



Boundary object as a trust buffer. The study of an open source code repositoryⁱ

Małgorzata Ciesielska

Teesside University Business School, United Kingdom

ciesielska@ciesielska.eu

Gregers Petersen

University of Copenhagen, Denmark

gregers@wireless-ownership.org

Keywords

Trust

Distrust

Open source software

Boundary

Abstract

This paper discusses the impact of the Internet as communication technology and work environment on changing professional relations and work practices. In particular the paper contributes to a better understanding of collaboration processes enabled by the web and hereby draw a field for further research on trust within Open Source Software (OSS) communities. Having two contradictory interpretations of the technical structure of the project development we indicate its ambiguity to make a point about the meaning of trust and distrust in crating the space for cooperation. Neither trust is necessary nor distrust destructive thanks to the boundary object – the source code repository – that serves as a sole meeting point between OSS community and the company.

Setting the stage

Creative processes may emerge from a process of negotiating multiple and potentially competing interests (Drazin, Glynn & Kazanjian, 1999:291) and identities (Jemielniak, 2010). This is in particularly visible within a group of the new knowledge workers, like software engineers (Hunter et al, 2010), which we focus on. In this setting we discuss the concept of trust as a basis for cooperation both in the open (Ciesielska, 2010; Ciesielska & Iskoujina, 2012; Chesbrough, 2006, Westenholz, 2012) and virtual collaboration (Greendberg et al., 2007; Jarvenpaa, et al., 1998).

This paper will take a look at the example of the source repository structure of a specific open source software project – here called Goblin – in order to make sense of it in terms of trust relations. A source repository is a form of dynamic online storage of the source-code a project is producing, and access is typically regulated through individual user-rights. We use an empirical example of an open source project in which business participants are working from within semi-

closed subprojects integrated into the repository structure. While the normal project developers – individual hackers - are working on a fully open basis. The main source-code tree is synchronized (upgraded) from both sources through the review process of the core group of developers.

We found the structural form of the Goblin project to contain ambiguous elements. In this paper, driven by the empirically rooted curiosity, we aim at explaining what the source tree of the application is for different participants and what purpose it serves in terms of enabling collaboration and building/communicating trust and or distrust. To analyze this ambiguous structure we are using a theoretical concept of a “boundary object” (Wenger, 2000). Although we believe that the source repository is much more than just a boundary object, taking this standpoint together with interpretative paradigm let us to focus on sense-making aspects and trust-distrust play undertaken on the junction between different communities of practice. We are placing our stories of trust and distrust in a particularly challenging context of commercial partners of open source software (OSS) community (Weber, 2004; Ciesielska, 2010; Ciesielska & Iskoujina, 2012). On the one hand, business participators with their specific rationales and fear of being too open and thereby to benefit their competitors. On the other hand, the open source logics, which take a stand against "free riding" and violation of licensing requirements, often associated with business conduct. At the same time, as a consequence of social and technological changes it is observable that pre-existing boundaries between corporate and community partners are being blurred (Jaaksi, 2006; 2007). This makes the field of OSS communities and public collaborative processes relevant for discussing trust building processes in the Internet era. The type of source repository structure described in this paper seems to share similarities with other OSS communities. Therefore it contributes not only to the theoretical discussions, but also offer empirically embedded analysis, which practitioners may find relevant for their work (Jemielniak, 2006).

Trust, façade of trust and boundary objects

Rothstein (2007) argues that the literature addresses different conceptualisations of trust, although the main differences allow for looking at it as a result of utilitarian calculations (figuring out who is worth being trusted in terms of persons' interests) or as a moral orientation (being trustful towards others). Similar remarks can be found in Greendberg *et al.* (2007), who refer to two ‘traditional’ sources of trust: (1) a cognitive trust that arises from the assessment of another person's integrity and abilities and (2) an affective trust that is linked to social bonds and benevolence. According to Lewis and Weigert (1985), cognitive trust provides a basis for the creation of affective trust relations and, therefore, can be considered as a positive antecedent of affective trust (McAllister, 1995; Johnson and Grayson, 2005).

Hardin (2006) argues that only cognitive explanations of trust emergence give interesting insights. According to him, other conceptions of trust, like moral commitments and psychological or character disposition, do not lead to a better understanding or creation of general trust theories. Although some may not completely agree with Hardin, as certain elements of the affective trust might have an impact on interpersonal and intraorganisational relations, this paper focuses primarily on the analysis of the cognitive aspect of trust—encapsulated interests, as Hardin would call them (i.e., perceived common interests of two parties)—judged by the common collaboration history and “proper” behaviour of the other party over a certain period of time.

The trust is believed to be a prerequisite for successful collaboration (Gambetta, 1988; Hardin, 2002; 2006; Lane & Bachmann, 1998/2000, Ciesielska & Iskoujina, 2012) resource for building social capital (Rothstein, 2007) and potential condition for knowledge sharing between communities of practice. In traditional, rational theories, trust is understood at interpersonal level and explained via positive relationship and common interests.

On the other hand distrust is considered as a destructive force that has to be dealt with. When distrust comes onto the scene there must be a replacement introduced. For instance, when a company does not trust its subcontractor, they refer to contracts and the trust they have in the institution of law (Latusek, 2007). Or alternatively, if one of the partners has more control over the situation, they can suppress distrust by creating a façade of trust (Ciesielska, 2010).

Hardy, Phillips and Lawrence (1998/2000) adds to this discussion a concept of facades of trusts, which are simply a creation of illusion of trust while being in reality power based relationship. According to their model trust can be either spontaneously created or generated on purpose but it is equally possible to use asymmetrical power structure to create façade of trust. Spontaneous trust emerges naturally during cooperation as predictable behaviour ensures partners in their good will and engagement. There are two likely sources of shared meaning: it can be institutionalized in the community of practice or it can be a by-product of social interaction. Trust can exist then as institutionalized (system trust) or communal identity (personal trust). However, rarely trust is natural and fully spontaneous. On the contrary participants often generate

it through communicative activities. They try to ensure that shared meaning is mutually constructed and in order to bring about common benefit.

However, in both cases trust is based on shared meaning, which either already exists (institutional order) or is mutually constructed during collaboration (generated trust). Whenever meaning is shared but is distorted or imposed by one partner Hardy, Phillips and Lawrence (1998/2000) refer to this situation as facade of trust. Cooperation emerges through management of meaning (manipulation) or through dependency and socialization (capitulation). They argue that relationship and co-operation may seem like trust when it is based on imbalance and power of one partner over the others. The development of the Internet communication and open source project in particular, make these kinds of explanations not fully applicable or at least do not lead to explaining how individuals and companies make sense of online open collaboration. However, cooperation may as well emerge in a situation of lack of trust or even distrust, even when there is no real power difference between partners. The existence of commercial partners in open source projects is the best example of this occurrence.

It became questionable if impersonal contacts; online/virtual teams and geographical distributions of participants automatically lead to the opening of developing other ways of reducing risks and building trust. Those specific aspects are raised in the literature on open source movement (Matzat, 2004; Osterloh & Rota, 2004) in form of institutional or system trust (the way open source projects are run) or swift trust (based on the assumption of common engagement in the project 'good'). It is also an interesting perspective mentioned in studies of the commercial IT/software sector (study of distrust by Latusek, 2007) as well as online communities (Wiertz & Ruyer, 2007). However, enabling conditions and dynamics of trust relations between business and independent participants in OSS projects has not been given enough focus.

Here we believe the concept of boundary object becomes helpful. After Star & Griesemer (1989) we define boundary object as abstract concepts or concrete "things" which exists in several intersecting social worlds. Although they have different meanings and serve various informational requirements of different parties, they serve as a means of translation. In order to do so, boundary object must be in the same time plastic and robust. That is being adaptable to the needs and constraints of several perspectives but maintain common identity between communities of practice (Wenger, 2000).

The boundary object perspective is especially useful in understanding collaborative situations between heterogeneous partners. We believe that in contrary to many trust theories, it enables to see that full consensus is not necessarily a condition for successful conduct of work. Because communities of practice may see, interpret and understand boundary object differently and the sense-making process between partners do not need to be fully coherent and leading to commonly accepted agreement on all aspects of cooperation. Therefore the boundary objects can be used to forge the collaborating links between groups. Being on the link between differentiated social worlds:

(...) effective boundary object are in constant flux of actual (and potential) co-construction and re-construction at the hands of interacting actors. This boundary constructing, in return, leads to enriched organizational sense-making (Holford et al., 2008:1).

The boundary object framework is very closely linked to sense-making perspective (Weick, 1979, 1995) and communication studies. Boundary construction is an integral part of the sense-making process. Boundaries are "shared social, organizational, and discursive spaces" (Wilson & Herndl, 2007:131). However, boundary objects as artefacts of bounding activity are very ambiguous constructs. On the one hand, they distinguish communities of practice. But at the same time, working out boundary objects is also part of the rhetorical activity that may foster communication and cooperation. Integration and mutual understanding is achieved by common points of reference (Chrisman, 1999). That is why we believe that the existence of the boundary object can help sustaining balance between different interests.

Methodological note

This study used ethnographic methods (Kostera, 2007; Jemielniak & Kostera, 2010). The authors have engaged open source software in terms of an empirical fieldsite (Kelty et al 2009). Understood more closely in terms of the; production and reproduction of open source software in particular settings. The here presented material has been assembled under the umbrella of a larger research project concerned with the intersections between open source software and institutional entrepreneurs. The two authors have in parallel worked on individual sub-projects, respectively the significance of trust and the exploration of property relations and expression of ownership– the investigated sites have not been identical, but

strong associations uncovered themselves through discussions and linking. The material underlying this paper is extracted from a *bricolage* (Lévi-Strauss 1968) of notes from a long-term ethnographic fieldwork, personal interviews, emails and the reading of source-code repository commit-logs. Issues in regard to encompassment (LiPuma 2001) and how to establish trustful relations kept emerging during the discussions between the authors. Ethnographic details pointed towards a schism between on the one side; independents developers who 'fear' encompassment - encirclement - and on the other side; a corporation which lacks trust due to a fear of intellectual piracy (Johns, 2009). Collaboration and exchange would seem an impossible thought.

Despite this it was evident, through the reading of commit-logs and source-code header-files that something was crossing back and forth across an invisible borderline. To understand source-code one has to acknowledge it as an entangled object (Thomas, 1991) which comes with a complete biography embedded in its text. Source-code is distributed in the form of simple text, it always starts with information about the author(s), license and practical comments. Commit-log messages explain whom it was who committed a particular file into a repository and often explain why this was/is done. Looking across time it was visible that larger "chunks" of new source-code were pushed into the projects repository – and that the origins pointed across the border. Something made it possible to solve the mentioned schism, transgressing the border through the common creation of a boundary object. The methodological approach, which led to this understanding, was by itself guided by open collaboration and sharing of empirical material.

Open source software

Open source software is, at least from a legal point of view, the product of a certain kind of ingenuity. The story goes, at least according to Richard Stallman (1999) that for the longest time of the existence of 'computer' software it has been free and open for use and modification. The introduction of proprietary software, on a large scale during the 1980's, confronted individuals like Stallman with a moral choice. They could either betray their fellow hackers by entering the world of nondisclosure agreements – or, work towards a change in the basic premises. Stallman realised that what was needed, was first a new operating system. This made him coin the slightly obscure term GNU which translates into Gnu Not Unix, as a punt on at that time dominating Unix operating system. He writes how; "the goal of GNU was to give users freedom, not just to be popular, so we needed to use distribution terms that prevented GNU software from being turned into proprietary software. The method we use is called "copyleft" (Stallman, 1999: 59). These thoughts and actions quickly lead to the definition of free software and creation of the Free Software Foundation. Quickly explained, a program is free software if:

- You have the freedom to run the program, for any purpose.
- You have the freedom to modify the program to suit your needs.
- You have the freedom to redistribute copies.
- You have the freedom to redistribute modified versions of the program.

In this context "free" refers to freedom and not to price, there is nothing, which hinders the selling or commercial use of free software – as long as the software, and any subsequent modifications, are redistributed under its original free software license. The importance of the legal aspects of licensing brings us back to the previously mentioned notion of "copyleft". The Free Software Foundation implemented, through an ingenious process of reversal, a completely new licensing system which at the same was based on existing copyright law – this was the GNU GPL, or GNU General Public License (still the most common of the diverse range of open source licenses). Existing legal notions of copyright, when it comes to computer software, is aimed at excluding others from access to, redistribution and modification of the underlying source-code. Whereas the copyleft turns this upside-down and enforces circulation, rights of use and right to change, as long as the software is re-distributed under the original license (e.g. GPL or similar). This might leave the impression that this particular kind of software is simply given away, but this is not the case than as Stallman has paraphrased it: "free as in freedom, not as in free beer". The ambiguity of this saying did on the other hand ignite a feeling of evangelism – with associations to hostility towards intellectual property rights, communism and similar aspects – which made commercial interests sceptical towards use and involvement with software licensed under GPL. In the year 1998 a new initiative, known as the Open Source Initiative, was created to clarify these misunderstandings and aimed at introducing the far less politically charged terminology; Open Source Software (Raymond, 1999).

Independent of all these legal and ideological aspects the explosive emergence and adoption of Free and Open Source Software has manifested a new mode of production, fuelled by the ubiquitous availability of personal computers and Internet access. Some voices claim that the open source has impact well beyond programming (Schweik and Semenow,

2003) and as a new mode of production will help create the post-capitalist world – but, it is clear that present institutional forms and relations are being re-worked and changed.

One significant area where this has taken place is in the new kinds of collaboration between open source software projects and commercial research and development structures. The boundaries between closed and public environments are being blurred and are breaking down through these kinds of involvements, and this brings with it a need for new kinds of trust relations (not based on contracts or other commercial agreements). This cooperation is especially interesting in the setting of open source software communities where contributors-coders participate both as private persons and as company's employees. Moreover, the boundaries of known identities, such as 'IT producer' and 'IT user' and entities as a company and a community are being transgressed as a consequence of interaction between the two worlds (Westenholz, 2003; 2009). This creates a situation when groups ruled by competing logics (closed-proprietary products vs. quasi-public goods) are to work on the single OSS project and raises a question of possible basis for this cooperation.

Goblin

This paper is based on a study of boundary interactions between an open source project and a number of commercial companies, who together are collaborating on development and use. Due to the reality that certain facts about identity and subsequent interests of the various involved parties require anonymity (the authors of this paper also have trust relations which have to be honoured) all names used or entities referred to are fictitious. The open source project will, in this respect and from now on, be named; the Goblin project. The focus of Goblin is the development of a Linux based system for networked devices and the source-code is licensed under GPL.

The majority of open source projects are organizational nightmares, not in the sense of being dysfunctional just quite non-organized (Woods & Guliani 2005). The biggest and more significant projects are structured, with appointed positions, boards and even paid employees, but the vast number of projects are just a group of people working on "something". From the point of view of a commercial company this is often a very difficult situation; then how do you interact and collaborate with something, which is "not there"? "Not there" is to be understood in the sense that the people involved in the project are spread across several continents, 99% of the interaction takes place via email or IRC (Internet Relay Chat) and there is no office to visit or phone number to call. There is often only a website and a source-code repository visible from the outside.

The source repository of the Goblin project – somehow the metaphor of a tree keeps being the standard way of explaining the structure – is where all the work takes place. It is both the hub around which everything revolves and the pounding heart. A source-code repository is by itself a simple object, it is a storage "container" for the numerous text files which constitutes the source-code.

Computer software is in its original form text files written in one of the many different programming languages, and these text files have to be compiled into binary format (machine readable form) to create the actual application, such as a word-processing program. The repository stores all of these files and integrates a system for the continuous updating of the whole "tree" when new files or changes to existing files are committed by individual developers. The normal workflow is that an individual downloads, or synchronizes, a copy of the complete source-code tree to a local computer, makes the wished changes and then commits the new version of the file back to the repository. Hereafter, the repository automatically updates itself, and "stamps" a new revision number onto the source tree.

On the other hand, this was only a description of the procedure and what really counts in reality is who has the rights to commit directly, who has to submit patches to a review-process and who is without access. Understood in the way that it is free for everyone to download a copy of the source-code, but not everyone can upload files to the repository. This brings the subject of this paper into clear view – then who can be entrusted with having direct, and potentially fatal, access to the heart of a project? The existing group of developers who all have commit rights trust each other, and there are long traditions for how an individual becomes a developer. But, how is trust created and maintained between a project and a commercial company?

The source-code repository itself becomes a kind of boundary object, and functions as a meeting and merging point. The repository is in ways the only place to meet then it is the only visible representation of work and existence. If someone wants to involve himself or herself with the Goblin project they will eventually have to find their way to the repository. Over time it happens that companies in one or another way emerges in the horizon – but, this is a situation filled with a particular kind of tension. The company wants to have access to the source-code in the repository in such a

way that it can work on its “own things” (without to many others who are snooping around). But, at the same time conflict can arise if the company decides to close off, or directly create an independent fork of, the Goblin source-code. The project wants to protect their work from appropriation as well as attracting new developers, submissions and relations.

This brought quickly the before mentioned tension into the play of organizing the process of collaboration. Though, before stating the solution it would be interesting to take a look at what kind of interests had brought this meeting about.

One particular “day” this problem became immanent. An employee of the company Extensiva approached several of the Goblin developers, and told that the company would like to use the projects software in a new line of products. Inquiries of this kind are always very exciting, and there was clear interest from both parties in collaboration. But, from the point of view of the Goblin developers there was a risk for the integrity of the repository, and the independency of the project itself. Added to this, there had been little actual contact between the Goblin project and Extensiva prior to the contact, and as one Goblin developer stated: “...We deal with people, and we don't really know any of the Extensiva people...”. Meaning that a whole new relationship had to be build and ways of maintaining it.

Extensiva initiated the contact, and it came as a slight surprise to the developers from Goblin, then from the beginning on the company made it clear what they wanted and what they could offer. Within Extensiva there had been a series of development meetings, which had come about due to existing customers, beginning to ask about other options and alternatives to the software solution the company offered. Both research and management was confronted with the same problem – if they didn't respond to these requests/suggestions then somebody else would. But, how to go about this request for diversity? From the management point it would be extremely difficult to enter into collaboration with another commercial company, and then could you trust others not to exploit such a relationship? Finding an independent partner kept being the point around which the discussions revolved. It was needed to look elsewhere. One of members of the Extensiva research department eventually suggested looking towards one of the existing open source project, who already “played” with some of the devices manufactured by Extensiva. This suggestion suddenly opened a new direction. If Extensiva, in some way or another, collaborated with an open source project it would both ensure independence from other commercial actors and make it possible to maintain the parallel development. In respect to the development, it would not require that the Extensiva research department had to do all the work themselves – but, if the numerous developers of an open source project should take part they would need technical information from Extensiva normally not “submitted” into the open. Again a prior question returned: who could you trust? A small group, or taskforce, from the company began investigating the activities and development directions of a number of different open source projects. This process eventually lead to the decision to contact the Goblin project, and offer technical information on one of the Extensiva devices while asking for collaboration. This strategy did contain some risk for the company in terms of the spreading of information, but the creation of a “partnership” with an independent actor was seen as being more important.

The Goblin project sees a lot of these things in a different light, they have their own interests and goals. The primary focus of the project is to ensure that the “wealth” of the source-code repository is not appropriated and exploited, despite the requirements of the GPL license, by outside people and entities. In these days it is needed to defend cultural products from the dominant forces of commodification (Strathern, 2000, 2004), and this does not become less explicit when the eyes are turned towards the particular form of multiple ownership, which prevails in the open source world (Strathern, 2005). One of the fundamental aspects of this defence is the constant circulation of source-code (Petersen 2008). This circulation constantly reproduces the ownership, through visibility, and it can only be broken if source-code (the cultural product) is taken out of circulation, as would happen if someone denied the requirements of the GPL license and placed the code in a closed “environment”. In this sense, the Goblin project might give away the source-code (their work) but it always keeps it, in terms of ownership – it is a particular model of giving-while-keeping (Petersen 2008). Another central aspect is the question about who has the rights to access the Goblin source-code repository, who can add changes and updates. Not everyone is a developer and there is what you might have to recognize as a set of rituals or traditions which both guides and defines how an individual becomes a developer – a recognized individual – with rights of access. The individual developer creates her/himself through work done, in accordance to these guidelines. These comments highlights the interests of the Goblin project, then it would not be possible just to offer access to the repository to a company like Extensiva - the company could in the worst case scenario erase the complete content of the repository.

On the other hand, Extensiva would be free to copy the content of the Goblin repository and create a closed mirror, which then could hinder the free circulation. The latter would be a more or less direct threat to the vitality of the project. Again, who can you trust? Despite all these worries, and as one informant from amongst the Goblin developers explained: “.. We all need to recognize the differences between us and them and use it in a productive way. In many ways, when

someone is clearly different from you it somehow becomes much easier to find a way to collaborate..”. The members of the Goblin project and Extensiva likewise recognise the inherent difference, and came to the same conclusion that it would be worthwhile to make an attempt at building a productive relationship.

Such a process of creation and building is neither straight forward and it took some time and work. In the end the result came to be that the Goblin project created a synchronized mirror of the source-code tree to which only Extensiva and a few of the Goblin developers had access.

In this way Extensiva could work freely with the source-code and the Goblin developers would be able to merge interesting solutions and changes into the main repository. The solution is interesting in many ways then it continues the ambiguity of the aspect of trust – then, who do you trust and in what way?

Between trust and distrust

The Goblin project and Extensiva found a model of collaboration based on the creation of a sub-project, a source-code mirror “sitting” next to the primary repository. The Goblin repository, with its new extension, became re-created as boundary object – at the same time ensuring that Extensiva could maintain its needs for “clouding over” its involvement, and a way for the Goblin developers to secure that the work would be added to constant flow/circulation of the source-code. It would seem that a balance had been found, but this does not equal that the tension between trust and distrust is gone. The Goblin developers will never fully be able to know if Extensiva is telling them about all the company is doing, and Extensiva are constantly wondering how much they can tell the Goblin developers. To this come numerous intricacies, which arise when business logics clash with logics of free circulation – all adding to the complexity of the constant act of keeping the balance. One interesting example goes as follow: The new collaboration had the very direct result that people on the different sides of the boundary got to the point that they began building relationships of a more personal nature. One Goblin developers, Axel, was doing a lot of work on one particular device from Extensiva – the hacker jargon uses the term 'porting' or ' to create a port', which means to add changes to an existing code-base in such a way that a hardware device is fully supported (everything works) – and his Extensiva contact, Carl, forwarded a lot of information and source-code to Axel to make the work easier. Axel, as all good hackers, has an extended information network through which he one day heard a rumour about Extensiva being ready to release a new version of parts of the software. Axel immediately contacted Carl and asked him to forward a copy of the source-code – Carl replied that he would get it organized. But, then he eventually had to reply to Axel that he was unable to forward a copy. Extensiva only offered copies of the source-code/software to their customers. Carl and Axel had no problem with agreeing that Extensiva had created a problem, which related to trust, distrust and relations of power. At first it looked like a knot tied to hard to untie, but then Carl remembered that he knew a contact in one of the companies, who where customers of Extensiva and had received a copy of the source-code. This everyday knowledge creates an opportunity and he leaped into action. The result was that Axel ended-up downloading the source-code from a public ftp-server belonging to the other company, with the help of Carl's friend. This little transaction increased the relationship between Carl and Axel, but it also contains a strong ambiguity. Axel, as Goblin developer, had been confronted with unhidden distrust, which very clearly could hurt the collaboration between the project and Extensiva if it became “public” knowledge. Carl and Axel trusted each other to the degree that things would be kept quiet, so that access to information could be secured for Axel and Carl would not lose his job – but, as Axel explained: “... I might have a good relation to Carl, but I don't really know what to think about the company.”

The subprojects within Goblin are physically and conceptually detached from the main source repository. They, in the same time, give possibility to hide internal company's work connected to OSS project and release (submit) proposed changes only if decided, and clearly indicates to project developers the origins of the proposed code. The sub-projects is ‘a hand’ given to the business partner who is perceived as more focussed on ‘hiding’ its work in progress. However participation in OSS development means that benefit is only possible if the code is actually submitted to the main source repository. Key developers have gatekeeper power to shape the main source repository and decide on releasing new version of the software. In consequence, there are trust, distrust and power relation mixed.

Discussion and Conclusions

The resource tree (software/project structure) serves as boundary object, although the boundary is actually getting blurred. There is constant play between trust and doubt, expectation of good cooperation serving both project and

participants confused with disbelief in other's party good intentions. It is mostly sense-making process where boundaries are drawn and overcome. It is not clear anymore which developer serves company's or community's interests more as the relations and collaboration within project are created in parallel based on institutional arrangements and personal contacts with each individual hacker and undergo constant redefining as the work goes on. This specific dialectics is enabled by the structure of the project, which serves as a boundary object connecting communities of practice. Those communities, of business and communal organization, are based on different logics of actions, but people working within them use the collaboration around a source-code tree as a transgressing activity. The question who to trust, thou, seems to be repeated like a mantra. Response is ambiguous; as for successful cooperation in IT mediated environment suspended and granted trust in the same time is needed. Therefore we believe there is another possible interpretation.

The situation of Goblin source code repository what really makes the collaboration possible might be explained with the macro-level system of trust - trust in social systems and institutions. Lindström and Janzon (2007:461) define institutional (vertical) trust as "the trust of the citizens in the institutions, particularly the public institutions of society". A special type of a system trust is swift trust (Meyerson, Weick and Kramer, 1996), which is not based on common history, but on the willingness to suspend doubt about the other party's involvement and good faith that they will act in the group's best interest:

Swift trust theory is related to early stage of group interaction when participants are acting upon their initial expectations of involvement of team members and their focus on effective task performance. It is often presented as emerging in the context of temporary task groups. The swift trust is then understood as trust initially present at the beginning of group formation when team has yet no history or relationship worked out what so ever. But, because of the character of their collaboration—temporary and task accomplishment oriented—they must swiftly form relationships and divide roles to be able to act (Meyerson, Weick and Kramer, 1996:167).

The existence of institutional and swift trust is determined by people's believe that those institutions will be effective in sustaining the system. The impersonal trust is sustained by the guardians who also "simulate the practices of risk spreading, personalizing, or contractually limiting agency relationship that principals ordinarily exert on their own behalf" (Shapiro, 1987:636). These guardians are themselves guarded by institutionalised trust (Kroeger, 2009). In Goblin case the combination of institutional (the OSS licences & their legal protection) and swift (the good of the project, share benefit assumption) trust antecedents, rather than personal or cognitive ones, facilitate the collaboration of various communities of practice within Goblin.

References

- Ciesielska, M. (2010), Hybrid organizations. A case of the open source-business setting, Copenhagen Business School Press, Frederiksberg.
- Ciesielska, M. and Iskoujina, Z. (2012), "Trust as a success factor in open innovation. The case of Nokia and Gnome", in Jemielniak, D. and Marks, A. (Eds) *Managing Dynamic Technology-Oriented Business: High-Tech Organizations and Workplaces*. Information Science Reference, pp. 11-29.
- Chesbrough, H. (2006). *Open Innovation. The New Imperative for Creating and Profiting from Technology*. Boston: Harvard Business School Press.
- Chrisman, N. (1999). Trading Zones or Boundary Objects: Understanding Incomplete Translations of Technical Expertise. *Conference paper 4S San Diego CA October 99'* available at: <http://chrisman.scg.ulaval.ca/Present/4S99.pdf>
- Drazin, R., M. A. Glynn & R. K. Kazanjian (1999). Multilevel theorizing about creativity in Organizations: A Sensemaking Perspective." *Academy of Management Review*, 24:2, pp. 286-307
- Gambetta, D. (1988). Can We Trust Trust? in: Gambetta, D. (ed.) *Making and Breaking Cooperative Relations*. NY-Oxford: Basil Blackwell Ltd., pp.213-237.
- Greendberg, P.S., Greendberg, R.H. & Antonucci, Y.L. (2007). Creating and sustaining trust in virtual teams. In: *Business Horizons*, 50, pp.325-333
- Hardin, R. (2002). *Trust and Trustworthiness*. New York: Russell Sage Foundation
- Hardin, R. (2006). *Trust*. Cambridge-Malden: Polity Press

- Hardy, C., Phillips, N. & Lawrence, T. (1998/2000). Distinguishing Trust and Power in Interorganizational Relations: Forms and façades of Trust. In: Lane, C & Bachmann, R. *Trust Within and Between Organizations*, New York: Oxford University Press, pp.64-87.
- Holford, D.W., M. Ebrahimi, O. Aktouf and L. Simon (2008). Viewing Boundary "Objects" as Boundary Constructions. Proceedings of the Proceedings of the 41st Annual Hawaii International Conference on System Sciences. Available at: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.115.5296&rep=rep1&type=pdf>
- Hunter, C., Jemielniak, D. & Postuła, A. (2010). "Temporal and Spatial Shifts within Playful Work", *Journal of Organizational Change Management*, vol. 23(1), pp. 87-102
- Jaaksi A. (2006). *Building consumer products with open source*. Available from: <http://www.linuxdevices.com/articles/AT7621761066.html> [accessed 19 June 2007].
- Jaaksi A. (2007). Experiences on Product Development with Open Source Software. in: Feller, J., Fitzgerald, B., Scacchi, W. and Sillitti, A. *Open Source Development, Adoption and Innovation*. Springer/IFIP, pp. 85-96.
- Jarvenpaa, S.L, Knoll, K. & Leinder, D.E. (1998). "Is Anybody Out There? Antecedents if Trust in Global Virtual Teams". In: *Journal of Management Information Systems / Spring 1998*, vol.14, no. 4, pp.29-64.
- Jemielniak, D. (2006). The Management Science as a Practical Field: In Support of Action Research", *International Journal of Knowledge, Culture and Change Management*, vol. 6 (3), pp. 163-170
- Jemielniak, D. (2010). "Software engineers or artists? Programmers' identity choices. "Tamara Journal for Critical Organization Inquiry, vol. 7 (1), pp.20-36
- Jemielniak, D. & Kostera, M. (2010) "Narratives of irony and failure in ethnographic work", *Canadian Journal of Administrative Sciences/Revue Canadienne des Sciences de l'Administration*, vol. 27 (4), pp. 335-347
- Johns, A. (2009). Piracy. The University of Chicago Press.
- Johnson, D. & Grayson, K. (2005). "Cognitive and Affective Trust in Service Relationships," *Journal of Business Research*, 58:4 (April), 500-507.
- Kelty, C. et al. (2009). Collaboration, Coordination, and Composition. In: *Fieldwork is not what it used to be*, Faubion, James D. & Marcus, George E. Cornell University Press.
- Kostera, M. (2007) *Organizational ethnography: Methods and inspirations*. Studentlitteratur, Lund
- Kroeger, F. (2009). "The institutionalization of trust: understanding the creation and collapse of escalating trust spirals in economic life". Paper presented at the annual meeting of the SASE Annual Conference, Sciences Po, Paris.
- Lane, C & Bachmann, R. (1998/2000). *Trust Within and Between Organizations*, New York: Oxford University Press.
- Latussek, D. (2007). Zaufanie i nieufność w relacji sprzedawca-nabywca w polskim sektorze IT. (The Trust and Distrust in Seller-Buyer Relation in the Polish IT Sector) Ph.D. Thesis. Warszawa: Wyższa Szkoła Przedsiębiorczości i Zarządzania im. Leona Kozłowskiego.
- Levi-Strauss, C. (1968). *The Savage Mind*. The University of Chicago Press.
- Lewis, J. D. & Weigert, A. (1985). Trust as a social reality. *Social Forces* 1985; 63(June); 967– 85.
- Lindström, M. & Janson, E. (2007). Social capital, institutional (vertical) trust and smoking: A study of daily smoking and smoking cessation among ever smokers *Scandinavian Journal of Public Health* 2007; 35; 460.
- LiPuma, E. (2001). *Encompassing others*. The University of Michigan Press.
- Matzat, U. (2004). Cooperation and Community on the Internet: Past Issues and Present Perspectives for Theoretical-Empirical Internet Research." In: *Analyse & kritik* 26/2004, pp.63-90.
- McAllister D. J. (1995). Affect- and cognition-based trust as foundations for interpersonal co-operation in organizations. *Academy of Management Journal*. 38(1); 24– 59.
- Meyerson, D., Weick, K. E., & Kramer, R. M. (1996). Swift trust and temporary groups. In Kramer, R. M., & Tyler, T. R. (Eds.), *Trust in Organizations: Frontiers of Theory and Research* (pp. 166–195). Thousand Oaks, CA: Sage Publications.
- Osterloh, M. & Rota, S. (2004). Trust and Community in Open Source Software Production. In: *Analyse & kritik* 26/2004, pp.279-301.
- Petersen, G. (2008). Circulating property. Paper presented at *Rethinking Economic Anthropology conference*, LSE, January 11-12th 2008.
- Raymond, E. S. (1999). The Revenge of the Hackers. In: *Opensources. Voices from the Open Source Revolution*, DiBona, C., Ockman, S. & Stone, M. (Eds.). O'Reilly, Sebastopol.

- Rothstein, B. (2007). "Trust and Social Concept." In: Goodin, R. B. & Pettit, P. (eds.) *A companion to Contemporary Political Philosophy*. Oxford University Press.
- Schweik, C. M. & Semenow, A. (2003). The Institutional Design of Open Source Programming. Implications for Addressing Complex Public Policy and Management Problems. In: *First Monday*, Vol 8/1, available at: <http://www.firstmonday.org/htbin/cgiwrap/bin/ojs/index.php/fm/rt/prinFRIENDLY/1019/2426>
- Shapiro, S. 1987. "The Social Control of Impersonal Trust." *American Journal of Sociology* 93(3): 623-58.
- Stallman, R. (1999). The GNU Operating System and the Free Software Movement. In; *Opensources. Voices from the Open Source Revolution*, DiBona, C., Ockman, S. & Stone, M. (Eds.). O'Reilly, Sebastopol.
- Star, S. L & Griesemer, J. R. (1989). Institutional Ecology, 'Translations,' and Boundary Objects: Amateurs and Professionals in Berkeley's Museum of Vertebrate Zoology, 1907 - 1939. *Social Studies of Science* 19: 387-420, Holford et al., 2008
- Strathern, M. (2000). Multiple perspectives on intellectual property. In: *Protection of intellectual, biological & cultural property in Papua New Guinea*, Whimp, K. & Busse, M. (Eds.). Asia Pacific Press, Canberra.
- Strathern, M. (2004). Introduction: Rationales of ownership. In: *Rationales of ownership; Transactions and claims to ownership in contemporary Papua New Guinea*, Kalinoe, L & Leach, J (Eds.). Sean Kingston Publishing, London.
- Strathern, M. (2005). Imagined Collectives and multiple authorship. In: *Code: Collaborative ownership and the digital economy*, Ghosh, R. A. (Ed.). MIT Press, London.
- Thomas, N. (1991). *Entangled Objects*. Harvard University Press.
- Weber, S. (2004), *The Success of Open Source*. Cambridge-London: Harvard University Press.
- Weick, K. E. (1979). *The Social Psychology of Organizing*. 2nd ed., New York: McGraw-Hill.
- Weick, K. E. (1995), *Sensemaking in organizations*. Thousands Oaks: Sage.
- Wenger, E. (2000). Communities of Practice and Social Learning Systems. *Organization*, Volume 7(2), pp. 225-246.
- Westenholz, A. (2003). Identity Work in the fractures between open source communities and the economic world. *Working Paper* No. 2003.16 IOA/CBS
- Westenholz, A. (2009). Institutional entrepreneurs transgressing ideological/material practices in two mature organizational fields. In: Renate Meyer, Kerstin Sahlin, Marc Ventresca, Peter Walgenbach (eds.) *Ideology and Organizational Institutionalism. Research in the Sociology of Organizations*. Emerald Group Publ. (Chap. 9).
- Westenholz, A. (ed.) (2012). *Janus Face of Commercial Open Source Software*. Frederiksberg, Copenhagen Business School Press.
- Wiertz, C. & Ruyter, K. (2007). Online Communities. Beyond the Call of Duty: Why Customers Contribute to Firm-hosted Commercial. *Organization Studies* 2007; 28; pp.347-376.
- Wilson, G. & Herndl, C. G. (2007). Boundary Objects as Rhetorical Exigence: Knowledge Mapping and Interdisciplinary Cooperation at the Los Alamos National Laboratory. *Journal of Business and Technical Communication* vol. 21, issue 2, pp. 129-154.
- Woods, D. and Guliani, G. (2005). *Open Source for the Enterprise: Managing Risks, Reaping Rewards*. O'Reilly.

ⁱ Earlier version of this paper was presented at EGOS Colloquium.