

# FREE AND OPEN SOURCE SOFTWARE HACKERS IN TURKEY

İbrahim İzlem GÖZÜKELEŞ  
izlem.gozukeles@emo.org.tr

August 11, 2006

## **Abstract**

In recent years, there has been growing interest of governments and firms for Free and Open Source Software (FOSS). However, FOSS is still a puzzlement for a wide spectrum of academic disciplines. Social scientists ask why FOSS hackers participate in FOSS if they do not get any monetary rewards. On the other hand firms ask, how a firm can earn money from FOSS. In this study, motivations of FOSS hackers and firms are taken as an interrelated phenomenon. This study discusses FOSS from the view of hackers in Turkey. In that respect, social conditions that make hackers voluntarily contribute to the FOSS projects and their relationship with the commercial world are explored.

# 1 Introduction

Availability of source code in Free/Open Source Software (FOSS<sup>1</sup>) is defined with its three essential features<sup>2</sup>.

- 1- Source code must be distributed with the software or otherwise made available for no more than the cost of distribution.
- 2- Anyone may redistribute the software for free, without royalties or licensing fees to the author.
- 3- Anyone may modify the software or derive other software from it, and then distribute the modified software under the same terms (Weber, 2004: 5).

However, FOSS is still a puzzlement for a wide spectrum of academic disciplines: software engineering, industrial engineering, economics, sociology, psychology, etc. Each academic discipline asks its own questions to understand the FOSS phenomenon and try to explain it in the boundaries of its own discipline. For example, recent literature focuses on two questions and seeks their answers in different disciplines. The first question is, 'what are the motivations of individuals for engagement in FOSS projects?' (von Krogh et al., 2003; Bitzer et al., 2004; Raymond, 1999, 2000; Zeitlyn, 2003) and the latter question is about FOSS's relationship with commercial world, 'in what level will FOSS have impact on the competitive strategy and organizations of firms?' (Baldwin and Clark, 2003; Lindman, 2004). Consequently, answer of the first question is mostly sought in the context of either social psychology or anthropology and latter is studied in the context of microeconomics. These studies give essential insights either about

---

<sup>1</sup>FOSS is used as a general term for both Free Software and Open Source Software, except where a specific distinction between the terms is explicitly made.

<sup>2</sup>Actually, there are differences between Free Software and Open Source Software. Yet, these are their common features.

the motivations behind individuals or about FOSS business models. However, they grasp the FOSS by an approach that makes parts of the whole as static and independent of one another.

Making of FOSS has been an active process which owes to agency of its members and to conditioning of technological innovations and market since 1980s. Even though needs of the market have been an important factor in emergence and development of the FOSS, FOSS firms such as Cygnus Solutions and Red Hat showed that there was something as a FOSS business model. Yet, what's more surprising was to see the world's major companies such as IBM and Oracle turn their attention to this new model as a new business opportunity (DiBona et al., 1999). First of all, it was against the conventional intellectual property law. However, it was not a marginal phenomenon. Instead, it was a widespread revolutionary way of doing and distributing software in the new environment, both technically and legally by inverting the logic of IPR (Intellectual Property Rights) (Valimaki, 2005). By natural, the questions about the hackers' motivation and firms' new business strategies were unavoidable.

For this reason, the widespread attitude in FOSS literature became investigating the answer of the motivation behind hackers' work and the viability of firms' new business models. Consequently, conceptualizations of the FOSS literature (e.g. FOSS as a social movement, FOSS Community and Gift economy) based on these attitudes. As discussed in the first part of the **second chapter**, although these studies give important insights about the hackers, they are ahistorical. More importantly, none of these theories conceptualize FOSS as a phenomenon in the informational capitalism, historically specific form of capitalism, "which information generation, processing, and transmission become the fundamental sources of productivity

and power” (Castells, 2001b: 21).

In that respect, second part of the second chapter is devoted to analysing FOSS from the perspective of hackers’ relation with their work in the informational capitalism.

Despite the popular view in media, hackers are not criminals who break computers, web sites and phone systems. On the contrary, hackers build things instead of breaking them:

Hackers built the Internet. Hackers made the Unix operating system what it is today. Hackers run Usenet. Hackers make the World Wide Web work (Raymond, 2001).

In this context, hackerdom is defined with “technical adeptness and a delight in solving problems and overcoming limits” (Ibid.). On the other hand, for Raymond, someone is really a hacker when people appreciate one’s work and start to call one as hacker. However, in this study, the term hacker is used in a wider context that represents the passionate workers of FOSS. Thus, a newbie who asks in a GNU<sup>3</sup>/Linux user mailing list:

Hi! I am new to GNU/Linux. What should I do to learn using it? What do you advice?

is considered as a hacker as well as a master programmer whom Raymond calls as a hacker. Furthermore, between these two extreme points, there are also hackers who test software, report bugs, offer new features, contribute to localization and give seminars. Each adds value to FOSS by their activity without getting any direct monetary reward.

This does not mean that there are not any differences between them and their relation to FOSS. Indeed, there are essential qualitative differences. For example, for programmers, sometimes their activity is “just for

---

<sup>3</sup>Recursive acronym for the phrase ‘GNU is Not UNIX’.

fun” (Torvalds and Diamond, 2001) as Linus Torvalds<sup>4</sup>. On the other hand, hackers who translate software manuals from English to Turkish are more interested in the product of their activity, making FOSS for more people rather than the activity itself. Moreover, a hacker, who has been contributing to FOSS for a long time, has more tight relation with FOSS.

Furthermore, this study diverges from most of the studies in the FOSS literature, because of its two premises about hackers. First, rather than presuming one always escapes from work, it presumes that under certain social circumstances one may also prefer work. Second, societies act and react to informational capitalism differently because of their cultural and historical diversity. For this reason, emergence and propagation of hackerdom is not the same in the all countries.

In this study, goal of the research was to gain deeper understanding on the relationship between the hackers and FOSS in Turkey. As stated above, this study does not consider hackers’ voluntary work as an anomaly. Furthermore, FOSS firms highly owe to the volunteer contribution of the hackers and they should understand hackers’ motivation for a viable software business. For this reason, the research focused on the certain social circumstances that make hackers preferred work to leisure. In this sense, the research asks five main questions:

1. What kind of FOSS works (or projects) do hackers engage in?
2. What are the differences between hacking and non-hacking activities of the hackers?
3. What are the differences between the hackers and non-hackers in the context of their relationship with their work?

---

<sup>4</sup>He is the creator of Linux.

4. What are the differences between the Turkish hackers and the other (European or American) hackers?
5. How do the Turkish hackers consider commercialization of FOSS?

In order to answer these questions, qualitative research methods, involving interviews and participant observation, were adopted. The interviews were conducted from February, 2005 to March 2006 with 66 Turkish hackers and observations include the time between the September, 2004 to May 2006. Additionally, data about the development of FOSS in Turkey was gathered from the mailing list<sup>5</sup> archives of LKD (Linux Users' Association-Linux Kullanıcıları Derneği). There has been a huge e-mail archive since December 1993. These lists' archive helped to comprehend how the general tendencies of hackers had changed since 1990s.

In this context, focus of the **third chapter** is the hackers in Turkey. Main findings of the research about the hackers in Turkey are presented in this chapter from different vantage points: types of work that hackers prefer, comparisons between *hacking and non-hacking activities*, *hackers and non-hackers*, *Turkish hackers and European hackers* and *FOSS licenses* in Turkey.

Lastly, in the **conclusion chapter**, hackers' relationship with business and firms in Turkey are discussed.

## 2 What is Free and Open Source Software?

### 2.1 A Brief Inquiry into the Literature on FOSS

FOSS is extraordinary and unprecedented. From the vantage point of competition, FOSS is a threat for proprietary software vendors. On the other

---

<sup>5</sup><http://liste.linux.org.tr>

hand, FOSS is also an interesting phenomenon for social scientists from the vantage point of relations of production which it contains, its work organizations and its culture. Indeed, what is problematic for proprietary software vendors is also problematic for social scientists: Money is not the primary motivation behind FOSS.

Despite there is a huge literature about FOSS, most of the studies are about "individual projects or internal phenomena of the field" (Lehman, 2004) and FOSS as a social whole "has been largely overlooked or was treated only superficially" (Ibid). Additionally, many studies repeat words *social movement* and *community* to describe FOSS phenomenon, but they have clear and distinct meanings in social sciences which may not correspond to FOSS phenomenon. For Lehman (2004), a more holistic approach is needed for comprehension of FOSS which includes both analysis of the field as a whole and interactions in it.

In social sciences, definition of *social movement* is controversial. For Blumer (1969), there is a relation between the dissatisfaction of enterprise and the emergence of movement. In this aspect, a social movement aims to establish a new order of life. Eyerman and Jamison's (1991) emphasis is collective creation of ideas, identities and ideals. According to Tarrow (1998), there is also a confrontation with elites, authorities and opponents

<sup>6</sup>. Social movements are important, because they

problematize the ways in which we live our lives, call for changes in our habits of thought, action and interpretation (Crossley, 2002:8).

In this sense, FSM (Free Software Movement) can be considered as a social movement since it problematize the ways in which people develop

---

<sup>6</sup>Crossley(2002) discusses the problems of these definitions in detail.

and use software. Its emergence and development is strongly related with the emergence of proprietary software and collapse of hacker community as its consequence. There is a collective creation of ideas, ideals and symbols (e.g. a manifesto <sup>7</sup>, a song <sup>8</sup>) There has always been a confrontation with proprietary software vendors, especially with Microsoft. FSM played an important role in the transformation of FOSS from its unconscious form to a conscious form, in other words to a goal oriented activity.

In European and the US literature, social movements are explained with different theories. In Europe, it is called as *new social movements* theory and based on social movements emerged after 1960s (e.g environmentalists, second wave feminists, the peace movement, gay rights movement, anti-globalization movement). This theory has essential role in the writings of post-marxists (Crossley, 2002). In FOSS literature, this theory is held mostly by The Oekonux Project. In the stated project, focus of debate is "whether the principles of the development of free software may be the foundation of a new economy which may be the base for a new society."<sup>9</sup>. Indeed, at the very start of GNU project Stallman's goal was not merely better software:

The principal goal of GNU was to be Free Software. Even if GNU had no technical advantage over UNIX, it would have a social advantage, allowing users cooperate, and an ethical advantage, respecting the user's freedom (Stallman, 2002).

It is obvious that FSM has some implications for a new society and these suggestions are contradicting with the capitalism (Pestimalcioglu, 2003).

---

<sup>7</sup>The GNU Manifesto, Retrieved December 5, 2005, from <http://www.gnu.org/gnu/manifesto.html>

<sup>8</sup>Free Software Song, Retrieved December 5, 2005, from <http://www.gnu.org/music/free-software-song.html>

<sup>9</sup><http://www.oekonux.org/>

On the other hand, FSM is a moment of FOSS and considering FOSS as a social movement has the possibility of underscoring its contemporary relations with the business world.

On the other hand, in the US literature, *resource mobilization theory* is widely used to explain social movements. In this theory, main concern is "balance of costs, rewards and incentives that provided agents with the motivation to become involved in the struggle" (Crossley, 2002: 12) For instance, Krogh et al. (2003), study on the rewards for individuals in FOSS projects. These rewards are reputation, control over technology and learning opportunities. Although this theory gives certain insights about the FOSS developers, it is criticized because of its base, rational actor model in "social movements" debate (Crossley, 2002). Additionally, its ahistorical approach does not help to understand the tendencies of FOSS. For example, it is presumed that people prefer leisure rather than working. However, as it is discussed in the second part of this chapter, man's attitude toward work may change under certain social circumstances.

Another frequently used term in FOSS literature is *community*. Tönnies (1958), who was one of the leading figures to study communities, used community (*Gemeinschaft*) to distinguish it from society (*Gesellschaft*). Community describes both similarity and difference. Members of community have something common within themselves and different from others (Cohen, 2000) and fulfill their needs by belonging to a community (McMillan and George, 1986). However,

The idea of community (in the sense of *Gemeinschaft* as described by Ferdinand Tönnies) emphasizes the non-rational sense of belonging among human beings, which is not bound to any specific purpose. This bond between individuals is based on sympathy, adaption or familiarization and memory, depending

on the exact nature of the relationship(Lehman, 2004).

Lehmann (2004) remarks that Tönnies's community theory focuses on organic groups (e.g. family) and "not fit FOSS developers well". According to Lehman (2004), on one hand, some of the basic assumptions of community theory (e.g. "close non-rational ties in a homogeneous, spatially delimited environment") does not fit to FOSS developers well. On the other hand, there are also community formation processes (e.g. "the writing of an endogenous history"). Studying FOSS developers as a virtual community is common in FOSS literature. Furthermore, some of these studies (e.g., Margret and Scacchi, 2003; Laurent, 2004) are related with the resource mobilization theory. Yet, the problems about change and tendencies of FOSS remain intact.

Both community and social movement based theories are too narrow abstractions to understand FOSS. First of all, they study only a short period of FOSS and generalize its results. Secondly, their analyses include only a few relations. For instance, most of the studies focus on a rational individual software developer. But for a long time, firms and governments are parts of the FOSS world. Young hackers transformed into venture capitalists in the second half of 1990s (Castells, 2001a). Thirdly, the conditions for existence of FOSS are part of what it is. In other words, the contemporary world is contained in FOSS. Yet, this fact is not fully discussed in the FOSS literature. For this reason, instead of merely asking 'What is FOSS?', additional question must also be asked 'In what kind of social and economic relations does FOSS exist?'

On the other hand, FOSS's different character from other goods in capitalism gave rise to conceptualization of FOSS in the gift economy. Smith

and Kollock (1999) described Linux as "the impossible public good". FOSS is nonrival and nonexcludable. It is nonexcludable, there can be unlimited numbers of users who can not restrict the use of others. For example, anyone can download the GNU/Linux freely. In addition by having its source code, anyone can add new facilities and distribute it with another name, such as Ali-x, Ayse-x myOS etc. It is nonrival, since it can be replicated with nearly infinite cost. In other words, there is not any significant marginal cost in its replication. Proprietary software is also nonrival but it is excludable. In this context, being nonrival and nonexcludable FOSS stays in the realm of public goods and proprietary software in the realm of private goods (it is excludable) with its tendency towards public goods (since it is nonrival) (Weber, 2004).

In this respect, being nonrival and nonexcludable, FOSS represents an economy of abundance that give rise to describe it as a form of gift economy. In the gift economy, one's social status is determined by what and how much one gives away (Raymond, 2000). For example, communities of scientists act through the principles of gift economy. In science, it is not important how much one knows, but how much one contributes to his/her field is essential among scientists. Moreover, if one had great knowledge and does not share with other scientists, it would be seen as waste of talent (Pinchot, 1999).

Furthermore, as Castells (2001a) remarks hacker culture inherited from the academic culture and hacker culture also bears the traits of the gift economy. Raymond (2000) defines hacker culture in the context of competence for prestige. According to Raymond, there is no serious shortage of bandwidth, disk space or computing power. Consequently, exchange economy became a pointless game without any serious scarcity. Reputa-

tion replaces monetary awards of exchange economy. In this sense, theory of gift economy explicitly removes the paradoxes of hackers' motivation in FOSS projects. Additionally, the gained reputation may transform into monetary rewards. Indeed, experiences of the hackers affirm the process of contribution-reputation-money.

However, theory of the gift economy has some problems and as Weber emphasizes that there is a only surface resemblance between the FOSS culture and the gift economy. According to him, it is true that FOSS

seems to bind people together, it encourages diffuse reciprocity, and it supports emotive feelings of stewardship for the gift that is both taken and given in return (Weber: 150).

Yet, Weber questions the extent of abundance. For him, despite the abundance of bandwidth and disk space, there is still some sense of scarcity as "the time, energy, and brain power of smart, creative people" (Weber: 151).

Thus, Weber is not convinced with the explanation of the gift economy. Even though it theorizes the practice of hackers, it is a contingent process and has the same problems with the conceptualizations of FOSS as a community and as a social movement. First of all, process of contribution-reputation-money may explain some hackers' practice, but it is not enough for a sustainable economy. Secondly, the gift economy, undervalues the social and economic conditions which it exists in. As Ghosh (1998) implies one should earn money to live in the capitalist economy.

Consequently, Ghosh (1998) avoids from describing FOSS economy from the vantage point of reputation. Instead, he uses an image of vast tribal cooking pot process which people put into something (e.g meat, onion, carrots etc.) and take out a bowl of stew. According to Ghosh (1998):

The cooking-pot model provides a rational explanation for people's motivations to produce and trade in goods and services, where a monetary incentive is lacking. It suggests that people do not only - or even largely - produce in order to improve their reputation, but as a more-than-fair payment for other goods - "ideas" - that they receive from the cooking-pot. The cooking-pot market is not barter, as it does not require individual transactions. It is based on the assumption that on the Net, you don't lose when you duplicate, so every contributor gets much more than a fair return in the form of combined contributions of others.

In this regard, Ghosh (1998) suggests a more rational framework than the contingent process of the gift economy. By describing FOSS, as a continuously accumulated social knowledge, he makes an important step to comprehending the existence of FOSS in the capitalist economy. He focuses on the development of FOSS in the contemporary economic relations rather than investigating an anomaly in hackers' relationship with their work.

## **2.2 FOSS in the Informational Economy**

### **2.2.1 FOSS as a Social Relation**

In this study, Castell's terms 'informational economy' and 'informational society' are rather preferred than the terms 'information economy' and 'information society'. Castells (2001b) uses the term informational instead of information because of their different significations. Information society denotes the role of information in society. In all societies information was critical. But in this stage of society, information conditioned by ICT became profound component of social relations. For this reason, the term informational indicates:

the attribute of a specific form of social organization in which

information generation, processing, and transmission become the fundamental sources of productivity and power, because of new technological conditions emerging in this historical period (Castells, 2001b: 21).

In this context, FOSS is conceived as a social relation which exists in informational society. Thus, instead of narrow (e.g FOSS Community) and ahistoric (e.g Gift Economy) conceptualizations, FOSS is discussed in the context of hackers' relationship with their work. Furthermore, it is studied as a process which contains its history and possible futures.

In that respect, rise of informational society (or economy) has also become the rise of software as a social phenomenon. Software, which is a type of information, emerged and developed in sophistication and commodification process of hardware (Valimaki, 2005). ICT reduced information to binary codes which is known as digitalization of information. As a consequence, digitalization eased and increased capacity for transmission and storage of information (Burnett and Marshall, 2003). Any information can be copied on cheap CD-ROMs or downloaded through Internet without any cost. For netizens<sup>10</sup>, especially for members of hacker culture, saying 'information wants to be free' was also their everyday practice on the Internet. According to Boyle (1996) this situation can be described as de-contextualization of information from any form or location.

However, saying 'software is a type information' is not enough to comprehend its specific role in the informational economy. Since as Castells remarks,

In the new, informational mode of development the source of productivity lies in the technology of knowledge generation, information processing and symbol communication. To be sure,

---

<sup>10</sup>It is concatenation of substring 'net' in Internet and substring 'izen' in citizen.

knowledge and information are critical elements in all modes of development, since the process of production is always based on some level of knowledge and in the processing of information. However, what is specific to the informational mode of development is the action of knowledge upon knowledge itself as the main source of productivity (Castells 2001b: 17).

In this respect, software is more inclined to be defined as knowledge which is a subset of information:

Knowledge: a set of organized statements of facts or ideas, presenting a reasoned judgment or an experimental result, which is transmitted to others through some communication medium in systematic form. Thus, I distinguish knowledge from news and entertainment (Bell, 1976: 175).

Additionally, its role in processing of information makes software essential in the informational economy. In this sense, software is a means of production for both itself and other information goods.

Thus, *free* may not have the same meaning in either *free information* or *free software*. In its first meaning, it evokes the meaning of one does not have to pay any money. One may download (or copy from friends) audio files, e-books or a whole operating system, GNU/Linux without paying any money. However, for software programmers, *free* means as it is in *freedom* or *free speech*. While first *free* is more related with consumption, latter is about production. Being free in its second meaning differentiates software from other information goods. There are also collaborative production of information goods other than software in which everybody can freely contribute and benefit from the contribution of others. However, this is more common in software development.

Software may act as database, editor or calculator for performing needs of its user. In this sense, FOSS may be free in its first meaning. On the

other hand, if you have the source code of software as in FOSS, it will be potential source code for a more functional software (Laurent, 2004). For instance, if you have the source code of a calculator application which performs only basic operations (addition, subtraction, multiplication, division), you may develop a more functional calculator in less time than building a calculator from scratch. In other words, free in FOSS as tendency of software has twofold aspects, use-value and means of production. One can use any FOSS according to his/her needs. Additionally, one can take the accumulated contribution of programmers and by adding value one can create more functional software. In other words, if a programmer develops a basic calculator in five days, another programmer can take its source code and develop a more functional calculator in three days. Yet, if he does not have chance to take the code of basic calculator, he can develop it in eight days.

However, software which has tendency to be free, also exists in capitalism as proprietary software. In this aspect, it has also tendency to commodify which made Bill Gates one of the richest people in the world. For this reason, freedom of software contradicts with capitalism in which profit maximization is profound. According to Gates:

a competitor who is free to review Microsoft's source code... will see the architecture, data structures, algorithms and other key aspects of the relevant Microsoft product. That will make it much easier to copy Microsoft's innovations which is why commercial software vendors generally do not provide source code to rivals (Gates, 2002)

Furthermore, globalization of information harmonizes toward more protection (Nayyer, 2001) and "building fences to keep people out" (Boyle, 1996: 18) by copyright laws. In this context, while FOSS developers want

software to be free as means of production, proprietary software firms want software to act as a commodity as anything in capitalism. As a result, while proprietary software build fences by copyright licenses, FOSS licenses try to remove these fences by their licenses.

In this context, FOSS contradicts with the characteristics of capitalism, profit maximization and it may seem staying away from capitalism. Yet, there are actual agents of FOSS in capitalism: FOSS firms and international companies (especially IBM and Novell). Additionally, hackers are real people who should sell their labor power in order to live. Even though hackers and firms have coinciding interests in promotion of FOSS, sometimes they pull FOSS in opposite directions. This reveals itself especially in the emergence of FOSS licenses. For this reason, hackers' relationship with their work and objectification of the contradiction between hackers and proprietary software business through FOSS licenses is essential for understanding historical specificity of FOSS.

### **2.2.2 Hackers**

Hackers' passionate relation with their work becomes a vital question for most of the studies in the FOSS literature. If the work in the informational economy is so corrosive on the characters of the workers<sup>11</sup>, why the hackers go on working in leisure time? Moreover, how can the hackers use the adjectives "fun", "joyous", "passionate" and "entertaining" to describe (Chance, 2005) their work?

Indeed, most of the FOSS researchers and hackers do not speak of the same kind of work. FOSS literature's conceptualization of work reminds

---

<sup>11</sup>Including high-skilled IT professionals (as IBM workers) (Sennett, 1998)

Marx's words about work that is avoided as a plague (Marx, 1993). However, Marx also emphasizes the essence of work in one's life (Marx, 2003).

This study points out that hackers' work is different from the contemporary meaning of work. The most important fact about the hackers is that they enter voluntarily into the work and this work includes operational autonomy, a variety of productive and creative activities. Chance describes this work as meaningful, since

hackers can choose what code to work on; they can influence the agendas that direct their work, both by entering into the open discussions about the direction of the project(s) they work on and by simply opting out of any projects whose agendas conflict with the hackers' own priorities; and finally though it is not forced nor prevalent in every productive forum, hackers can and often do enter into discussions about work itself (Chance, 2005)

On the contrary to contemporary society, hackers reject any distinction between work and leisure, instead they emphasize active realization of themselves as a creative and autonomous person. Consequently, instead of investigating about anomaly in hackers behavior, this study asks for the social conditions of hackers work and its sustainability in the informational capitalism.

Furthermore, societies act and react to informational capitalism differently according to their cultural and historical specificity. Although all societies are affected by informational capitalism, it happened in "different settings with specific cultural / institutional expressions" (Castells, 2001b: 21). For this reason, as one can not speak of homogeneity of informational society in the all countries, one can also not speak of homogeneity of the hackers. It is sure that there are some common traits among hackers. However, emergence and propagating of hackerdom are not the same in the

all countries. For example, in the US, "the culture of freedom, individual innovation, and entrepreneurialism that grew out from the 1960s culture of American campuses" (Ibid.) played essential role in the emergence of FOSS. European, Chinese or Indian hackers appropriation of FOSS happened in different historical and social contexts. So did the Turkish hackers.

Thus, despite the hackers' common attitude towards work, specificity of the hackers will be the second premise of the study about hackers.

### 2.2.3 FOSS Licenses

The common point of FOSS business models is, although FOSS firms sometimes contribute to the FOSS projects, they highly owe to the volunteer contribution of the hackers. In other words, while source code is conceptualized as crown-jewels which must be hidden from the others in the proprietary software business, in FOSS business source code is continuously accumulated social knowledge. Thus, it needs sharing for its existence and giving freedom for attracting contribution of hackers. In other words, firm's success depends on ongoing involvement of the hackers (Golden, 2005).

There are many studies which point out the role of hackers in the firms FOSS business model. For example, Fink warns the firms about dual licensing:

Depending on the project, the community may object to copyright assignment. Make sure you seek council with potential members of the community before assuming that everyone will agree to assign you their copyright (Fink, 2003: 182).

As Hecker remarks,

much of your potential success will depend on the efforts of others willing to work for you "for free" you need free-software

developers who will contribute their work to your company, and to the developer community at large, without demanding or receiving money or other tangible payment in return (Hecker, 1999).

Additionally, Onetti and Capobianco (2005) also underline the key role of hackers' work in FOSS business. There are also studies which emphasize about supporting the FOSS developers in order to keep away from the danger of freeloader image in hackers' eyes (DiBona et al., 1999).

However, despite these statements, possible conflicts of interests between individuals and firms are usually undervalued and not studied in detail. In one of the rare studies that discusses this conflict, Bonaccorsi and Rossi (2003, 2004, 2005) point out "altruistic individuals" and "selfish firms" from the perspective of relationship between their different sets of motivations.

Bonaccorsi and Rossi (2005) emphasize that while firms enter the FOSS field in order to profit from FOSS, monetary rewards are not so crucial for the hackers. In this respect, they make distinction between the extrinsic and intrinsic motivations. Extrinsic motivations are mostly related with monetary rewards and intrinsic motivations are related with the pleasure of activity itself. For this reason, while firms mostly have extrinsic motivations, intrinsic motivations are more essential than extrinsic motivations for the hackers. In other words, hackers are more interested in the activity itself than the product.

In studying the relationship between the hackers and the firms, the evolution of the FOSS licenses and contradiction between free and open source software are essential topics. Since, the term open source software was coined as a marketing strategy for free software.

In this context, first of all, a misconception, FOSS as a public good must be corrected. Being in public domain means there is not any copyright holder who has the right to restrict copying or using the software.

On the contrary, McGowan argues that

Open-source software production is not about the absence or irrelevance of intellectual property rights. Open-source production instead rests on the elegant use of contractual terms to deploy those rights in a way that creates a social space devoted to producing freely available and modifiable code...Open-source production, therefore does not take place within a true commons... (McGowan, 2001)

FOSS licenses are based on copyright and some of them (especially GPL) restrict copying or using according to the statements in their license agreements. This happens in two steps. First, they reverse the traditional copyright and give users copying, using and distributing rights. For example, according to GPL <sup>12</sup>:

- You have the freedom to run the program, for any purpose.
- You have the freedom to modify the program to suit your needs. (To make this freedom effective in practice, you must have access to the source code, since making changes in a program without having the source code is exceedingly difficult.)
- You have the freedom to redistribute copies, either gratis or for a fee.
- You have the freedom to distribute modified versions of the program, so that the community can benefit from your improvements.

However, in the second step , GPL puts restrictions<sup>13</sup>:

---

<sup>12</sup>Retrieved December 5, 2005, from <http://www.gnu.org/philosophy/free-sw.html>

<sup>13</sup>Retrieved December 5, 2005, from <http://www.fsf.org/licensing/licenses/gpl.html>

- Any derivative work—that is, any work containing a non-trivial amount of GPLed code—must itself be distributed under the GPL.
- No additional restrictions may be placed on the redistribution of either the original work or a derivative work. (The exact language is: "You may not impose any further restrictions on the recipients' exercise of the rights granted herein.")

Furthermore, these restrictions made GPL the most popular FOSS license by preserving the source code for hackers. As DeLanda (2001) emphasizes, GPL has been like a virus. If one developer uses a GPL'd software, the derivative work should also be licensed with GPL that will be source for the latter works. In fact, FOSS with GPL is the most suitable license for conceptualization of software as continuously accumulated social knowledge. On the other hand, OSI (Open Source Initiative) does not insist on GPL. Moreover, according to some FOSS advocates, GPL's reciprocal statements make it less free and an important barrier for attracting business world to the FOSS, so they encourage using less restrictive licenses (Fogel, 2005). Among these licenses BSD is the most popular. Its mere statement is writing the author of the code. It gives any right to user including making source code private (Laurent, 2004).

According to open source advocates, giving freedom to make source code private increases firms incentive to participate in the FOSS projects. Indeed, Microsoft also agrees with open source advocates. For Microsoft, open source has some positive elements, but GPL is a threat for IPR:<sup>14</sup>"The

---

<sup>14</sup>Retrieved December 5, 2005, from <http://www.microsoft.com/presspass/features/2001/may01/05-03csm.msp>

GPLs viral nature poses a threat to the intellectual property of any organization that derives its products from GPL source”.

In this context, although BSD and other open source licenses encourage firm’s participation in FOSS projects (by giving them right to proprietarize source code), they do not guarantee the continuity on social accumulation of knowledge. Additionally, they are less successful than GPL in building collaborative development around a project (Fink, 2003). On the other hand, if project’s goal is making software be accessible by the greatest possible number of users and implementing a standard in a part of software sector, BSD-style licenses gain importance.

### **3 Main Findings of The Research About Hackers**

The research focused on the social circumstances that made Turkish hackers work voluntarily. In this part, key findings of the search about Turkish hackers are presented by discussing these social circumstances and hackers’ attitudes according to the questions in the introduction.

Interview questions are open ended and can be categorized into two groups. In the first group, questions are mainly about how hackers perceive and appropriate FOSS. In this respect, practice of the hackers are investigated as well as the non-hackers, software engineers. In the second group, hackers’ attitudes toward commercialization of the FOSS in Turkey are investigated. In this context, emergence and development of the terms unrestricted software, free software and open source software are discussed. Mail archives of LKD since 1993 are investigated and hackers were asked about the contradiction between free software and open source software.

### 3.1 What kind of FOSS works (or projects) do hackers engage in?

Hackers' contribution to the FOSS in Turkey is not only by developing software. They localize FOSS, give seminars and organize social activities (festivals and meetings) in order to familiarize more Turkish people with FOSS. In this context, as one can not speak of the homogeneity of hackers' work, one can not also speak of the homogeneity of the social circumstances that make their voluntary contribution possible.

On the other hand, FOSS users are named as prosumers<sup>15</sup> because of their specific relation with FOSS that is based on reproducing and improving it. In this sense, they are not passive computer users as most of the Microsoft or Macintosh users. Thus, on the contrary to ordinary computer users, most of the hackers in the research were at least capable enough to write small scripts<sup>16</sup>. Although writing scripts is not a necessity, it gives important opportunities to automate or simplify long works that can not be done by Windows operating systems. For example, the new hacker who was a Windows user two months ago told that:

Assume that you have 10 mp3 files and you want to hide these files by making their extension .zip. How one can do it on windows? Right click the file, rename it and do it for each file. But on Linux, I just wrote a small script and after this script number of files was not important for me: 10 file or 1000 files, there is no matter. I was not a programmer, but Linux made me develop creative solutions.

In other words, most of the hackers at least program for themselves in their daily use.

---

<sup>15</sup>Concatenation of producer's pro- and consumer's -sumer

<sup>16</sup>Small programs used only for simple, repeated actions.

Weinberg discusses the psychology of the programmers from the perspective of different personality traits whether a person is compliant, aggressive or detached:

The compliant type is characterized by the attitude of "liking to work with people and be helpful." The aggressive type wants to "earn money and prestige," and detached type wants to be "left to myself to be creative." (Weinberg, 1998: 53)

According to Weinberg, programmers are more inclined to having a detached type. Moreover, this detachment is overdeveloped:

Although they are detached from people, they are attached to their programs. Indeed, their programs often become extension of themselves (Ibid.).

In this context, it can be argued that hackers make FOSS extension of themselves as they presume it. For this reason, availability of source code means for the hackers is its *prosumeability*. This is the most common point between the all hackers whether they contribute to FOSS projects by programming or in the other ways. Indeed, it is the main reason behind the distribution wars, since each hacker personalize their GNU/Linux distribution. However, for the hackers who program for the others<sup>17</sup>, challenge of work, autonomy and increasing realization of one's powers are essential in their relation with FOSS. Most of the hackers implied that they did not like all programming tasks. For example, a few said that they did not contribute to the Uludağ project<sup>18</sup> because developing a distribution was an easy work. As I observed from the mailing lists, when a member of the list asked a question, hackers' response to this question is determined by its challenging degree. When it is a so easy question, hackers do not reply

---

<sup>17</sup>Hackers who are releasing their product for public use.

<sup>18</sup>A local GNU/Linux distribution sponsored by Turkish Government

it or reply to mail as "It is simple question. Search on Google". When it is harder question, hackers themselves search on Google and compete for finding the best solution. Furthermore, if a well-known and technically superior hacker posts his problem (it is very rare), number of hackers who are eager to answer the question is higher, too.

For most of the hackers, hacking means autonomous work. They do not like to be told what to do and how to do. As some of the hackers said hacking was something for them as playing chess, solving puzzles or reading a book and difference was product might also be a valuable good in the market. For them, the most fortunate people were who could also hack at daily job. Furthermore, autonomous work also means work which is away from market pressure. Thus, in their view, Microsoft programmers are not enemies, but who suffer from competence and market pressure.

Increasing realization of powers is also related with challenge and autonomy. In each challenging autonomous work, they try to overcome their boundaries. Additionally, hackers preference for UNIX Operating Systems (e.g GNU/Linux and BSD) is not by chance. Raymond explains difference between the philosophy of UNIX and the other operating systems:

Many operating systems touted as more 'modern' or 'user friendly' than Unix achieve their surface glossiness by locking users and developers into one interface policy, and offer an application programming interface that for all its elaborateness is rather narrow and rigid. On such systems, tasks the designers have anticipated are very easy - but tasks they have not anticipated are often impossible or at best extremely painful.

Unix, on the other hand, has flexibility in depth. The many ways Unix provides to glue together programs mean that components of its basic toolkit can be combined to produce useful effects that the designers of the individual toolkit parts never anticipated (Raymond, 2003).

As UNIX-style Operating Systems, hackers prefer the software which can be extension of themselves by their work. Since, it does not give the tools to do something, but gives tools to make new tools.

On the other hand, for hackers who participate in mostly non-programming activities (localizing software, giving seminars, organizing meetings/festivals), although there may be pleasure in the activity itself, products of the activity are important. In this respect, these hackers were more inclined to use the words, sharing, freedom, helping others etc. for describing their relation with FOSS.

However, Uludağ project and release of its first product, Pardus, added another dimension to the development of FOSS: Nationalism. Project management team made the ways of contribution possible through mailing lists with a clear statement of goals. After this it was ordinary to see messages like:

Hello, I am a student. I know C, php and Java programming languages. I can spend 6 hours a week for the project. I can also translate English documents. What can I do for Uludağ?

On the other hand, nationalism is not enough to understand hackers' interest for it. Since there were also local GNU/Linux distributions before it. The most critical point about the project is project management team's strategy: Releasing the project with GPL and having good relationship with the hackers since the beginning.

### **3.2 What are the differences between hacking and non-hacking activities?**

To understand hacking, I wanted hackers compare it with their non-hacking activities. In this respect, they compared programming at work with hack-

ing at leisure. There were hackers who were also hacking at work time professionally, so they said that there was not any difference.

On the other hand, for the others, there was an important discrepancy between hacking and non-hacking. For Weinberg, the deepest difference between the professional and amateur work is the user of the program.

Almost invariably, the sole intended user of an amateur's program is the amateur himself, whereas professional is writing programs which other people use (Weinberg, 1998: 122).

It is also true to some extent for hacking and non-hacking activities of the hackers. Since, even though hackers mostly write programs for themselves as "scratching an itch" (Raymond, 1999), the product may be released for public use. For this reason, explaining the difference between these activities should depend on the voluntariness of the work. All hacking works (programming, localization etc.) can be considered from the perspective of voluntariness.

However, if the differences are only explained from the vantage point of hackers as the difference between voluntary and involuntary work, one can undervalue the specificity of the FOSS hackers in the informational capitalism. FOSS hackers relation with FOSS is different from the ordinary leisure time activities. Since they relate with their product through FOSS licenses. It is true that interviewees were not well-informed about the FOSS licenses. Nevertheless, they were inclined to use GPL. According to Fink,

While the development process is not about the license, it is the license that provides the boundary conditions that make the process work. These conditions establish a foundation that ensures communal and collaborative development. Since the GPL, LGPL, and other licenses that contain a reciprocity requirement provide that changes and modifications be returned to the community, this is the foundation that ensures collaborative development (Fink, 2003: 53).

In this context, by establishing their relationship with FOSS through FOSS licenses, first, they ensure continuous accumulation of social knowledge. On the contrary to the common sense, hackers are not people who dedicate their whole life to the FOSS. As I observed hackers from the e-mailing lists, most of the hackers' of the 1990s are not hackers now. However, as Ghosh's (1998) cooking-pot market, during their hacking process, they prosume FOSS without exhausting but with increasing its value by their small to large contributions. Second, hacking product with the FOSS license socializes the software.

... the license guarantees that the hacker and the user receive exactly the same rights with respect to the product, and that both are endowed with the product's full creative and productive potential. ... whereby the more hackers produce and relate to one another through sharing, the more productive and communicative powers they have (Chance, 2005).

In summary, the main difference between hacking and non-hacking activities is the voluntary work that is defined through FOSS licenses that both ensures continuity of the knowledge and socializes hackers work.

### **3.3 What are the differences between the hackers and non-hackers in the context of their relationship with their work?**

In order to answer this question, I tried to compare hackers' practice with the software engineers. As the answers of previous questions, key finding of the research was the major role of voluntary work in hackers' relationship with the software and the major role of obligation in software engineers' relation with the software.

First of all, as some of the interviewees emphasized, there was an essential difference between the formation of these relationships and role of the

voluntarism appearing at the first step. Students prefer computer engineering departments because of its popularity and higher income. For most of the students and software engineers, earning money from FOSS is not clear and they choose proprietary software business. Furthermore, they consider hackers' contribution as altruism. On the other hand, hackers' relation with FOSS starts voluntarily.

Secondly, software engineers' lives are similar to the characters in Senett's (1998) book. They are flex-timers and usually work under the pressure of project deadlines and competition in the market. When I asked if they programmed at home in their leisure and enjoyed programming, they started to complain about extra hours, project-based working, difficulties of catching up with the technology etc.

However, as their firms work on FOSS, some of the hackers did not declare any difference between hacking for the firm and hacking for themselves. They answered, humorously, "I worked 30 hours a day". But they never complained about anything.

Thirdly, there are significant differences between the work processes of the hackers and the software engineers. For the software engineers, the product is a whole entity. They can not say, "I want to develop this part rather than that part". On the other hand, hackers can do this in their personal hacking and some firms overlook<sup>19</sup> the hackers' attitudes. Indeed, there is again a difference between the kind of work: while software engineers seek for external rewards (money), hackers are mostly interested in the activity itself.

---

<sup>19</sup>This behavior of firms is discussed in the next chapter.

### 3.4 What are the differences between the Turkish hackers and the other hackers?

Despite the huge literature about the motivations of the hackers and the FOSS business models, there is an undervalued issue: Lower participation of the females in the FOSS activities. According to Levy, hackers

formed an exclusively male culture. The sad fact was that there never was a star-quality female hacker (Levy, 2001: 84).

Although I used the term, hacker, in a wider meaning, I could not interview with any female hacker<sup>20</sup>. Thus, I discussed lack of female hackers both with male hackers and female software engineers/programmers. They attributed the phenomenon to genetic and cultural differences. A few women explained the genetic differences as,

females prefer permanent relationship. Their first operating system was Windows. They are not like males. They can easily move from Windows to Linux.

Nevertheless, there is a slight difference between Europe and Turkey. According to a research in EU, "women do not play a role in the development of Open Source and Free Software; only 1.1% of the FLOSS sample is female." (Ghosh et al., 2002). There were less female hackers in Turkey. For example, there were not any female among the hackers who gave seminars until the Free Software and Open Source Days in 2006.

On the other hand, there are significant motivational differences between European and Turkish hackers. Results of research (Ghosh et al., 2002) in EU are presented in the Table 4.1.

---

<sup>20</sup>I tried, but some of them refused to answer because of their business

Table 1: Motivations of FOSS Developers in Europe

Motivation	%
Learn and Develop new skills	78.9
Share Knowledge and skills	49.8
Participate in a new form of cooperation	34.5
Improve FOSS products of other developers	33.7
Participate in the FOSS scene	30.6
Think that software should not be proprietary good	30.1
Solve a problem that could not be solved bu proprietary software	29.7
Improve my job opportunities	23.9
Get help in realizing a good idea for a software product	23.8
Get a reputation in FOSS community	9.1
Distribute not marketable software products	8.9
Make money	4.4
I do not know	1.9

Table 2: Motivations of FOSS Hackers in Turkey

Motivation	%
Learn and Develop new skills	10.9
Share Knowledge, collaboration, contribute to a better world	67.2
Nationalism	4.7
Think that software should not be proprietary good	7.8
Solve a problem that could not be solved bu proprietary software	4.7
Improve my job opportunities	1.6
Enjoy programming, fun	21.8

In the interviews, I asked open ended questions. For this reason, it is not possible to find one-to-one correspondence. However, Table 4.2 at least gives the tendencies of the Turkish hackers.

The first difference between the results of these studies is Turkish hackers' are more inclined to social motivations. Indeed, the EU report only included FOSS developers in its research. On the other hand, in this research, interviewees belong to a wider spectrum. Furthermore, 100% of the hackers, who were not developers, declared their interest in social aspects of FOSS. Yet, if the interviewees were only developers, social motivations of the Turkish hackers would be also higher than the Europeans.

Second difference is lack of monetary motivations in Turkey. Some hackers earn money in the process of contribution-reputation-money. Nevertheless, they do not contribute to the projects for money. Additionally, there are only few FOSS firms which may employ these hackers and only a

small fraction of them monitor hackers' activities for employment.

Last difference is nationalism. Despite some government's national software policy on FOSS, supporting FOSS in the name of nationalism makes Turkish hackers distinct. Percentage of the hackers who are motivated by nationalism may seem inconsiderable. However, percentage of the hackers who talked about Uludağ project with nationalist terms was more than this percentage. Furthermore, after announcement of the project, many new people started to pay attention to FOSS because of its national aspects.

In this context, although most of the hackers are not under the influence of nationalism, it can be argued that most of the hackers who contribute to the Uludağ project are motivated by nationalism. Continuity of their contribution may alter the formation of FOSS in Turkey. However, second possibility is more probable: Internationalization of FOSS may affect their national feelings in the opposite direction.

Although it is not observed from the comparison of tables, another big difference between Turkey and Europe is the organizational ability of people. For example, in most of the LKD panels and meetings, if the LKD's board of directors was there, one of the frequent question was,

Hello,  
We are students from X university. We want to organize at the university as GNU/Linux users. Does LKD have future plans for organizations in the universities?

As one of LKD activists told people organized by themselves in Europe and Turkish people had less organizational ability than the European. On the other hand, as I observed in my participation at the LKD work groups, they overcome the problems of organizational ability.

In a work group, they plan their work in detailed and document it for future activists. In case of any problem, they quickly reorganize in order to overcome this. For example, they create sub work groups immediately for the problem or they review the work process for finding problems and clearing unnecessary steps. In this sense, they organize as they program: Divide the long functions into sub functions and remove unnecessary statements in it <sup>21</sup>.

### **3.5 How do the Turkish hackers consider commercialization of FOSS?**

While conflicting interests of hackers and firms transformed FOSS into new forms, coinciding interests of hackers and firms made FOSS more widespread and as a network affect new actors with different interests engaged in FOSS.

In this context, considering the separation between Free Software Foundation and Open Source Initiative is necessary to analyze the tendencies of FOSS in Turkey. Open source software appeared as a need for "a marketing program to pitch it to the corporate world" (Fogel, 2005: 15). As a consequence of the separation between the free software and open source software, each side started to favor different types of licenses. For free software advocates, hackers' interests have been more essential than anything, so GPL which is based on reciprocity has been the most appropriate license for them. On the contrary, open source software advocates are more concerned with firms participation in FOSS. So they prefer BSD-style licenses<sup>22</sup> which give rights for firms to make the software proprietary. In the recent

---

<sup>21</sup>Indeed, this process is called as refactoring in software development.

<sup>22</sup>Licenses which do not enforce reciprocity.

times, Raymond, leader of the OSI, declared that for pragmatic reasons, hackers should prefer BSD license rather than GPL <sup>23</sup>.

The discussions are still going on in the world, but there is a different situation in Turkey. As I observed from the discussions in mailing lists and forums, although some Turkish hackers are aware of discussions around the world, FOSS as a relationship between hackers and firms is still in the formation process and being conditioned by the discussions in the world.

In this context, hackers were asked for the relation between Free Software and Open Source Software to understand how they perceived the discussions in the world:

Which term do you prefer? "Free Software" and "Open Source Software"? Why?

In the world, it is observed that there is a tendency for using "Open Source Software" rather than "Free Software". What do you think about this tendency?

Soon, this relation is investigated from the perspective of licenses:

Do you have any license preferences for the products that you develop? GPL, LGPL, BSD, Mozilla etc?

There were significant inconsistencies between the answers of the hackers. While some preferred to use Open Source Software instead of Free Software, they preferred to use Free Software License, GPL. On the other hand, while some preferred Free Software, they talked about using BSD licenses. Indeed, most of the hackers were not aware of the debate between Free Software and Open Source. They said that they preferred GPL license, because they did not know anything about the other licenses.

---

<sup>23</sup>Retrieved December 5, 2005, from

<http://www.myfreebsd.com.br/static/raymond-20050604.html>

More interestingly, after a few interviews I had to change the question and added the term, unrestricted beside free and open source<sup>24</sup>. In English, the main problem about free was the confusion between different meanings of free. Thus, free software hackers explain it, "free as in free speech, not as in free beer". In Europe, 'libre' is also used for free software to escape from the confusions of free. However, in Turkey, even though the word, 'free' has not any problem as 'free' in English, some hackers started use the word 'unrestricted'. First of all, the word, 'unrestricted', is not capable enough to explain the philosophy of free software. Second, it is inclined to be understood as 'zero-cost' more than 'free'. Moreover, some of the hackers insistence on translating free software as unpaid software<sup>25</sup> shows that they consciously avoid from using the Turkish word, özgür.

A hacker explained the problems of 'özgür' in Turkish:

In Turkey, unfortunately, 'free' means rebelism, anarchism, terror. Additionally, firms may not consider the word 'free' as a serious discourse.

And some hackers, especially Debian hackers, were strongly opposed to the term 'Unrestricted'. They said they did not talk about public good, so using the term 'unrestricted' was nonsense.

Additionally, LKD's first governing statute was revealing its unawareness about the international debate between the free software and the open source software. Neither of these terms were used in the first governing statute. Instead, the term, unrestricted was preferred.

However, after three years, in its second general committee, the term unrestricted was replaced with the term, free. Additionally, after Gates's

---

<sup>24</sup>in Turkish, unrestricted:Serbest, free:Özgür, open source:Açık Kaynak Kod

<sup>25</sup>Retrieved December 5, 2005, from

<http://listweb.bilkent.edu.tr/linux/06/0116.html>

visit in 2005, in the platform, which was led by LKD, they preferred the term open source and started talking about business models.

In fact, the change in LKD's discourse and internalization of the international debate did not happen by coincidence. On the contrary, the change is strongly related with two processes.

First process is similar to the trend in the world: transformation of some hackers to entrepreneurs. When they graduated from their schools, they chose to make money from what they knew well, FOSS. In this sense, while nature of FOSS was changing from hacking object to business object, a change in LKD was inescapable. Since, these transforming hackers were also LKD's activists. This is discussed in detail in the next chapter.

Second process is LKD's and other Turkish FOSS organizations' practices. As discussed in the previous chapter, the terms free software and open source software, consequently the debate between them, happened and evolved under some social circumstances. For this reason, without any social base, appearance of this debate would sound artificial. If there were not any hackers who desired continuous availability of source code and if there were not any firms who wanted to benefit from hackers' work, there would not be any conflict between hackers and firms. However, LKD's activities and internationalization works of small hacker groups, helped rise and making of the hackerdom in Turkey. Furthermore, success of the FOSS products and firms affected the IT firms' business strategies. Additionally, Uludağ Project is accelerating the process by making licenses a current issue for hackers.

Furthermore, hackers were bothered by being known as enemies of Microsoft. For this reason, there is a general attitude among old hackers that rather than pronouncing the name of Microsoft, they call it as "the other

operating system”.

## 4 Conclusion

In this study, instead of merely answering the questions of the FOSS literature, questions themselves were questioned because of their ahistoric presumptions. Since, one may also prefer work as well as leisure under certain social circumstances. Furthermore, as in the case of hackers’ relation with their work, duality between work and leisure is not a historic necessity. Therefore, work, as a productive activity of the hackers, is also leisure for them.

On the other hand, in FOSS business models, software is considered as continuously accumulated social knowledge which needs sharing rather than hiding. In this context, FOSS business models depend on voluntary work of the hackers. However, as discussed in the study, hackers do not contribute to and support all FOSS projects.

In that respect, types of works/projects that hackers prefer were investigated. In this investigation, it was concluded that hackers could be grouped into two broad categories. In the first category, there were hackers who were interested in challenging tasks in FOSS projects and realization of their passion in programming activity. In the second group, there were hackers who were interested in the product rather than the activity itself. Members of this category were, particularly, localizing software and giving seminars. Indeed, there was overlap between each category. Nevertheless, it can be said that while members of the first category were mostly attracted by the pleasure in the activity itself, hackers in the second category were interested in the social aspects of FOSS and in the interviews they talked

about, freedom, sharing, helping others etc. In this sense, widespread use of Debian, which is well-known about its social sides, is not a coincidence.

On the other hand, all hacking activities have three common points. First, they are voluntary activities. Second, relationship between the hackers and their products is established through FOSS licenses to ensure continuity in the accumulation of the knowledge. On the contrary to common intuitive, hackers are not always very high-skilled programmer. There are many ordinary hackers who make modest contributions to the FOSS projects. Most of them do not dedicate their entire life to FOSS. They prosume FOSS and their small contributions to the software continuously accumulate (Ghosh, 1998). Third, by making the software public, software is socialized that anybody can prosume and reproduce it (Chance, 2005).

Additionally, voluntary work is the essential difference between hackers and non-hackers (especially software engineers). In Turkey, software engineering is one of the most popular professions because of its higher income. In other words, software engineers' relationship with their work is conditioned by monetary rewards at the beginning. On the contrary, hackers' relationship with FOSS begins voluntarily.

However, there are some differences between European and Turkish hackers. First of all, Turkish hackers are more interested in the social aspects of FOSS than European hackers. For some hackers, FOSS is a national issue. Although general hacker culture is more inclined to internationalism than nationalism, Uludağ Project attracted new hackers because of its national aspects.

Furthermore, it was also true for Turkish hackers that money was not an essential motive for them. This fact can not be explained merely by the altruism of Turkish hackers. When the Turkish hackers are compared with

the European hackers, it is true that Turkish hackers were more interested in the social aspects of FOSS and less interested in monetary rewards. However, it was also true that earning money from FOSS is not still obvious in Turkey. Hence, as discussed in the context of software engineers, participating in FOSS projects is seen as a kind of altruism. Additionally, there are not many FOSS firms in Turkey which might employ hackers.

On the other hand, most of the hackers did not show any tendency against commerce or market. For example, they did not deny that they disliked Microsoft, but they did not consider it as an enemy. Furthermore, some considered Microsoft and proprietary software as a complementary for FOSS. Young student hackers were more inclined to social sides of FOSS and sometimes they gave anti-commercialist answers. For the other hackers, commerce and market were the fact of life that they could not deny.

Additionally, despite hackers' inclination to GPL, they were not well-informed about the FOSS licenses. However, Uludağ Project triggered the FOSS licenses issue among hackers. The project management team's insistence on GPL and free software philosophy increased awareness about GPL. The project management's team message to the Turkish hackers was very clear: "Uludağ is a GPL'd project. Pardus is yours. Let's hack!". In this sense, hackers learned GPL in their practice.

Nevertheless, in Turkey, firms should consider social motivations of the hackers in most of the projects. In that point, greedy salesman image is an essential problem in FOSS business. In other words, firms should escape from free-riding problem, such that they should not have an image of getting from FOSS, but giving back nothing. Since if hackers have an impression that their contribution is just for firm's profit, it is highly probable that they give up contribution to the project. Hackers are not very tolerant

for the businessmen who considered FOSS merely as a profitable business. This type of businessmen are not supported by the hackers.

In summary, it can be stated that contradicting interests of hackers and firms do not seem to have significant negative effects on the future development of FOSS in Turkey. Instead, hackers and firms reinforce each other. FOSS, as continuously accumulated social knowledge, may present important opportunities for Turkish software firms and firms can conduct hackers work for their own benefit. However, in order to be successful in FOSS business, they should always consider type of voluntary work they need (e.g. challenging works vs. ordinary works), basic motivations of volunteers (e.g. pleasure in the activity itself vs. social motivation) and ways of fostering volunteers' motivations.

## References

- Baldwin, C. Y. and Clark, K. B. (2003). *The Architecture of Cooperation: How Code Architecture Mitigates Free Riding in the Open Source Development Model*. Harvard Business School.
- Bell, D. (1976). *The Coming of Post-Industrial Society: a Venture in Social Forecasting*. New York: Basic Books, 2nd edn.
- Bitzer, J., Schrettl, W. and Schroder, P. J. (2004). Intrinsic motivation in open source software development. Retrieved October 3, 2005 from <http://opensource.mit.edu/papers/bitzerschrettlshroder.pdf>.
- Blumer, H. (1969). *Collective Behaviour in A. McClung Lee(ed), Principles of Sociology*. New York: NY:Barnes and Noble.
- Bonaccorsi, A. and Rossi, C. (2003). Why open source software can succeed. Retrieved October 3, 2005 from <http://opensource.mit.edu/papers/rp-bonaccorsirossi.pdf>.
- Bonaccorsi, A. and Rossi, C. (2004). Comparing motivations of individual programmers and firms to take part in the open source move-

- ment. from community to business. Retrieved October 3, 2005 from <http://opensource.mit.edu/papers/bnaccorsirossimotivationlong.pdf>.
- Bonaccorsi, A. and Rossi, C. (2005). Intrinsic motivations and profit-oriented firms in open source software. do firms practise what they preach? *Proceedings of the OSS 1st International Conference on Open Source Systems July 11 - 15, 2005 Genova, Italy*.
- Boyle, J. (1996). *Shamans, Software, and Spleens: Law and The Construction of Information Society*. Cambridge, Mass. : Harvard University Press.
- Burnett, R. and Marshall, P. D. (2003). *Web Theory An Introduction*. Routledge.
- Castells, M. (2001a). *The Internet Galaxy: Reflections on the Internet, Business, and Society*. New York: Oxford University Press Inc.
- Castells, M. (2001b). *The Rise of The Network Society*. Blackwell Publishers Ltd.
- Chance, T. (2005). The hacker ethic and meaningful work. <http://opensource.mit.edu/papers/chance.pdf>.
- Cohen, P. A. (2000). *The symbolic construction of community*. London ; New York : Routledge.
- Crossley, N. (2002). *Making sense of social movements*. Buckingham ; Philadelphia, PA : Open University Press.
- DeLanda, M. (2001). Open-source: A movement in search of a philosophy. presented at the Institute for Advanced Study, Princeton, New Jersey, Retrieved November 21, 2005, from <http://www.cddc.vt.edu/host/delanda/pages/opensource.htm>.
- DiBona, C., Ockman, S. and Stone, M. (1999). Introduction. *Open Sources: Voices from the Open Source Revolution, 1st edition*, Retrieved January 31, 2005, from <http://www.oreilly.com/catalog/opensources/book/toc.html>.
- Eyerman, R. and Jamison, A. (1991). *Social Movement: A cognitive approach*. Cambridge Politics.
- Fink, M. (2003). *The Business and Economics of Open Source and Linux* (pp. 175–208). New Jersey: Prentice Hall.
- Fogel, K. (2005). *Producing Open Source Software*. O'Reilly.

- Gates, B. (2002). State of new york vs microsoft corporation, direct testimony. <http://www.microsoft.com/presspass/legal/billgates/billgates.asp>.
- Ghosh, R. A. (1998). Cooking pot markets: an economic model for the trade in free goods and services on the internet. *First Monday*, Retrieved October 3, 2005 from [http://www.firstmonday.org/issues/issue3\\_3/ghosh/index.html](http://www.firstmonday.org/issues/issue3_3/ghosh/index.html).
- Ghosh, R. A., Krieger, B., Glott, R. and Robles, G. (2002). Free/libre and open source software: Survey and study. *International Institute of Infonomics, University of Maastricht, The Netherlands*, Retrieved November 21, 2005, from <http://www.infonomics.nl/FLOSS/>.
- Golden, B. (2005). *Succeeding with Open Source* (pp. 1–39). Addison Wesley Professional.
- von Krogh, G., Haefliger, G. S. and Spaeth, S. (2003). Collective action and communal resources in open source software development: the case of freenet. <http://opensource.mit.edu/papers/vonkroghhaefligerspaeth.pdf>.
- Laurent, A. M. S. (2004). *Understanding Open Source and Free Software Licensing*. O'REILLY.
- Lehman, F. (2004). Floss developers as a social formation. Retrieved November 21, 2004, from <http://www.firstmonday.org/issues/issue9-11/lehmann/index.html>.
- Lindman, J. (2004). *Effects Of Open Source Software On The Business Patterns Of Software Industry*. Master's Thesis, Helsinki School Of Economics (HSE) Department of Management, Retrieved November 21, 2005 from <http://opensource.mit.edu/papers/lindman.pdf>.
- Margret, E. and Scacchi, W. (2003). Free software development: A case study of software development in a virtual organizational culture. <http://opensource.mit.edu/papers/eliottscacchi.pdf>.
- Marx, K. (1993). *1844 Elyazmaları* (pp. 138–204). (K. Somer, Trans.). Sol Yayinlari.
- Marx, K. (2003). *Grundrisse Ekonomi Politiğin Eleştirisinin Temelleri 2. Cilt* (pp. 94–99). (A. Gelen, Trans.). Sol Yayinlari.
- McGowan, D. (2001). Legal implications of open-source software. *University of Illinois Law Review*, Vol. 2001, p103.

- McMillan, D. W. and George, D. M. C. (1986). Sense of community: A definition and theory. *Journal of Community Psychology*, 14, 6–23.
- Nayyer, K. (2001). Globalization of information: Intellectual property law implications. *First Monday*, Retrieved October 3, 2005 from [http://www.firstmonday.org/issues/issue7\\_1/nayyer/index.html](http://www.firstmonday.org/issues/issue7_1/nayyer/index.html).
- Pestimalcioglu, G. (2003). A study of free software movement: Towards a new society? *Unpublished master thesis, Department of Sociology, Graduate School of Social Sciences, METU.*
- Pinchot, G. (1999). The gift economy. *Business On A Small Planet*, IC 41, p.49, Retrieved October 2, 2005, <http://www.context.org/ICLIB/IC41/PinchotG.htm>.
- Raymond, E. (1999). *The Cathedral and the Bazaar*. Retrieved August 15, 2004 from <http://www.catb.org/esr/writings/cathedral-bazaar/>.
- Raymond, E. (2000). *Homesteading the Noosphere*. Retrieved October 3, 2005 from <http://www.catb.org/esr/writings/cathedral-bazaar/homesteading/>.
- Raymond, E. (2001). *How To Become A Hacker*. Retrieved December 4, 2005 from <http://www.catb.org/esr/faqs/hacker-howto.html>.
- Raymond, E. (2003). *The Art of Unix Programming*. Retrieved October 3, 2005 from <http://www.faqs.org/docs/artu/index.html>.
- Sennett, R. (1998). *The Personal Consequences of Work in the New Capitalism*. London: W. W. Norton & Company.
- Smith, M. A. and Kollock, P. (1999). *Communities in Cyberspace* (p. 230). London:Routledge.
- Stallman, R. M. (2002). *Free Software, Free Society: Selected Essays of Richard M. Stallman*. GNU Press [www.gnupress.org](http://www.gnupress.org) Free Software Foundation Boston, MA USA.
- Tarrow, S. (1998). *Power in Movement*. Cambridge University Press.
- Tonnies, F. (1958). *Community and Society*. The Michigan State University Press.
- Torvalds, L. and Diamond, D. (2001). *Just for Fun: The Story of an Accidental Revolutionary*. HarperBusiness.

- Valimaki, M. (2005). *The Rise of Open Source Licensing: A Challenge to the Use of Intellectual Property Licensing in the Software Industry* (pp. 13–49). Turre Publishing.
- Weber, S. (2004). *The Success of Open Source*. Harvard: Harvard University Press.
- Weinberg, G. M. (1998). *The Psychology of Computer Programming, Silver Anniversary Edition*. New York: Dorset House Publishing.
- Zeitlyn, D. (2003). *Gift economies in the development of open source software: Anthropological reflections*. Retrieved November 21, 2005 from <http://opensource.mit.edu/papers/rp-zeitlyn.pdf>.