



THE USE OF FREE AND OPEN SOURCE SOFTWARE BY ICELANDIC SOFTWARE DEVELOPERS

Research report
Hildur Björns Vernudóttir

Spring 2010
B.Sc. Computer Science

Supervisor: Marta Kristín Lárusdóttir
Co-supervisor: Tryggvi Björgvinsson
Examiner: Hannes Högni Vilhjálmsson

T-619-LOKA
School of Computer Science

Abstract

This thesis presents a study made on the use of Free and Open Source software (FOSS) in software development in Iceland. Structured interviews with 12 practitioners from 12 Icelandic software development companies were conducted. The results show that seven out of twelve companies use a combination of FOSS and proprietary software in development. There are six main reasons practitioners give for using FOSS in their software development. First, it is cheaper; second, it can be implemented on any computer without licensing issues; third, it uses open standards; fourth, it is flexible; fifth, it is portable, and sixth and finally it contributes to less vendor lock-in. On the other hand, there are five main reasons practitioners give for not using FOSS for software development. First, the user interface of proprietary software is more appealing than of FOSS; second, it is easier to exchange data outside the company; third, the knowledge is already inside the companies because almost everybody learns to use proprietary software in school, since at least today proprietary software is dominant in the university environment; fourth, there are more functions or features in proprietary software than FOSS, and fifth and lastly, proprietary software has more reliable quality. The results show that more small-sized companies (EU SMEs definition) are using FOSS than micro-sized companies which is very surprising for it was assumed that micro-sized companies (usually start-up companies) had lesser funds and would therefore use more FOSS than larger companies.

Samantekt

Þessi ritgerð fjallar um niðurstöður rannsóknar á notkun á frjálsum og opnum hugbúnaði hjá íslenskum hugbúnaðarfyrirtækjum. Það voru tekin viðtöl við 12 aðila frá 12 íslenskum hugbúnaðarfyrirtækjum. Niðurstöðurnar sýna að sjö af tólf fyrirtækjum nota bæði frjálstan og opinn hugbúnað og séreignarhugbúnað við þróun á hugbúnaði. Viðmælendurnir gáfu upp sex megin ástæður fyrir notkun á frjálsum og opnum hugbúnaði í hugbúnaðarþróun. Í fyrsta lagi, því hann er ódýrari, í öðru lagi er hægt að setja hann upp á hvaða tölvu sem er án þess að þurfa að huga að leyfismálum, í þriðja lagi þá styðst hann við opna staðla, í fjórða lagi er hann sveigjanlegur, í fimmta lagi er hann flytjanlegur og í sjötta og síðasta lagi er síður hætt á því að læsast inni hjá ákveðnum söluaðila. Hins vegar gáfu viðmælendurnir upp fimm megin ástæður fyrir því að nota ekki frjálstan og opinn hugbúnað í hugbúnaðarþróun. Í fyrsta lagi þykir viðmót séreignarhugbúnaðar vera meira aðlaðandi, í öðru lagi er auðveldara að skiptast á upplýsingum utan fyrirtækis, í þriðja lagi liggur þekkingin þegar fyrir innan fyrirtækjanna þar sem flestir læra að nota séreignarhugbúnað í skóla vegna þess að hann er fremur ráðandi í háskólum í dag, í fjórða lagi er meiri virkni eða fítusar í séreignarhugbúnaði, og í fimmta og síðasta lagi eru áreiðanlegri gæði í séreignarhugbúnaði. Niðurstöðurnar sýna að fleiri lítil fyrirtæki nota frjálstan og opinn hugbúnað heldur en agnarsmá fyrirtæki sem er fremur óvænt þar sem gert var ráð fyrir því að agnarsmá fyrirtæki (yfirleitt sprotafyrirtæki) búa yfir minna fjármagni og þess vegna væru þau líklegri til að nota frjálstan og opinn hugbúnað fremur en fyrirtæki sem eru stærri.

Contents

Glossary	3
1 Preface	4
2 Introduction	5
3 Background	6
3.1 The History of Free and Open Source Software	6
3.2 The Free Software Foundation	8
3.3 The Open Source Initiative	8
3.4 Main difference	9
3.5 Motivation for the study	9
4 Method	10
4.1 Selection of interviewees	10
4.2 The questionnaire	11
4.3 Analysis of interviewees background	13
5 Results	15
5.1 Defining FOSS	15
5.2 The use of FOSS	15
5.2.1 FOSS development tools	17
5.2.2 Reasons for choosing FOSS	18
5.2.3 Quality factors	19
5.2.4 Advantages and disadvantages of FOSS	20
5.2.5 Security of FOSS	22
5.3 The use of proprietary software	23
5.3.1 Proprietary software development tools	23
5.3.2 Reasons for choosing proprietary software	25
5.3.3 Quality factors	26
5.3.4 Advantages and disadvantages of proprietary software	28
5.3.5 Security of proprietary software	29
6 Analysis	33
6.1 The main reasons for Icelandic software developers to use Free and Open Source Software	33
6.2 The main reasons for Icelandic software developers do not use FOSS	34
6.3 The difference in the use of FOSS between those who develop CM software vs. those who develop COTS software	35
7 Conclusion	36
8 Future work	38
9 Acknowledgements	39
A Lessons from The Cathedral and The Bazaar	42

B	The Open Source Definition	43
C	Questions for the structured interview	44

List of Tables

1	SMEs categorization (in million) (EU, 2005)	10
2	Companies categorized by SMEs	11
3	Number of companies selected	11
4	Unique ID of each interviewee	13
5	Number of persons working in software development	13
6	Annual Turnover	13
7	Annual Cost of Software	14
8	Methodology	14
9	Operating Systems on the Servers	15
10	Defining FOSS	16
11	Companies using FOSS	16
12	FOSS development tools	17
13	Quantity of FOSS used by each company	18
14	Reasons for choosing FOSS	18
15	ISO 9126 (ISO, 2001)	19
16	Average rating of the importance of quality factors of FOSS	19
17	Advantages of FOSS	21
18	Disadvantages of FOSS	21
19	Why is FOSS secure (or not)	22
20	List of Proprietary Software	24
21	Quantity of proprietary software used by each company	25
22	Reasons for choosing proprietary software	25
23	Ruling factors why proprietary software chosen	26
24	Average rating of the importance of quality factors of Proprietary Software	26
25	Advantages of proprietary software	28
26	Disadvantages of proprietary software	29
27	Reasons why proprietary software is secure (or not)	30
28	Can FOSS replace some proprietary software	31
29	Quantity of FOSS used by the type of software developed	35
30	Quantity of proprietary software used by the type of software developed	35
31	Use of FOSS divided by the type of software developed	36

List of Figures

1	Average rating of the importance of quality factors of FOSS, sorted by types of software	20
2	Average rating of the importance of quality factors of proprietary software, sorted by types of software	27

Glossary

Commercial Off The Shelf (COTS)

Is ready-made software, available for sale, lease, or licensed to the general public (Wikipedia, 2010a).

Custom-Made software (CM)

Is custom-made software which provides the buyer with some unique system benefits and value that custom-off-the-shelf software cannot (Wikipedia, 2010b).

Free Software (FS)

Is a *social movement* that fights for users *freedom* to share, study and modify software. FS is always Open Source Software (OSS) but OSS is not always FS because OSS gives it self a flexibility to add/remove freedom or copyright privileges (Stallman et al., 2002; Fogel, 2005).

Hacker

An individual who is an expert at programming and solving problems with a computer (Britannica, 2010a).

Non-Disclosure Agreement (NDA)

Is a legal contract that creates a confidential relationship between parties and bind them not to disclose information covered by the agreement and therefore restrict it from generalized use. (The Free Dictionary By Farlex, 2010).

Open Source Software (OSS)

Is a *development methodology* and *business process*. It permits users to use, change and improve software, and to redistribute it in modified or unmodified forms, no discrimination is allowed and it must not restrict other software. Free Software(FS) is always OSS but OSS is not always FS because OSS gives it self a flexibility to add or remove freedom or copyright privileges (OSI, 2010b; Fogel, 2005; Scacchi, 2007).

Proprietary Software

Software that is neither free nor open source, it is protected by patent and/or copyright against free competition and there are often major restriction for its use (Britannica, 2010b).

Small and Medium-Sized Enterprises (SMEs)

Enterprises that are categorized by their size, micro, small or medium-sized. They have to comply with a certain criteria such as the staff headcount and the turnover or balance sheet ceiling (EU, 2005). The guidelines are shown in table 1 on page 10.

Software developer

Is a person who designs and writes software (PCMagazine, 2010).

1 Preface

This thesis is a final B.Sc. project with a research emphasis in the School of Computer Science at the University of Reykjavik. It presents the result of a survey which was conducted in March and April 2010 on the use of Free and Open Source Software in Icelandic software development. It reveals the reasons why Icelandic software practitioners choose to use or not use Free and Open Source Software in software development.

It is written for all those who are interested in Free and Open Source Software, and specifically those who are interested in knowing more about the use of Free and Open Source Software by Icelandic software companies.

The thesis starts with a brief introduction of the research. Then it continues with a look at the history of both Free Software and Open Source Software, how they define themselves and what separates them. Thereafter, the thesis describes the research method used to conduct the research. The results of the survey are presented in chapter 5, and the thesis ends with an analysis of the results and conclusions that can be drawn from them.

2 Introduction

In recent years, awareness about Free and Open Source Software (FOSS) has increased, both with the general public and amongst companies. Over the last year there have been three conferences on this subject in Reykjavik alone, and according to them, many companies look at Free and Open Source Software as a real option today, instead of proprietary software. This should especially apply to start-up companies due to the lower cost of such software. Even though the general cost is lower, Free and Open Source Software has a growing reputation for better quality, stability, and help on on-line forums with possible solutions to problems. Furthermore, many IT service companies are servicing Free and Open Source Companies as well as proprietary software. Because of this, it is interesting to see what direction Icelandic software development companies are taking with respect to FOSS.

This research report focuses on the main reasons why software development companies are making use of FOSS for developing software. Could it be because of higher cost of the proprietary software, easier access to FOSS, open access to the source code, some idealistic views, because of the development methodology or some other reasons? Furthermore, if they are not making use of FOSS, what are the main reasons they use proprietary software? Can it be because of better quality, more security, more services provided by proprietary companies, some beneficial contracts they have with the proprietary companies or some other reasons? These are the questions this research intends to answer.

To take the research a step further, a theory is set forward to compare the use of Free and Open Source Software development tools in different companies. The theory, set forward as the third question, is that when companies develop software, they tend to use less of FOSS when developing *Custom-Made Software (CM)* compared to making *Commercial Off the Shelf Software (COTS)*. This suspicion is based on the assumption that when developing CM software companies want to use generally accepted software of good quality, to fulfil the requirements of their broad customers base.

To summarize, these are the three research questions to be answered in this research:

1. Given that Icelandic software developers use Free and Open Source Software (FOSS), what are the main reasons behind the decision?
2. If they are not using Free and Open Source software, then what are the main reasons for doing so?
3. Is there a difference in the use of Free and Open Source Software between those who develop Custom-Made software (CM) versus those who develop Commercial Off The Shelf software (COTS)?

3 Background

Many use the names Free Software (FS) and Open Source Software (OSS) interchangeably but there is a fundamental difference between these two concepts. The study itself combines those two concepts and uses the term Free and Open Source Software (FOSS) to describe both.

In the following section the history of both FS and OSS will be described to gain better understanding of the meaning and origin of Free and Open Source Software.

3.1 The History of Free and Open Source Software

The history of Free and Open Source Software runs back to the beginning of software development, before men realised that software could be commercialized. For many years the academic society was a software-sharing community, where software development was a collaboration to improve software (Salus and Reed, 2008). In the 1970s, a computer hacker¹ named Richard Stallman worked at the Artificial Intelligence lab at MIT among other academic hackers. They worked in harmony with each other, exchanging source code and collaborating on software projects. Then things started changing, and in the 1980s companies started hiring many skilled hackers from the universities leading to an inexperienced staff replacing them that did not possess the skills nor interest to maintain the software on the old computers. The system started breaking and the reaction was to start using proprietary software which Stallman opposed (Levy, 2001).

Stallman was not comfortable in this new environment. He did not like this development, that fellow hackers were signing non-disclosure agreements (NDA), which is a legally binding contract that restricts sharing software and therefore restricting users' freedom. Signing in on the NDA forced them to deny their fellow hackers access to the source code. Stallman felt that this was morally wrong and that it divided users in the sense that they were not allowed to distribute programs freely (Williams, 2002). Hackers were betraying the key element of the "Hacker Ethic" which is the free flow of information² (Levy, 2001). Because of this, Stallman started calling himself *the last true hacker*.

The proprietary business developed dramatically and very skilled software developers were being hired from the universities and signing NDAs. Codes which had previously been open were being closed, like for example the UNIX operating system and others. What finally encouraged Stallman to act on this changing environment was an incident with a Xerox printer the university was given. The printer had a jamming problem that Stallman wanted to fix with a simple addition to the source code which notified the person using it that the printer was jammed and that the person should go and fix it. To be able to do that, he had to be able to access the source code. After months of frustration over the printing problem he finally contacted a person in another university he knew had worked on the code at Xerox. This person denied him of the source code because of NDA. Stallman got so angry, he turned away feeling victimized and walked out (Williams, 2002). For Stallman, all these changes were

¹Hacker in the positive sense of the word as "an expert at programming and solving problems with a computer". It is often confused with the term cracker which is an individual trying to gain unauthorised access to a computer system (Britannica, 2010a)

²The six codes of honour of the Hacker Ethics are: i)Access to computers-and anything which might teach you something about the way the world works should be unlimited and total. Always yield to the Hands-On Imperative. ii)All information should be free. iii)Mistrust Authority - Promote Decentralization. iv)Hackers should be judged by their hacking, not bogus criteria such as degrees, age, race, or position. v)You can create art and beauty on computer. vi)Computer can change your life for the better. (Levy, 2001, p. 40-45)

just to much for him to bare. He decided to fight back the only way he knew, by programming. He wanted to be able to use a non-proprietary operating system to withhold his own ethical standard and to give other people that opportunity as well. Therefore, he quit his job at MIT and started working on his own operating system which he called GNU³, later joined by others (Williams, 2002). While working on that project he founded the *Free Software Foundation* (FSF) which advocates for freedom of all software. The GNU kernel, dubbed the Hurd, is a complicated micro-kernel system, which at the time caused the GNU developers some trouble and is even today, twenty years after they started working on it, not fully ready.

Going back to the year of 1991, in Finland, a man named Linus Torvalds was also trying to build an operating system which he said would not be as big and impressive as the GNU (Torvalds and Diamond, 2001). At the time he was a student at the University of Helsinki and had bought himself a computer earlier that year. He used the Minix operating system (micro-kernel) but it did not fulfil his needs so he started his own pet-project, building himself an operating system using a monolithic-kernel architecture. Torvalds got an access to a ftp server in the University of Helsinki where he published his first public version of the Linux kernel version 0.01⁴ in September 1991 and invited everyone to join in on the project. It went really well, many participated and after many minor releases, Linus released version 1.0 in 1993. The Linux kernel was issued under a GPL license as free software⁵. Because of the lack of the Hurd-kernel in the GNU operating system, the GNU and Linux were combined, which resulted in the first entirely free operating system, called the GNU/Linux operating system.

The hackers themselves were stunned over the rapid development of Linux, no one could have anticipated that Linux would be developed at that speed and with that much success. One hacker, named Eric S. Raymond, was fascinated by the development method Torvalds used to build the kernel and decided to repeat that process with the open community, but on a much smaller scale, with an email system. He wanted to see if that development process could be repeated with good results. When he had successfully finished his experiment he published an article named "The Cathedral and the Bazaar" (Raymond, 2001) where the Cathedral is a metaphor for the developing community where only a few people have the complete oversight of the source code (like GNU) and the bazaar as a metaphor for the whole development society as Torvalds did it (Raymond's guidelines can be seen in Appendix A). The article was a huge success, and it led to Netscape contacting Raymond because they wanted his help to open the source code for their web-browser. Eric was excited about the project's potential, i.e. to open the eyes of the companies for what the hacker community could teach them about the open *development methodology*. From there, Eric Raymond, Bruce Perens, and others, founded the Open Source Initiative (OSI). They declined the moralizing ideals of the Free Software Foundation which they felt did not comply with ongoing business strategies. They wanted to be more business friendly at the cost of some freedom imprisonment.

A year later, the co-founder of OSI, Bruce Perens resigned from the OSI. He said that the OSI was originally founded to introduce the non-hacker world to FS and that the world had noticed so they should start teaching them about FS. He said that FS and OSS are in fact two words with the same meaning, just intended for a different audience, and that the definitions of FS and OSS are entirely compatible. Furthermore, he thought Raymond was drifting away from the free software values and

³GNU is a recursive acronym for "GNU's not Unix"

⁴Torvalds first called it Freax but got a negative reaction to that name and a teacher, Ari Lemmke, at Helsinki changed it back to Linux. Torvalds comments on this in his biography: "I admit that I didn't put up much of a fight. But it was his doing. So I can honestly say I wasn't egotistical, or half-honestly say I wasn't egotistical. But I thought, okay, that's a good name, and I can always blame somebody else for it, which I'm doing now." (Torvalds and Diamond, 2001, p. 88)

⁵GPL is a license maintained by the Free Software Foundation

that OSI was abusing the certification mark in a ways he could not agree upon. He concluded by saying that he sided with FSF (Perens, 1999).

It was clear that the hacker world had split in half.

3.2 The Free Software Foundation

The Free Software Foundation (FSF) was established in 1985 by Richard M. Stallman. FSF is a *social movement* that advocates for users' freedom to share, study and modify a software. Moreover it refers to free as in freedom and free speech, not cost. Free software can be charged for but there cannot be any restraints on sharing, studying and modifications according to its free license. FSF define their goals with four essential freedoms (Stallman et al., 2002):

freedom 0 The freedom to run the program, for any purpose.

freedom 1 The freedom to study how the program works, and change it to make it do what you wish. Access to the source code is a precondition for this.

freedom 2 The freedom to redistribute copies so you can help your neighbour.

freedom 3 The freedom to distribute copies of your modified versions to others. By doing this you can give the whole community a chance to benefit from your changes. Access to the source code is a precondition for this.

The FSF issues Free Software licenses and the most popular license is the GNU General Public License (GNU GPL), a free, copyleft license⁶ for software. It prohibits any derivative work of the software issued under its term to be made proprietary, and any distribution limitations (Laurent, 2004).

The FSF does not only issue licenses but also focuses on producing software, as well as it focuses on internal publications which are inspired by Richard M. Stallman (Feller and Fitzgerald, 2002).

3.3 The Open Source Initiative

The Open Source Initiative (OSI) was established in 1998 by Eric S. Raymond and Bruce Perens. It stands for open source software (OSS) as a *software development methodology* and *business process* (Fogel, 2005). The Open Source Initiative (OSI) refers to the term "Open Source Software" as access to the source code and distribution that has to comply with certain criteria. The main terms of the criteria is to permit the users to use, change and improve software, and to redistribute it in modified or unmodified forms. However, no discrimination is allowed and it must not restrict other software. A more detailed definition can be seen in Appendix B. The OSI does not issue licenses but they can verify an Open Source Software license. If a software license complies with their definition it is then certified by the OSI.

As well as placing the OSI certification mark on OSS, the OSI strategically promotes its reason for existing to the business community (Feller and Fitzgerald, 2002).

⁶All derivative work of a copyleft licensed software is also a free software.

3.4 Main difference

The main difference between FS and OSS is a *difference in licensing* where FS is always OSS but OSS is not always FS. This is because OSS gives itself the flexibility to add or remove some freedoms or copyright privileges which are similar (Scacchi, 2007). They also have different ideological view where the FSF is a *social movement* while the OSI is a *development methodology* and a *business process*.

Many software developers, in the free and open community, want their product to be licensed with both parties. Therefore, the licensing lists of the FSF and OSI are very similar (FS, 2010) and (OSI, 2010a).

Despite the differences between the FSF and the OSI, they also compliment each others work. The OSI does not produce software but FSF produces GNU software and therefore fuels much of the FOSS development in the world. Also, while the FSF focuses on internal publication, the OSI focuses on external publication, i.e. outside the FOSS hackers community. Also, the OSI supplements the FSF publication by legitimating the FS/OSS movement to the business community (Feller and Fitzgerald, 2002).

As stated before, when using these terms together as "Free and Open Source Software" (FOSS), in this thesis it is an inclusive term to cover both FS and OSS certified software.

3.5 Motivation for the study

Coming from a very Microsoft centred academic environment lead to a growing interest in getting familiar with other kind of software, to see what else is out there.

The chance came when series of conferences were held about FOSS over the last year which were very inspiring and raised some question about to what degree Icelandic companies are using FOSS and also what motivates them to use it, what are the advantages and disadvantages? These questions lead to this research report.

The main focuses of researchers exploring this field are the motivations for participating in FOSS projects, some specific software, the social structure, etc. A study, similar to this one, focusing on the actual use of FOSS in software development companies seems not to have been conducted before and therefore it could be an interesting addition to the field. The conclusions drawn from this study can widen our understanding on the advantages and disadvantages of FOSS and proprietary software from the practitioners' perspective.

4 Method

This is an exploratory study using structured interviews with 12 practitioners from 12 Icelandic software development companies. The outline of the chapter is as follows; first, selection of the interviewees is explained; second, the questionnaire is described, and third and last, the chapter details an analysis of the interviewees.

4.1 Selection of interviewees

The population is from a list on the website of The Federation of Icelandic Industries (*is. Samtök Iðnaðarins*)⁷. The list contains a total of 49 Information Technology (IT) companies, mostly in IT development but some in IT services (wholesale and telecommunications). To separate the software development companies from the others, all websites of the companies were visited and basic information was gathered about the companies and their products. If information was lacking the companies were contacted and asked if they were developing software. It turned out that of those 49 companies, 32 of them are in software development.

The European Union's (EU) definition of small and medium-sized enterprises (SMEs) is shown in table 1. Micro, small and medium-sized enterprises represent about 99% of all enterprises in the EU (EU, 2005). For a company to fall under the SMEs definition it has to comply with a certain criteria such as the staff headcount and the turnover or balance sheet ceiling.

Enterprise category	Headcount	Turnover	OR Balance Sheet Total
medium-sized	< 250	≤ €50 m	≤ €43 m
small	< 50	≤ €10 m	≤ €10 m
micro	< 10	≤ €2 m	≤ €2 m

Table 1: SMEs categorization (in million) (EU, 2005)

The Federation of Icelandic Industries website displays the number of employees within each company and that information was verified by contacting all the companies. More detailed grouping was added to the SMEs table as the companies are further categorized by the type of products they develop, that is Commercial Off The Shelf software (COTS), Custom-Made software (CM)⁸ or both. The 32 companies are categorized by the definition in table 1 and the type of software they develop, and the result is shown in table 2. The most common type of software developed is CM software (solely) or in 19 out of the 32 companies.

⁷<http://www.si.is/starfsgreinahopar/upplysingataekni/fyrirtaekin/>

⁸See the definitions in the glossary

	COTS	CM	Both COTS/CM
Larger	1	0	2
Medium-sized	0	1	3
Small	1	13	3
Micro	3	5	0
Total	5	19	8

Table 2: *Companies categorized by SMEs*

One of the main focus of this study is to compare FOSS use in software companies according to whether they are making COTS, CM or both type of software. Additionally the study examines whether the size of the company has an effect on FOSS use, assuming that micro-sized companies (usually start-up companies) use more of FOSS because of lesser funds than larger companies. To be able to have a good sample in both these categorizes, the size of the company and type of software, it was decided to focus on selecting interviewees only from micro and small sized companies and from all types of software developed. From what has been said, regarding the two categorizations, it was decided to interview practitioners in 3 companies who develop COTS software, 6 companies that develop CM software and 3 who develop both COTS and CM software, a total of 12 companies, as shown in table 3. Five of those are micro sized and 7 are small sized companies.

	COTS	CM	Both	Total
Small	1	3	3	7
Micro	2	3	n/a	5
Total	3	6	3	12

Table 3: *Number of companies selected*

One person at each company was interviewed, usually the CEO, the development director or a substitute. When making contact, the company was telephoned and they were asked to forward the call to the person which *has the most knowledge of the software used to develop their product*. Either the call got transferred to that person or the contact information was given. In all cases, an introduction email in Icelandic, as seen in Appendix C, was sent. The interviews were tape-recorded and an interview report was made after each interview.

4.2 The questionnaire

The questionnaire for the structured interview has 9 general questions, 10 questions about FOSS and 10 questions about proprietary software. A total of 29 questions, which can be seen in Icelandic, in Appendix C. If a company was not using FOSS then the FOSS questions were not included in the interview and vice versa if the company was not using proprietary software.

The general questions include, e.g. question on the number of practitioners working on software development in the company, questions on financial issues, what type of servers the companies are using, what develop methodology is used, what kind of products they develop, and finally the interviewees were also asked to define FOSS.

The questions on FOSS include, e.g. what FOSS the company uses, how it was chosen, who makes the decision on using FOSS, what quality factors they value most when choosing a software, if they make changes to the FOSS and whether they contribute the changes back to the community, and what are the advantages and disadvantages of the FOSS they use, etc.

The questions about proprietary software are very similar to the ones on FOSS, with some variations, e.g. whether the company can replace some proprietary software for FOSS.

The answers to the questions asked can be found in the chapter "analysis on the interviewees background" and also in the result chapter.

To be able to trace answers between interviewees, each interviewee is marked by a unique ID, as shown in table 4.

	COTS	CM	Both
Small	S-COTS-1	S-CM-1	S-Both-1
		S-CM-2	S-Both-2
		S-CM-3	S-Both-3
Micro	M-COTS-1	M-CM-1	n/a
	M-COTS-2	M-CM-2	
		M-CM-3	

Table 4: Unique ID of each interviewee

4.3 Analysis of interviewees background

The number of persons working on software development in the companies ranged from 3 to 13 people, where 3 was the most common number, as seen in figure 5.

Number of persons working in software development	Quantity	Interviewees
1-5	6	[S-CM-3][S-Both-2][M-COTS-1] [M-CM-1][M-CM-2][M-CM-3]
6-10	4	[S-CM-1][S-CM-2][S-Both-1] [M-COTS-2]
11-15	2	[S-COTS-1][S-Both-3]

Table 5: Number of persons working in software development

There was some trouble getting answers to the financial questions because either the interviewee did not know the answer, or post-interview requests for answers were not replied to. This does not affect the result of the survey.

All but three companies answered the question about the annual turnover of the companies where the most common turnover is from €588 thousand - €1 million⁹ as shown in table 6.

Annual Turnover (approx)	Quantity	Interviewees
No answer	3	[S-Both-3][M-CM-1][M-CM-3]
€0-588 thousand	3	[S-COTS-1][M-COTS-1][M-CM-2]
€588-1,175 thousand	4	[S-CM-2][S-Both-1][S-Both-2][M-COTS-2]
>€1,175 thousand	2	[S-CM-1][S-CM-3]
Total	12	

Table 6: Annual Turnover

⁹As of April 2010 €1 = 170 ISK (approximately)

The most common annual cost of software is < €5,800, as seen in table 7. In relation to the annual software cost, 4 interviewees¹⁰ said that they had contracts (like MSDN or Bizspark) with Microsoft or other companies which resulted in lower software cost. The interviewee which pays the most annual cost for its software said that the company could lower its cost extensively by changing its database from Microsoft SQL server to MySQL.

The ratio between the software cost and total operational cost was less than 1% in one instance, all others interviewees were not able to answer this question and commented that those numbers were too low to focus on them.

Annual Cost of Software (approx)	Quantity	Interviewees
No answer	3	[S-Both-3][M-COTS-1][M-CM-1]
< €5,800	5	[S-Both-1][S-Both-2][M-COTS-2][M-CM-2][M-CM-3]
€5,800-29,400	3	[S-COTS-1][S-CM-2][S-CM-3]
> €29,400	1	[S-CM-1]
Total	12	

Table 7: Annual Cost of Software

All but one company used some kind of software development methodology when developing software. Most common use was Scrum Agile, in 7 occasions, or some kind of variation of Agile. Also, some companies used mixed methodologies depending on which projects they were working on.

Software Development Methodology	Quantity	Interviewees
Scrum Agile, adjusted to company's needs	7	[S-COTS-1][S-CM-2][S-CM-3][S-Both-2][S-Both-3][M-COTS-1][M-CM-2]
Extreme Programming	1	[S-CM-1]
Mixed methodology, Scrum, Kanban, TDD, Continuous Integration and others	3	[S-Both-1][M-COTS-2][M-CM-3]
Nothing	1	[M-CM-1]
Total	12	

Table 8: Methodology

The companies are developing all kinds of products, for example business software, i.e. salary system, accounting software, human resources management tools, purchasing system, job registration, time recording system, retail software, dispatch system, and software for issuing invoices. The companies also develop systems for pension funds, trade unions, the public health care system, the school community, web content management systems, all kinds of web-oriented systems, and last but not least wishes from customers.

The companies target market range from individuals to institutions, and small and medium-sized enterprises.

When asked about licenses on the companies products, 9 companies said that they sell subscription to their centrally run software and 3 companies either sell their software to customers and then customers can buy an upgrade or sign a service contract.

¹⁰[S-Both-1], [S-Both-3], [M-COTS-2] and [M-CM-2]

Seven companies have Microsoft Windows running on their servers, one company runs on Linux and the remaining four companies run on combinations of different servers, as shown in table 9.

Servers Operating System	Quantity	Interviewees
Windows	7	[S-COTS-1][S-CM-2][S-CM-3][S-Both-3] [M-CM-1][M-CM-2][M-CM-3]
Linux	1	[S-CM-1]
Combinations of Windows, Linux and others	4	[S-Both-1][S-Both-2][M-COTS-1][M-COTS-2]
Total	12	

Table 9: *Operating Systems on the Servers*

5 Results

The results presented in this chapter are based on 12 individual responses from the structured interviews. It contains answers, based on the focus of the questions in the questionnaire, i.e. the general questions, the questions on FOSS and the questions on proprietary software, respectively. But most answers to the general questions asked can be found either in the chapter "analysis on the interviewees background" or in this chapter.

5.1 Defining FOSS

Asking if the interviewee knew something about Free and Open Source Software, all 12 respondents were in some way familiar with the concept, ranging from comments like "*a little*", "*something*" to "*very much*".

Then they were asked to define FOSS and the answers were categorized by their contents as seen in table 10. Most of the interviewees defined it as a software one can use without paying any price or license fee.

5.2 The use of FOSS

There are 7 companies out of 12 using FOSS in their software development and the distribution is shown in table 11. The table shows that all but one company are small sized and the use of FOSS is almost equally distributed by the type of software they develop.

The 5 companies that are not using FOSS in their software development had a few comments on the subject. One interviewee¹¹ said that they had plans of using more FOSS but they had not gone into that process yet. Also, in 2 instances¹², companies were using one FOSS mind-mapping tool (one used XMap but the other could not remember its name) that was used to capture some ideas but not exactly in software development, by their remarks. Furthermore, they were using those tools because they worked fine and were free of charge. One company¹³ had founded its business only

¹¹[M-CM-1]

¹²[S-Both-3] and [M-CM-3]

¹³[M-COTS-1]

Defining Free and Open Source Software (FOSS)	Quantity	Interviewees
Software that you can use without paying any cost or licensing fee.	3	[S-Both-3][M-CM-1][M-CM-2]
Everybody can look at the source code.	2	[S-CM-1][M-CM-3]
Access to source code and is without licensing fee.	1	[M-COTS-1]
Outgrowth from GNU-GPL licensing.	1	[S-Both-2]
Software access via the internet, source code available <i>and you can add to the code.</i>	2	[S-COTS-1][S-CM-2]
Software access via the internet, source code available and you can add to the code, <i>many people can participate in the development but no one owns the software. Anyone can add their code to the directory.</i>	1	[S-CM-3]
Software access via the internet, source code available and you can add to the code, many people can participate in the development but no one owns the software. Anyone can add their code to the directory. <i>Freedom to use if you follow certain terms.</i>	1	[S-Both-1]
Open Source Software can be used given certain terms, you can change the source for own use and it is not a condition that everyone else can use the changes. Software can be open even though you pay for it. Free Software is "free as in freedom, not free as in free beer".	1	[M-COTS-2]
Total	12	

Table 10: Defining FOSS

	COTS	CM	Both	Total
Small	S-COTS-1	S-CM-1 S-CM-2 S-CM-3	S-Both-1 S-Both-2	6
Micro	M-COTS-2			1
Total	2	3	2	7

Table 11: Companies using FOSS

using Microsoft solutions but later the company established a subsidiary built exclusively on FOSS solutions to approach a different target market.

5.2.1 FOSS development tools

The companies are using many variations of FOSS, and the software listed in table 12 (in alphabetical order) are the ones that the interviewees said they were using, even though some of them do not fall under the definition. Some companies are using a lot of different components of FOSS and could not name them all, therefore this list only an indicator of what tools are being used, based on software the interviewees remembered. Also, some companies named, for instance, their operating system or web browsers as a developing tool while others did not. As a result, the quantity of developing tools used by each company are not absolute.

Developing Tools used	Quantity	Interviewee
Access script (part is open)	1	[S-COTS-1]
Apache	2	[S-Both-1][M-COTS-2]
Business Intelligence and Reporting Tools (BIRT)	1	[M-COTS-2]
Chrome	1	[S-CM-3]
Confluence Wiki from Atlassian	1	[M-COTS-2]
CVS	2	[S-CM-1][S-Both-1]
Eclipse	3	[S-CM-1][S-Both-1][M-COTS-2]
Firebird database	1	[S-Both-2]
Firefox	2	[S-CM-3][S-Both-1]
FreePascal	1	[S-Both-2]
Gimp	2	[S-CM-2][S-Both-2]
GIT	1	[S-Both-1]
Java	2	[S-COTS-1][M-COTS-2]
Jira from Atlassian	1	[M-COTS-2]
Linux	3	[S-COTS-1][S-Both-1][M-COTS-2]
Maven	1	[S-Both-1]
Meeting Room Booking System (MRBS)	1	[S-Both-2]
MySQL	3	[S-COTS-1][S-Both-1][M-COTS-2]
Netbeans	1	[S-Both-1]
OpenOffice.org	3	[S-CM-1][S-Both-1][S-Both-2]
Opera	1	[S-Both-1]
PHP	1	[S-COTS-1]
PSPad	1	[S-Both-2]
Python	2	[S-Both-2][M-COTS-2]
Safari	1	[S-Both-1]
Scite	1	[S-Both-1]
ScrumWorks	1	[S-CM-2]
Sonar to analyse the build	1	[S-Both-1]
StarOffice	1	[S-CM-2]
Subversion	3	[S-CM-3][S-Both-1][M-COTS-2]
Textpad	1	[S-Both-2]
Tomcat	2	[S-Both-1][M-COTS-2]
Trac	1	[S-Both-1]
Various components from different parties	2	[S-COTS-1][S-Both-1]
View Your Mind (VYM)	1	[S-Both-2]

Table 12: FOSS development tools

Table 13 lists the quantity¹⁴ of FOSS used by each company. As previously said, the quantity of developing tools used by each company are not absolute and therefore only a reference.

Company	Quantity
[S-CM-1]	3
[S-CM-2]	3
[S-CM-3]	3
[S-Both-1]	18+
[S-Both-2]	9
[S-COTS-1]	6+
[M-COTS-2]	11

Table 13: *Quantity of FOSS used by each company*

5.2.2 Reasons for choosing FOSS

When asked how the companies made the choice of using FOSS, it was usually someone inside the company who knew something about FOSS but choices were also based on software comparisons and recommendations from others. This is presented in table 14. The main factors that ruled their decision to use FOSS were in fact that the software is *known*, it is *free of charge or costs less*, it *does what it is supposed to do*, the software is *widely spread*, it is *secure* and finally that it *integrates sufficiently with other software, operating systems and browsers*.

The reasons the companies choose FOSS	Quantity	Interviewees
Comparisons and recommendations	3	[S-COTS-1][S-Both-1][S-CM-3]
Someone inside the company knew about FOSS	3	[S-CM-1][S-CM-2][M-COTS-2]
It was free of charge and it works	1	[S-Both-2]
Total	7	

Table 14: *Reasons for choosing FOSS*

When asked if the companies made some changes to the source code of the FOSS they used, 4 out of 7 companies replied positively that they made some changes to FOSS and all of them submitted the changes back to the community.

¹⁴The "+" sign refers to "various other components"

5.2.3 Quality factors

The ISO 9126 standard (ISO, 2001) is used to rate a software quality and it is defined by six characteristics which are, *Reliability*, *Portability*, *Usability*, *Efficiency*, *Maintainability* and *Functionality*. Each characteristics is explained in table 15.

Characteristics	Explanations
Reliability	The efficiency of a software to manage its level of performance under certain conditions.
Portability	The competency of a software to be transferred between environments.
Usability	The efficiency of a software to be usable by an individual estimate.
Efficiency	The level of performance of a software plus the resources used under certain conditions.
Maintainability	The efficiency of a software to make specified modifications.
Functionality	The efficiency of a software to endure a set of functions under certain conditions.

Table 15: *ISO 9126 (ISO, 2001)*

The interviewees were asked to rate the quality of Free and Open Source software products by the ISO 9126 standard. The result is displayed in table 16 where the *Reliability* of software scored highest with an average of 7 out of 7, and the lowest score was *Portability* and *Maintainability* with an average of 5 out of 7.

Average Quality factors of FOSS	Rating
Reliability	7
Portability	5
Usability	6
Efficiency	6
Maintainability	5
Functionality	6

Table 16: *Average rating of the importance of quality factors of FOSS*

Furthermore, in figure 1, the average rating of the quality factors are sorted by the *types of software developed*. It shows that the COTS software developers value *Portability* and *Maintainability* considerable lower than the other developers.

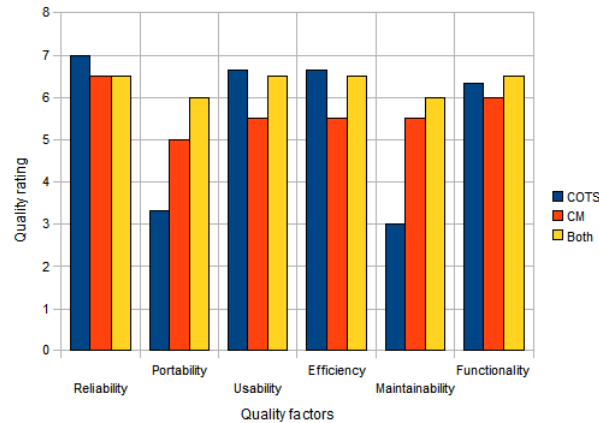


Figure 1: Average rating of the importance of quality factors of FOSS, sorted by types of software

The interviewees were asked if the company had enquired about their employees' views towards using FOSS. In all cases, an informal survey was conducted where the will of everyone was either known, or meetings were held where everyone could express their mind. Often, programmers recommended the use of some software, and in small environments they could easily express their thoughts about the quality (or lack of quality) of some software and decide from there if it was usable or not. The final decision to choose FOSS was taken by the development director on 2 occasions¹⁵, by the programmers on 3 occasions¹⁶, and by the executives on 2 occasions¹⁷.

FOSS did not meet all of the requirements of software that the companies had. Some commented that it did indeed fulfil its purpose very well and other comments were that no software could fulfil all of the needs of a company, i.e. there is always something missing.

5.2.4 Advantages and disadvantages of FOSS

The interviewees were asked to name the advantages of FOSS they were using and their responses can be viewed in table 17. The most common advantage mentioned is that it is *free of charge* and the *ability to download a piece of software and leave it on a computer without breaking any licensing agreements*.

¹⁵[S-CM-2][S-CM-3]

¹⁶[S-CM-1][S-Both-1][M-COTS-2]

¹⁷[S-COTS-1][S-Both-2]

Advantages of FOSS	Quantity	Interviewees
Free of charge	3	[S-COTS-1][S-CM-2][M-COTS-2]
You can download and leave it on other computers without violating licensing agreement	3	[S-CM-2][S-Both-2][M-COTS-2]
Cheaper than proprietary software	2	[S-COTS-1][S-Both-1]
It is widespread and many people are using it	2	[S-CM-3][M-COTS-2]
You can study and learn from the source code	2	[S-COTS-1][S-Both-1]
Open source communities are good to acquire knowledge and information and if you run into trouble you can usually find some Forums to get answers	2	[S-COTS-1][M-COTS-2]
It is there, it works, logs all changes and traces them	2	[S-CM-1][S-CM-2]
A lot of supply of software	1	[S-Both-1]
Do not have to pass the cost of software on to the product	1	[S-Both-2]
Employers who have interest can download and try out software without getting permission to buy something, it is good working knowledge	1	[M-COTS-2]
Flexibility to make changes and to add functionality	1	[S-Both-1]
If an open source project dies then you have the source code	1	[S-Both-1]
Non-stop development	1	[S-CM-3]
Open source code	1	[S-CM-3]
Open standards	1	[M-COTS-2]
Portability	1	[S-Both-1]
Security glitches are quickly detected and fixed	1	[M-COTS-2]
You can download a software and try it and throw it if you do not like it without being vendor locked	1	[M-COTS-2]

Table 17: Advantages of FOSS

As well as naming the advantages of the FOSS they used, the interviewees were also asked to name the disadvantages, which are listed in table 18. The only disadvantage named more than once is the less user-friendly interface of FOSS.

Disadvantages of FOSS:	Quantity	Interviewees
Interface is more user-friendly with proprietary software	2	[S-CM-1][M-COTS-2]
Because you do not pay for the software you cannot get immediate service if you run into trouble. Maybe you have to wait for days or weeks before the open source community picks it up	1	[S-COTS-1]
Different formats in text documents, there is a lack of some kind of filters to read the newest text formats. It disturbs communications outside the company	1	[S-Both-2]
Fewer people know FOSS, there is more cost and trouble teaching people	1	[S-CM-2]
Lack of service	1	[M-COTS-2]
Missing features or functionality	1	[S-Both-1]
Open source projects are released more often but sometimes before it is ready to be released	1	[S-COTS-1]
Some open source projects die or someone buys them	1	[S-Both-1]
Sometimes it does not fulfil our demands	1	[S-Both-1]
There is a clearer vision with proprietary software	1	[M-COTS-2]
When using software from, for example Microsoft, it intertwines it software well together. There is a lack of that in FOSS	1	[S-CM-3]
You cannot hold someone responsible if the system does not work	1	[M-COTS-2]
You have to acquire certain knowledge in open source software	1	[M-COTS-2]

Table 18: Disadvantages of FOSS

5.2.5 Security of FOSS

All the interviewees thought FOSS to be secure, with comments like "Yes, I think it is okay", "Yes, what we are using is secure" and "Yes, it is secure". Furthermore, they were asked to clarify their answer in more detail and the results are shown in table 19. The most common response was that it is more secure if a stable and widespread release is used.

Reasons	Quantity	Interviewees
It is more secure to use a stable and widespread release.	4	[S-COTS-1][S-CM-2][S-Both-2][M-COTS-2]
There is FOSS that is not secure, for example, Joomla has a lot of security glitches.	2	[S-Both-1][M-COTS-2]
This is a survival of the fittest. If the software is not secure it will die else it gets distributed and becomes safer and safer, more mature.	2	[S-CM-3][M-COTS-2]
If the source code is open and many are working on it, a security glitch is easily detected and it does not go unattended for many years.	2	[S-CM-3][S-Both-1]
FOSS is more secure than proprietary software	1	[S-Both-2]
If big companies or organizations, like NASA, use FOSS then it must be secure, like for example the Wiki systems.	1	[S-COTS-1]
It is easy to find security glitches in web application. Open source demands certain methods of work that makes it operational secure.	1	[S-Both-1]
Not one company owns the code and fixes all the errors. Company can go bankrupt.	1	[S-CM-3]
Our experience is that proprietary software has not better service because they do not necessarily fix software problems if it does not affect many.	1	[S-Both-2]
Software is more secure if it is bought from one of the big companies	1	[S-CM-2]
The procedure of filling the security glitches is very active within the FOSS community. If a security glitch is detected, someone posts the problem and/or patches it.	1	[S-Both-1]
The software is always alive while he returns what he is supposed to do, then he does not die.	1	[S-CM-3]

Table 19: Why is FOSS secure (or not)

The final question on FOSS was whether the companies had ceased using some FOSS, and if so, what software and the reason for it. Additionally they were asked what software replaced the FOSS product.

The result shows that 3 companies¹⁸ out of 7 had not ceased using any FOSS. The remaining 4 companies¹⁹ had all ceased using some FOSS and they named a few reasons for it.

One company²⁰ is always trying out new software and un-installs it if it does not meet their requirements. They ceased using an application server named *Resin* from *Caucho* because it did not meet their requirements, but they were running it parallel with *Tomcat* which they continued using.

Another company²¹ installs some software for certain projects and then un-installed them if they did not fulfill its purpose. For example, they ceased using *RedHat* server and replaced it with *Mandrake* which was then replaced with *Gentoo* and finally they installed *Ubuntu* which they think is very stable and are currently satisfied with it.

¹⁸[S-CM-1][S-CM-2][S-CM-3]

¹⁹[S-COTS-1][S-Both-1][S-Both-2][M-COTS-2]

²⁰[S-Both-1]

²¹[S-Both-2]

The third company²² ceased its plan to use some *Wiki systems* a few years ago because it was too much of a hassle implementing them on Microsoft Windows system and no-one in the company had the knowledge to set it up. Later when they got themselves a Linux they bought *Confluence Wiki from Atlassian* because it was highly recommended, and they are very satisfied with that choice. They commented that most FOSS seems to be for Linux systems, or at least the best FOSS. They further stated that FOSS seems to be linked together.

Fourth and finally, a company²³ ceased using *Joomla* because they thought it was not cool enough for their image and now they use *plain HTML* but they plan to port it to *Ruby on Rails*.

5.3 The use of proprietary software

All of the companies, 12 out of 12, are using Proprietary software in their software development. Some interviewees commented on not being satisfied with Internet Explorer, and there were also comments that if using Microsoft software it usually meant that more Microsoft software is needed because they are so interrelated.

5.3.1 Proprietary software development tools

The companies are using many variations of proprietary software, and are listed in table 20 (in alphabetical order) are the ones that the interviewees said and remembered was the proprietary software they used. This list is only an indication of what tools are being used, because some companies named, for instance, their programming language, operating system or web browsers as a developing tool while others did not. As a result, the quantity of developing tools used by each company is not absolute and should only serve as a reference.

²²[S-COTS-1]

²³[M-COTS-2]

Developing tools used	Quantity	Interviewees
Adobe	1	[S-CM-2]
Adobe Creative Suit	1	[S-COTS-1]
Adobe Flash	1	[S-Both-1]
Adobe Photoshop	1	[S-Both-1]
Air Studio	1	[M-CM-3]
Atmel	1	[M-CM-3]
AVS Studio from Atmel	1	[M-CM-3]
Balsamiq for making mockup	1	[S-Both-1]
Borland C++ builder	1	[S-Both-3]
C#	1	[S-Both-3]
C++ (eitthvað gamalt kerfi)	1	[S-Both-3]
Codec	1	[S-COTS-1]
Confluence Wiki from Atlassian	4	[S-COTS-1][S-CM-1][S-CM-3][S-Both-1]
Cubase	1	[S-COTS-1]
Delphi	1	[S-Both-2]
Genero from 4Js	1	[S-CM-1]
Google Web Tool Kit	1	[S-CM-1]
Informix database from IBM	1	[S-CM-1]
IntelliJ	1	[S-COTS-1]
JIRA from Atlassian	3	[S-COTS-1][S-CM-3][S-Both-1]
Mac Os X systems	1	[S-Both-1]
Microsoft browser, Internet Explorer	1	[S-Both-1]
Microsoft Office	10	[S-COTS-1][S-CM-2][S-CM-3][S-Both-1][S-Both-3][M-COTS-1][M-COTS-2][M-CM-1][M-CM-2][M-CM-3]
Microsoft operating system	4	[S-CM-3][S-Both-1][M-COTS-1][M-COTS-2]
Microsoft SourceSafe	1	[M-CM-2]
Microsoft SQL	5	[S-CM-3][S-Both-2][M-COTS-1][M-COTS-2][M-CM-2]
Microsoft Visio	3	[S-CM-2][S-CM-3][M-CM-2]
Microsoft Visual Basic for Application (VBA)	2	[S-Both-3][M-CM-1]
Microsoft Visual Studio	9	[S-COTS-1][S-CM-2][S-CM-3][S-Both-2][S-Both-3][M-COTS-1][M-COTS-2][M-CM-2][M-CM-3]
.NET	3	[S-CM-3][M-COTS-1][M-CM-2]
Orcad	1	[M-CM-3]
Paint	1	[S-CM-2]
Rstudio	1	[M-CM-3]
SendStudio	1	[S-COTS-1]
Storefront	1	[M-COTS-1]
TeamCity from NetBeans	1	[S-COTS-1]
Telerik	1	[M-CM-2]
Windows Communication Foundation (WCF)	1	[M-COTS-2]

Table 20: *List of Proprietary Software*

Table 21 lists the quantity of proprietary software used by each company. As previously stated, the quantity of developing tools used by each company is not absolute and is therefore only a reference.

Companies	Quantity
[S-COTS-1]	10
[S-CM-1]	4
[S-CM-2]	5
[S-CM-3]	8
[S-Both-1]	9
[S-Both-2]	3
[S-Both-3]	6
[M-COTS-1]	6
[M-COTS-2]	5
[M-CM-1]	2
[M-CM-2]	7
[M-CM-3]	7

Table 21: *Quantity of proprietary software used by each company*

5.3.2 Reasons for choosing proprietary software

When asked how the company made the choice of using proprietary software the responses were quite different and therefore the answers listed in table 22 show the response of each interviewee.

Reasons for choosing Proprietary software	Interviewees
Recommendation from those who have more experience in software development.	[S-COTS-1]
The projects were supported by certain software therefore we were bound to use it. But we chose the Google environment because we thought it was the best.	[S-CM-1]
Word of mouth and the knowledge inside company	[S-CM-2]
There was a lot of hype around .NET when it first came on market. We rewrote our entire system into .NET. Personally I do not like Microsoft products nor the company but no-one can conceive of changing technology. It works well and our sale-pipes are somewhat based on it.	[S-CM-3]
We try out many systems and we use what is convenient at each time. The know-how of our employees plays a critical role and in the case of using Photoshop, there is only one Photoshop, you have no choice.	[S-Both-1]
Ease of use, does it work. Easier to sell products that are based on more known name like Microsoft because middle-managers want to buy products that are known and secure. They won't get screwed for taking risk or do something bold.	[S-Both-2]
Our products are based on Microsoft solutions and therefore we have to be strong in the Windows platform and these are the best development tools for it.	[S-Both-3]
The founders graduated from school where the focus was on Microsoft environment therefore it was the only software they were familiar with.	[M-COTS-1]
We just chose what we thought was powerful and effective.	[M-COTS-2]
Localized foreign software	[M-CM-1]
This is the environment we come from and where our knowledge lies.	[M-CM-2]
It depends on the CPU we use, the software comes with it. We choose our CPU depending on what service accompanies it.	[M-CM-3]

Table 22: *Reasons for choosing proprietary software*

The main factors that ruled the companies' decision to use proprietary software were *that it fulfilled their demands, software quality and the software knowledge inside the company*. More detailed responses is listed in table 23.

Ruling factors why proprietary software chosen	Interviewees
That it fulfilled our demands, not to expensive and portable between systems.	[S-COTS-1]
Quality, distribution and future security.	[S-CM-1]
Mainly compatibility.	[S-CM-2]
Delphi was phasing out and .NET was a powerful tool.	[S-CM-3]
Usability, not chosen by its price.	[S-Both-1]
Ease of use, does it work.	[S-Both-2]
First and foremost to fulfil our users demands as well as we could.	[S-Both-3]
Personnels know-how.	[M-COTS-1]
Because Visual Studio is a very powerful development tool and we are developing in .NET.	[M-COTS-2]
It was given to use Proprietary software as a development tool because the system we were localizing was developed in that environment.	[M-CM-1]
Personnels know-how and widely used.	[M-CM-2]
Price, quality and the community.	[M-CM-3]

Table 23: *Ruling factors why proprietary software chosen*

5.3.3 Quality factors

The interviewees were asked to rate the quality of proprietary software products by the ISO 9126 standard²⁴. The result is displayed in table 24 where the *Reliability* of software scored highest with an average of 7 out of 7, and the lowest score was *Portability* with an average of 5 out of 7.

Average Quality factors of Proprietary Software	Rating
Reliability	7
Portability	5
Usability	6
Efficiency	6
Maintainability	6
Functionality	6

Table 24: *Average rating of the importance of quality factors of Proprietary Software*

²⁴The ISO 9126 table is shown in table 15 on page 19

Furthermore, in figure 2, the average rating of the quality factors are sorted by the *types of software developed*. It shows that the *Portability* factor scores lowest with all the companies.

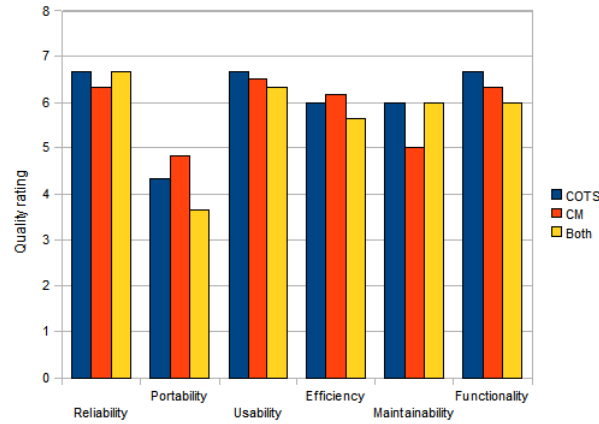


Figure 2: Average rating of the importance of quality factors of proprietary software, sorted by types of software

The interviewees were asked if the company had enquired about their employees' view towards using proprietary software. There were no formal inquiries but with 7 companies there were some meetings where everyone could express their mind. In 5 cases there were no formal meetings. The final decision is taken by the programmers and other employees²⁵ in 5 companies, and by the development directors and executives in 7 companies²⁶.

The Proprietary Software met all the requirements of the software the companies had in 7 out of 12 occasions²⁷. Some commented that "*It fulfilled its expected task*"²⁸ and that "*It fulfilled its expectations but not all needs, no software does*"²⁹.

²⁵ [S-CM-1][S-Both-1][M-COTS-2][M-CM-2][M-CM-3]

²⁶ [S-COTS-1][S-CM-2][S-CM-3][S-Both-2][S-Both-3][M-COTS-1][M-CM-1]

²⁷ [S-COTS-1][S-CM-3][S-Both-1][S-Both-3][M-COTS-1][M-CM-2][M-CM-3]

²⁸ [S-Both-1]

²⁹ [S-COTS-1]

5.3.4 Advantages and disadvantages of proprietary software

The interviewees were asked to name the main advantages of proprietary software they were using, and their responses are shown in table 25. The most common advantage mentioned is the know-how of the employees.

Advantages of proprietary software	Quantity	Interviewee
The knowledge of the use of it is within the companies, students graduate with knowledge on the software so there is less to teach	3	[S-CM-2][M-COTS-1][M-CM-2]
Good interface	2	[M-COTS-2][M-CM-2]
It is widespread	2	[M-CM-1][M-CM-3]
Microsoft is big and everybody knows them and most people are using their software	2	[S-CM-3][M-COTS-2]
Microsoft software is very mature, good and powerful	2	[S-CM-3][M-COTS-2]
Service provided	2	[S-COTS-1][M-COTS-2]
There are more functions than you will ever need	2	[S-CM-3][M-CM-2]
Clearer vision with proprietary companies.	1	[M-COTS-2]
Easy to exchange data outside company because everybody got Microsoft	1	[M-CM-3]
If there is an update on the software, all the documentation is ready	1	[S-COTS-1]
It can be run widely on many different computers	1	[M-CM-1]
It works	1	[S-Both-2]
Microsoft has a big software development community which they take a good care of	1	[M-COTS-2]
Powerful marketing at Microsoft in Iceland, conferences are frequently held and that makes it more fun	1	[M-COTS-2]
Quality	1	[S-Both-3]
The functionality is the main thing. Tools that work well	1	[S-CM-1]
The proprietary companies foresee the total solution for their users	1	[M-COTS-2]
We get proprietary software for things we do not get FOSS	1	[S-Both-1]
You can scold someone in the service department if something goes wrong	1	[S-COTS-1]
You can use the proprietary software from the idea until the product is ready	1	[S-CM-2]

Table 25: *Advantages of proprietary software*

As well as naming the advantages of the proprietary software they used, the interviewees were also asked to name the disadvantage, which are listed in table 26. The most common disadvantaged mentioned is the price or high license fee of proprietary software.

Disadvantages of proprietary software	Quantity	Interviewee
The price or high licence fee	7	[S-COTS-1][S-CM-1][S-Both-1][S-Both-2] [M-COTS-1][M-COTS-2][M-CM-3]
If some-things get broken or if something does not work correctly, you cannot fix it by yourself	2	[S-Both-1][S-Both-2]
You have to learn to use many different tools with more features than you need and sometimes hard to learn	2	[S-CM-2][S-CM-3]
Adding more features to sell the latest version	1	[S-CM-3]
Closed standard	1	[M-COTS-2]
Interface is not nice	1	[S-CM-3]
It can be complicated to realise what you may or may not do	1	[M-COTS-1]
It is disadvantage to have to struggle through all the trouble of licensing and to renew them etc.	1	[M-COTS-2]
Microsoft is not very good in making software, often there are bugs	1	[S-CM-3]
Other proprietary software companies might not have as accessible information as Microsoft	1	[M-COTS-2]
Portability can be a hassle	1	[S-Both-1]
Rapid development in this sector and sometimes it is hard to follow up	1	[S-Both-3]
Sometime you want to have something done different	1	[S-Both-2]
Sometimes some simple things are not simple to do like changing primary keys in MS SQL, you have to go around thing so it will work	1	[S-Both-2]
Sometimes there are long periods between new releases	1	[S-COTS-1]
SourceSafe is not the best version control system	1	[M-CM-2]
That Access is not SQL database	1	[M-CM-1]
Vendor locking	1	[M-COTS-2]
We are using development tool that not many know and therefore it is difficult to get programmers who have used it before	1	[S-CM-1]

Table 26: *Disadvantages of proprietary software*

5.3.5 Security of proprietary software

All the interviewees thought Proprietary Software was sufficiently safe. One interviewee said "*It is medium safe*"³⁰, another said that "*It was not safer than other software*"³¹. About 10 interviewees thought it safe and there were comments like "*Yes, but it depends on the manufacturer and the maturity of the software*"³², and "*Yes, the ones that we are using are safe*"³³.

Furthermore, the interviewees were asked to clarify their answer in more detail and the results are shown in table 27. The most common response was that they were dependent on the service provided when it comes to fixing errors in the proprietary software and also that the Microsoft operating systems are fairly safe today.

Two companies³⁴ made a contract with few of their customers, which stated that the customers would gain access to their source code if they would go bankrupt. The customers wanted some guarantee to be able to access the source code to be less vulnerable if the company would run out of business.

³⁰[M-CM-3]

³¹[M-CM-1]

³²[M-COTS-2]

³³[S-Both-1]

³⁴[S-Both-1][S-CM-3]

Reasons why proprietary software is secure (or not)	Quantity	Interviewee
Closed source can have errors that their customers cannot fix by themselves and they are helpless until the company sends an upgrade.	2	[S-Both-1][S-Both-2]
The Microsoft operating systems are fairly safe.	2	[S-Both-1][M-COTS-1]
All the software that we are using is well known and has been under development for a long time, therefore problems are very known.	1	[S-CM-2]
Closed source can have security glitches, which not many know of, for a long time and crackers can easily find a way through.	1	[S-Both-1]
Components from 3rd party is not always good.	1	[S-Both-3]
Flash and Adobe pdf are the biggest security glitches on the Internet today.	1	[S-Both-1]
If there is a paying customer the software companies spends time on it and releases software when they are confident about it. Else there will be a lot of unsatisfied customers.	1	[S-COTS-1]
It has faults, it can crash and it can be buggy.	1	[M-CM-1]
It is experienced, known products and many are using it.	1	[S-CM-2]
It is not safer than FOSS.	1	[M-CM-1]
It is safe because it so much used.	1	[M-CM-2]
Microsoft has been efficient in fixing the security glitches in its software.	1	[M-COTS-1]
Microsoft products are always of good quality.	1	[S-Both-3]
Secure developing tool would be one with no memory problem because of some kernel we could not control. So I would say that propriety software is fairly safe.	1	[S-CM-3]
Secure operating system would be one with a good uptime, security measures would be enforced and it would be unbreakable. So I would say that propriety software is fairly safe.	1	[S-CM-3]
SQL is a very good database.	1	[S-CM-3]
The most popular software are the ones who get the most attacks. Not to be mainstream makes you a bit safer. Today, Windows systems are attached but not so much.	1	[S-Both-2]
The security of a software depends on the manufacturer. So you chose a software depending on his maturity and the manufacturer.	1	[M-COTS-2]
The software crashes a lot but it does not cause a risk of accident therefore you are secure of using it. But it can ruin files if it crashes.	1	[M-CM-3]
Visual Studio is fairly good but a bit buggy.	1	[S-CM-3]
We try to select systems that are safe or we can make safer.	1	[S-Both-1]
What we are using is secure but there are propriety software that are not safe.	1	[S-Both-1]
With us, Microsoft software has been faulty but other Propriety software has been stable.	1	[S-CM-1]

Table 27: *Reasons why proprietary software is secure (or not)*

The interviewees were asked if FOSS could replace some proprietary software that the company is using, and 4 out of 12 interviewees said that FOSS could replace all of their proprietary software. The results are displayed in table 28.

One interviewee had this comment "*But it could be too expensive because of the inner knowledge in the company, we would have to retrain our personnel and it would cost a lot of licenses to make the exchange*"³⁵. And he continued saying, "*We would need someone inside the company who knows how the system works and can take care of it - but I think that in a few years we will have a lot more of FOSS*". A few said "*The customers decide*"³⁶, meaning that the customers which are using Microsoft products want systems that complies with it, and therefore the companies have to use Microsoft products to fulfil their needs.

Three interviewees said they could replace their software for documentation but the only problem was that they have to be able to open all the documents the customers send to the company³⁷.

³⁵[M-COTS-1]

³⁶[S-Both-1][M-COTS-2][M-CM-3]

³⁷[S-Both-1][S-Both-2][M-CM-3]

One interviewee³⁸ said the company could change their database to a FOSS database and save a lot of money³⁹. Another said that they could change their development environment but it would require substantial effort and it would not benefit them because of their Microsoft development licenses.

Another said that they could probably replace some software as long as it was not in their final product because they sold Microsoft solutions⁴⁰.

Two interviewees said they could not change anything, the reasons were that they were in a complete Microsoft environment, one⁴¹ in an Bizspark program for start-up companies. The other, which is running its firewall on Linux and mail-scanner, said that they were developing desktop environments in Microsoft solutions and that FOSS was not good enough for their customers⁴².

Can FOSS replace some Proprietary Software	Quantity	Interviewees
It can replace all other software we use	3	[M-COTS-1][M-COTS-2][M-CM-1]
Documentation	2	[S-CM-2][M-CM-3]
Nothing	2	[S-Both-3][M-CM-2]
Database	1	[S-CM-1]
Development environment	1	[S-Both-2]
Flow chart tool	1	[S-CM-3]
Operating System	1	[S-Both-1]
Server	1	[S-COTS-1]
Total	12	

Table 28: *Can FOSS replace some proprietary software*

The final question on proprietary software was whether the companies had ceased using some proprietary software, and if so, what software and what was the reason? Furthermore, what software replaced it?

The conclusion was that 5 out of 12 companies had ceased using some proprietary software and 7 answered negatively. One interviewee⁴³, which answered negatively, said that they were saving their budget and did not buy service nor upgrade their software. Another⁴⁴ said that they prepared carefully what software they bought because they did not have a lot of money as a start-up company and three companies⁴⁵ said that they were relatively young and therefore it had not come to that.

The Proprietary Software that the companies replaced was diverse and the reasons were many. Two companies replaced software which they were phasing out, such as COBOL which they said was being replaced by Access⁴⁶, 4thDimension was replaced by Microsoft and Oracle software⁴⁷, and Delphi was replaced by .NET⁴⁸. The version control system SourceSafe from Microsoft was replaced

³⁸[S-Both-2]

³⁹[S-CM-1]

⁴⁰[S-CM-3]

⁴¹[M-CM-2]

⁴²[S-Both-3]

⁴³[M-CM-3]

⁴⁴[S-COTS-1]

⁴⁵[S-COTS-1][S-CM-1][M-CM-2]

⁴⁶[M-CM-1]

⁴⁷[S-CM-2]

⁴⁸[S-CM-3]

by some FOSS version control system⁴⁹, because they thought that it was better. One company tried many job control systems before finding JIRA from Atlassian⁵⁰. A Windows host was replaced by Linux CentOS⁵¹, IntelliJ was replaced by NetBeans⁵² and Microsoft Office was replaced by Thunderbird/KDE email because Microsoft Outlook did not support IMAP⁵³.

⁴⁹[S-CM-3]

⁵⁰[S-Both-1]

⁵¹[S-Both-1]

⁵²[S-Both-1]

⁵³[S-Both-2]

6 Analysis

This chapter analysis the results presented in the previous chapter and answers the research questions set forth in the introduction chapter.

6.1 The main reasons for Icelandic software developers to use Free and Open Source Software

There are 7 out of 12 companies using FOSS in their software development. Those 7 interviewees named 18 advantages of the FOSS they used, and if the advantages are compared to the disadvantages of proprietary software the reasons for using FOSS can be summarized as follows.

- FOSS is free of charge or cheaper than proprietary software
- FOSS can be installed on any computer without breaking any licensing issues
- FOSS supports open standards better
- FOSS provides the flexibility to make changes
- FOSS is more portable
- FOSS reduces vendor lock-in

When software is free of charge it gives the employees the ability to download, learn and study a new software, and therefore adding to their knowledge which hopefully results in more valuable and competent employees. Furthermore, the companies do not have to worry about breaking any licensing issues because the software can be freely distributed to all the computers in the company without any restraints.

FOSS supports open standards which makes it easier for various different software to be integrated together. On the other hand, proprietary software relies more on corporate standards and thus only followed by one or a few companies.

FOSS gives the users a flexibility to make changes to the software, to adjust it to the users' needs, to fix some bugs or security glitches. This can be done by the user without being dependant on the proprietary software developer to fix problems or add features.

One of the qualities of using FOSS is that it is portable between platforms which is a key issue for cost reduction in development. If software cannot be run in different environments it forces the user to use another software, with additional cost, or bind him and all other users to use the same platform.

Finally, the last reason for using FOSS is that a company does not get locked in to a specific vendor if choosing to use specific software and can easily change its mind and start using something else without additional cost.

Based on the interviews conducted in this research, the six reasons mentioned above appear to be the main reasons Icelandic software developers use Free and Open Source Software in their software development.

6.2 The main reasons for Icelandic software developers do not use FOSS

There are 5 out of 12 companies interviewed not using FOSS in their software development. Those interviewees named 6 advantages using the specific proprietary tools they used, and if the advantages are compared to the disadvantages of FOSS, the reasons for using proprietary software can be summarized as follows:

- The interface of proprietary software is better than FOSS
- It is easier to exchange data outside company because almost everybody are using Microsoft Office
- The knowledge is already inside the companies because almost everybody learn to use proprietary software in the school environment
- More function or features with proprietary software than FOSS
- More reliable quality of proprietary software⁵⁴

Interface of FOSS has for a long time been thought of as an inferior design compared to proprietary software. Raymond supports that in his book *The Cathedral and the Bazaar* when he talks about that FOSS developers need to make software more appealing to common users outside the hacker community

"If this were primarily a technical problem, the outcome would hardly be in doubt. But it isn't; it's a problem in ergonomic design and interface psychology, and hackers have historically been poor at these things. That is, while hackers can be very good at designing interfaces for other hackers, they tend to be poor at modelling the thought processes of the other 95% of the population well enough to write interfaces that J. Random End-User and his aunt Tillie will pay to buy"(Raymond, 2001, p. 190-1).

Even though some software has taken tremendous advancement for the last decade, for instance, GIMP and Ubuntu, the FOSS appears not to be placing enough effort on usability issues.

Microsoft is dominant on the market for office tools, and therefore the same document format is used by many, which is convenient, except for those who want to use other tools. Some interviewees said they used Microsoft Office to be able to receive documents from customers, even though using OpenOffice.org for other documentation.

Gathering knowledge can be expensive, and therefore it is valuable for companies to hire people who possess the knowledge to use specific tools. On the other hand, when students are graduating and establish their own companies they most probably continue using the tools they are accustomed to. It becomes a circular environment where everybody uses proprietary software.

There are often more functions or features implemented in proprietary software than FOSS. But as some interviewees mentioned it can also add some complications to have too many features, and also many features are never used.

The interviewees thought proprietary software to be more reliable in the sense that it is serviceable and therefore someone to scold if the software is not functioning. As for quality, Microsoft software is mature and intertwines its software well together.

⁵⁴Quality is subjective where different people experience the quality of the same software very differently (Wikipedia, 2010c).

Common advantages with proprietary software and FOSS software is that they are widespread, many people are using it and it works.

The reasons mentioned above are what Icelandic software developers claim to be the main reasons why they do not use Free and Open Source Software in their software development.

6.3 The difference in the use of FOSS between those who develop CM software vs. those who develop COTS software

There is a huge difference in the quantity used by the type of software developed as shown in table 29. The companies who develop CM software seem to use much less of FOSS than the other types of developers.

	FOSS quantity used	Number of companies using FOSS	Number of companies
COTS	17+	2	3
CM	9	3	6
BOTH	27+	2	3
Total	53+	7	12

Table 29: *Quantity of FOSS used by the type of software developed*

Those 9 software products used to make CM software are 2 browsers, 2 office bundles, 2 version control systems, and each of the following: development environment, image manipulation program and a project management tool.

The average use of FOSS is 3 tools for developing CM software compared to 8+ when developing COTS software. Even though the sample is not large it gives a small insight into the use of FOSS tools.

	Proprietary quantity used	Number of companies using Proprietary software	Number of companies
COTS	33	3	3
CM	18	6	6
BOTH	21	3	3
Total	72	12	12

Table 30: *Quantity of proprietary software used by the type of software developed*

Again, the difference is substantial between the quantity used by the type of software developed as shown in table 30. The average use of proprietary software is again 3 tools for developing CM software compared to 11 when developing COTS software.

When comparing table 29 to table 30 it abolishes the assumption set forth in the introduction chapter which stated that when developing CM software companies want to use generally accepted software of good quality, according to their customers requirements and broad customers base. From what has been said it does not make difference whether CM software developers use FOSS or proprietary software in software development. However there seems to be a difference in the amount of software products used in general by COTS and CM software developers.

7 Conclusion

According to the results, more small-sized companies are using FOSS than micro-sized companies which is a very surprising conclusion for it was assumed that micro-sized companies (usually start-up companies) had lesser funds and would therefore use more FOSS than larger companies. One explanation is that at least two micro-sized companies made a MSDN/BizSpark contract with Microsoft and are therefore using their products.

On the other hand, there is no measurable difference in the type of software developed where the division is fairly even as seen in table 31, which indicates that the type of software developed does not affect the use of FOSS.

Type developed	Number of companies
COTS	2
CM	3
Both	2

Table 31: Use of FOSS divided by the type of software developed

When conducting the interviews, it was interesting to see how many companies are using FOSS in software development with combination of proprietary software, where some of them used proprietary software where they could not find good enough FOSS. Furthermore, 4 out of the 7 companies which are using FOSS had their practitioners making changes to the software and also contributing the changes back to the community. When giving back the changes they are strengthening the FOSS community which they themselves look to when running into trouble, looking for information and possible solutions.

All of the companies that used FOSS thought it was secure. With FOSS, security glitches can be easily detected and fixed because of the source code is open. Opposed to security by obscurity, that is, when source code is closed, the software can have security glitches for years before detected, in other words, a false sense of security. The users have no means to either protect themselves by pointing the software developer to the problem or fixing it by themselves, like they would if they had access to the source code. These circumstances make the companies more dependent on the proprietary software developer to fix the problem and to provide the inevitable service. As Raymond put it in his book, "Given enough eyeballs, all bugs are shallow"(Raymond, 2001, p. 30), which means when more people are looking into the source code there is always someone who knows an answer to the problem.

It was also mentioned that if FOSS projects stopped being maintained or developed, *they still have the source code*. Therefore they are in a way more secure than when a proprietary software development company goes bankrupt or stops developing their software. Two of the companies interviewed had made some kind of an agreement with a few of their customers, if they went out of business the customers were able to access their source code. Therefore, some customers of software developers need some security of accessing the source code, if it is closed.

The interviewees said that FOSS does what it is supposed to do, it integrates sufficiently with other software and is portable between platforms. It complies with the results of the importance of quality factors of FOSS where reliability and functionality scored high, but not so much with the portability

factor which scored relatively lower than the other two factors. There is therefore not complete consistency between the portability factor assessment based on the ISO 9126 where portability was least important but still considered an advantage of FOSS.

Yet another interesting factor is the problem of receiving documents between companies and customers. Companies are bound to have Microsoft Office even though using OpenOffice.org in all other documentation inside the company because of outside communications. This might explain why open standards support was considered an advantage.

There is a general misunderstanding that if software is free-ware, i.e. can be downloaded from the internet without being paid for, it is either Free Software (as in Freedom) or Open Source Software. But to be valid as such the software has to comply with the FSF and the OSI definition. Therefore, some of the software mentioned in table 12 on page 17 is neither FS or OSS, i.e. the software for Atlassian (Confluence Wiki and JIRA), Opera, Safari, TextPad or StarOffice. Some interviewees which had the definition of FOSS fairly correct still named some proprietary software as FOSS. The reason for that misunderstanding could be that FOSS is often accessible free of charge through the internet, or the name "Free" leads to this misunderstanding, or that the advocacy of the FSF and the OSI have not generated a success in bringing the message of FOSS to the computer software development community.

A surprising result is that people who graduate in computer science do not learn more about FOSS. For instance, when conducting two interviews the interviewees said that they had an open mind towards FOSS but that they came from a school environment that focused mainly on Microsoft software so when they started their own companies they did not have the knowledge to use any other software. It would have demanded too much time from them to get to know another system so they used software that they already knew. Today they said they have more knowledge about FOSS after tampering with it themselves. This is a particularly interesting result since the Icelandic government published its policy in December 2007 on the use of Free and Open Source Software. One aspect of the policy refers to the academic society and states the following:

"Students in Icelandic educational institutions shall be given the opportunity of learning about and using free and open-source software on a par with proprietary software" (Forsætisráðuneytið, 2007, p. 3).

The academic society should support FOSS awareness among their students and faculty because focusing on proprietary software in teaching clearly results in more use of proprietary software when the students start working, where FOSS could manage the task.

The conclusion is that FOSS and proprietary software can easily compliment each other in this rapidly changing environment, software developers should keep their mind open and select tools that fit the task. To be able to make an informed decision they have to know different tools, not just the ones from the leading software manufacturer.

We have the freedom, let us use it.

8 Future work

From the research more questions arose which can be for future work. The following examples could be interesting to look further into:

- Does the quantity of software developers affect the use of FOSS?
- Is the software development cost lesser in software development than the cost for general users?
- Licensing understanding of software developers
- How does the development licenses affect its users?
- Is there an increase of demand from customers of the software developers to want an access to the source code?
- How can FOSS developers gain better understanding on the importance of usability and implement it on their software?

9 Acknowledgements

I would like to thank several people who assisted me in various ways while I was working on the thesis.

The people who agreed to be interviewed for this research, for their valuable time spent on the interviews and openness during the interview.

Rósa Hansdóttir for lending me her most precious dicta-phone which proved its purpose.

Sunna G. Marteinsdóttir for reading the thesis (and actually liking it :)) and for giving good comments.

My instructor Marta, who met me once a week during the semester, informing me on how to conduct a good research as well as interviews, for reading the thesis and giving comments, and also for her many good advices.

Lastly, I would like to thank my instructor Tryggvi Björgvinsson for all his invaluable lectures on the subject, for lending me many interesting books on the subject, for his weekly trips to RU to attend meetings, for being inspiring, giving ideas, comments and advice, and time spent recursively reading the thesis.

References

- Britannica (2010a). Hacker. <http://www.britannica.com/bps/dictionary?query=hacker>. Accessed on the 14th of April 2010.
- Britannica (2010b). Proprietary. <http://www.britannica.com/bps/dictionary?query=proprietary>. Accessed on the 14th of April 2010.
- EU (2005). European union. http://ec.europa.eu/enterprise/policies/sme/facts-figures-analysis/sme-definition/index_en.htm. Accessed on the 26th of February 2010.
- Feller, J. and Fitzgerald, B. (2002). *Understanding open source software development*. Addison-Wesley Longman Publishing Co., Inc. Boston, MA, USA.
- Fogel, K. (2005). *Producing open source software: how to run a successful free software project*. O'Reilly Media.
- Forsætisráðuneytið (2007). Policy on free and open-source software: Government policy of iceland. http://eng.forsaetisraduneyti.is/media/English/Free_and_Open_Source_Software_-_Government_Policy_of_Iceland.pdf. Accessed on the 12th of May 2010.
- FS (2010). List of fs approved licenses. <http://www.gnu.org/licenses/license-list.html>. Accessed on the 24th of April 2010.
- ISO (2001). Iso/iec 9126-1:2001. http://www.iso.org/iso/iso_catalogue/catalogue_ics/catalogue_detail_ics.htm?csnumber=22749. Accessed on the 28th of April 2010.
- Laurent, A. M. S. (2004). *Understanding open source & free software licensing*. O'Reilly Media, Inc.
- Levy, S. (2001). *Hackers: Heroes of the computer revolution*. Penguin Books.
- OSI (2010a). List of osi approved licenses. <http://www.opensource.org/licenses/alphabetical>. Accessed on the 24th of April 2010.
- OSI (2010b). The open source definition. <http://www.opensource.org/docs/osd>. Accessed on the 14th of April 2010.
- PCMagazine (2010). Software developer. http://www.pcmag.com/encyclopedia_term/0,2542,t=developer&i=41187,00.asp. Accessed on the 23rd of April 2010.
- Perens, B. (1999). It's time to talk about free software again. <http://lists.debian.org/debian-devel/1999/02/msg01641.html>. Accessed on the 15th of April 2010.
- Raymond, E. S. (2001). *The Cathedral and the Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary*. O'Reilly Media, Inc.
- Salus, P. and Reed, J. (2008). *The Daemon, the Gnu, and the Penguin*. Reed Media Services.
- Scacchi, W. (2007). Free/open source software development: Recent research results and methods. *Advances in Computers: Architectural Issues*, 69:243.

- Stallman, R. et al. (2002). The free software definition. *Free Software Free Society: Selected Essays of Richard M. Stallman*, pages 41–44.
- The Free Dictionary By Farlex (2010). Non-disclosure agreement. <http://encyclopedia.thefreedictionary.com/Nondisclosure+agreements>. Accessed on the 14th of April 2010.
- Torvalds, L. and Diamond, D. (2001). *Just For Fun: The Story of An Accidental Revolutionary*. Texere Publishing.
- Wikipedia (2010a). Commercial off the shelf (cots). http://en.wikipedia.org/wiki/Commercial_off-the-shelf. Accessed on the 24th of February 2010.
- Wikipedia (2010b). Custom software developer. http://en.wikipedia.org/wiki/Custom_software_developer. Accessed on the 24th of February 2010.
- Wikipedia (2010c). Software quality. http://en.wikipedia.org/wiki/Software_quality. Accessed on the 8th of May 2010.
- Williams, S. (2002). *Free as in Freedom: Richard Stallman's Crusade For Free Software*. O'Reilly Media, Inc.

A Lessons from The Cathedral and The Bazaar

1. Every good work of software starts by scratching a developer's personal itch.
2. Good programmers know what to write. Great ones know what to rewrite (and reuse).
3. "Plan to throw one away; you will, anyhow" (Fred Brooks, *The Mythical Man-Month*, Chapter 11)
4. If you have the right attitude, interesting problems will find you.
5. When you lose interest in a program, your last duty is to hand it off to a competent successor.
6. Treating your users as co-developers is your least-hassle route to rapid code improvement and effective debugging.
7. Release early. Release often. And listen to your customers.
8. Given a large enough beta-tester and co-developer base, almost every problem will be characterized quickly and the fix obvious to someone.
9. Smart data structures and dumb code works a lot better than the other way around.
10. If you treat your beta-testers as if they're your most valuable resources, they will respond by becoming your most valuable resources.
11. The next best thing to having good ideas is recognizing good ideas from your users. Sometimes the latter is better.
12. Often, the most striking and innovative solutions come from realizing that your concept of the problem was wrong.
13. "Perfection (in design) is achieved not when there is nothing more to add, but rather when there is nothing more to take away" (Antoine de Saint-Exupère).
14. Any tool should be useful in the expected way, but a truly great tool lends itself to uses you never expected.
15. When writing gateway software of any kind, take pains to disturb the data stream as little as possible - and *never* throw away information unless the recipient forces you to!
16. When your language is nowhere near Turing-complete, syntactic sugar can be your friend.
17. A security system is only as secure as its secret. Beware of pseudo-secrets.
18. To solve an interesting problem, start by finding a problem that is interesting to you.
19. Provided the development coordinator has a communications medium at least as good as the internet, and knows how to lead without coercion, many heads are inevitably better than one.

This list is from the book *The Cathedral and the Bazaar*.

(Raymond, 2001, p. 23-54)

B The Open Source Definition

1. Free Redistribution

The license shall not restrict any party from selling or giving away the software as a component of an aggregate software distribution containing programs from several different sources. The license shall not require a royalty or other fee for such sale.

2. Source Code

The program must include source code, and must allow distribution in source code as well as compiled form. Where some form of a product is not distributed with source code, there must be a well-publicized means of obtaining the source code for no more than a reasonable reproduction cost preferably, downloading via the Internet without charge. The source code must be the preferred form in which a programmer would modify the program. Deliberately obfuscated source code is not allowed. Intermediate forms such as the output of a preprocessor or translator are not allowed.

3. Derived Works

The license must allow modifications and derived works, and must allow them to be distributed under the same terms as the license of the original software.

4. Integrity of The Author's Source Code

The license may restrict source-code from being distributed in modified form only if the license allows the distribution of "patch files" with the source code for the purpose of modifying the program at build time. The license must explicitly permit distribution of software built from modified source code. The license may require derived works to carry a different name or version number from the original software.

5. No Discrimination Against Persons or Groups

The license must not discriminate against any person or group of persons.

6. No Discrimination Against Fields of Endeavor

The license must not restrict anyone from making use of the program in a specific field of endeavor. For example, it may not restrict the program from being used in a business, or from being used for genetic research.

7. Distribution of License

The rights attached to the program must apply to all to whom the program is redistributed without the need for execution of an additional license by those parties.

8. License Must Not Be Specific to a Product

The rights attached to the program must not depend on the program's being part of a particular software distribution. If the program is extracted from that distribution and used or distributed within the terms of the program's license, all parties to whom the program is redistributed should have the same rights as those that are granted in conjunction with the original software distribution.

9. License Must Not Restrict Other Software

The license must not place restrictions on other software that is distributed along with the licensed software. For example, the license must not insist that all other programs distributed on the same medium must be open-source software.

10. License Must Be Technology-Neutral

No provision of the license may be predicated on any individual technology or style of interface.

This list is from the website of The Open Source Initiative.

(OSI, 2010b)

C Questions for the structured interview

Kynning

Góðan dag. Ég heiti Hildur Vernudóttir og er að gera lokaverkefni í tölvunarfræði í Háskólanum í Reykjavík. Í lokaverkefninu er ég að gera rannsókn á notkun Frjáls og Opins hugbúnaðar í íslenskum hugbúnaðarfyrirtækjum þar sem starfsmenn eru færri en 50. Úrtakslistinn er unninn út frá upplýsingum frá Samtökum Iðnaðarins.

Spurningarnar eru til þess fallnar að svara þremur eftirfarandi rannsóknarspurningum:

1. Að því gefnu að íslenskir hugbúnaðarframleiðendur nota Frjálsan og Opinn hugbúnað, hverjar eru þá helstu ástæður þess?
2. Séu þeir ekki að nota Frjálsan og Opinn hugbúnað, hverjar eru þá helstu ástæður þess?
3. Er munur á notkun á Frjálsum og Opnum hugbúnaði hjá þeim sem annars vegar framleiða sérsniðin hugbúnað (e.Custom Made) og hins vegar þeim sem framleiða hilluvarning (e.Commercial Off The Shelf)?

Til þess að geta svarað þessum rannsóknarspurningum þyrfti ég að fá að ræða við þann aðila sem hefur mesta þekkingu á þeim hugbúnaði sem fyrirtækið notar við þróun á vörum fyrirtækisins. Þetta er *nafnlaus rannsókn* og er *fullum trúnaði heitið*. Skráning niðurstaða verður *hvorki tengd við nafn viðmælanda né fyrirtækisins*. Ég geri ráð fyrir því að viðtalið taki ekki lengri tíma en 1 klukkustund.

Með von um góðar undirtektir,
Hildur Vernudóttir
mailto:hbv@ru.is
s.863-8757

Inngangsspurningar:

1. Þekkir þú eitthvað til Frjáls og Opins hugbúnaðar?
2. Hvernig myndir þú skilgreina Frjálsan og Opinn hugbúnað?
3. Hvað eru ca. margir að vinna við hugbúnaðarþróun hjá fyrirtækinu?
4. Hvað er fyrirtækið að velta miklum fjármunum?
5. Hver er hugbúnaðarkostnaður fyrirtækisins?
 - (a) Hvert er hlutfall(%) kostnaðar vegna hugbúnaðar af heildarrekstrarkostnaði fyrirtækisins?
6. Notar fyrirtækið eitthvert ákveðið þróunarferli?
 - (a) Ef svo er, hvernig er það?
7. Hvernig vörur er fyrirtækið að þróa/framleiða?
 - (a) Hvernig leyfi (copyright) eru á vörum fyrirtækisins?
8. Hver er markhópur þess hugbúnaðar sem fyrirtækið framleiðir?
9. Hvaða stýrikerfi keyrir á servernum ykkar?

Spurningar um Frjálsan og Opinn hugbúnað

1. Er fyrirtækið að nota Frjálsan og Opinn hugbúnað í þróunarferli á hugbúnaði/um?
 - (a) Ef já. Ef þú lítur á allan ferilinn *frá hugmynd að lokavöru*, hvaða Frjálsa og Opna hugbúnað er fyrirtækið að nota?
 - (b) Ef nei, fara í spurningar um séreignarhugbúnað.
2. Hvernig fór valið á Frjálsum og Opnum hugbúnaði fram?

3. Til hvaða þátta var litið við val á þeim búnaði (kostnað, aðgengi, kunnátti o.s.frv.)?
4. Gerið þið einhverjar breytingar á Frjálsa og Opna hugbúnaðinum?
 - (a) Ef fyrirtækið gerir einhverjar breytingar, eru þær gefnar út í samfélagið aftur?
5. Þegar valinn er hugbúnaður, hvaða **gæðapættir** finnst þér vera **mikilvægastir** á skalanum 1-7 (Setjið hring við hvert svar)?

Gæði	minnst		meðal			mest	
Áreiðanleiki (Reliability)	1	2	3	4	5	6	7
Flytjanleiki (Portability)	1	2	3	4	5	6	7
Nytsemi (Usability)	1	2	3	4	5	6	7
Skilvirkni (Efficiency)	1	2	3	4	5	6	7
Viðhaldspægni (Maintainability)	1	2	3	4	5	6	7
Virkni (Functionality)	1	2	3	4	5	6	7

6. Voru viðhorf starfsmanna könnuð?
7. Hver eða hverjir sáu um að velja Frjálsa og Opna hugbúnaðinn sem þróunartól?
8. Uppfyllir Frjálsi og Opni hugbúnaðurinn allar þarfir fyrirtækisins?
 - (a) Ef þú dregur saman notkun á þeim hugbúnaði sem þú nefndir áðan, hverjir eru þá helstu kostirnir?
 - (b) En helstu ókostirnir?
9. Finnst þér Frjáls og Opinn hugbúnaður öruggur?
 - (a) Getur þú útskýrt svarið þitt nánar?
10. Hefur fyrirtækið hætt með einhvern Frjálsan og Opinn hugbúna einhverra hluta vegna?
 - (a) Ef svo er, hvaða hugbúnaður var það og hver var ástæðan fyrir því?
 - (b) Hvaða hugbúnaður var settur upp í staðinn?

Spurningar um séreignarhugbúnað

1. Er fyrirtækið að nota séreignarhugbúnað í þróunarferli á hugbúnaði/um?
 - (a) Ef já. Ef þú lítur á allan ferilinn *frá hugmynd að lokavöru*, hvaða séreignarhugbúnað er fyrirtækið að nota?
2. Hvernig fór valið á séreignarhugbúnaði fram?
3. Til hvaða þátta var litið við val á þeim búnaði (kostnað, aðgengi, kunnátti o.s.frv.)?
4. Þegar valinn er hugbúnaður, hvaða **gæðapættir** finnst þér vera **mikilvægastir** á skalanum 1-7 (Setjið hring við hvert svar)?

Gæði	minnst		meðal			mest	
Áreiðanleiki (Reliability)	1	2	3	4	5	6	7
Flytjanleiki (Portability)	1	2	3	4	5	6	7
Nytsemi (Usability)	1	2	3	4	5	6	7
Skilvirkni (Efficiency)	1	2	3	4	5	6	7
Viðhaldspægni (Maintainability)	1	2	3	4	5	6	7
Virkni (Functionality)	1	2	3	4	5	6	7

5. Voru viðhorf starfsmanna könnuð?
6. Hver eða hverjir sáu um að velja séreignarhugbúnaðinn sem þróunartól?
7. Uppfyllir séreignarhugbúnaðurinn allar þarfir fyrirtækisins?
 - (a) Ef þú dregur saman notkun á þeim hugbúnaði sem þú nefndir áðan, hverjir eru þá helstu kostirnir?
 - (b) En helstu ókostirnir?
8. Finnst þér séreignarhugbúnaður öruggur?
 - (a) Getur þú útskýrt svarið þitt nánar?
9. Í hvaða tilvikum getur fyrirtækið séð að Frjáls og Opinn hugbúnaður geti komið í stað einhvers séreignarhugbúnaðar sem fyrirtækið notar?
10. Hefur fyrirtækið hætt með einhvern séreignarhugbúnað einhverra hluta vegna?
 - (a) Ef svo er, hvaða hugbúnaður var það og hver var ástæðan fyrir því?
 - (b) Hvaða hugbúnaður var settur upp í staðinn?

Mig langar að þakka þér kærlega fyrir þátttökuna í könnuninni. Ef í ljós kemur að ég þarf frekari upplýsingar mætti ég fá að hringja eða senda tölvupóst?



School of Computer Science
Reykjavík University
Menntavegi 1
101 Reykjavík, Iceland
Tel. +354 599 6200
Fax +354 599 6201
www.reykjavikuniversity.is