LEARNING TO REMAIN - EVALUATING THE USE OF MENTORING FOR THE RETENTION OF FLOSS DEVELOPERS

Andreas Schilling  
*University of Bamberg*

Sven Laumer  
*University of Bamberg*

Follow this and additional works at: [http://aisel.aisnet.org/ecis2012](http://aisel.aisnet.org/ecis2012)
LEARNING TO REMAIN – EVALUATING THE USE OF MENTORING FOR THE RETENTION OF FLOSS DEVELOPERS

Schilling, Andreas, Centre of Human Resources Information Systems, University of Bamberg, Feldkirchenstraße 21, 96052 Bamberg, Germany, andreas.schilling@uni-bamberg.de
Laumer, Sven, Centre of Human Resources Information Systems, University of Bamberg, Feldkirchenstraße 21, 96052 Bamberg, Germany, sven.laumer@uni-bamberg.de

Abstract

The retention of newcomers is of vital relevance for initiatives developing Free Libre Open Source Software (FLOSS). Based on previous FLOSS research which repeatedly highlights the amount of knowledge developers build in the beginning of their project participation for their continuance and experiences from the organizational domain, we evaluate the use of mentoring as an appropriate knowledge transfer and retention strategy for FLOSS projects. Combining FLOSS and organizational literature, we develop our research model and hypothesize that (i) mentoring facilitates novices in their learning, which, in turn, increases their retention (ii) and (iii) that mentoring has direct effects on protégés’ continued participation. The evaluation of 91 newcomers to the KDE project supports our hypotheses and finds a strong direct and indirect association between mentoring and novices’ project retention. On the one hand our analysis shows that mentoring significantly increases novices’ achieved level of knowledge after their ramp up period which, in turn, enlarges their project permanence. On the other hand, we find evidence that there is also a strong and significant direct association between mentoring and protégés’ retention behaviour, which could be the result of the strong interpersonal relationship which is formed between mentors and protégés.

Keywords: FLOSS, Open Source Development, Turnover, Retention, Mentoring, Knowledge Transfer.
1. Introduction

Initiatives developing Free Libre Open Source Software (FLOSS) are of high relevance for our daily life. For example, today most mobiles worldwide run based on FLOSS (Gartner Inc., 2011) and nearly two thirds of European and American companies rely on FLOSS for their mission critical operations (Gold, 2007). At the same time, however, the huge majority of all FLOSS initiatives fail, most commonly due to a lack of sustained contributors (Colazo and Fang, 2009). While previous research repeatedly identified learning as the most important driver for FLOSS developers’ on-going project permanence, there are no conceptual approaches - we are aware of - on how FLOSS projects can support the knowledge building process of their newcomers and, therefore, foster their retention behaviour in order to compensate the temporal commitment of the existing contributor base (Robles et al., 2009) for guaranteeing their continuity.

Likewise, organizations depend today more than ever on the skills and knowledge of their employees. According to a recent estimate, organizations loose up to a third of their knowledge every year (Anderson and Lee, 2007). Hence, the continuous training of the workforce is a top priority for employers. Also IT companies invest considerable amounts of their resources for the training of their workforce and develop several measures for sharing and transferring knowledge to new employees. One of these training initiatives is mentoring. It describes a dyadic training relationship in which an experienced senior (the mentor) provides technical advice and interpersonal support to an inexperienced newcomer (the protégé) (Kram, 1988). In the case of organizations, this special training method has been shown repeatedly to be very effective for assisting newcomers by advancing their knowledge and strengthening their intention to remain (Parise and Forret, 2008). As a result more than 80 per cent of top organizations rely for the training of their personnel on mentoring (Subrahmanian and Anjani, 2011).

Drawing on the similarity with the challenges organizations are facing in terms of knowledge transfer and retention, we evaluate in the following the use of mentoring for FLOSS projects. Mentoring requires only the time of the respective mentor, therefore, it is predestined for the knowledge transfer in FLOSS initiatives. Based on the positive effects organizations experience with the use of this special training method in terms of educating and retaining their workforce, we will evaluate mentoring as a potential strategy for FLOSS projects to assist newcomers in building new knowledge and increasing their sustained commitment. Consequently, we will examine the following research question:

Is mentoring an appropriate strategy for FLOSS projects to support the knowledge building of newcomers and facilitate their retention behaviour?

By providing an answer to this question our research has theoretical and managerial contributions to FLOSS and organizational research. We contribute to existing FLOSS literature by evaluating if and how project members’ learning and their on-going commitment can be facilitated through the use of mentoring. Furthermore, our research provides managers of FLOSS initiatives with practicable advice on how to assist newcomers’ knowledge building and their continued participation which, in turn, enhances their projects’ chances to succeed. Besides extending FLOSS literature, our research addresses the repeated call for further studies on the effects of mentoring on the continuance behaviour of protégés (Parise and Forret, 2008). Finally, our work provides practical advice for organizations, considering their need to continuously monitor the employees’ learning progress (Gartner Inc., 2007).

In order to evaluate the use of mentoring as a training and retention strategy for FLOSS projects, this paper is structured as follows: In the next section we describe the theoretical background of our research by detailing the concepts and effects of mentoring and presenting a review of relevant FLOSS literature. With this background information, we develop our research model and formulate the corresponding research hypotheses. For evaluating our research model, we describe in section three our research methodology, in which we analyse the retention behaviour of mentored newcomers at the
KDE project, and put it in contrast to non-mentored project novices. Finally, we discuss the findings of our evaluation and their implications for future research and present a conclusion of our research.

2. **Research Model Development**

Within this section we develop the research model and describe the concepts of mentoring and FLOSS developers’ knowledge building in more detail.

2.1 **Mentoring**

Mentoring describes a teaching method according to which an experienced professional (the mentor) provides technical assistance and psychological support to an inexperienced individual (the protégé) (Kram, 1988). Giving technical guidance, mentors support their protégés in their doing and help them in advancing their knowledge. In addition, mentors provide protégés with psychological support in from of counselling or friendship, which creates a strong interpersonal bond between the two. This dyadic relationship is a special characteristic of mentoring relationships, which separates them to traditional teaching settings such as between teachers and their students or between supervisors and their employees. As a consequence of this intense one-to-one training, mentoring is seen as most effective measure for assisting novices in building different forms of knowledge (Hale, 2000). In the first place, mentors help their protégés acquiring declarative knowledge by assisting them in understanding the necessary facts and routines for doing their jobs successfully. Besides fostering protégés’ knowledge of what to do, mentoring relationships also enable the transfer of procedural knowledge, the understanding of how to accomplish a particular task. This tacit form of knowledge arises typically only through practical experience and is therefore difficult to convey in traditional teaching relationships. Based on the strong interpersonal bond which is formed between mentors and their protégés, mentoring relationships, however, have been found very effective to convey procedural knowledge in addition to declarative knowledge. Previous research repeatedly attests the superiority of mentoring as teaching method, especially in the case of organizations (Hale, 2000). Existing literature which applied mentoring in the organizational context finds not only advances in protégés learning but also stimulating effects on employees’ job satisfaction and their intention to stay in the organization (Hale, 2000; Brashear et al., 2006). Driven by these positive effects on protégés’ knowledge building and their talent development, organizations use formal mentoring initiatives as a measure to train and retain their employees effectively. In contrast to informal mentoring relationships, which may arise spontaneously between colleagues for an uncertain amount of time and in various forms and intensity, formal mentoring programs are initiated by the organization, have a concrete goal and a defined timeline. Although empirical research on this aspect is very scarce, previous studies suggest that formal mentoring is a highly effective organizational measure for the training and retention of employees. With their qualitative analysis Eby and Lockwood (2005) suggest that formal mentoring programs not only realize the expected learning advancements of mentoring but have, in addition, unique benefits for protégés. Employees who have been mentored report of advancements in their career planning and value the participated mentoring program because it provided them with new networking opportunities (Eby and Lockwood, 2005). In addition to the achieved knowledge gains, the authors also report that friendship is commonly formed through formal mentoring initiatives between protégés and their mentors. The positive effects resulting from formal mentoring which are suggested by Eby and Lockwood (2005) are supported by empirical research of Lentz and Allen (2009). According to their findings, mentoring has sustainable benefits for both employees and employers. Besides advancing protégés’ knowledge and fostering their talent, the researchers identified that mentoring programs positively affect job satisfaction, the intention to continue and alleviates negative experiences of career plateauing.
2.2 Knowledge Building and Developer Retention in FLOSS Projects

An important reason for project members’ sustained participation in FLOSS projects is their wish to extend their knowledge. Studying the motivation of FLOSS contributors, Ghosh (2005) shows that members continue in their projects primarily because they want to build new knowledge. Consistent with this finding, research results by David and Shapiro (2008) and Wu et al. (2007) emphasize the relevance of members’ learning for FLOSS projects in order to retain their commitment on a longitudinal base. According to Fang and Neufeld (2009) FLOSS developers’ learning behaviour is not an isolated information absorption process but rather consists of individuals’ interpersonal and practical experiences within the project. Drawing on Legitimate Peripheral Participation Theory, the authors propose situated learning as a theoretical foundation to explain FLOSS developers’ knowledge building. Situated learning regards knowledge building as a social process which is situated in individuals’ everyday practices. Thereby, new knowledge is built through problem oriented interaction with others such as participation in related discussions, giving and receiving feedback and making proposals regarding possible solutions (Edmondson, 1999). Beside the consideration of the social context, situated learning puts a strong emphasize on one’s practicing in order to build new knowledge. According to the theory, learning occurs through one’s practical and purposeful problem oriented interactions. Although only few subsequent literature focuses on FLOSS developers’ learning behaviour, recent research by Singh et al. (2010) support the relevance of both relational and situational aspects for FLOSS developers’ knowledge building. Based on a detailed analysis of FLOSS developers’ contribution history, the authors propose a Hidden Markov Model to capture the learning states of project members. Consistent with the theory of situated learning, the results of Singh et al. (2010) demonstrate that FLOSS developers’ learning is constituted of their conversational and practical project experiences. Extending existing research, the authors provide further insight into FLOSS developers’ learning activities. In contrast to prior expectations, the authors found that FLOSS developers gain only little knowledge through code contributions. Rather than contributing code, Singh et al. (2010) found that inexperienced project members benefit most from reading mailing list conversations and participating in problem related discussions opened by others. Initiating mailing list conversations on the other hand, is the best means for advancing the learning of developers who already acquired a basic understanding of the project and its development. Finally, developers in a high learning state extend their knowledge best when helping others with their problems. Although Singh et al. (2010) found that project members can experience positive learning effects resulting from their activities in the project, their results are alarming for managers of FLOSS initiatives. They show that it is very unlikely that members ever change their initial knowledge state. This finding is supported by Adams et al. (2009). The researchers show in their analysis that it requires significant efforts for newcomers to become effective contributors. Based on the monitored project activities of newcomers, the researchers estimate that it would take them up to 60 weeks for becoming an effective contributor. However, as a result to their unmet learning expectations, many novices leave the project before experiencing advances in their knowledge. In light of the temporary commitment of the existing contributor base (Robles et al., 2009), the inability of FLOSS projects to retain their newcomers becomes a serious threat for their future existence.

While prior FLOSS literature repeatedly highlights the importance of learning for members’ continuance and warns of possible learning barriers which hinder the acquisition of new knowledge, there is no conceptual approach, we are aware of, which enables project managers to actively support the learning behaviour of their novices. In their research Li et al. (2006), examine how project managers’ transformational and transactional leadership styles affect members’ motivation and their future contribution intensity. Thereby, the researchers attest newcomers’ individualized consideration and intellectual stimulation as most effective measures to enhance their motivation to contribute. Building on these findings, we present in the following section a research model for investigating the effect of the one-to-one training technique mentoring on newcomers’ knowledge building and retention behaviour.
2.3 **Hypotheses Development**

Building on companies’ experiences, we propose in the following mentoring as a potential strategy for FLOSS projects in order to facilitate the learning of their novices and retain their commitment on longitudinal base. Although there are substantial differences between organizations and FLOSS projects regarding remuneration and regulation, the talent development of individuals in both contexts is considered alike (Crowston et al., 2007). Furthermore research findings by Fang and Neufeld (2009) suggest that novices to FLOSS projects and newcomers in organizations are similar in the ways they acquire new knowledge. Based on this fitness and the stimulating effects of mentoring in the case of organizations, we evaluate this special training method in the following for FLOSS initiatives. Our research model is visualized in Figure 1.

In line with organizational literature, we expect that mentoring relationships advance the learning behaviour of protégés such that mentored newcomers show a higher level of project related knowledge than non-mentored novices. Based on previous research by Fang and Neufeld (2009), mentors are supposed to facilitate FLOSS novices’ learning in two ways. First mentors provide newcomers to FLOSS projects with help on theoretical and practical aspects which are relevant for them to accomplish their programming work. Similar to the experiences in corporate software development this technical support is supposed to advance protégés’ declarative knowledge building (Chen and McQueen, 2010). Second, the strong relational bond between the mentors and the protégés is supposed to facilitate the transfer of procedural knowledge. As a result, we assume that protégés acquire with the support of their mentors much sooner an understanding for the necessary routines and practices to contribute to the FLOSS project efficiently. Consequently, we hypothesize that mentoring programs in FLOSS projects advance the knowledge building of protégés.

**H1:** Mentored project members show higher levels of knowledge building than non-mentored project members.

Following existing literature, we assume that newcomers’ advancement in their knowledge building stimulates them to continue their commitment in the FLOSS project. According to the theory of situated learning, individuals’ continuance intention is mutually reinforced by their learning progress and the practicing of their knowledge (Fang and Neufeld, 2009). Hence, a higher knowledge level not only increases new project members’ capabilities to contribute but also strengthens their wish to remain in the project. Consistent with this theoretical assumption of situated learning, FLOSS studies conducted by Ghosh (2005), David and Shapiro (2008) and (Wu et al., 2007) identify contributors’ learning as most important for their project continuance. Building on previous research results by Adams et al. (2009) which emphasize especially newcomers acquired knowledge level after their ramp-up period for their future contributions to the project, we hypothesize:

**H2:** Newcomers with higher levels of knowledge after their ramp up period retain longer in a FLOSS project

In addition to facilitate the acquisition of new knowledge, mentoring relationships have been found to increase the retention behaviour of protégés. According to Reid et al. (2008) mentoring initiatives strengthen protégés’ intention to stay by increasing their commitment for the job. Similarly, other researchers support that mentoring relationships are an effective measure for decreasing protégés’ turnover behaviour and increasing their satisfaction with the job (Thatcher et al., 2002). Based on these findings from organizational research, we assume that mentors in the FLOSS domain can help their protégés to align their project work with their personal interests and goals. As a consequence to this guidance, mentored newcomers to FLOSS projects are supposed to perceive their work much more satisfying and enjoying, which in turn has been shown to strengthen their retention behaviour (Ke and Zhang, 2009). Hence, we hypothesize that:

**H3:** Mentored project members remain longer in the FLOSS project than non-mentored members.
3. Research Methodology

For validating the proposed hypotheses an annual mentoring event in which newcomers to FLOSS initiatives are mentored by senior developers has been investigated. Google Summer of Code (GSoC) is a popular mentoring event every year which is designed for students over 18 who would like to contribute to a FLOSS project during their summer break. For participating in the event and receiving a 3-month stipend, students have to choose one of the participating FLOSS projects in their application and detail what they would like to contribute to this project. Based on the received applications and the amount of available stipends which can be assigned, each project has to select the students who are granted stipends for GSoC. KDE, the default working environment on many Linux distributions, is one of the projects students can contribute to during the GSoC event. For examining the effects of mentoring in the FLOSS context, we choose to compare the knowledge gains between mentored and non-mentored newcomers to the KDE project. The KDE project provides us with several distinct advantages for our evaluation. First, KDE consists of a rich variety of subprojects and contributors. In more than 300 subprojects, KDE members with different cultural and intellectual background are working together for developing FLOSS. Another characteristic of the KDE project from which we benefit for our evaluation is its popularity in the community. It was the top project to which members of the last two events (GSoC-09 and GSoC-10) contributed two. Hence, it is the GSoC project with the most mentored FLOSS developers. Another consequence of KDE’s popularity is its attraction of many new contributors which would like to become active in its development.

Eby et al. (2004) previously revealed that protégés’ subjective evaluations of their training relationships are a major threat to validity for studies on mentoring. With respect to these findings we use in the following archival data for examining mentored and non-mentored newcomers’ level of acquired knowledge after their first 30 weeks in their project (which is according to Adams et al. (2009) the necessary ramp-up period for KDE developers) and their continued project permanence. By evaluating archival data on members’ learning progress and their retention behaviour, we are not only able to adequately abstract from their subjective perceptions but also study the effects of mentoring in retrospect.

3.1 Measurement

Similar to formal mentoring initiatives in the organizational domain, mentors in the context of GSoC provide their protégés with both technical advice and interpersonal support. For accomplishing their programming projects mentors help their students to understand the necessary programming techniques and the routines used to compile and integrate their code into KDE’s codebase. In complement to the transfer of declarative knowledge, mentors assist their protégés also in gaining procedural knowledge by talking about their experiences in the projects’ development and the lessons they have learned. During the mentorship, mentors develop an intense relationship with their protégés, which often gives protégés the necessary comfort to engage in related mailing list discussions and
makes them more robust for reacting to problems occurring in their work. To identify mentored and non-mentored novices to the KDE project, we used for both groups a twofold extraction approach. For identifying newcomers to the project who have been mentored, we selected from all participants of the last two GSoC events those who are no longer than at most four months before the beginning of their mentoring initiative registered as a KDE developer, so that the mentoring event is included in their ramp up period. To do so, we used KDE’s central member directory to extract the registration dates of all participants of the last two GSoC events. To identify non-mentored project novices we approached similarly. First we mined the log file of KDE’s central member directory for all new entries between 1st January 2010 and 1st July 2010. Next, we filtered for those members who are interested in contributing to the project and, hence, submitted at least one code commit during their first month to KDE’s code repository. To do so we queried the online service markmail.org which indexes the mailing list kde.cvs-commit to which every commit to KDE’s subprojects gets propagated. To separate the considered developers in our data analysis, we use the dichotomous variable “mentored” to indicate whether the KDE newcomer has been mentored or not.

To develop a measure for quantifying and comparing the amount of knowledge which has been built by mentored and non-mentored newcomers during their ramp up period, we orient on the comprehensive analysis of FLOSS developers’ learning behaviour by Singh et al. (2010). In line with the theory of situated learning, the authors show in their longitudinal examination of FLOSS developers’ contribution and interaction dynamics that novices’ learning depends on their practical and their social interactions within the FLOSS project. On the one hand Singh et al. (2010) find that newcomers advance their knowledge through their conversations with other project members by initiating and engaging in discussions on the projects’ mailing list. On the other hand, novices advance their knowledge through practicing, which in the case of FLOSS is represented by their participation in the code development of the project. Building on these research findings of Singh et al. (2010), we model the knowledge gains which project novices acquire during their ramp up period as a formative latent variable; this variable is constituted based on their conversational interactions through opening of and engaging in mailing list discussions and by their practicing behaviour in form of issued code commits to their project. In order to determine novices’ knowledge level, we queried the online service markmail.org which indexes all of KDE’s project mailing lists for the necessary conversation and contribution logs.

For measuring project members’ retention behaviour at their corresponding KDE subproject, we adopt a measurement technique which has previously been employed by Colazo and Fang (2009). The time developers stay active in projects’ development is thereby calculated based on the period between their most recent commit and their first commit to the corresponding project. Although mentored students participate voluntarily in GSoC and are free to leave the event at any time, we modified this measure to account for the monetary rewards they receive for their commitment. To do so, we excluded all commits of mentored students which have been issued during the GSoC event from this calculation. To compute the relevant timespan, we wrote a program routine which identifies newcomers’ very first and their most recent project commits based on the extracted commit data and identifies the corresponding timestamps. In the case of mentored novices, this data is filtered in a second step. Then, the relevant time difference in days between the extracted timestamps is computed.

To separate the considered developers in our data analysis, we use the dichotomous variable “mentored” to indicate whether the KDE newcomer has been mentored or not.

To develop a measure for quantifying and comparing the amount of knowledge which has been built by mentored and non-mentored newcomers during their ramp up period, we orient on the comprehensive analysis of FLOSS developers’ learning behaviour by Singh et al. (2010). In line with the theory of situated learning, the authors show in their longitudinal examination of FLOSS developers’ contribution and interaction dynamics that novices’ learning depends on their practical and their social interactions within the FLOSS project. On the one hand Singh et al. (2010) find that newcomers advance their knowledge through their conversations with other project members by initiating and engaging in discussions on the projects’ mailing list. On the other hand, novices advance their knowledge through practicing, which in the case of FLOSS is represented by their participation in the code development of the project. Building on these research findings of Singh et al. (2010), we model the knowledge gains which project novices acquire during their ramp up period as a formative latent variable; this variable is constituted based on their conversational interactions through opening of and engaging in mailing list discussions and by their practicing behaviour in form of issued code commits to their project. In order to determine novices’ knowledge level, we queried the online service markmail.org which indexes all of KDE’s project mailing lists for the necessary conversation and contribution logs.

For measuring project members’ retention behaviour at their corresponding KDE subproject, we adopt a measurement technique which has previously been employed by Colazo and Fang (2009). The time developers stay active in projects’ development is thereby calculated based on the period between their most recent commit and their first commit to the corresponding project. Although mentored students participate voluntarily in GSoC and are free to leave the event at any time, we modified this measure to account for the monetary rewards they receive for their commitment. To do so, we excluded all commits of mentored students which have been issued during the GSoC event from this calculation. To compute the relevant timespan, we wrote a program routine which identifies newcomers’ very first and their most recent project commits based on the extracted commit data and identifies the corresponding timestamps. In the case of mentored novices, this data is filtered in a second step. Then, the relevant time difference in days between the extracted timestamps is computed.

| Mentoring  | Dichotomous variable, separating mentored newcomers (1) from non-mentored project novices (0). |
| Knowledge level | Formative latent variable describing novices’ acquired knowledge level after their ramp up period. |
| Project retention | Timespan in days between project novices’ first and most recent commit. |

Table 1: Measurement Model
In addition to the analysed variables we control in our examination for the effects of factors, which have previously been attested to affect FLOSS developers’ retention behaviour. Following the findings of Midha (2008) and Colazo and Fang (2009), we account for the effects of the projects’ age, its total number of active developers and its size (measured by lines of code) of the corresponding KDE subproject. For assessing these control variables, we collected additional contextual information by using the online platform ohluh.net, which maintains detailed statistics for the contributor base and the code development of all analysed KDE subprojects.

3.2 Hypothesis Testing

We designed three regression models in order to evaluate the hypothesized effects of mentoring on newcomers’ knowledge building and their project permanence. The first model is the baseline model which includes exclusively the control variables. Accounting for the existence and non-existence of mentoring relationships, the second regression model incorporates, in addition to the control variables the direct effects of the dichotomous variable mentoring. In the third regression model, we incorporate furthermore the mediation effect on newcomers’ retention behaviour via their knowledge building.

An appropriate regression technique for testing our developed models is the Cox proportional hazard model. This form of survival analysis measures the simultaneous influence of the examined variables on the time to a specified event. To do so, a hazard function consisting of the analysed variables is constructed which defines the probability that a newcomer experiences a specified event at a given point in time (in our case, this event is the end of a developers’ project participation). In contrast to other regression techniques, the hazard function is not required to follow any particular shape and can include ordinal scaled variables. In addition to accounting the simultaneous effects of multiple influences, the Cox proportional hazard model can also incorporate the mediation effect (a-b-c). To do so, however, the model requires proving the positive relationship of the first relationship term (a-b) in a preliminary step.

Using the extraction and filtering strategy described above our evaluation sample consists of 91 newcomers to the KDE project. 41 of them have been mentored (16 in GSoC-09, 25 in GSoC-10) and 50 of them were non-mentored newcomers to the KDE project. Before we can start evaluating the hypothesized relationships, we have to calculate the latent variable describing individuals’ acquired knowledge after their ramp up period. In order to determine the unobserved knowledge gains from individuals’ amount of started discussions, posted replies, and the submitted code contributions, we employed Exploratory Factor Analysis (EFA). This statistical method describes the derivation of underlying unobserved factors based on the correlation between interdepended observed variables. For ensuring the adequacy of our observed variables for the use of EFA, we relied on the Kaiser-Meyer-Olkin (KMO) measure and Bartlett’s test of sphericity. Both tests supported the use of EFA for our data. The KMO value of our data is 0.618, which exceeds the required limit according to Field (2005) for the appropriate use of factor analysis. Moreover, Bartlett’s test of sphericity was significant (p<0.000), indicating that the correlation matrix is not an identity matrix. Based on the findings of these two tests we used Principal Component Analysis (PCA), which is a common extraction method for EFA that relies on linear transformations. The results of the PCA revealed that there is one factor with an eigenvector of 1.555 which accounted for 51.85% of the total variance of three knowledge building activities. Based on these results, we calculated the individual factor scores representing novices’ learning progress. With respect to the ordinal scale of the derived knowledge state variable Table 2 describes Kendal-Tau correlation coefficients for all used measures.

For evaluating the hypothesized positive relationship between mentoring and members’ learning state, we conducted a Mann Whitney U-Test. This non-parametric test assesses whether two groups differ significantly in their distribution of an ordinal value. Under the null hypothesis that the distribution of the analysed variable is in both groups alike the ranks of the analysed value in each group are summed up. If the cumulative value of both groups’ ranks is above a defined threshold, this means that one of the two groups has a significant tendency towards higher ranks than the other. In the case of mentored
and non-mentored project members, the Mann Whitney U-Test finds that the two groups differ significantly in their achieved learning state (two-sided \( p = 0.047 \)). Next, in order to evaluate the direction of the identified differences, we computed the 1-tailed significance level of the Mann Whitney U-test and multiplied it by two. The results of this significance test support hypothesis 1 by showing that mentored project novices achieved significantly \( (2\times (one-sided \ p) = 0.046) \) higher learning states after their first seven month than non-mentored project newcomers.

Based on the proven positive relationship between mentoring and members’ achieved learning state, we calculate the developed Cox proportional hazard models. We started with computing our baseline model which has the lowest explanatory power of all considered models \( (\chi^2=7.261) \). Contrary to previous research, we did not find a significant association between project size or the number of developers and newcomers’ retention. Only project age has a significant weak association with novices’ retention \( (B = 0.007, \ p = 0.025) \). Next, we analyse the second model which incorporates the considered control variables and the dichotomous classification in mentored and non-mentored newcomers. This regression model explains significantly more variance than the baseline model \( (\chi^2=19.677) \) and finds beside the weak positive association with project age, a strong and significant \( (B = -0.829, \ p = 0.001) \) relationship between mentoring and project novices’ retention. Finally, the third model, which also includes the mediating effect of knowledge building, has the highest explanatory power of all analysed models \( (\chi^2=32.294) \). It attests a significant and strong mediating association between newcomers’ acquired level of project knowledge and their continued commitment while also showing a strong direct association between mentoring and the risk that newcomers’ leave the project \( (B = -0.539, \ p = 0.003) \). Moreover, the third model finds a significant weak positive and weak negative association between novices’ permanence and the project age respectively the number of developers in the project. The detailed results of all regression models are presented in Table 3.

### 4. Discussion and Implications

#### 4.1 Discussion

The results of our evaluation have several implications for research and practice. In addition, our findings are very encouraging for initiatives developing FLOSS. In light of prior research which points to learning barriers in FLOSS projects and regards them as existential threats for the existence of the initiatives, we show that mentoring is an appropriate strategy for FLOSS projects to educate their contributor base and actively increase their project permanence. Although the KDE project consists of a rich variety of subprojects and developers our results can only suggest that mentoring is also an appropriate measure for other FLOSS projects. Future research is, therefore, necessary to examine the effects other projects are experiencing with the use of mentoring. Moreover, the application of alternative training methods (such as traditional teaching relationships) in the FLOSS context and their effects on newcomers’ knowledge building and continued commitment is necessary to evaluate and
compare the effectiveness and efficiency of mentoring in the FLOSS context.

The results of the conducted Mann-Whitney T-test support the use of mentoring in FLOSS projects as an effective education strategy. New members to the KDE project who were mentored achieved significant higher learning states than non-mentored newcomers after their first seven month in the project. This observation is consistent with organizational literature in which mentoring is especially valued for the education of new employees. While the advancing effects on novices’ knowledge building are similar to organizations’ experiences, subsequent research is necessary to build a comprehensive understanding for the observed knowledge gains. Besides extending the number of the analysed developers and broadening the project focus, future research on the effects of mentoring may also account for newcomers’ education.

The strong and significant association between novices’ learning state and their on-going participation has implications for both FLOSS research and practice. While previous research by (Singh et al., 2010) focused exclusively on the effects on contributors’ commitment intensity, no prior study - we are aware of - has linked FLOSS developers’ project permanence to their advances in knowledge building. Our study suggests that project members’ on-going commitment is strongly affected by the acquired amount of project knowledge after their ramp up period. In the case of KDE, the effects are in particular eminent. Newcomers who achieved a higher knowledge state after their ramp up period are much less likely to leave the project. This finding not only helps to understand FLOSS developers’ retention behaviour in more detail but also invites FLOSS initiatives to develop and launch training programs in order to retain their newcomers.

In addition to protégés’ learning advancements, our study in the context of KDE shows that there is also a strong and significant direct association between newcomers’ mentorship and their project retention. While controlling for knowledge gains, mentored newcomers are still nearly 52.5 per cent (1-exp(B)) more likely to remain in the FLOSS project than non-mentored novices. This finding is in support to the effects reported in the organizational domain. Probably, the strong relational bond which is formed between mentors and their protégés facilitates not only the transfer of knowledge but also stimulates them to continue in the project. While we assume that this interpersonal relationship between protégés and their mentors is also the main reason for their enlarged project permanence, future studies examining this aspect are required. Although previous researchers associate serious validity threats with the extraction of perceptual data for mentoring studies, it seems necessary for gaining further insights into the reasons why mentored FLOSS contributors remain longer in their projects than non-mentored developers.

\[
\begin{array}{cccc}
\text{Model 1} & \text{Model 2} & \text{Model 3} \\
\hline
\text{Project Size} & 0.000 & 0.000 & 0.864 \\
\text{Project Age} & 0.007 & 0.003 & 0.025 \\
\text{No. of developer} & -0.013 & 0.007 & 0.075 \\
\text{Mentored} & -0.829 & 0.243 & 0.001 \\
\text{Knowledge level} & -0.539 & 0.147 & 0.000 \\
\hline
\chi^2 & 7.261 & 19.677 & 32.294
\end{array}
\]

\textit{Table 3: The results of the Cox regression models}
4.2 Limitations

While our findings have relevant theoretical and managerial implications, we would like to outline the limitations of our evaluation. The most important constraint of our analysis was the exclusive consideration of newcomers to the KDE project. Although the KDE project consists of a rich diversity of subprojects and attracts developers with various motivations and backgrounds, future research with a broader spectrum of independent FLOSS projects and a richer set of experienced and inexperienced newcomers is necessary to analyse the representativeness of our findings. In addition, the extrinsic incentives (such as the three month stipend or the prestige associated with Google) which participants of the GSoC event are exposed to might influence them in ways which cannot be reproduced by non-sponsored mentoring initiatives in FLOSS projects. Furthermore, our analysis only examined the retention behaviour of newcomers to their first KDE project; it is possible that the analysed developers show different retention behaviour at other projects within or outside KDE’s development.

4.3 Conclusion

The use of FLOSS is becoming ubiquitous for private households and organizations worldwide. Due to lacking sustained contributors, however, the huge majority of initiatives developing FLOSS fail. Existing research results indicate that the level of project knowledge novices acquired in their beginning is most relevant for their on-going commitment. However, prior studies show also, that most developers are very unlikely to ever achieve a higher learning state on their own. Driven by compatible experiences in the organizational domain, we examined the use of mentoring as a potential knowledge transfer and retention strategy for FLOSS projects. Consistent with organizational literature we argue that mentoring positively affects newcomers’ retention behaviour directly through the relational bond which is created between protégés and their mentors. Moreover, we hypothesize that mentoring relationships facilitate newcomers’ project permanence indirectly by advancing their knowledge level. An evaluation of 91 mentored and non-mentored newcomers to the KDE project gives support to our hypotheses. The results suggest that mentoring is an effective measure for FLOSS projects to retain newcomers by advancing their knowledge. Furthermore we find a strong direct association between mentoring and newcomers’ retention behaviour.

References

Chen, J., McQueen, R.J., 2010. Knowledge transfer processes for different experience levels of knowledge recipients at an offshore technical support center. Information Technology & People 23 (1), 54–79.


