

The power of gifts: organizing social relationships in open source communities

Magnus Bergquist & Jan Ljungberg

The Viktoria Institute, Göteborg, Sweden, email: magnus@viktoria.informatik.gu.se,
janl@viktoria.informatik.gu.se

Abstract. *In writings on the open source software development model, it is often argued that it is successful as a result of the gift economy that embraces activities in online communities. However, the theoretical foundations for this argument are seldom discussed and empirically tested. Starting with the 'classic' theories of gift giving, we discuss how they need to be developed in order to explain gift-giving practices in digital domains. In this paper, we argue that the gift economy is important, not only because it creates openness, but also because it organizes relationships between people in a certain way. Open source software development relies on gift giving as a way of getting new ideas and prototypes out into circulation. This also implies that the giver gets power from giving away. This power is used as a way of guaranteeing the quality of the code. We relate this practice to how gifts, in the form of new scientific knowledge, are given to the research community, and how this is done through peer review processes.*

Keywords: Gift culture, open source, peer review

INTRODUCTION

In writings on open source software, it is often argued that software development is founded on gift relationships, in which pieces of code are shared in the form of gifts to the community of open source developers. Raymond (1999a), for instance, examines the open source hacker milieu as a difference between exchange cultures and gift cultures. Gift cultures differ from exchange cultures in that the latter are characterized by scarcity and the former by abundance. In gift cultures, Raymond argues, social status is determined 'not by what you control but by what you give away' (Raymond, 1999a: 99).

Vaguely referring to the classic work by Mauss (1950/1999), Raymond catches some fundamental and important aspects of the cultural logic of gift-giving and sharing practices that are vital for the success of open source software development. However, he does not really dig into the theoretical foundations that lead to these conclusions. In this paper, we seek to develop Raymond's arguments by relating them to the work of Mauss and his followers.

However, the role of gift economy in modern society is not well researched, and a barrier to this seems to be a tendency to see them as archaic customs (Cheal, 1988). Where the analysis of Marcel Mauss of gift giving in 'primitive' cultures is based on the giving of material objects, the understanding of gift giving in open source communities has to be understood as a form of gift giving in a digital world interconnected via global networks.

As a consequence of the understanding of these differences, three themes are going to be discussed:

1. Living in a society dominated by commodity transactions, open source activists have to be socialized into the culture of gift economy. We use some examples to show how the basic norms and values are taught to newcomers (so called 'newbies').
2. Gift giving is, according to Mauss and others, a way of creating and maintaining relationships of power between groups and individuals. Open source gift giving transforms these relationships to interdependencies based on the idea of reputation.
3. Producing good code is important to open source developers. Making the source code open for inspection by peers is regarded as a guarantee of the high quality of source code. Practices have evolved to assure that the best code produced is integrated into the software distributions. Classic theories of gift economies do not clarify how gifts can be treated as parts of quality assessment processes. Here, we will relate open source gift giving to another kind of gift economy – the academic society – and suggest a theoretical framework for how gift giving and peer review systems relate to one another.

The paper is organized in the following way. First, we discuss methodological implications concerning the different data used in the study. A brief overview of open source actors and stakeholders is then given, followed by a more theoretical discussion about gift economy, where we discuss how this theory needs to be developed in order to capture the intricacies of gift giving in digital domains. Two cases are then presented to highlight different dimensions in gift-giving practices. These examples are followed by a discussion in which we relate open source gift giving to theories of peer review in the academic society. This also adds a new dimension to classic theories on gift giving.

METHOD

The empirical data have been collected over a period of 2 years in different domains, such as various kinds of web pages, discussions lists and from publications about open source communities and activities.

The empirical data are of two kinds. One is based on texts dealing with the philosophy and ideology of the open source movement, for instance Raymond's (1999b) writings about open source software and why it has become successful. These texts provide important contributions on the open source phenomenon but, at the same time, they have become icons in the community and should rather be read as descriptions of the public image that the community presents to the surrounding world. Raymond's texts and other documents published on community centres, such as <http://www.opensource.org>, have also been influential in discussions

and debates about the future of open source activities that can be found in discussions lists.

In order to grasp the more dynamic everyday life among open source developers, we have regularly scanned news groups and discussion lists of various sorts. Here, dimensions other than the ones found on 'official pages' (such as <http://www.opensource.org>) can be found. The official pages are characterized by homogeneity and harmony: the open source community seems to be a gathering without social dynamics and conflicts. In discussion lists and comments, a more agitated representation is found. It is important not to consider this image as more 'real' than the formal and official representation of the movement. They exist alongside and shape the discourse of what the community is about.

The method used can be associated with what Hine (2000) describe as virtual ethnography. Examples are used in the text to illustrate and highlight different claims made by actors who relate themselves to open source software development. We are interested in the cultural logic (values, norms) behind different positions and activities undertaken by actors. For example, when we discuss the habit of flaming developers, we are not interested in the flaming *per se*, but why certain activities give rise to this action.

OPEN SOURCE – ACTORS AND STAKEHOLDERS

We will describe briefly the different roles that may occur in open source projects (for overviews of open source, see Raymond, 1999b; Feller & Fitzgerald, 2000; Ljungberg, 2000). Typically, open source projects start with a single programmer solving a small problem affecting his own work. For example, the origin of WWW was the work by Tim Berners-Lee to help high-energy physicists share their work. Another example is Larry Wall, who wrote the script language Perl to solve some problems in systems administration (O'Reilly, 1999). When open source solutions turn out to be significant and attract attention from others, a plethora of different roles arises. Often, people play several roles, e.g. as both developer and user.

The owner of an open source project is the person (or group) who started the project and has the exclusive right, recognized by the community at large, to redistribute modified versions of the software. According to Raymond (1999b), there are basically three ways to acquire ownership of an open source project: to found it; to have it handed over by the former owner; or to volunteer to take over a dying project in which the former owner has lost interest. As the owner attracts contributors, i.e. people that discover the software and want to contribute to its development, he becomes more of a coordinator or project leader.

Often, a group of core developers write most of the code concerning new functionality, review submitted code and make most of the decisions about releases (Mockus *et al.*, 2000). This could be of a more formal nature as in the Apache core development group, where frequent contributors might be nominated as new members to the group and approved by anonymous voting by the group members. In the Apache case, the core group accounted for more than 80% of the contributions (Mockus *et al.*, 2000), and the Orbiten free software survey showed that 10% of the contributors accounted for 72% of the total code base (Ghosh & Prakash, 2000). This means that almost all the new functionality is implemented and maintained by a small group of core developers.

Defect repair involves a much wider development community, an order of magnitude larger than the core group. Problem reporters are an even wider group. Mockus *et al.* (2000) showed that, in the Apache project, the role of system tester is almost reserved for the wide community of Apache users.

User support is a more mundane task not so much linked to reputation or other motivational factors affecting the will to contribute. This task seems to be performed mainly by some product users voluntarily providing answers to the questions of other users (Lakhani & von Hippel, 2000).

Today, there are a huge number of users of open source software, most of them not actively contributing to its development. Also, large organizations have become users. By using the software, these actors can, however, still be seen as important contributors. Creating a critical mass of users is important both for the usability of a system or software and for the construction of a symbolic attraction surrounding open source development style.

Commercial businesses tied to open source projects provide additional resources for developing the free components of the software but, even more importantly, it helps to promote the open source packages and drive them into the mainstream (Osterhout, 1999). Companies make money out of open source software by distributing the open source software, by adding value to the open source software by additional proprietary products or by relating to open source software in different ways, such as bundling it with their own products (O'Reilly, 1998).

GIFT ECONOMY

The open source community is organized around a large stock of devoted hackers who are both producers and users. As there is often no monetary compensation to be expected for efforts conducted, the will to contribute to the community has to be explained in terms other than being based on a more traditional cost–benefit rationality. Raymond (1999a) proposed the idea of open source being a gift culture. Gift cultures are based on gift economies, in which social relations are not regulated by the possession or exchange of money or commodities. Gift cultures are characterized by the creation and maintenance of social relationships based on the economy of gift exchange.

But the idea of gift cultures is a puzzling one. What is it exactly that is given away and why?

The gift-giving relations could be of different kinds, ranging from the gratuitous gift at one extreme to the exploitative relationship at the other (Sahlins, 1972; Godbout, 1998; Godelier, 1999). The classic analysis of the social and cultural context of gift giving is Mauss' *The Gift* (1950/1990). He sees a gift as the transaction of objects coordinated by a system of rules. The rules are in fact symbolic translations of the social structure in a society or a group of members. Mauss does not provide us with a romantic view of gift giving. On the contrary, he argues that giving a gift brings forth a demand for returning a gift, either another object or, in a more symbolic fashion, forces of power connected to the objects. Gift giving therefore creates social interdependencies and becomes a web upon which social structure is organized. To give away something is to express an advantageous position in relation to the recipient.

Mauss frames the general idea surrounding the practice of giving gifts as a culturally defined and socially determining activity in societies. We will come back to this discussion and also take it a bit further by a more close analysis of power relations involved in gift giving in the open source community. But first it is important to distinguish the general idea of gifts and the habit of giving gifts in the 'primitive' societies described by Mauss and the highly technological and, to some extent, virtual societies in which the open source community is located.

Rheingold (1994) often ascribed to the notion of understanding the virtual community custom of giving away pointers, texts, advice or, as is common in open source communities, source code as a special 'gift economy' that is inherent in a more general 'gift culture'. Rheingold describes this as 'a marriage of altruism and self-interest' (Rheingold, 1994: 58).

He also separates two kinds of information-sharing practices. The first can be found among people who are interested in new cultural phenomena that have led them to explore different virtual worlds. They share information about almost everything. But sharing can also mean emotional support. The second consists of professionals who rely on having information constantly at their disposal. This kind of giving away is based on a hunger for intellectual companionship, initially found most commonly among professionals who work more or less on their own, e.g. journalists (see Millen, 2000), freelance artists and designers, programmers, etc. (Rheingold, 1994). Besides a need for social stimulation, they have a shared and immediate demand for accessing relevant information that could not be locally stored or achieved to the extent that is required. By creating a network of contacts, the chances of getting to relevant information sources increases dramatically. At the same time, social relationships are continually developed, creating a kind of social contract in the particular community that arises around a certain type of professional activity. Others, some of them networking only for social reasons, have followed these early adopters.

But the importance of common interests and the Internet as a medium for gift giving is not enough to explain the kind of gift culture that encompasses different online communities such as the open source movement. Kollock (1999) adds to this the character of digital information as an explanation for the intensive sharing of work, social experiences and other forms of knowledge between members of the community. He focuses on the possibility of producing 'an infinite number of perfect copies of a piece of information, whether that be a computer program, a multimedia presentation, or the archives of a long email discussion' (Kollock, 1999: 223). The contributor can give away an infinite number of copies of a document or software without losing it or diminishing its value (which is the case with material objects). The Internet also provides an easy way to distribute large amounts of information. The cost of providing one person with some information is not a greater expense than providing 10 million people with the same information, no matter where in the world the consignees are. And, from the receivers' point of view, the transaction cost of downloading a document is almost the same irrespective of where in the world it is originally posted.

With this in mind, Mauss' discussion of gifts can be seen differently. His argument about the social character of gifts is based on how gifts are perceived and handled in a world of material objects. The scarcity of material objects and the costs associated with transactions

are important for how the value of gifts and the giving of gifts are socially constructed. Information and executable software in combination with the Internet radically transforms the context in which we should understand this relation. But, is then the social character of gifts disappearing when anyone can give anything to anybody with a transaction cost close to zero? The answer to this is no, but this does not mean that gifts are used and experienced the same way in virtual communities as in the kind of societies Mauss described.

Two tendencies can be spotted. The almost total reduction of scarcity problems concerning the dissemination of information has increased the consciousness in the community of the presence of other forms of scarcity. This has led to the development of different practices aimed at decreasing time cycles of innovation and development and scarcities inherent in the organization of cooperation and coordination. Further, the community is primarily a loosely coupled network of individuals with no organizational forces in terms of economy or management that can force other individuals to behave in a certain way. Mauss argues that gifts express, but also create, power relations between people. One of the norms of gift giving is the rule that gifts be accepted. Therefore, the receiver becomes subordinate to the giver. In the open source community, this can be the fact. But the opposite is also important. Refusing to accept a gift can, in some situations, be a way of showing superiority.

In the previous text, we discussed how we should understand theories of gift giving when applying them in digital domains. We will now focus on the practice of gift giving by presenting two cases in the form of excerpts from news group discussions. First, we illustrate how the idea of gift giving is implemented in the community in the form of socialization processes, in which newcomers, so called 'newbies', are taught the rules of the gift economy by more experienced members. We will then relate this to how practising power is linked to gift giving.

Learning the art of gift giving: the newbie case

Newsgroups, frequently asked questions (FAQs) and digests are ways to condense time by making series of questions and answers, mistakes and successes available in a concentrated form. They are easy to access world wide and also cheap.

But, in order to make these tools for information distribution and knowledge sharing the flexible and manageable channels they are intended to be, it is important to socialize new members of the community into a set of norms and behaviours for a certain kind of communication and cooperation practice. A culture of sharing has evolved, which constitutes the community netiquette. The following example is drawn from a Usenet Linux News Group called *alt.Linux* showing how the basic principles in the culture of sharing are established and made sense of. A question is posted by Darren who is going to install a Linux distribution on his computer and wants some advice from the community on which distribution to choose:

A Linux Newbie

Umm hello, before I get to my point, I'd like to say that I'm pretty new to newsgroups in general and much newer to the Linux community, so please forgive any mistakes I make and, if I offend, point it out to me and I will be more than happy to say that I am sorry.

Now, with that out of the way, I'd like to ask anyone who feels like giving an opinion on what distribution of Linux I should get. I am a first-time user of Linux but I have used Windows since windows95 and I have done work in DOS (I am proficient in DOS but hardly fluent). I am currently considering using Linux because I hate having to reboot my P2450 10 times a day just to get Direct X to work properly. To this end, I have set aside a little under 4 gigs of hard drive space for Linux. I am currently considering Red Hat, Caladra, or SuSE. Any opinions? Comments? Reasons to have me committed?

Now if you are going to reply to this (hey it could happen), can I convince anyone to send me an E-mail directly as opposed to just posting to the group? Outlook Express 5 is pretty damn flighty and I don't want to risk losing a valuable opinion. If you would prefer replying to the group that is fine; I am just stating a lose preference.

Also, as I am completely new to Linux, I was wondering if anyone could recommend a good book on it? Maybe on Linux in general and the packages I mentioned before? Good lordy this is a long post; I hope no-one flames me . . . Thank you for your time and I hope I haven't earned your malice.

Sincerely, Darren

It is obvious that Darren is familiar with some of the basic rules for socializing in newsgroups, even if he says that he is new to newsgroups in general and the Linux community in particular. He knows that there are certain rules for communicating and that different communities have different rules. He also knows that postings should be kept short and concentrated in order to create a current flow of questions and answers. He articulates a fear of having exceeded an invisible prescription for this specific community and that he is running the risk of being punished the usual way: being flamed (getting hundreds of angry mails from upset 'listeners') in a rude and arrogant way. The strategy used is to approach the community in a humble and respectful way, trying not to upset anyone, even if he breaks some of the rules. He 'pretends' that he does not even expect an answer from the community but, if he does, he wants to make sure that the information (which he presupposes is very valuable) is not lost because of technical errors.

The term 'newbie' is used widely to describe both others and oneself as a dilettante. It is used here in a self-stereotyping way to play down the odds that the community would look upon Darren as a self-sufficient person. But, having stated his relation and attitude towards the community, Darren attempts to establish a relationship with the potential listeners by telling them something about his personal situation. This is achieved by describing what kind of hardware and software he is using and by stating the problems running the present system, tools and applications.

The strategy is to establish a sense of loyalty towards the community. By 'talking technology', he tries to make himself accountable as fairly knowledgeable when it comes to handling computers in general. The will to abandon the existing platform in favour of a Linux distribution is a kind of a symbolic token given to the community in order to get something valuable back. But this has to be done in a reasonably smooth way. The decision to change platform is not based on blind idealism, but because of bad experiences with the existing equipment.

Instead of naively celebrating the community and its members, Darren appeals to a certain kind of rationality that, he seems to expect, is addressing a set of community-relevant values.

Instead of reading the posting 'as is', we can understand it as a way to become an insider and thereby gaining access to the knowledge and experience that resides in the community. The answers threaded in the discussion also confirm that some members of the community have accepted Darren as an earnest enquirer and that he has acted in an appropriate way. Twenty minutes after the question is posted, 'Leonard' posts a message that contains not only the information sought after by Darren, but also an additional netiquette handbook appropriate to community standards, pinpointing the rules that the members of the community relate to when networking in newsgroups:

Re: A Linux Newbie

Well Darren, I think you did ok . . .

You didn't post in HTML

You didn't cross-post

You didn't post a binary you weren't a jerk and were generally polite . . .

I'm going to assume you took the time to catch up with the newsgroup before posting to make sure your question hasn't been answered so all proper netiquite (sp.?) has been obeyed. How nice . . .

I used Redhat Linux 5.2 when I got started. If I was starting today I would buy the 6.0 or more likely the Mandrake as it is the same stuff for 1/3 the price . . . and I hear it's easier to install.

The learning curve is mighty steep at first, but it gets better and you'll wonder why you waited so long.

As far as books, I found 'Mastering Linux' to be basic, outdated and at times inaccurate . . . and it was far too reliant on the X interface. I actually found the 'Linux for Dummies' book had exactly the same information in much less space, and therefore was much more helpful (despite the title and the stigma attached to it) and dealt more with the command line. I'm looking for a good intermediate book now . . .

Good luck

Leonard

As we can see, humility is a strategy to get acceptance from the community, and this is also what the more seniors encourage. In order not to get flamed, inexperienced users often headline their question 'newbie' followed by a topic that is developed further in the message. The norms and values of the community are internalized through these kind of narrative practices.

Network information is important. Being accepted by the community means getting answers to proposed questions. But more important from the point of view of socialization is the interchange of information as a way of creating mutual alliances. Instead of seeing the situation as a kind of 'information retrieval' situation in which one person receives a piece of information from another, we should understand it as a mutual interchange where one gift is given for another. The newbie gets important information about a subject, or information about how to

get further into the community, but also pays back by being willing to mould himself or herself according to community norms and values. 'Looking good' on a Discussions List, with respect to community values and rules of behaviour, can open up doors to further networks of information and contacts with relevant people. But it also means that a certain kind of power relationship is made explicit by some and internalized by others. As a result, newbies are socialized into the specific norms of the community.

The newbie status has, however, developed as a project of its own. An important site for developing, discussing, formalizing and branding the status of newbies is Linuxnewbie.org, a portal created with the explicit aim of being a space where new but generally inexperienced Linux users can meet, get information, pose questions and socialize. Norms are not accepted without protest, and countermovements are being developed.

Power in a gift economy

One easily gets the impression that the sharing of gifts in online communities creates a very friendly and altruistic atmosphere. And indeed it does, to some extent. But it does not mean that social stratification and struggles over power cease to exist. When giving is easy and gifts become non-reducible, dependencies are reshaped and transformed to new kinds of relationships. Gifts are often not given to anyone in particular. They are made public (on web pages) and thereby made available to anyone who cares to make use of them. An application or some information does not really become a gift until someone finds it and makes use of it. If a giver manages to get attention, people will turn the things offered into gifts, which means that a relationship is created between the giver and the user. As gifts do not imply a monetary compensation, virtual community gift giving is managed through acknowledgement: the giver is 'paid' by the community by receiving a certain amount of fame and respect.

As discussed above, giving digital gifts does not remove the interdependencies between givers and receivers. But social relationships and obligations between the parties involved in the exchange are restructured based on the character of digital information and the Internet as a social world. In this section, we will discuss this process of social restructuring.

The difference between gift giving and commodity transactions is that the giving of gifts should not involve explicit bargaining. In bargaining, no obligation exists after the exchange is accomplished. Where commodities are exchangeable, gifts are unique. A commodity that is purchased, and then given as a present, is transformed from a product to an obligation. By giving away, the giver shows superiority and the receiver becomes dependent. There is an obligation to repay the gift in the future. As Mauss argued, a gift can be a way of showing social status. The giving of artefacts creates asymmetrical social relationships, especially if the gift can never be refunded and the giver knows it. Cheal (1988) therefore talks about gift giving as based on a moral economy.

On the Internet, however, the receiver is often unknown to the giver (Kollock, 1999). Gifts are placed on various homepages and ftp sites, and anybody can download a piece of information or an executable file and use it for various purposes. But this only counts for the Inter-

net in general. The interesting question is in which social context gift giving on the Internet gets its social meaning. The focus for the production of meaning in the gift economy on the Internet is the various kinds of communities in which people share some understanding of the context they are involved in. They are not unknown to each other, which does not mean that they have to be personally acquainted. The open source community has also become a major player in the development of norms and behaviours related to gift economy as a result of this being something officially dealt with in the community. In these communities, the value of the gifts is dependent on the amount of attention the giver gets from the receivers that choose to make use of the gift.

Rather than taking place in a world of objects or artefacts, gift giving in online communities takes place in a world of information. There is no obligation to pay back but, for the ordinary end-user, there is a moral obligation to pay back in the form of having some solidarity to the community by praising the software, the project owners and/or the major contributors.

There are differences between the meaning and use of gifts in purely social communities and the more product-oriented communities such as the open source. In social MUDs, there are 'only' social relationships to 'deal' with. In some ways, open source gifts look more like traditional commodities and can be used as both a product and a gift for creating and maintaining relationships. For the end-users that do not want to participate in the community, the difference is not so great. Internet technology makes exchange open. Anybody can get hold of the products. It must work this way when the idea is to spread the products in order to make it possible for as many as possible to take part in the development process and add some of their expertise to the ongoing project.

Becoming an owner of a project that becomes successful gives the right to decide who should be entitled to give back and who should not. The more attention an open source project owner gets from the members of the community, the more status and reputation he achieves. And the more a project gets attention, the more users would like to contribute to it and become a part of the project and, as a result of this, get some of the attention (Raymond, 1999a). It is important to note that our use of the concept attention differs considerably from the way it is commonly used in the context of attention economy, e.g. as attention span of an individual (Goldhaber, 1997) or 'focused mental engagement' (Davenport & Beck, 2001: 20). In this paper, attention relates to the logic of how social capital and reputation are formed and gained. Attention in the context of open source is not about being bold and loud, but about being recognized or 'seen' as a good programmer and contributor to the community. The right sort of attention then creates status and reputation. The power structure, compared with gift giving in a world of artefacts, is therefore restructured according to the logic of how reputation is formed in the community.

How then do people who are, or try to be, active in the community relate to this circumstance? In the online magazine *Freshmeat*, an editorial argues for the importance of giving gifts and thereby helping the community to develop high-quality software for the benefit of mankind. The article is followed by 15 comments from readers in which values and norms in the open source gift culture are discussed in relation to how new kinds of power structures are established on the Internet. It is worth noticing that the discussion of power is a reaction

to the editor's rather ideal formulation of why hackers should contribute unselfishly to different software development projects.

The discussions reveal that some of the developers/users experience power relationships that are expressed as 'an elitism of the inner circle' and exercised as the right to hinder a person in contributing to the common good. It is argued that the openness of the open source movement is overstated. Practices such as flaming are not only used to exclude people who overrule the core values of the community. In the newsgroup example below, the author of the posting claims that it is also used to exercise power over those who want to contribute to the community by delivering code to different projects:

Giving Back

siberian – April 01st 1999, 21:31 EST

More than once I have had the urge to begin contributing to the community. I have written code, documented it and gained authorization for its release. But at the last minute I always hesitate and then stop. Why? I think I fear the fangs of the community. At this point, everywhere I turn, it's a big flamefest and getting quite tiresome. Its gotten to the point where it seems one has to be some sort of Jedi Master level coder to contribute. On more then a few mailing lists, I have seen contributors flamed for their contributions! Flaming someone for GIVING something away. Its incredible.

On the one hand, there is a message of openness, not only when it comes to source code, but also that openness is a fundamental feature of open source as an organizational entity. And indeed there is openness, in the sense that everything that is shared among the community members is on display on web pages and discussion lists. But, on the other hand, does the honouring of the successful projects and the heroes behind them create boundaries of in-groups and out-groups? There are the ones who are part of projects, in terms of being accepted as code providers, and there are the ones who want in but are rejected by the project owners or flamed by other members of the community.

Another commentator offers some advice on the discussion list. The creation of power relations is recognized but located in a certain social context in the community:

Flaming

Charles P. Wright – April 02nd 1999, 02:56 EST

You really don't turn into some sort of lightning rod for flaming if you release your software. That is really only people who put themselves in high positions (esr, rms) who are subject to that. If you start your own project you should feel safe from that kind of crap. And if you get a less than polite email every once and a while you can just delete it. Knowing people like your project and having dozens of people contribute code is a real ego boost, not to mention you get better software than you had yesterday because of it!

Some people, it is argued, place themselves in high positions ('esr' and 'rms' stand for Eric S. Raymond and Richard M. Stallman, who are two of the most influential people in the community). The strategy proposed in the posting above is to ignore people in high positions and people who flame for no reason. The advice is that, if a person is afraid of flaming, it is more 'safe' to start a project of your own than to deliver code to an existing project.

If the project is not successful, no one will notice it. If it is a hit, a lot of people will try to join and contribute. However, there is a contradiction in this statement. If everybody starts projects of their own rather than helping others with existing projects, no collaboration will occur. Another and perhaps more important topic is how a person should relate to flaming, and what flaming really is about.

Virtual collaboration puts high demands on people having trust in one another. Giving away the best piece of code a person has produced demands strong social ties between the giver and the receiver. The receiver has to present himself and act in a trustworthy way, otherwise the giver will not give away his code. At the same time, the demands for high-quality source code give the project owner the moral right to judge whether a contribution has high enough quality to be considered as part of the final distribution. It becomes important for the receiver to make himself trustworthy in the eyes of the givers. Virtual collaboration and gift giving must thus be based on trust in order to make criticism regarded as something that can contribute to the overall quality of the products:

Please code!

jimduchek – April 01st 1999, 22:54 EST

Don't let that stuff put you off! Don't *_ever_* listen to anything from anyone you don't respect. The only way to have your code get really good is peer review. If they flame it, they flame it. So what? I've never seen this happen myself, though. Ignore the flames and listen to the REAL comments on your code. As a developer myself, I rarely use patches exactly as they come from user/developers, but I do use almost all of them in some form or another. *_NEVER_* write something and don't submit it, it goes against the whole philosophy of free software. Even if it sucks, you'll likely get at least one mail that says 'Your code won't work for us for this reason, and this reason, and this reason.' you just learn from that and keep going. And if not, it gets used, and you get a big ego boost! Ignore the script kiddies, they don't matter a damn bit.

The advice presented is to develop an ability not to listen to critique from someone one does not respect. Filter the flames from good critique, ignore the flames and listen to the good comments.

A problem with the openness of the Internet and the open source community is that anybody can comment even if they are not really serious about the matter in question. One must have the ability to be indulgent towards weirdoes, overlook 'script kiddie' critique, shrug one's shoulders and move on. Good critique should be taken into account, because it is dangerous to ignore critique, as good development is dependent on peer review processes. One has to fight against the 'reinventing-the-wheel' syndrome that seems to be lurking in the virtual bushes. The two strategies proposed are 'trust', if the critique is good, and 'ignorance', if not serious. Talking about a certain kind of critique as 'flaming' creates a mental distance that becomes a resource for the individual in this highly exposed environment:

Nobody has the right . . .

Angler – May 23rd 1999, 18:41 EST

There are always rude people around. Of course they'll take their sweet time out to flame

newbies, or act snobbish. But they seem to be mostly outweighed by the people who are nice, helpful, and patient. One rude email does not a group attitude make. But they are the most vociferous, and the ones we tend to remember with bile. But it's unfair to generalize. I think though that just saying that no one has the right to criticize your software is a dangerous path. Criticism can be constructive – that is, administered with the minimum amount of flame (i.e. none). And you won't learn as quickly if you can't take criticism.

The other kind of criticism – flaming – you can just ignore. But even some rude emails have valid, if stretched, points.

As for newbies getting flamed . . . I believe it happens. But then, such things always happen. There are always jerks in any community, but I don't think it's as prevalent for Linux as some of us believe through rotten experience. If you don't believe me, join a LUG.

If a development process is to produce quality software in a compressed time frame, there must be a hard critique. The problem is who has the right to decide over others. The critique against those who feel that their attempts to give to the community have been ignored is brought back to the fundamental principles of the community. The only reason to deny or refuse a contribution is the ambition to maintain and develop the good quality of open source software development, which is also the core argument for spending time developing such software:

tight code

thi – May 29th 1999, 21:00 EST

flaming newbies happens so newbies can learn how to flame. the essence of improvement is managed change. when the newbie learns to apply the flame to the code and not to their ego, the code is reduced and enhanced. cruft is vaporized, bloat melted, misdesigns highlighted. the remaining skeleton is tight code, on which users drape their cpu cycles to their own needs.

flames by newbies should be culled for the good parts and the rest ignored. this is similar to how one should treat code from newbies. thus the only difference between newbie and non-newbie is completeness.

completeness is rarely well-defined, unfortunately.

This also becomes a way of socializing members. Supporting the norms surrounding the culture of gift giving and freedom of getting in and out of the community and, at the same time, making actors conform to a certain elitism is a tricky business, where different ideologies and individual standards for behaviour occur in sometimes agitated discussions. Being a project owner of a development project that gets noticed is often met with respect. But this esteem is always questioned by the celebration of 'the culture of individuality' and the weak imperative necessities in the form of social ties between contributors and owners.

GIFT GIVING AS PEER REVIEW

As we have seen, it is important in the open source community regularly to return to a basic theme that concerns the question why gift giving is important in order to support open source

software development. Values and norms surrounding quality assurance and the sharing of knowledge become a key theme in understanding the open source community's cultural foundations. Here, we argue, academic research has much resemblance to the gift economy, in the sense of non-altruistic gift transactions.

For instance, both Raymond (1999a) and Bezroukov (1999) have noticed the similarities between the open source communities and research communities in academia. If we view open source from the theoretical perspective of peer review, we could gain a more in-depth understanding of both the quality aspect of open source software development and how power relations are socially organized.

In the academy, you give away your knowledge, not because you are altruistic, but because that is the way of career progression within the academic field. You give away knowledge and information in return for status and reputation. The acceptance of a gift by a community implies recognition of the status of the donor and the existence of certain reciprocal rights. Scientific contributions are gifts, as authors normally do not receive royalties or other payments for the publication of results in a journal. Manuscripts for which scientific authors do get paid as textbooks and popular writings are certainly held in much lower esteem (Hagstrom, 1982).

By sharing knowledge and being open about results and methods, the results can be justified and replicated. Others can give contributions by responding or by continuing on the published work, pushing the scientific frontier. By writing and publishing papers and by being referred to by others, you not only share your knowledge, but also become visible in the academic community. The reputation is secured by the rule that one can use knowledge produced by somebody else, but it must always be clear from whom the idea originates. The more other researchers quote your publications, the more your reputation will grow. In scientific communities, it is a mortal sin to use somebody else's knowledge without acknowledging the person who came up with the original idea. Thus, the failure to recognize previous work of others may give rise to strong antagonisms and intense controversy.

The principles for handling the contributions to the scientific community, peer review, were developed through the establishment of a referee system in the first scientific journals in seventeenth- and eighteenth-century Europe (Ziman, 1968; Chubin & Hackett, 1990). It is by now a fairly established academic designation for marking the scientific quality of a piece of work, for assigning a candidate to a position or for approving or rejecting a research proposal. All established researchers are more or less involved in peer review processes, as reviewers of others' articles, project proposals or CVs or by being reviewed themselves.

The peer review system makes it possible to judge whether contributions are good enough to enter the field. Among many contributions, the best ones are chosen. When people make selections, they try to be reasonable in their selections. But what seems reasonable for one person is not always reasonable for another. There are no objective criteria for what counts as 'relevant', 'interesting' or 'new' research. Peer review is thus a social mechanism, through which a discipline's experts or the core members of a community maintain control over new knowledge entering the field (Merton & Zuckerman, 1973; Chubin & Hackett, 1990). Peer

review can thus be seen as a way of organizing power relationships within a given community or in relation to other fields of activity.

The open source communities are driven by similar norms. You write a piece of software and provide it to the community. Your contribution is peer reviewed by the owners of a software development project and, if it is good enough, you get your credits in the open source gift economy. A good idea is usable in further research but also gives the owner credits. There is an interesting relationship between how usable an idea is and how much attention it can get. In the academic society, a good theory seem to be more tradable than a partial analysis. Theories can potentially be of general benefit to large parts of the research community. There seem to be a similar relationship concerning the rules governing how ideas are valued in the open source movement. A generally applicable service such as Linus Torvald's Linux operating system becomes, like a theory in the academic society, a general base for others to act upon. This makes Linux famous and Torvalds a respected member of the community. Writing a driver for an application is similar to, let us say, an empirical study in which a theory is used.

CONCLUSION

In this paper, we have discussed the relationship between the habit of giving gifts and the organization of social relationships between members in the open source community.

It seems to be true that the Internet and the increasing use and importance of digital information have changed the way social relationships are created and maintained in virtual environments compared with those face-to-face settings once studied by anthropologists such as Marcel Mauss, for instance. Understanding gift-giving practices also calls for an understanding of the fundamental social principles on the Internet as a context for human activities. Gifts become valuable when everybody wants to be part of giving them.

Giving away ideas and source code is the base for different kinds of activities in the open source community, which also becomes fundamental for the creation of culture. Values and norms are organized around a giving and/or getting axis. However, this does not mean that people are unable to execute power or make a career, which presupposes some kind of hierarchy. The hierarchy in this virtual environment is a matter of giving or receiving more or less attention. As we have seen, the giver–receiver relationship also changes over time and must be understood as a dynamic process with no fixed dependencies.

Finally, we have discussed gifts and reputation in relation to theories on peer review in order to understand the developers' ambition to guard the quality of the code produced. Producing high-quality source code is the key motivator for members when understanding the open source community as a social actor. The system of giving credits to those that the community thinks deserve it is therefore accepted and even favoured among the community members. One could understand this culture as a kind of amalgamation of collectivism and individualism: giving to the community is what makes the individual a hero in the eyes of others. Heroes are important influences but also powerful leaders. Their presence is important as a way of converging different values and goals in this heterogeneous context, focusing it on common goals, but also legitimizing the asymmetries in power relationships.

REFERENCES

- Bezroukov, N. (1999) Open source software development as a special type of academic research (critique of vulgar Raymondism). *First Monday*, 4 (10). [http://firstmonday.org/issues/issue4_10/bezroukov/index.html].
- Cheal, D. (1988) *The Gift Economy*. Routledge, London.
- Chubin, D.E. & Hackett, E.J. (1990) *Peerless Science. Peer Review and US Science Policy*. State University of New York Press, Albany.
- Davenport, T.H. & Beck, J.C. (2001) *The Attention Economy*. Harvard Business School Press, Boston, MA.
- Feller, J. & Fitzgerald, B. (2001) A framework analysis of the open source software development paradigm. In: *Proceedings of the International Conference on Information Systems (ICIS)*, Brisbane, December 2000.
- Godbout, J.T. (1998) *The World of the Gift*. McGill-Queen's University Press, London.
- Godelier, M. (1999) *The Enigma of the Gift*. Polity Press, Cambridge.
- Ghosh, R. & Prakash, V.V. (2000) The Orbiten free software survey. *First Monday*, 7. [http://www.firstmonday.org/issues/issue5_7/ghosh/index.html].
- Goldhaber, M.H. (1997) The attention economy and the Net. *First Monday*, 2. [www.firstmonday.dk/issues/issues2-4/goldhaber/].
- Hagstrom, W. (1982) Gift giving as an organizing principle in science. In: *Science in Context*, Barnes, B. & Edge, D. (eds). The Open University Press, Stony Stratford.
- Hine, C. (2000) *Virtual Ethnography*. Sage Publications, London.
- Kollock, P. (1999) The economies of online cooperation. In: *Communities in Cyberspace*, Smith, M. & Kollock, P. (eds). Routledge, London.
- Lakhani, K. & von Hippel, E. (2000) *How Open Source Software Works: 'Free' User-to-User Assistance*. Working Paper 4117. MIT Sloan School of Management, Boston, MA.
- Ljungberg, J. (2000) Open source movements as a model for organising. *European Journal of Information Systems*, 9 (4).
- Mauss, M. (1950/1999) *The Gift. The Form and Reason for Exchange in Archaic Societies*. Routledge, London.
- Merton, R.K. & Zuckerman, H. (1973) Institutionalized patterns of evaluation in science. In: *The Sociology of Science*. Storer, N.W. (ed.). The University of Chicago Press, Chicago.
- Millen, D. (2000) Community portals and collective goods: conversation archives as an information resource. In: *Proceedings of HICSS33*.
- Mockus, A., Fielding, R. & Hersleb, J. (2000) A case study of open source software development: the Apache server. In: *Proceedings of the International Conference on Software Engineering (ICSE) 2000*. Limerick, Ireland.
- O'Reilly, T. (1998) *The Open Source Revolution. Release 1.0, Esther Dyson's Monthly Report*. [www.edventure.com/release1/1198.html] (22 August 1999).
- O'Reilly, T. (1999) Lessons from open source development. *Communications of the ACM*, 42 (4), 33–37.
- Osterhout, J. (1999) Free software needs profit. *Communications of the ACM*, 42 (4), 38–39.
- Raymond, E.S. (1999a) Homesteading the Noosphere. In: *The Cathedral and the Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary*, Raymond, E.S. O'Reilly and Associates, Sebastopol, CA. [www.tuxedo.org/~esr/writings/homesteading/homesteading.html].
- Raymond, E.S. (1999b) *The Cathedral and the Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary*. O'Reilly and Associates, Sebastopol, CA.
- Rheingold, H. (1994) *Virtual Community*. Minerva, London.
- Sahlins, M. (1972) *Stone Age Economics*. Aldine-Atherton, Chicago.
- Ziman, J.M. (1968) *Public Knowledge. An Essay Concerning the Social Dimension of Science*. Cambridge University Press, Cambridge.

Biographies

Magnus Bergquist holds a PhD in cultural anthropology. He works at the Viktoria Institute, Gothenburg, Sweden, where he is active in the Knowledge Management Group. He is currently involved in the research project entitled Knowledge Organizations of the Future. Research topics include: knowledge organizations, relationships between organizational and cultural contexts, virtual organizations and scenario planning.

Jan Ljungberg holds a PhD in informatics. He is the manager of a research group on knowledge management at the Viktoria Institute and also Associate Professor at the Department of Informatics at Göteborg University. His current research interests include knowledge management and virtual forms of organizing.