Rajanen, M., Iivari, N. & Lanamäki, A. (2015) Non-Response, Social Exclusion, and False Acceptance: Gatekeeping Tactics and Usability Work in Free-Libre Open Source Software Development. In Proceedings of 15th IFIP TC.13 International Conference on Human-Computer Interaction (INTERACT 2015). Bamberg, Germany

Non-response, Social Exclusion, and False Acceptance: Gatekeeping Tactics and Usability Work in Free-Libre Open Source Software Development

Mikko Rajanen[1], Netta Iivari[1] and Arto Lanamäki[1]

[1] University of Oulu, Department of Information Processing Science,
PO Box 3000, FI-90014 Oulun yliopisto, Finland
{mikko.rajanen, netta.iivari, arto.lanamaki}@oulu.fi

Abstract. Usability is an important aspect of Free-Libre Open Source Software (FLOSS), but barriers exist for usability specialists' participation in such projects. Organizational boundary literature is interested in the tensions of online communities, including FLOSS communities. While this literature recognizes the importance of managing boundaries in online communities, little empirical research has been conducted on actual gatekeeping tactics project members perform against outsiders' contributions. Based on several years of engaged research with FLOSS projects, we characterize three gatekeeping tactics in FLOSS projects: non-response, social exclusion, and false acceptance. They all have hindered usability work. We also offer examples of usability specialists and their contributions succeeding in avoiding these gatekeeping tactics in FLOSS projects. This paper provides an important contribution to the boundary management literature through detailed examination of gatekeeping tactics in action, as well as to the Human Computer Interaction literature interested in contributing to FLOSS projects through usability work.

Keywords: Usability specialist, FLOSS, boundary management, gatekeeping

1 Introduction

A young and enthusiastic usability expert wanted to volunteer in a Free-Libre Open Source Software (FLOSS) project developing a media-center software. He had seen how the project was strongly oriented in technical functionalities, while the usability aspects were lacking. Therefore, he saw how the project could benefit from his usability expertise. He went on the project's internet relay chat (IRC) channel and found active discussions going on. After a period of time, he introduced himself and proposed to volunteer as a usability expert in the project. No one answered him. After a while, he kindly restated his message. Soon, he received an answer: "don't be so *spammy*." He was confused and sad that the community disregarded his kind will to volunteer in the project, and to contribute to it using his expertise.

This example is not how FLOSS projects are painted by the apologists. They, instead, emphasize the basic freedoms of software users: freedom to run software, freedom to study software, freedom to change software in any way that a user sees as necessary and freedom to distribute copies of software with or without changes to it [1] and characterize FLOSS projects as participatory and egalitarian settings in which individuals develop FLOSS due to their personal need, but also voluntarily reveal their solutions to be used and further developed by others [2]. In this kind of setting, user innovation flourishes as the reputation and status that can be gained as well as the community development model motivate people to reveal the solution also for others' use and further development [3]. The continuous improvement and refinement of the solution done by the community members is one driving force of FLOSS development [4]. All these indicate that FLOSS projects are to be seen as participatory and egalitarian places where people collaboratively develop FLOSS to serve their own needs as well as the needs of the others.

However, less attention has been paid to the aspects of inaccessibility and inequality in FLOSS projects. In a sense, Human Computer Interaction (HCI) research has already acknowledged FLOSS projects as not such participatory and egalitarian places; many HCI studies have revealed that usability experts are experiencing challenges when trying to enter FLOSS projects. HCI research has already established that usability specialists and their contributions are easily ignored, both in commercial software development (e.g. [5, 6]) and in FLOSS development (e.g. [7,8,9,10,11,12]. Hence, it is evident that power and politics play a role as regards usability work also in FLOSS projects; not all members get equal treatment in FLOSS projects, and usability specialists are often among the 'power-weak' in this respect [11,12].

This paper tries to explain the challenges involved with usability specialists entering FLOSS projects with a focus on boundary management as the theoretical framework. Boundary management is generally interested in the "activities involved in defining, negotiating, and protecting organisational resources and domains of action, as well as managing relationships with external stakeholders, to achieve the organisational goals" [13]. This framework has already been utilized in the context of various kinds of online communities, including FLOSS communities (e.g. [14]). Online communities are considered as new forms for organizing, related to which boundaries are more permeable and dynamic [13,15,16]. Unlike traditional organizations that are hierarchical with quite clear boundaries, online communities are fluid objects "where boundaries, norms, participants, artifacts, interactions, and foci continually change over time" [15]. However, the framework has not yet been utilized to make sense of usability work in FLOSS projects. We consider the framework as a suitable lens to be utilized in this novel context: for examining the boundary management activities carried out by FLOSS project members when usability specialists are attempting to enter FLOSS projects. Here, we will especially rely on the concept of 'gatekeeping' to make sense of the challenges involved with usability specialists' work. This concept addresses the filtering and moderation of participation and content production in online communities [16]. Based on empirical analysis, we identify three gatekeeping tactics that have hindered usability work in our cases as well examples of situations in which usability specialists have succeeded to enter into and contribute to the FLOSS projects in questions, i.e., they and their contributions have succeeded in becoming ‘filtered in’. This study contributes both to the boundary management literature through examination of actual gatekeeping tactics in action and to the HCI literature interested in contributing to FLOSS projects through usability work.

The paper is structured as follows. The next section reviews research on boundary management and gatekeeping in the context of online communities and more specifically in FLOSS communities, indicating that these concepts have relevance also for FLOSS usability research. The third section describes the research method involved in this study, introduces the cases involved in this study as well as the procedures for data collection and analysis. The fourth section presents the empirical results of our analysis. The fifth section discusses the implications of the results while the last section summarizes the results and their implications for research and practice as well as outlines the limitations of the results and based on those, identifies a number of interesting paths for future work.

2 Background

This section introduces literature on boundary management and gatekeeping, and relates it to the FLOSS and HCI research on usability work in FLOSS projects.

2.1 Boundary management and gatekeeping in FLOSS communities

FLOSS is computer software that is freely available as source code, and often also as a precompiled binary file. The license permits users to read, change, and modify the source code as derived works, recompile the modified source code to binary form, and distribute the modified source code as a derivative under the same license as the original source code. [17]

While the letter ‘O’ in ‘FLOSS’ refers to openness *of the source code*, openness is often seen to apply to the whole ethos of development [18]. A fundamental idea of FLOSS is to enable software to evolve freely through community participation. In principle everyone can participate in FLOSS projects, and the more people participate, the better results will be achieved. This is reflected in Eric Raymond’s [17] formulation of Linus’ Law: “Given enough eyeballs, all bugs are shallow”. This insouciant depiction of openness is visible in, for example, the article by Bach and Terry [19] who note that “members of the FLOSS community are highly accessible”, and that “there are rarely ‘gatekeepers’ that control access.”

On the other hand, studies have also shown that power and politics picture in FLOSS projects, too (e.g. [11]). The FLOSS community is often depicted with an onion model (e.g. [20]) with different layers representing the level of involvement within that particular FLOSS community. The layers also depict the level of decision power at each layer. In a typical FLOSS community, there is a lead developer or a small group of developers forming the core team that controls the overall architectural design and the course of the project [20,21,22,23]. Some FLOSS projects are relatively democratic, while in others the project leaders make decisions as “benevolent dictators". Indeed, one of the most common models in coordinating FLOSS development projects is that several contributors work under a single 'benevolent dictator' who is usually the founder of the project and who attracts committed and talented contributors [24]. An alternative to having one single benevolent dictator is rotating dictatorship or forming a voting committee from the developers [24]. Despite these differences, the core developers nevertheless have a significant position in FLOSS project as decision-makers: they make both low-level decisions regarding whether or not a particular contribution is accepted to the code, mid-level decisions regarding software features to be included in individual releases, and strategic decisions regarding the direction of the development in the future as well as the development roadmap.

For a developer to become an accepted contributor or even an acknowledged member of a FLOSS project certain procedures or 'joining scripts' need to be relied on [14,25,26]. A developer may have to provide *feature gifts*, i.e. whole modules or features as his contribution during that phase [26]. The core developers assess the value of the contribution and the contributor before accepting it. There is significant prestige motivation to get own contributions accepted and to become a member of the development team [20,23]. However, many obstacles have been witnessed as regards entering FLOSS projects. In a recent systematic literature review, Steinmacher and colleagues [25] identified 20 published studies dealing with contribution barriers that newcomers face in FLOSS projects. These barriers represent 5 main types: social interaction, newcomers' previous knowledge, technical hurdles, finding a way to start, and documentation. Social interaction was the biggest category with 12 studies, and it divided into three subcategories: "lack of social interaction with project members", "receiving an improper answer", and "not receiving a (timely) answer" [25].

This leads us to consider how FLOSS projects balance between openness and control. Do FLOSS projects welcome newcomers and their contributions, stimulate creativity, and allow diversity? Or are they protecting the establishment, and relying solely on its tradition? These questions have already been addressed in the literature on boundary management, where it has been argued that in online communities "boundary management involves trade-offs between openness (attracting external participation, stimulating innovation, creativity and organisational growth) and control (over platform activities and content production and appropriation), or trade-offs between standardisation and formalisation of production processes and availability and accessibility of diverse resources" [13]. Here, we focus on a particular type of boundary management, namely gatekeeping. We adopted the concept of gatekeeping from Shaw [16], whose definition of gatekeeping reads as "the systematic reproduction of an unequal and regular flow of valued resources—especially influence—to an incumbent group or organization". Shaw borrowed the concept from Clayman and Reisner [27], who in turn credit its origin to Lewin [28]. Following Shaw [16], in this paper we formulate gatekeeping tactics to refer to the specific actions carried out by FLOSS project members relating to filtering and moderating outsiders' participation in and contribution to FLOSS projects to achieve the goals of the projects. We will offer an empirical analysis of it in relation to usability work in FLOSS projects.

2.2 Usability and FLOSS Development

HCI research has already addressed the FLOSS development context and HCI researchers have already argued that usability specialists should participate in FLOSS projects [7,9,10,29,30,31,32,33]. Different kinds of usability methods have been suggested for FLOSS development. Several articles recommend conducting usability testing [7,10,32,33,34], while some argue also for usability inspections in FLOSS projects [33,34]. For usability design, user interface design using blogs has been suggested [32], as has the establishment and use of usability infrastructure such as discussion forums, mailing lists, and design areas [8,10,30,32].

Nevertheless, researchers have also revealed that problems might arise when combining the traditional usability methods and recommendations with the FLOSS development and philosophy [8,9,10,31,33,35,36]. Reluctance to heavyweight corporate usability processes might arise in FLOSS development; i.e., decentralized and engineering-driven FLOSS development might be in contrast with heavyweight corporate usability processes [8,9,31,33,35]. In addition, usability specialists might not be available in FLOSS projects, and, even if they were, they might be very few in number and they might be working in isolation [7,8,9,10,11,12]. Furthermore, the software development in FLOSS projects is almost always already ongoing when the projects gain enough visibility to be spotted by the usability specialists, making it difficult to apply usability methods from the beginning of the development when it is easiest [11,37]. Usability specialists might have difficulties in showing their merits and gaining authority in FLOSS projects [7,12,30,36]. They may be welcomed into the consultative role as feedback providers (cf. [6]), but they might find it difficult to gain any decision-making power regarding the design solution, which is recommended in the participative role endorsed in the HCI literature (cf. [6]). In FLOSS projects, as mentioned, the core developers make all of the decisions regarding what will be included in the code base. The usability specialists should try to utilize various kinds of lobbying, persuasion, and allying strategies in order to have an impact [12,36,38,39].

Hence, several challenges for usability specialists' participation in FLOSS projects have already been identified. Our empirical data, along with these other studies, pinpoints these challenges. However, this paper contributes through making sense of these challenges. Three gatekeeping tactics hindering usability work are identified from FLOSS projects. Additionally, a few cases are identified in which usability specialists have succeeded in overcoming these challenges, and becoming 'filtered in'.

3 Research Method

Our arguments about boundary management and gatekeeping tactics in FLOSS projects are based upon intensive engagement in different FLOSS projects since 2007, with an overall aim of enabling usability specialists to enter into and contribute to FLOSS projects. Walsham [40,41] has discussed different roles researchers can adopt in qualitative, interpretive research, i.e. the 'outside researcher' and the 'involved researcher' roles [40]. The former refers to the researcher having "no direct involvement in action in the field or in providing significant feedback to field participants", while the latter refers to researcher acting as a participant observer or action researcher who consciously and explicitly aims at changing things in the field. We have adopted the latter position, within a large research program that has included intensive work with a number of FLOSS projects during a 7 years timespan. The overall purpose has been the improvement of usability processes in FLOSS projects. The research program has resulted in the development and experimentation of various kinds of methods outlining for usability specialists how to enter into and contribute to FLOSS projects (reported in [11,12,35,39,42]). The methods have placed particular emphasis on participative approach of the usability specialists: they need to try to gain a thorough understanding the project in question and to actively collaborate with the developers [12,35,39]. Additionally, the results from the research program implies that usability work is more likely to have an impact if the results are provided by encultured insiders that have modified their usability work to fit the particular OSS project, because every OSS project is unique [42]. During our analyses, it has also become evident that power and politics sometimes truly complicate usability work in FLOSS projects [11]. The present paper aims to explain some of the challenges faced by usability specialists in the projects.

Our research program has included altogether 16 subprojects within which junior researchers, organized into usability teams, have introduced usability activities into FLOSS projects as part of their university studies. This has occurred under the guidance of more experienced HCI researchers. All junior researchers have had usability background from at least two previous usability courses about usability evaluation methods (e.g. heuristic evaluation and usability testing), user-centered design, and user interface design in both theory and practice. Each of the subprojects has consisted of three to five junior researchers working between 200 and 300 hours each in planning the usability activities, carrying out these usability activities in the selected project, communicating with the project, following up the impact of these usability activities, collecting empirical data, and writing project reports. The senior HCI researchers have guided the usability teams during the entire process, including the selection of the project and the selection of suitable entrance strategies and usability methods to use. Altogether, our role has been that of involved researcher [40], i.e., we have consciously and explicitly tried to change things in the field and to make a valid contribution. During the 7 year research program and 16 different interventions organized, various strategies and methods have been experimented with, resulting in differing outcomes in involved projects with different domains, communities and cultures. After the interventions, the HCI researchers have been responsible of analyzing the collected data and developing and further refining the methods outlining ways for usability specialists to utilize their expertise and gain recognition in FLOSS projects.

In this paper, the empirical data included has been collected from six of these usability projects. This data includes online material of the involved FLOSS projects, such as websites, mailing list and discussion forum posts, IRC discussion logs, etc. In addition, the junior researchers, during their interventions, have produced numerous kinds of reports of their work and of the selected case projects that are included as research material. For the purposes of this paper, we inductively identified and analyzed the instances we encountered from this empirical data that somehow related to power and politics as regards usability specialists entering into and contributing to FLOSS projects. We identified problems in that process as well as successes achieved. The data itself led to the categorization of the three different gatekeeping tactics, while only after the identification of those, the literature outlined in the previous section relating to boundary management and especially to gatekeeping was utilized as a sensitizing device to make sense of our findings.

Before describing our empirical findings on the gatekeeping tactics in FLOSS projects, we briefly introduce the six case projects involved in this analysis and our associated interventions into these case projects.

Case A was developing a media application, targeted at non-technical end users without programming skills or interest. The project was started in 2004 and had a total of about 30 developers. The usability team observed this FLOSS project for five months in 2007, while conducting heuristic evaluations, cognitive walkthroughs, and usability testing. The usability team reported the findings in the form of a report, which was sent to the core developers and mentioned in a post in the main discussion forum of the community.

Case B was developing a game targeted at non-technical end users. This project, started in 2003, had a total of 15 developers. The usability team observed this FLOSS project for five months in 2008, while performing heuristic evaluation and usability testing. The usability team was in close contact with the lead developer regarding their findings and possible redesign solutions, and also participated in discussions in the project's IRC channel. After the evaluations, the usability team wrote a usability report. This included suggestions for changes to fix the identified usability problems.

Case C was developing a 3D content creation software targeted at end users with 3D content creation skills but without skills or interest in programming. The project, started in 2002, had a total of 40 more or less active developers. The usability team observed this project for six months in 2009. During that time span, they carried out usability testing and heuristic evaluation and wrote several reports about usability problems and their suggestions for changes to fix those problems. These reports were made available on the usability team's blog and advertised in the project's IRC channels and discussion forums.

Case D was developing a media center software with target users of ordinary people. The project started in 2003 and had about 20 active developers. The usability team observed this FLOSS project for five months in 2009, while performing heuristic evaluations and usability testing. In this case the results report was sent to the FLOSS developers by email in a similar manner as in case A.

Case E was developing a game targeted at non-technical end users without programming skills. This project started originally in 1995, but the development team had changed many times since then. This project had 20 active developers with commit rights. The usability team observed this FLOSS project for four months in 2010, while conducting heuristic evaluations using game usability heuristics and usability testing. The usability team wrote preliminary and final usability reports about the usability issues and their suggestions for changes to the user interface to fix them. The final usability report was delivered to the wiki of the FLOSS project. In addition, the usability team submitted code patches and level design work, including new user interface menus and a new tutorial for the game.

Case F was developing a vector graphics software targeted at non-technical end users. The project, started in 2003, had a total of 6 core developers and 14 developers. The usability team observed this FLOSS project for six months in 2009 and 2010, while performing heuristic evaluation and prototyping. The usability team wrote a usability report based on the results from heuristic evaluation and redesigned an improved user interface as a mock up. These deliverables were sent to the core developers through email and discussion forum.

4 Gatekeeping tactics in FLOSS development

Our work with the FLOSS development projects has revealed that gatekeeping tactics to keep outsiders' unwanted contributions away truly take place. The gatekeeping tactics identified from each case are summarized in Table 1.

Table 1 - Three gatekeeping tactics identified in the FLOSS cases

	A	B	C	D	E	F
Non-response	x			x		x
Social exclusion	x	x	x			
False acceptance					x	

Next we offer empirical illustrations of the tactics, followed by examples that reveal that usability specialists occasionally have also been able to get 'filtered in'.

4.1 The Gatekeeping Tactic of 'Non-response'

In our first intervention, involving the case A, the junior researchers carried out two types of expert usability evaluations: heuristic evaluation and cognitive walkthrough. Next, they planned and executed usability tests based on the findings from these expert evaluations. A report of usability findings was written and sent to the developers by email. This was the first contact between the developers and the usability team, as it was planned. The purpose of this approach was to mimic the way the software patches are submitted in the FLOSS development projects: somebody writes the patch, which is then shared with the community. Eventually the core developers either accept this patch into the main branch or reject it.

As it turned out to be, the work of the usability team had no impact, but their message fell into deaf ears. At first, no answer was received from the emailed core developers. Thereafter, the same document containing the usability findings was posted to the discussion forum of the project. Then one of the core developers answered that they were discussing this document and its findings internally and could comment on it later. However, there has not been any answer or further communication from the developers and there are no signs of changes to the software that could be traced back to these usability findings. One can argue that the gatekeeping tactic of non-response was evidently utilized by the core-developers. This resulted in totally ignoring the attempts of the usability team.

A validation test was conducted in case D, in which a similar kind of FLOSS project was selected and the usability team followed a similar type of approach in their work. Hence, the result was also the same. The results report was sent to the FLOSS developers by email, they replied they had received it, but no further communication from their side emerged and the FLOSS in question has not been changed according to the results reported.

In another intervention, involving the case F, the usability team conducted heuristic evaluation and redesigned the user interface based on the results from the evaluation. The redesigned user interface was prototyped as a mock up so that it would be easy for the developers to understand the proposed changes. The results from heuristic evaluation and the prototype mock up were sent to the core developers by email and discussion forum posts. However, no answer was received from the core-developers despite multiple communication attempts. Not surprisingly, the work of the usability team had no impact and no signs of changes to the user interface could be noticed. Also in this case the gatekeeping tactic by the core-developers was simply to not respond to the communication attempts.

Finally, similar kind of behavior has been observable towards some users in FLOSS project discussion forums. In the case A, some users had expressed criticism towards the user-interface of the application and offered certain usability improvement suggestions. Also those had been disregarded by the developers who had only commented that the application “is not meant for girlfriends”. This kind of response does not invite further discussion on the matter. Interestingly, this project had nevertheless stated on its website that it wanted to target ‘non-technical end-users’. However, likely the suggestions provided by the users were not such that were preferred by the core developers, and thus they were hushed down.

4.2 The Gatekeeping Tactic of ‘Social Exclusion’

The usability interventions arranged were not entirely ignored in all case projects. For example, in case B, after the failure encountered in case A, the HCI researchers decided that the junior researchers should familiarize themselves with the project before their usability intervention. Hence, the junior researchers followed the project’s IRC channels and discussion forums for some time before making themselves and their intentions known to the project. Thereafter, they contacted the lead developer through email and offered their help. In this project, there was no prior knowledge about usability, but the usability team explained the concept of usability and its potential benefits to the project, and identified some possible areas for usability evaluation. They carried out expert usability evaluation and usability testing for the software. Additionally, they continuously communicated with the core developer and the community through the project’s IRC channel. Also here, they delivered their results by email to the core developers. In this case, however, the core developers accepted and implemented the changes to the next version of the software. The usability team was even later on contacted and requested to carry out another usability evaluation.

However, although the work done by the usability team seemed to be a success, some problems were observed. The developers also gave some negative feedback on the work of the usability team. One of the main problems was the rapid development of the software. The pace of the usability evaluation was slower, and therefore some of the usability team's findings were obsolete by the time the report was ready. The usability team was in this case treated as an external resource and the core developers did not help the team to fit the usability activities into their overall development plan and bug fixing process. The usability activities were welcomed and encouraged, but the usability team did not become integrated into the community. In addition, the developers wanted concrete suggestions about how to fix the user interface problems and not just general comments about what the problems were. However, once the developers had a list of concrete improvement suggestions, they considered only those suggestions that they saw as fixing issues they saw as problems. If the core developers thought that some of the usability team's findings were not truly problems, these findings got a very low priority and were eventually discarded. All in all, one can say that in this case the usability team was allowed to work and was acknowledged by the FLOSS project, but they were excluded from the actual decision-making and planning processes, hindering the usability team's ability to offer meaningful and timely usability contributions.

Another observation can be connected with case C, within which another usability team consisting again of junior researchers carried out their intervention. Again, the junior researchers were expected to familiarize themselves with the project before their usability intervention. However, in this case there was a vast number of communication channels available both in the project's website and third party websites; e.g. mailing lists, IRC channels, wikis, and discussion forums. The usability team searched and followed multiple communication channels (e.g., various IRC channels, message boards, project news sites, and wiki pages) for a couple of weeks getting to know the project. Next, they contacted the core developers and offered their usability expertise in particular area of software that had already raised some discussion about complicated user interface and difficulties in use. The usability team conducted usability tests and expert usability evaluations, and documented them in open source fashion on a website, which was promoted in community forums and IRC channels, and also offered to several community news sites for publication. The reactions were mixed. One core developer was very supportive to the usability activities, while other core developers and community ignored the usability issues. The reports were downloaded about fifty times from the website, but no further discussion was generated. The news about the usability activities and their results were quickly buried beneath other discussions and news. Eventually, the usability intervention did not have an impact on the software in question. This FLOSS project had a multilayer hierarchical structure in which the leading core developer as the benevolent dictator was inaccessible to the usability team. The leading core developer communicated with other trusted core developers as his lieutenants. The usability intervention did not catch the attention of them, albeit the usability team was able to carry out their work and gain one supportive core developer on their side. However, their usability intervention results ultimately ended up as being socially excluded.

4.3 The Gatekeeping Tactic of 'False Acceptance'

An example that we label as "false acceptance" occurred in case E. The initial usability team working with the project clearly succeeded in having an impact on the software under development. The usability team, similarly to case B, followed the project IRC channels and discussion forums for some time before their intervention. Again, they conducted an expert usability evaluation and empirical usability testing. They sent their report to the mailing list of the project as well as to the community wiki. In this case, the results aroused a lot of interest. The developers actively commented the results. In addition, the usability team submitted code patches and game level design work that were accepted into the code repository of the project. Moreover, the usability report was referenced directly in commit messages of the core-developers four times. One of these commit messages asked for an input and contribution from the usability team. After this, changes were made based on their recommendations. One of the usability team members even got commit rights and gained a status as a developer. He was participating actively in discussions and was recognized within the community as being a skillful and committed user of the software.

However, the gatekeeping tactic of false acceptance can be connected with this case. Related to it, we emphasize that it is important to acknowledge that it may take place during a longer time span and in a discreet way, potentially leaving the usability contributors with an impression that they succeeded in improving the usability of the software. The usability team concentrated their efforts into the tutorial, which was found incomprehensible and frustrating to new users. The usability team streamlined the tutorial, cut the amount of data and descriptions presented to user and polished it with innovative new level design. The new tutorial performed well in usability tests. The core developers and the broader community were enthusiastic about it. However, later on it was revealed that the creator of the original tutorial had reverted the tutorial almost back to its previous version in the next major release. Hence, the usability improvements were obliterated. This did not involve any discussion on the matter or explicit criticism on the work of the usability team. The original creator of the tutorial had the right to modify the software the way he wished, and he exercised this right.

Subsequent usability teams working with the same project noticed later on that parts of the user interface and functionality changes requested by the first usability team working with the case project two years earlier that had already accepted and implemented by the core-developers, had been reverted almost completely to the original. This discarded the usability improvements. One core developer hinted to the later usability teams that another core developer had some very strong opinions on a certain part of the user interface and this core-developer had reverted back the changes to bring back this part of the user interface as it was originally and as he liked it, without a collective decision by all core-developers. None of the other core-developers had reacted to this in any way.

4.4 Usability Contributions Becoming Filtered In

Despite encountering these gatekeeping tactics when aiming to enter into and contribute to FLOSS projects, many successes have also been achieved by our usability teams. In particular, we view cases B and E as largely successful ones. In these two cases, the usability specialists and their contributions, at least partly, have succeeded in avoiding the gatekeeping tactics and become 'filtered in'. In case B, the usability team managed to avoid the gatekeeping tactics of non-response and false acceptance, albeit the gatekeeping tactic of social exclusion became visible through the usability team working on issues that already had become obsolete in the project due to the recent developments. We maintain that this success was due to the small size of development team and community as well as due to their willingness to have new people contributing to the development; the usability team was able to access the core developers and managed to contribute to the community in such a way that their contributions were accepted and appreciated. They were able to get in contact and interact with the core developers and the community before and during their intervention and they were able to address and convince the true decision-makers in this community. However, it may have been that the usability team accidentally succeeded in producing such a solution that the developers and the community were satisfied with, or at least beforehand the usability team did not specially consider the risk of false acceptance emerging nor tried eliminating it. In this case project there was no prior knowledge about usability, but instead the usability team educated the developers about the matter. Through this, however, it seems that the usability team succeeded in convincing the developers and the community of the value of their work and their redesigned solutions also happened to be compatible with the existing goals and visions of the decision-makers.

In case E, moreover, the usability specialists also managed to avoid most of the gatekeeping tactics, except for the false acceptance tactic. Also here we assume that this was due to the developers and community being very open to outsider contributions and the decision-makers being easy to contact and interact with. However, despite of this apparent openness for outsider contributors, the gatekeeping tactic of false acceptance was eventually utilized by at least some of the developers to block the contributions that they did not like. This happened even when these contributions were accepted by the other developers. In these instances, the usability team succeeded in arousing interest in the community and convincing the core developers initially, but this was not enough for their contribution to survive. This would likely have required another kind of design solution, but the usability team was unaware of their solution not meeting the needs or desires of some of the decision-makers.

5 Discussion

Several HCI studies have addressed the introduction of usability activities into FLOSS development context [7-12,29-36,38,39,42], but the challenges as regards this still seem to prevail. Moreover, there has not previously been reported this kind of long-term research intervention involving multiple FLOSS case projects, spanning multiple years and focusing on the boundary management and gatekeeping aspects of the interactions between the usability specialists and FLOSS developers. Based on the results of our analysis of usability work in FLOSS development, this study identified and characterized three tactics of gatekeeping: *non-response*, *social exclusion,* and *false acceptance*. Even though guidelines for introducing usability activities into FLOSS development have been proposed and experimented with earlier (see e.g. [12,35,39]), the gatekeeping tactics performed by FLOSS developers and communities against proposals to change have not been studied and categorized through this kind of longitudinal multiple case studies. Our studies imply that the usability specialists interested to conduct usability activities in FLOSS development context have to seriously take into account these forms of gatekeeping by the FLOSS developers and to be prepared for them. Next, the tactics are first summarized, after which some characteristics from successful cases are recapped.

As mentioned earlier, in cases A, D, and F the developers initiated a reply to usability specialists about the identified usability issues and proposed usability changes only after multiple attempts of contacting them. In these cases, the developers wanted not to open a public discussion about these proposed changes, but to silence the issue down. This kind of attitude was also indicated by certain discussion forum messages, where developers sometimes expressed open hostility towards any criticism or improvement suggestions about the user interface by the end-users. These occurrences indicate that the developers used the gatekeeping tactic labeled *non-response* to silence usability discussions even before they started. This tactic gains further support from the FLOSS literature in which "receiving an improper answer", and "not receiving a (timely) answer" have been reported as contribution barriers to newcomers [25]. These results have not been produced related to usability work, while our study shows that these are relevant also as regards usability work

On the other hand, in cases B and C, the developers replied to and interacted with the usability specialists, but the usability specialists were, nevertheless, left out from the decision-making and planning processes. In case C, the usability specialists were free to conduct usability activities, but even though there were some words of encouragement and comments from core-developers regarding usability work, the usability specialists were left completely isolated from the decision-making arena, their voice was not heard within the community and the software was not changed as recommended. In case B, the planning process should have been visible also to the usability team to enable them to contribute in a meaningful way. Additionally, they were not allowed to take part in decision-making process in this project; hence the developers had the sole authority to decide which usability improvements ended up as being implemented and which ones were disregarded. These occurrences show that the developers used the gatekeeping tactic of *social exclusion*. Also this gatekeeping tactic can be connected with the existing FLOSS research that has indicated that "lack of social interaction with project members" acts as a contribution barrier for new members in FLOSS projects [25]. In our cases, the usability team should have been integrated into the planning and decision-making processes of the FLOSS projects or at least they should have been better informed of the outcomes of such processes. The lack of integration led to not keeping in pace with the development. As the rhythms of the development and usability were different, usability specialists ended up solving problems of yesterday. At the point of delivery, the development had already moved several steps ahead, and the usability efforts were obsolete. However, we acknowledge that it is a true challenge for any usability person to acquire a highly influential position in a FLOSS project and hence the gatekeeping tactic of social exclusion is very likely encountered by usability specialists working in FLOSS projects also in the future.

In cases B and E, on the other hand, the developers and the whole community were very enthusiastic about the work of the usability team and praised their reports of good quality. The developers implemented many of the suggested usability improvements and in the case E, the usability specialists and their work was referred to also in commit messages. However, in this case unfortunately some of the changes were rolled back in the next major release. These changes were not discussed in the community and even though other developers must have noticed the changes, they did not comment about them, at least not in public. All in all, in this case individual developers used the gatekeeping tactic of *false acceptance* directly to nullify the changes introduced by the usability team and it can be argued that also other developers utilized this tactic indirectly, because of their silence.

Despite all these challenges, many successes have also been achieved: the usability teams have succeeded in avoiding the gatekeeping tactics and become 'filtered in'. However, these gatekeeping tactics have been identified only after the fact and hence the usability teams could not prepare for them before or during their intervention. Still, these successful cases can be discussed here, as they may provide help for planning future interventions. It seems that the small size of development team as well as their willingness to have new people contributing to the development played a role in these successes. The usability team in these cases was able to access and directly interact with the core developers. In case B, it seems that the usability team also succeeded in convincing the developers of the value of their work and their redesign solutions also happened to be compatible with the project goals and visions.

However, we do not only wish to emphasize the negative aspect of boundary management and gatekeeping in online communities in general or in FLOSS projects in particular. Along with the existing literature, we highlight that boundaries and boundary management play a critical role in such communities: "Boundaries have an important dual nature. They *enable* the community to grow and thrive, but they also *protect and secure* the community from external threats." [13] Online communities need to protect their boundaries and ensure that only meaningful contributions get filtered in. For usability specialists, thus, it is of essential importance to understand what is considered meaningful in the particular community.

Regarding recommendations for practice, the boundary management literature can be utilized for offering advice for usability specialists on what kinds of things to consider when trying entering into a FLOSS project. The boundary management literature discusses different kinds of boundary logics that can be considered also in our context. The boundary logic of *identity* concerns the coherence of the community and its activities, the boundary logic of *power* relates to controlling key resources, defining the suitable domains of activity and influence in the community and managing external relationships, the boundary logic of *competence* addresses the critical competences for participation in these communities and the boundary logic of *transactional efficiency* relates to the resource view of the communities and economically efficient production models [13]. Applying these findings into our context leads to the following considerations. For enthusiastic usability specialists it might be essential to try to understand the identity of the community in question before entering into it and trying to introduce changes into it. This includes understanding the identity of the community as well as that of the software in question. In FLOSS projects there may be critical issues and ideological underpinnings underlying the design solution and usability specialists should understand those before messing with the solution. Otherwise the gatekeeping tactic of false acceptance may emerge when the decision makers notice that important aspects of their solution have become neglected. As regards power and politics in FLOSS projects, they have already been acknowledged as critical concerning usability work [11]. However, how to address them is still a question mark. Based on the boundary management literature we suggest that for usability specialists, figuring out the power related issues of interest may include finding out who controls the key resources in the project, defines the suitable domains of activity, has influence in the community and manages external relationships as well as how all these issues are accomplished in practice (cf. [13]). This should help usability specialists targeting right people at right time through right means. Finally, the issue of competence has already been brought up in the HCI literature on FLOSS projects: usability specialists tend not to have valued type of competence in FLOSS projects [11,29,30,36,38]. We do not have a solution for this problem, but our data points out (in case E) that usability specialists may benefit from having also technical skills. In our data one of the usability team members succeeded in getting commit rights and status as a developer. This usability team also contributed code to the FLOSS project in question. However, we do not wish to claim that technical skills are mandatory for usability specialists entering FLOSS projects. Many times usability specialists do not possess such skills and they should be allowed to contribute to FLOSS projects despite that. We suggest that usability specialists bring this issue to be openly discussed within the FLOSS project in question. Even this kind of a procedure may help the situation, albeit surely many challenges will still prevail for usability specialist attempting to enter into and to contribute to FLOSS projects.

One major takeaway from this study is the finding that FLOSS projects as fluid communities with blurred boundaries make also usability specialists' participation "blurry". In traditional closed source software development organizations, one knows who belongs to a project through an employment contract or other arrangement. In FLOSS development the boundary between inclusion and exclusion of potential participants is not that clear. The tactic of non-response signals that even when a newcomer has joined the project discussion forum, it may be difficult to actually enter the project and contribute to it in practice. The tactic of social exclusion shows that even after usability specialists have gained access within the boundary, they may remain on the peripheral edge of the FLOSS development onion unable to gain access to the deeper layers, and therefore be out of sync from the rhythm of how things develop in the inner layers of the FLOSS development onion. The tactic of false acceptance underscores that, even after usability has been accepted, integrated, and synchronized with the activities of the core developers, it may be disregarded retrospectively.

All in all, this paper contributes to the boundary management literature by providing a detailed examination of boundary gatekeeping tactics in action in FLOSS development context. By analyzing the FLOSS projects balancing between openness and control, we utilized the boundary management concept of gatekeeping and found that the concept of gatekeeping tactic applies in FLOSS development context in the sense of limiting outsider contributions, even though this goes against the philosophy of FLOSS movement, which highlights openness, participation, and lack of gatekeeping. Furthermore, this paper also contributes to Human Computer Interaction literature where there has been interest and several attempts to find ways to contribute to FLOSS projects through usability work. By identifying these gatekeeping tactics that were used by FLOSS projects to silence, block or revert usability work, we have made these kinds of actions visible in FLOSS projects. We recommend gatekeeping tactics as analytical lenses to study further the issue of contributing to FLOSS projects through usability work.

6 Conclusion

HCI research has identified numerous barriers for usability specialists' participation in FLOSS projects. Organizational boundary literature is interested in the tensions of online communities, including FLOSS communities. While this literature recognizes the importance of managing boundaries in online communities, little empirical research has been conducted on actual gatekeeping tactics project members perform against outsiders' contributions. Based on several years of engaged research with FLOSS projects, we characterize three gatekeeping tactics in FLOSS projects: *non-response*, *social exclusion*, and *false acceptance* that have hindered usability work.

By following the idea of Wittgenstein that "whereof one cannot speak, thereof one must be silent", we propose that the identification of the three tactics will have much practical relevance. As we have given names and visibility to these gatekeeping tactics, it is now possible to talk about them, to analyze them, and to be better prepared for them. When usability specialists face these barriers in action, they can try to identify the possible causes of them and communicate these issues on the broader forum.

Our choice of studying the FLOSS cases from the usability specialists' perspective has provided us with valuable insights, by allowing us to identify these three gatekeeping tactics. We can now analyze and understand better how online communities, including FLOSS projects, balance between openness and control. This choice of perspective is, however, also a limitation. The tactics may look different through the eyes of the insiders. Therefore, future research questions may involve: "How intentional are the gatekeeping tactics from within the boundary?", "Is it possible to create a greater transparency to alleviate the need of such tactics?", and "What are the strategies to help usability processes keep pace with the onwards moving core development?" Additionally, the usability specialists might use usability cost-benefit considerations to highlight the value of their usability work [37]. Although our research program has already proposed methods for usability specialists on how to enter into and contribute to FLOSS projects, the methods placing particular emphasis on understanding the project in question and active collaborate with the developers [12,35,39], many challenges still prevail. Further studies relying on a ‘user-centered approach’, the users here referring FLOSS developers, should be initiated to understand better the needs of the FLOSS developers, both as regards their work and as regards being more open to usability work.

7 References

1. Stallman, R.: The GNU Manifesto. Dr. Dobb’s Journal of Software Tools 10(3) (1985)
2. Von Hippel, E., Krogh, G.: Open source software and the “private-collective” innovation model: Issues for organization science. Organization science, 14(2), pp. 209-223 (2003)
3. Von Hippel, E.: Innovation by User Communities: Learning from Open-Source Software. MIT Sloan Management Review 42(4): pp. 82-86. (2001)
4. 4 Franke, N., Shah, S.: How communities support innovative activities: an exploration of assistance and sharing among end-users. Research policy 32(1), pp. 157-178 (2003)
5. Gulliksen, J., Boivie, I., Göransson, B.: Usability professionals – current practices and future development. Interacting with Computers 18(4), pp. 568-600 (2006)
6. Iivari, N.: Discourses on ‘culture’ and ‘usability work’ in software product development. Acta Universitatis Ouluensis, Series A, Scientiae rerum naturalium 457 (2006)
7. Andreasen, M., Nielsen, H., Schrøder, S., Stage, J.: Usability in Open Source Software Development: Opinions and Practice. Information Technology and Control 25:3A (2006)
8. Bødker, S., Nielsen, L., Orngreen, R.: Enabling user-centered design processes in open source communities. In: Proc. HCII, Part I: Usability and Internationalization, LNCS 4559, pp. 10-18 (2007)
9. Cetin, G., Verzulli, D., Frings, S.: An Analysis of Involvement of HCI Experts in Distributed Software Development: Practical Issues. In: D. Schuler (ed.): Proc. Human Computer Interaction International: Online Communities and Social Computing. LNCS 4564, (2007)
10. Nichols, D., Twidale M.: The Usability of Open Source Software. First Monday (2003)
11. Rajanen, M., Iivari, N.: Power, Empowerment and Open Source Usability. In: Proceedings of CHI 2015 (2015)
12. Rajanen, M., Iivari, N., Anttila K.: Introducing Usability Activities into Open Source Software Development Projects – Searching for a Suitable Approach. Journal of Information Technology Theory and Application 12,4, pp. 5-26 (2011)
13. Jarvenpaa, S. L., Lang, K.R.: Boundary Management in Online Communities: Case Studies of the Nine Inch Nails and ccMixter Music Remix Sites. Long Range Planning 44(5/6), pp. 440-457 (2011)
14. O'Mahony, S., Ferraro, F.: Managing the Boundary of an 'Open' Project. Harvard NOM Working Paper No. 03-60 (2004)
15. Faraj, S., Jarvenpaa, S. L., Majchrzak, A.: Knowledge Collaboration in Online Communities. Organization Science 22(5), pp. 1224-1239 (2011)
16. Shaw, A.: Centralized and Decentralized Gatekeeping in an Open Online Collective. Politics & Society 40(3), pp. 349-388 (2012)
17. Raymond, E.: The cathedral and the bazaar. Knowledge, Technology & Policy 12:3 (1999)
18. Coleman, G.: The Political Agnosticism of Free and Open Source Software and the Inadvertent Politics of Contrast. Anthropological Quarterly 77(3), pp. 507-519 (2004)

19. Bach, P. M., Terry, M.: The future of FLOSS in CHI research and practice. In: CHI '10 Extended Abstracts, Atlanta, Georgia, USA. (2010)
20. Aberdour, M.: Achieving Quality in Open Source Software. IEEE Software 24:1 (2007)
21. Barcellini, F., Détienne, F., Burkhardt, J.-M.: A Situated Approach of Roles and Participation in Open Source Software Communities. Human–Computer Interaction 29(3), (2014)
22. Feller, J., Fitzgerald, B.: A framework analysis of the open source software development paradigm. In: Proceedings of the ICIS, pp. 58-69 (2000)
23. Mockus, A., Fielding, R.T. Herbsled, J.: A Case Study of Open Source Software Development: The Apache Server. In: Proc. ICSE, pp. 263-272 (2000)
24. Ljungberg, J.: Open Source Movements as a Model for Organizing. European Journal of Information Systems 9,4, pp. 208-216 (2000)
25. Steinmacher, I., Graciotto Silva, M.A., Gerosa, M.A., Redmiles, D.F.: A systematic literature review on the barriers faced by newcomers to open source software projects. Information and Software Technology 59(0), pp. 67-85 (2015)
26. Von Krogh, G., Spaeth, S., Lakhani, K.R.: Community, joining, and specialization in open source software innovation: a case study. Research Policy, 32(7), pp. 1217-1241 (2003)
27. Clayman, S. E., Reisner, A.: Gatekeeping in Action: Editorial Conferences and Assessments of Newsworthiness. American Sociological Review 63(2), pp. 178-199 (1998)
28. Lewin, K.: Frontiers in Group Dynamics: II. Channels of Group Life; Social Planning and Action Research. Human Relations, 1(2), pp. 143-153 (1947)
29. Bach, P., Carroll, J.: FLOSS UX Design: An Analysis of User Experience Design in Firefox and OpenOffice.org. In: Proc. OSS, pp. 237-250 (2009)
30. Bach, P., DeLine, R., Carroll, J.: Designers Wanted: Participation and the User Experience in Open Source Software Development. In: Proc. CHI 2009, pp. 985-994 (2009)
31. Benson, C., Müller-Prove, M., Mzourek, J.: Professional usability in open source projects: GNOME, OpenOffice.org, NetBeans. In: Extended Abstracts of the CHI2004, (2004)
32. Nichols, D., Twidale, M.: Usability Processes in Open Source Projects. Software Process Improvement and Practice 11, pp 149-162 (2006)
33. Zhao, L., Deek, F.: Improving Open Source Software Usability. In: Proc. of the 11th Americas Conference on Information Systems (AMCIS2005), Omaha, USA (2005)
34. Zhao, L., Deek, F.: Exploratory inspection: a learning model for improving open source software usability. In: Extended Abstracts CHI 2006, pp. 1589-1594. (2006).
35. Rajanen, M., Iivari, N.: Open Source and Human Computer Interaction Philosophies in Open Source Projects – Incompatible or Co-Existent? In: Proc. of Academic MindTrek 2013. Tampere, Finland (2013)
36. Terry, M., Kay, M., Lafreniere, B.: Perceptions and Practices of Usability in the Free/Open Source Software (FOSS) Community. In: Proc. CHI 2010, pp. 999-1008 (2010)
37. Rajanen, M., Iivari, N.: Traditional Usability Costs and Benefits - Fitting them into Open Source Software Development. Proc. ECIS 2010 (2010)
38. Bach, P., Twidale, M. Social participation in open source: what it means for designers. Interactions 17(3) (2010)
39. Rajanen, M., Iivari, N., Keskitalo, E. Introducing usability activities into open source software development projects: a participative approach. Proc. NordiCHI, 683-692. (2012)
40. Walsham, G.: Interpretive case studies in IS research: nature and method. European Journal of information systems 4(2), pp. 74-81 (1995)
41. Walsham, G.: Doing interpretive research. European journal of IS 15(3) (2006)
42. Iivari, N., Rajanen, M., Hedberg, H.: Encouraging for Enculturation - An Enquiry on the Effort of Usability Specialists Entering OSS Projects. Proc. ACIS 2014. (2014)