called hacker community, grew around the FSF and GPL. However, the GNU project, which mainly focused on developing a free operating system, still depended too much on the Unix code. In order to be freed up from the Unix constraint, a free version of the kernel had to be developed. A kernel is the heart of an operating system that controls memory, disk, process and task management.

In 1990, a computer science student at the University of Helsinki, Linus Torvalds, started to carefully study the Unix operating system. Torvalds had a general interest in the kernels of operating systems and in particular in multi-tasking techniques. His choice to study Unix depended on the fact that DOS (Microsoft), a more accessible and less expensive operating system, did not implement multi-tasking. At that time Unix was still relatively expensive and a student could certainly not afford it. Furthermore, the University of Helsinki could not afford to buy enough computers using Unix for all its students. Torvalds spent much of his time waiting for a computer to free up in the University's computer labs and found this annoying and frustrating. Therefore, he decided to work on developing his own Unix-like operating system. He had no idea how he would manage such a big project, but decided to give it a try anyway. Torvalds did not have the resources, time, tools or skills to develop an operating system on his own. His inexperience and inability to realistically evaluate the effort required actually proved to be an important factor in the starting up and success of the project. Torvalds himself later said that had he fully understood the efforts required to carry out the project he never would have started in the first place.

His efforts initially focused on studying Minix, a Unix clone, developed for didactic purposes by Andrew Tanenbaum at the University of Amsterdam. Torvalds did not plan on developing the whole kernel. In fact, much of his work at the beginning focused on developing some simple task-switching programs, i.e. the multi-tasking that had originally got him interested in operating systems. Torvalds diffused the news of his project in an online newsgroup with the following message on August 25, 1991:

Hello everybody out there using minix – I'm doing a (free) operating system (just a hobby, won't be big and professional like GNU) for 386(486) AT clones. This has been brewing since April, and is starting to get ready. I'd like any feedback on things people like/dislike in minix, [...] I'd like to know what features most people would want. Any suggestions are welcome, but I won't promise I'll implement them.<sup>7</sup>

Within the following month, Torvalds managed to develop a stable version of his product, which he decided to call Linux (Linus+Unix). This first version of the kernel included a simple and primitive file system but it was still incomplete. Once again, he called on the on-line community for its indispensable collaboration in helping the project evolve into a more complete form. To get help from this virtual community, Torvalds posted another message about his project on October 5, 1991, explicitly asking for help in developing his kernel. The announcement by Linus Torvalds was accompanied by this message:

I'm working on a free version of a Minix look-alike for AT-386 computers. It has finally reached the stage where it's even usable (though it may not be, depending on what you want), and I am willing to put out the sources for wider distribution.... This is a program for hackers by a hacker. I've enjoyed doing it, and somebody might enjoy looking at it and even modifying it for their own needs. It is still small enough to understand, use and modify, and I'm looking forward to any comments you might have. I'm also interested in hearing from anybody who has written any of the utilities/library functions for minix. If your efforts are freely distributable (under copyright or even public domain) I'd like to hear from you so I can add them to the system.<sup>8</sup>

The response to Torvalds's message was immediate and extraordinary. A large community began to develop around the Linux project. By this time, the internet was becoming more and more diffused making the development and expansion of an online community of software developers possible. The increase in the size of the community automatically lead to an increase in productive capacity both in terms of number of projects that could be developed and the quality of the products.

Collaboration between the Linux community and the Free Software Foundation lead to the development of GNU-Linux, a complete non-commercial operating system. GNU-Linux would prove to be one of the most extraordinary results created by a community of voluntary software developers.<sup>9</sup>

The work carried out by the community made it possible for Torvalds to release the first official version of the Linux operating system in 1994. During the first few years of development, the growth of the operating system was exponential; the code lines that made up Linux went from the 10,000 developed by Torvalds to 1.5 million lines in 1998. This project, which had been started to satisfy the curiosity of one single person, had managed to involve 12 million users and 120 different countries. Today Linux is one of the few free products in direct competition with an already established proprietary product, Microsoft's Windows operating system, that has managed to widen its own market share.

#### **1.4 Diffusion**

Following the rapid development of the GNU-Linux system, some companies began to get involved in the project by using it and/or contributing to its development. In 1994, Bob Young and Mark Ewing founded Red Hat with the aim of improving some of the main drawbacks to Linux, namely to create a more user-friendly interface and increase the number of application programs that could be used with this new operating system.<sup>10</sup> Red Hat provides technical assistance, documentation and training and develops more user-friendly configurations to distribute Linux to non-expert users.

Red Hat is one of the most important companies involved in the development of software products in a copyleft context. Red Hat is a so-called *pure player*, i.e. a company whose business model is completely focused on deploying a copyleft product and related services. Only a fraction of the company's profits come from the direct sale of software packages and the value of these packages does not depend on the cost of the individual products but rather the cost of integrating the products and packaging them. The result is that the products Red Hat deploys are much cheaper than other software packages built around proprietary software products. At the same time, anyone can copy and distribute Red Hat products for free since they are protected by copyleft.

Red Hat follows the evolution of the Linux operating system very closely. Every time Torvalds and his collaborators distribute a new

version of the kernel, Young and Ewing take advantage of the new release to build a more complete and operational software package based on it. Red Hat's aim is to complete the Linux platform by adding and integrating the necessary tools and most useful application programs. These additions are made in order to make the operating system actually usable for a non-expert clientele and, as a consequence, to promote the mass diffusion of Linux.

By 1999, Red Hat had proved so successful that in one of the US Justice Department's various antitrust cases against Microsoft, the software giant used Red Hat's Linux system as proof that it did not have a monopoly on the operating system market. Microsoft basically involuntarily publicized and legitimized Linux and Red Hat. Red Hat then decided to widen its market by going abroad, first to Europe and then to Asia. It now occupies a significant share of the Chinese market.

In 2001, Red Hat proposed a solution to one of the numerous antitrust cases by the US Justice Department against Microsoft. The solution was that the government allow Red Hat to provide the Linux operating system, office application programs and many other products for free to all American schools, private and public. In exchange, Microsoft would have to provide the necessary computers. The aim of this proposal was not to save Microsoft from being found guilty, but rather to deploy non-proprietary products in every school district in the United States. Obviously Microsoft's managers understood Red Hat's real intentions and did not accept the offer. In fact, the proposal was an attempt to open school districts to non-proprietary software and by accepting the proposal Microsoft would have risked losing its control over the market. In other words, had Microsoft accepted this solution, students and, therefore, future generations of computer users would have grown up knowing how to use non-proprietary products rather than Microsoft products. Bill Gates and the managers at Microsoft decided to face the antitrust case and the possible consequences it might have considering even a heavy fine to be the lesser of the two evils.

In reality, the use of non-proprietary products and Red Hat Linux in particular is already widespread in many universities. North Carolina State University was the first in a series of universities to make an agreement with Red Hat for the supply of products to its Engineering

Departments. Red Hat's aim is to deploy these products in all schools in order to free educational institutions from the influence of proprietary products and standards. The idea is, therefore, to establish the Linux operating system, and in general non-proprietary software, as alternative standards.

As the Linux project grew, so did another project for a free product: the web server software Apache. In 1994, a group of programmers in the California Bay Area formed the Apache Group to develop a free, reliable web server. By December 1995 the group had developed and was able to release Apache 1.0. Apache had immediate success and quickly became the most popular HTTP server on the Internet. In 1999, the members of the Apache Group formed the Apache Software Foundation (ASF) to provide organizational, legal and financial support for the Apache web server.<sup>11</sup>

### **1.5 Institutionalization**

During the mid-nineties, members of the various free software development communities began to feel the need to overcome the misunderstandings the word "free" was causing among many users. A group of programmers, many of whom had built up their reputations within the Free Software Foundation, began to work together to try and find a strategy to make the concept of free software more appealing to companies. The aim of this initiative was to avoid the widespread association of the word "free" with the meaning "free of charge". The Free Software Foundation had tried to disassociate itself from this interpretation by using the motto "free speech, not free beer". The Free Software Foundation's mission was to promote the freedom of information and research, not to destroy the commercial value of software. This misinterpretation of the word "free" was the weak point of the free software philosophy and kept many software companies from becoming involved in the community. In order to change this, companies had to believe that involvement in the free software development company had something to offer them as well.

In 1997, at a Linux Congress, Eric S. Raymond presented a paper called “The Cathedral and The Bazaar”. This paper compared the commercial software development model (the cathedral) to the new development model based on a community of dispersed software developers and on the free distribution of codes (the bazaar). Raymond proposed the use of the term “open source” as the best alternative to “free”. This paper marked the beginning of the study of voluntary development communities, their behaviours and the economic sustainability of this model.

1998 was a very important year in the history of the open source community and open source software. The “open source” label was decided on during a strategy session in February 1998 that included several members of the Linux community and Raymond himself. In the same month Bruce Perence and Eric Raymond founded the Open Source Initiative (OSI)<sup>12</sup> with the main aim of creating a document that would precisely define open source software (The Open Source Definition).<sup>13</sup> In April of the same year, the OSI officially approved the term “open source” to indicate the accessibility, efficiency, reliability, flexibility and innovation that would characterize open source software.

The new definition managed to make free software much more appealing to companies even if the Free Software Foundation and the Open Source Initiative did not have the same philosophy and did not agree on the methods of software development. Regardless of these differences, however, the majority of open source projects were originally supported by and developed in collaboration with the Free Software Foundation.

On January 23, 1998, an announcement was made that surprised everyone in the IT world: Netscape intended to make the source code of its browser Navigator public, i.e. to share it with the open source development community.<sup>14</sup> Netscape Navigator was quickly losing market share to its main competitor Microsoft Explorer. Consequently, Netscape made the difficult decision to open Navigator’s source code and started a project called Mozilla. The announcement created a lot of interest within the open source community and especially among the main supporters of the Open Source Initiative and the Free Software Foundation.

Netscape's group of managers had to face a series of very challenging problems. First of all, the property of some parts of Navigator were shared with other companies. Therefore, Netscape had to communicate its change in strategy and, if possible, convince the partner companies to support the Mozilla project. Netscape set a deadline, February 24, 1998, by which each partner had to communicate its decision. They could choose to participate in the Mozilla project or leave it. Since some companies backed out, Netscape had to eliminate some code.

Another problem Netscape had to deal with was what license to choose to best respect the aims of the project. Mozilla needed a license that would motivate the community of volunteers to contribute to the project and at the same time protect the economic interests of Netscape. None of the licenses that existed at that time, including the open source licenses, met these requirements. Therefore, on March 5, 1998, Netscape proposed a new license created by the company itself. The license, called the Netscape Public License (NPL), was immediately presented to the open source community for feedback. Unfortunately, the reaction of the community was very negative. The community did not like the presence of some norms that reserved Netscape some special rights. One of these was the possibility Netscape would have to not share with the community some of the parts of the new code developed as a part of the Mozilla project. Netscape, believing the support of the entire open source community to be indispensable to the success of Mozilla, tried to remedy the situation. The license was reviewed and modified and on March 21, 1998 Netscape proposed a second license called the Mozilla Public License (MPL). This license was different from the NPL in that it did not reserve Netscape any particular privileges. The community accepted the MPL. This new license convinced the open source community that the Mozilla project was serious and credible.

The official presentation of the project was quite unique. The location had to attract the curiosity of the community of developers, which is notoriously made up of anti-conformist and eccentric hackers. A party, open to all, to celebrate the project was held on April 1, 1998 in one of the biggest nightclubs in San Francisco. Netscape wanted the substantial and credible involvement of the open source community. The success of the Mozilla project was considered to be directly related to the ability of

the company to obtain the consensus and contribution of the best programmers in the community. Netscape had to move fast. The fact that Netscape was the first major company to take part in the open source community was one of the greatest advantages the Mozilla project gave Netscape.

Netscape wanted to be responsible for managing the Mozilla project and so, on May 24, 1998, it registered the domain mozilla.org. A series of support tools for the development community were provided on this site. Netscape wanted to play an active role in the decision making process and, therefore, had to create an efficient structure around Mozilla. At the same time, however, this structure had to be separate from the rest of the company. Netscape had to make a significant effort to reach a compromise between influencing the project and allowing the development community the freedom it needed. At the end of this initial planning stage, Netscape was finally able to state: “the lizard [Mozilla] is free”.<sup>15</sup>

Following the foundation of the Open Source Initiative and the beginning of the Mozilla project, hundreds of open source projects were started and numerous companies began to directly participate in the development and diffusion of these projects. For example, in 1999 IBM announced that it wanted to use the Linux operating system and Apache web server in its e-business solutions. Other companies followed IBM’s example and began to become directly or indirectly involved in the open source community. For example, in the same year Sun Microsystems launched a project to share the development of its own office product, StarOffice, with the open source community. This led to the development of OpenOffice.org, an open source office project and product. OpenOffice.org is a complete office suite with features comparable to Microsoft Office features.

The involvement of companies validated open source concepts and strategies. Companies in the IT industry were forced to start comparing their own business model to open source software development. Even Microsoft, regardless of its position on the market, had to begin to carefully evaluate the open source phenomenon and study strategies to effectively respond to the threat it posed. In fact, the “Halloween Documents” already mentioned at the beginning of the chapter, indicated

that Microsoft considered Linux to be a real threat to its own business model. Microsoft eventually announced its intention to release part of the source code of some of its proprietary software products exclusively to some governments and institutions.

Table 1.1 shows the main events that played a role in the creation and evolution of open source software.

Table 1.1 – Significant historical events in the development of the open source software.

|      |                                                                                                                                                                                                              |
|------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1968 | Arpanet                                                                                                                                                                                                      |
| 1969 | First version of Unix                                                                                                                                                                                        |
| 1972 | “C” language                                                                                                                                                                                                 |
| 1973 | TCP/IP                                                                                                                                                                                                       |
| 1979 | Sendmail<br>AT&T starts selling Unix<br>BSD first version                                                                                                                                                    |
| 1984 | GNU project is launched                                                                                                                                                                                      |
| 1985 | Free Software Foundation is launched                                                                                                                                                                         |
| 1991 | Linux Project is launched                                                                                                                                                                                    |
| 1994 | Red Hat is founded                                                                                                                                                                                           |
| 1995 | Apache Software Foundation is launched                                                                                                                                                                       |
| 1998 | Open Source Initiative is founded<br>Mozilla Project is launched                                                                                                                                             |
| 2000 | Linux included in IBM products<br>Sun launches Open Office                                                                                                                                                   |
| 2003 | Microsoft first announcement of limited access opening of parts of Windows<br>EU-IDA publishes Open Source Software guidelines for Governments<br>The city of Munich announces Open Source Software adoption |
| 2004 | French Government announces large scale migration to Open Source Software (ADELE)                                                                                                                            |

Thanks to the open source movement, the development model that had been developed within the context of the Free Software Foundation found new outlets. The way the Open Source Initiative interpreted this

model made it possible to involve companies in open source projects, favouring the diffusion of open source products on the market. This stage in the evolution of open source software saw the creation of an efficient network of companies involved in offering support services for open source products and products complementary to open source software. Furthermore, some governments and public administrations began to show interest in the community and its products. Some governments are currently considering the possibility of using open source software in place of proprietary software to manage public information. Governments manage large quantities of information, deal with the public and have legislative power. All of these factors make the idea of government involvement in the open source community very appealing to programmers

### **1.6 Recent Developments**

The open source phenomenon has led to the creation of free, open software products that can compete with proprietary products on the market. It has also introduced a new way of developing software. However, the success of both open source products and the open source model have made the open source community the object of many legal attacks. The aim of these attacks is to limit the diffusion of open source software.

On March 7, 2003 the SCO Group filed a \$1 billion lawsuit in the US against IBM. It warned 1500 IBM clients that the software they were using was illegal. SCO claimed that the Linux operating system sold by IBM included Unix code that was protected by copyright. In June 2003 the amount was increased to \$3 billion and then to \$5 billion. IBM reacted to the lawsuit with counter-claims against SCO. This, in turn, caught the attention of many companies that were using open source code such as Red Hat. On August 4, 2003 Red Hat started legal action against SCO claiming that some of the statements made by SCO indicated that anyone using a version of Linux that was not covered by a SCO UNIX license was violating copyright. Red Hat then created the Open Source Now Fund to financially support any legal expenses

programmers and non-profit institutions that use the GPL might incur. On September 24, 2003, Hewlett-Packard (HP) also decided to create its own fund to guarantee continued use of its products to future users. Sun Microsystems guaranteed users of its products legal protection as well. On January 13, 2004, Novell followed suit. Novell's involvement was important because this company holds the copyright on Unix. The Unix system created by AT&T was sold to Novell, which then conceded some copyright to other companies, including SCO. Novell claims that there are no parts of the Unix code in Linux, and even if there were, since Novell holds the copyright, it would give its clients permission to use Linux. Novell's decision to side with Linux could lead other companies to do the same.

Following many warnings, on March 4, 2004, SCO filed lawsuits against two corporate users of Linux: AutoZone, the largest American retailer of automotive parts and accessories, and DaimlerChrysler, the automobile and truck manufacturer.

On July 19, 2004, Newsforge published a article reporting an HP memo that forecasts Microsoft waging a war on free software by using software patent infringements as its weapon. The memo stated: "Microsoft is going to use the legal system to shut down open-source software".<sup>16</sup> According to the article, Microsoft could attack not only Linux distributors but open source programmers as well. Microsoft has been spending an increasing amount of money on filing patents for even the most elemental computing components in its software products. Furthermore, Marshall Phelps, hired by Microsoft to help develop its intellectual property strategy, "...is on record as saying that Microsoft intends to spend \$7bn annually on IP issues."

One proposal that has been made to help avoid these legal problems in the future is that from a certain moment on only code whose origin is verified by a certificate of origin will be included in the Linux kernel. The aim is to be able to more easily and precisely retrace the names of the authors of every single part of the code that makes up the heart of Linux. Using a Developer's Certificate of Origin (DCO), every developer has to "sign" every contribution he/she makes. In addition, developers have to certify one of the following three points:

- the contribution is either completely original,
- the contribution is based upon a previous work that is nonetheless covered under an open source license,
- the contribution comes from someone else who has certified one of the previous two points and has not been modified from its original form.

In another attempt to help avoid future legal actions by SCO or others, Pamela Jones, creator of pro-open source information site Grolaw.net, has started a project to clarify what the origins of Unix are. In her letter presenting the project on February 4, 2004, Jones states:

I want to do a systematic, comprehensive, and carefully documented history timeline relating to Unix and the Linux kernel, based, with his kind permission, on Eric Levenez's Unix History timeline chart, but from the perspective of tracing the code by copyright, patents, trade secret, and trademark. The idea is that the final timeline will be a publicly-available resource, released under a Creative Commons license, that will assist the community in defending against - or better yet in deterring - future lawsuits against GNU/Linux code.

On April 19, 2004 there was actually the first court ruling in favour of open source software. A three-judge panel in a Munich court ordered a preliminary injunction against the company Sitecom for violating GNU GPL. Sitecom was an open-source networking software distributed without attaching the GPL text and the source code. This ruling by a German court, which recognizes the legal validity of the GPL, could set a precedent in the history of this license. In the past, Microsoft has often questioned the legal validity GPL and, more recently, SCO has claimed that in its case against IBM it will show the weaknesses of this license. Although there are ambiguities in the license that will most likely be fought over, this case gives the community of GPL users hope for the future.

Open source software is particularly vulnerable to the sorts of attacks SCO has made. SCO, and anyone else for that matter, can see the source code of open source products such as Linux whereas the open source

community does not have access to the code of proprietary software. This puts the open source community at a disadvantage. At the heart of all these legal cases is the issue of the relationships between intellectual property rights and software. To better understand these issues, in the next chapter we will discuss what intellectual property rights are and how they have been and are being applied to software.