AN ABSTRACT OF THE DISSERTATION OF

Becka S. Morgan for the degree of Doctor of Philosophy in Science Education presented on November 20, 2012.

Title: How does a Collaborative Community Affect Diverse Students' Engagement with an Open Source Software Project: A Pedagogical Paradigm

Abstract approved: ____

Lawrence B. Flick

Open Source Software (OSS) communities are homogenous and their lack of diversity is of concern to many within this field. This problem is becoming more pronounced as it is the practice of many technology companies to use OSS participation as a factor in the hiring process, disadvantaging those who are not a part of this community. We should expect that any field would have a population that reflects the general population given no constraints. The constraints within OSS are documented as being a hostile environment for women and minorities to participate in. Additionally OSS communities rely predominately on volunteers to create and maintain source code, documentation, and user interface as well as the organizational structure of the project. The volunteer nature of OSS projects creates a need for an ongoing pool of participants.

This research addresses the lack of diversity along with the continual need for new members by developing a pedagogical paradigm that uses a collaborative environment to promote participation in an OSS project by diverse

students. This collaborative environment used a Communities of Practice (CoP) framework to design the course, the indicators of which were used to operationalize the collaboration. The outcomes of this course not only benefit the students by providing them with skills necessary to continue participation and experience for getting a job, but also provide a diverse pool of volunteers for the OSS community. This diverse pool shows promise of creating a more diverse culture within OSS.

In the development of this pedagogical paradigm this research looked primarily at student's perception of the importance of their group members and mentors provided to guide their participation in and contribution to an OSS community. These elements were used to facilitate the formation of a CoP. Self-efficacy was also used as a measure; an increase in self-efficacy is associated with the successful formation of a CoP. Finally the intent to continue, as reported by students, was measured to determine the potential contribution to the OSS community overall.

This research was designed to use collaboration to support the formation of a CoP within the groups formed between students based on common interests in the OSS project. Additionally students were provided with a mentor from the community to assist in finding paths to contribute. The Ubuntu project was chosen for its commitment to diversity and its reputation for being a welcoming environment to newcomers, reducing the risk of negative community interactions for students. Written reflections were gathered at mid and end of term and used in conjunction with transcripts or reports of group meetings as well as emails between mentors and mentees. Additionally self-efficacy was measured at the beginning and end of the term.

The results of this study show that this pedagogical paradigm supports student contribution. Contribution levels were found to be associated with the level of the formation of a CoP within each group and the use of mentors, as well as attending a live, hands-on bug triage demo and the Global Jam, to gather resources. It was also evident that students intend to continue participating at a rate higher than the average rate for newcomers trying to contribute without the type of support offered by this class. Further research into the examination of the use of reflective dialogue with mentors is recommended. It is also recommended that the results from the operationalization of the indicators of the formation of a CoP be used to assist in a more consistent formation of this important resource across more groups within the class.

The results of this research point to the effectiveness of this paradigm to promote contributions to an OSS community. These contributions provide the skills students need to improve their attractiveness to future employers. This class also produced a number of students who intend to continue participating in OSS, providing a diverse pool of potential volunteers to the OSS community.

How does a Collaborative Community Affect Diverse Students' Engagement with an Open Source Software Project: A Pedagogical Paradigm

by

Becka S. Morgan

A DISSERTATION

submitted to

Oregon State University

in partial fulfillment of

the requirements for the

degree of

Doctor of Philosophy

Presented November 20, 2012

Commencement June 2013

Doctor of Philosophy dissertation of Becka S. Morgan presented on November 20, 2012

APPROVED:

Major Professor, representing Science Education

Dean of the College of Education

Dean of the Graduate School

I understand that my dissertation will become a part of the permanent collection of Oregon State University libraries. My signature below authorizes release of my dissertation to any reader upon request.

Becka S. Morgan, Author

ACKNOWLEDGEMENTS

I owe a debt of gratitude to my teachers, students, mentors, peers, friends and family who all supported me in my journey to this culminating achievement. First to my mentor Dr. Carlos Jensen, this would not have been possible without you. Your tireless work with me over the last three years helped me focus my research and design and carry out my study. I could not have made it to this point without your invaluable input. And to Dr. Larry Flick for your assistance in the writing of my dissertation I am grateful. Your suggestions and feedback brought me to a deeper understanding of my work; I thank you for this great insight. To Dr. Maggie Niess I am forever grateful for your advocacy on my behalf. Without you I would literally not be here. I also thank you for listening to me when I struggled and the spot on advice you offered. Dr. Mirabelle Fernandes-Paul, you always provided gentle support in the midst of the tumult that was often a daily part of grad school. I am grateful for those reprieves.

To my peers and colleagues, Teresa Wolfe, Laura Good, Kathryn Stofer, Matt Campbell, Mike Furuto, Tobias Irish, Chi-Chang “Michael” Lui, Jennifer Wyld and Scott Pattison I am forever grateful for the support and laughter while we met the challenges that were graduate level courses. We learned to unpack the knowledge and provided scaffolding for each other. You all made this process easier with your invaluable input, but more importantly all of you made it amusing. The laughter we shared makes each of you a smile that will stay with me always.

To my students who participated in the OSS course that was the basis for this research, I cannot express how grateful I am to you all. You stayed engaged with the project and worked tirelessly on making contributions. Many of you went above and beyond in your free time to make this research work. I owe a special thanks to Zach Knowles for spending his free time helping me test the teaching techniques for the introduction of the tools necessary for contributing to Ubuntu. I also wish to thank Chris Weber for his help in setting up the flash drive environment that held the Ubuntu O/S before the course started.

Finally I wish to thank my friends and family for their never ending love and support during the time I have been working on all three of my degrees. Gail Dossett, Debbie Williams, Mary Gordon, Kelly Anne Bowler and Barbara Curtis I love you all, and I know you know this, but your support has been the rock that has provided the foundation for my daughters and I to live this most amazing life. You are my family of choice. Most importantly I wish to thank my daughters, Michaelyn Adrian Kairi Morgan and Nikita Kassondra Morgan. I am grateful every minute of every day that you are both a part of my life. This journey would not only have been more difficult without the two of you, it would have been far less fun. I thank you for your patience, for the food you put in front of me so I would not forget to eat, and, most importantly, the laughter until our sides hurt

and our eyes leaked. There are no words so I will simply say to you both – you are my I love you ♥

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

TABLE OF CONTENTS (Continued)

TABLE OF CONTENTS (Continued)

LIST OF FIGURES

LIST OF TABLES

How does a Collaborative Community Affect Diverse Students' Engagement with an Open Source Software Project: A Pedagogical Paradigm

Chapter 1 : Introduction

Background of the Problem

Open Source Software (OSS) communities are homogenous, lacking gender and racial diversity. OSS projects are predominately volunteer driven communities, a situation that produces a high turnover rate, which in turn creates a necessity for a continual source of contributors. The culture of OSS is considered by many to be unwelcoming to women and minorities, either subtly or blatantly. Practices creating barriers include the use of flaming, a heated/hostile response to someone's contribution or comment within a project, and ad hominem abuse, the practice of attacking someone's character rather than addressing the issue. Additionally, conferences have been sites for inappropriate presentations (Geek Feminism, various), and reports of sexual assaults (Shirley, 2010) and death threats (Sierra, 2007). Although the lack of diversity in OSS is the subject of a great deal of debate, there has been very little research into ways to mitigate these barriers. The research detailed here considered a solution that addresses the high turnover and hostile culture by providing a source of contributors, concentrating on increasing the gender and racial diversity of OSS, using an educational model that promotes a culture of collaboration within the classroom. The model used a design grounded in the learning theory of communities of practice (CoP). The use of this educational model in providing a diverse pool of contributors is supported by the work of Ye and Kishida, who show learning to be a primary motivator of OSS participation (Ye & Kishida, 2003). The work of Ye, et al. uses a communities of practice (CoP) model to support the use of legitimate peripheral participation (LPP) as a way to view the path into a project. The research described in this paper used this paradigm as the basis for the creation of a collegiate curriculum designed to promote participation in an OSS community (Figure 1.1).

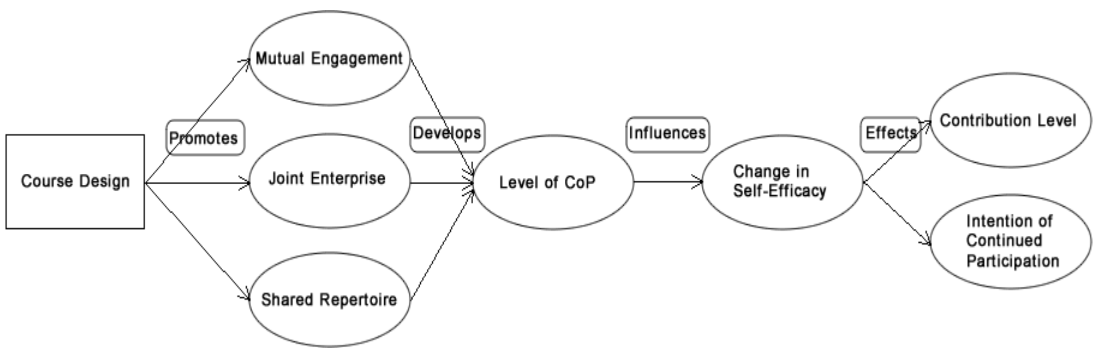

Figure 1.1: Course Design Model

OSS adoption is on the increase. The world is increasingly shaped by technology and the use of OSS has seen tremendous growth over the last years. Deshpande and Riehle found that the doubling time for both total number of lines of code and total number of projects is 14 months (2008). Two separate surveys of both private and public sector organizations found that over half of all respondents reported a complete commitment to using OSS while an additional third are experimenting with the use of OSS (Trapasso & Vujanic, 2010; Wurster, Igou, & Babat, 2011). OSS is being used in critical systems, including on Wall Street (Schmerken, 2011), by the medical community to facilitate compliance with the passage of ARRA (the American Recovery And Reinvestment Act of 2009) (Vaughan-Nichols, 2009), the White House website (Walker, 2002) which is also contributing code to an open government platform in collaboration with India (VanRoekel & Chopra, 2011).

If you could look across the OSS community, visually taking in the landscape of people creating technology in this community, you would see a white, male portrait. This representation has roots in Science, Technology, Engineering and Mathematics (STEM) education and is reflected in the students who pursue these career paths in academia. In Mead and Metraux's 1957 study high school students were asked to write a description of a scientist. From the data gathered a composite emerged. "The scientist is a man who wears a white coat and works in a laboratory. He is elderly or middle aged and wears glasses. He is small, sometimes small and stout, or tall and thin. He may be bald. He may wear a beard, may be unshaven and unkempt. He may be stooped and tired" (Mead & Métraux, 1957, pp. 386 – 387). This image was reflected in the classrooms of the day. In 1950 if you walked into a science class you would be greeted by rows of attentive, serious and intelligent students. The image of these students is universally presented as black horn-rim glasses, pocket protectors, and crisp white lab coats. You would also notice that most, if not all, were white males.

Time passed and these images persisted. In 1983 Chambers created a research tool, Draw-A-Scientist Test (DAST), for determining how students pictured scientists. This test looked for seven characteristics that were indicators of a stereotypical scientist: Lab coat (usually but not necessarily white), eyeglasses, facial growth of hair (including beards, mustaches, or abnormally long sideburns), symbols of research: scientific instruments and laboratory equipment of any kind, symbols of knowledge: principally books and filing cabinets, technology: the "products" of science, and relevant captions: formulae, taxonomic classification, the "eureka"! syndrome, etc. (Chambers, 1983). The use of drawing to show children's understanding is supported by research (White & Gunstone, 1992) and can be said to represent mental models that not only express a person's beliefs, but also are used for their predictive qualities (Norman, 1983). The DAST test initially showed that students overwhelmingly represented scientists as male and that the number of stereotypical images increased with the age of the students. The DAST was used in continuing research and in 1995 (Finson, Beaver, & Cramond, 1995) it was shown that, although students entered science with a stereotypical image of a scientist, there were interventions that altered that perception. Studies put students who participated in a summer program in a treatment group that allowed them to interact with science faculty, work on research with a faculty mentor and participate in field research for one week. The pre-test of the control group, those without this exposure, and the treatment group showed similar stereotypical representations. However, the post-test showed significant differences in the number of stereo typical indicators used by the treatment group. The control group had no significant difference in the pre- and post-tests (Finson et al., 1995).

While the field of science had an image problem, research in mathematics was producing research that claimed men were superior to women at mathematics (Bonser, 1910; Eells & Fox, 1932; Lund, 1932; Stroud & Lindquist, 1942). In 1980 the work of Benbow and Stanley refocused mathematics educators on the concept that men were superior to women at mathematics. In their study of academically advanced 7 – 10th grade students they found that "A large sex difference in mathematical ability in favor of boys was observed in every talent search" (Benbow & Stanley, 1980, p. 1263). They further claimed the students in their study had, up to this point, received identical instruction in mathematics, thus negating early research that hypothesized the difference in ability could be attributed to a difference in the courses taken (Fennema, 1974). In the end Benbow and Stanley conclude that "We favor the hypothesis that sex differences in achievement in and attitude toward mathematics result from superior male mathematical ability" (Benbow & Stanley, 1980, p. 1264). This research not only influenced the teachers who were teaching mathematics through academic journals, but also appeared in popular media like *The New York Times* as well as *Science*, *Time*, and *Newsweek*. The popular media ran articles with the titles "Do Males Have a Math Gene?" (Williams & King, 1980) and “The gender factor in math: A new study says males may be naturally abler than females" (*Newsweek,* 1980). Family weekly reported in its January 1981 article that the Benbow and Stanley research "concludes that boys are born with greater math ability" and *Time* concluded that "males inherently have more mathematical ability than females" (as cited in Jacobs & Eccles, 1985). Some of the articles in popular media also included images that represented males being superior to females in mathematics. These images, along with the information being disseminated, created a mental model of males being mathematically superior. Most children, much like with scientists, draw mathematicians as older, white and male (Picker & Berry, 2000).

The media reports of the Benbow and Stanley study affected the way parents viewed their children’s abilities. Most notably women saw their daughters as having less mathematical ability after being exposed to the media coverage (Jacobs & Eccles, 1985). Jacobs and Eccles research suggested that media confirms pre-existing beliefs therefore women may still be influenced by earlier studies and when these beliefs were confirmed it changed their views of their daughters’ abilities. If true, this evidence would demonstrate the pervasive effect of these beliefs.

Computer science and engineering both have their foundations in science and mathematics. Not surprisingly that the mental model students have of computer scientists and engineers reflects, overwhelmingly, the white male

(Knight & Cunningham, 2004; Martin, 2004). These images persist with a slight variation in computer science where often the males are drawn as unkept, sloppy and unhealthy (eating junk food and drinking soda) (Martin, 2004). In addition to the images students draw, the lack of female images in textbooks for STEM courses perpetuates these images (Brush, 1991).

These images are not just a mental model of something abstract; these stereotypes may contribute to early academic choices that serve to limit career choices among girls. Weber and Mitchell state the following:

> Images are constructed and interpreted in attempts to make sense of human experience and to communicate that sense to others. Images in turn become part of human experience, and are thus subject to reconstructions and reinterpretations. While images always maintain some connection to people, places, things, or events, their generative potential in a sense gives them a life of their own, so that we not only create images, but are also shaped by them (Weber & Mitchell, 1995).

Across the STEM fields a majority of students draw figures in these disciplines as white males, suggesting that women and people of color do not picture themselves as eligible for these career choices. Research into the self-efficacy of the non- white/non-male students in STEM shows that the underrepresented groups who are staying away or drop out do so, in part, because they have low self-efficacy in these STEM areas (Busch, 1995; Margolis & Fisher, 2002; Miura, 1987; Wilson, 2002). The major side effect of this erosion of self-efficacy is the declining participation of women and people of color in computer science (CS) and information technology (IT) fields. The decline in participation is demonstrated by statistics compiled by both the National Center for Education Statistics (NCES) and the National Science Foundation (NSF). In the academic year 1983 - 1984 37.1% of Bachelor's degrees in CS/IT were awarded to women in the U.S. (Snyder, Hoffman, & Geddes, 1997). This number fell to 17.7% by 2007 – 2008 (NSF, 2011). According to NSF (2009), the participation in CS by people of color during 2006 – 2007 is as follows: Asian/Pacific Islander 8.4%, Black 10.8%, Hispanic 7.0%, American Indian / Alaska Native 0.6%, and other unknown race/ethnicity 8.8%.

Through the lens of the statistics in CS education, reflecting numbers were declining and the subject of a great deal of research, the landscape of OSS is even bleaker. Surveys on the OSS community report that women make up approximately 1.5% of the contributors to OSS (David, Waterman, & Arora, 2004; F/LOSS-POLS, 2006; Ghosh, Glott, Krieger, & Robles, 2002; Lakhani, Wolf, Bates, & DiBona, 2002; Robles, Scheider, Tretkowski, & Weber, 2001). It stands to reason that the same issues that plague CS, mathematics, and science education also contribute to the low number of women and minorities in OSS. We should be mindful that all this is coming about in a time when the projected growth in the number of jobs in software development is expected to outpace every other STEM field (Lockard & Wolf, 2012).

Bandura states that self-efficacy is "people's judgments of their capabilities to organize and execute courses of action required to attain designated types of performance" (Bandura, 1986, p. 391) and self-efficacy has a direct influence on "how much effort people will expend and how long they will persist in the face of obstacles and aversive experiences" (Bandura, 1977, p. 194). Research has shown that a lack of self-efficacy in women is correlated with academic and career choices leading to jobs in low income brackets, usually in caretaking roles, and away from often higher paying science and mathematics based careers (Hackett & Betz, 1981).

Bandura outlines four elements that are important to developing higher self-efficacy:

1. Performance accomplishments; the successful completion of tasks engendering the belief that one has the skills needed for the task.
2. Vicarious experience; seeing someone that is the same as you succeed. Live or symbolic modeling.
3. Verbal persuasion; the encouragement of influential others.
4. Physiological states; how a person interprets a physiological response (e.g. anxiety) to a given situation.

Underrepresented groups frequently enter collegiate CS courses with less experience. This lack of experience can be attributed not only to less exposure, but also to curriculum that is insufficient to provide these experiences (Gürer, 2002; Margolis & Fisher, 2002; Margolis, 2008). This lack of previous experience often results in lower self-efficacy due to an absence of a history of mastery of skills pertinent to CS. This lack of minority presence is often coupled with a learning environment dominated by white males, both within the student population as well as instructors. Furthermore, this lack affects underrepresented groups by denying them vicarious experiences and role models. Programs that have at least one female faculty member have shown improved retention of female students (Cohoon, 2001).

A large body of research that supports the use of mentoring to increase the self-efficacy of women (Cohoon, 2001; Gürer & Camp, 2002; Margolis & Fisher, 2002), leveraging both vicarious experiences as well as verbal persuasion. Vicarious experience and verbal persuasion have been shown to be especially important to women and non-whites (Smith, 2001). Underrepresented groups often are not encouraged to engage in mathematics and science related fields, by both their teachers and their parents, and this verbal persuasion has been shown to be more valuable long term (Zeldin & Pajares, 2000). Women come to college with a higher expectation of creating personal bonds with faculty and “failure to establish a personal relationship with faculty represents a major loss to women” (Seymour & Hewitt, 1997, p. 267).

One effect of lowered self-efficacy is increased anxiety, which creates a negative physiological state. High anxiety levels can also be exacerbated by the attitudes of the majority students. Margolis and Fisher found that women interviewed reported male students make fun of female students with less experience when they are not familiar with jargon or concepts, suggesting to them that these should be well known to anyone who is a CS major (Margolis & Fisher, 2002). Self-efficacy is based on self-judgment and often does not reflect actual skills or abilities. Both men and women have misperceptions about the academic competence of both sexes, often overestimating the grade point average (G.P.A.) of males and underestimating female G.P.A.s. Although this inaccurate perception has been seen across many disciplines, it is most pronounced in male dominated majors, like computer science. Both men and women incorrectly believe that men have a higher G.P.A. within this major, a belief that can cause women to be more discouraged in male dominated majors, often causing them to change majors

(Beyer, Rynes, Perrault, Hay, & Haller, 2003; Beyer, 1999). The beliefs fostered by the research showing that men were superior to women in mathematics related tasks (Benbow & Stanley, 1980) have become ingrained in our culture, this example demonstrates their pervasive effects, even though they have been shown to be incorrect (J. E. Jacobs, 2005).

In addition to this misperception of academic competence, women also see men as better suited to computer science because they perceive that the male driven, myopic, and singular focus on computers is necessary to be a computer scientist, a focus many women do not share (Margolis & Fisher, 2002). It is well known that the image of computer science is highly competitive and solitary in nature. This image is confirmed in collegiate courses where labs are to be completed individually and working together is considered cheating. This solitary work coupled with the male dominated hacker culture, a culture where males focus myopically on computers, talking of nothing else, spending every waking moment in front of the screen, working, eating, even falling asleep there, is especially detrimental to the attraction and retention of underrepresented groups (Margolis & Fisher, 1997, 2002). It has been shown that within this highly competitive, solitary environment students do not form study groups in order to collaborate and support each other, which leads to dropping out of the major (Seymour & Hewitt, 1997).

There is a growing body of research that shows that a collaborative environment is attractive to a wider number of potential CS majors and is especially important to underrepresented groups. The paradigm of using pairs programming in introductory programming labs has been shown to improve the confidence and enjoyment of students as well as increase the number of students who continue in the major and pass future classes on the first try (McDowell, Werner, Bullock, & Fernald, 2006; Nagappan et al., 2003; Simon & Hanks, 2008). Pairs programming is adopted from the industry model of Agile Development. Students using this model are assigned to pairs and complete labs together trading off who writes code and who helps guide and correct what is happening at timed intervals. Labs using this model were found to be more interactive, showing a lower level of student frustration and encouraging more problem solving among students rather than relying on the lab instructors to help solve problems (Nagappan et al., 2003).

The success of pairs programming in CS education points to the importance of collaboration and community in CS classrooms. As learning new programming skills and expanding existing skills has been shown to be a primary motivator for joining an OSS project (Ye & Kishida, 2003) it stands to reason that a collaborative class using a CoP model would promote participation in an OSS project. Research shows that a collaborative classroom fosters a sense of community which in turn factors into higher self-efficacy for both genders (Margolis & Fisher, 2002; Townsend, Menzel, & Siek, 2007). Studying the model of the Women@SCS (School of Computer Science) that emerged from the longitudinal study at Carnegie Mellon (Frieze & Blum, 2002) the following relationships between self-efficacy and CoP were evident:

- Mutual Engagement – In the case of Women @CSC mutual negotiation was focused on creating a culture in CS that was inviting to women. Women created a space that served to provide a place for follow students to come and get advice, participate in professional conferences and events, and find mentorship. Verbal persuasion can be seen as an element of these relationships as negotiated in the practice, and vicarious experience can be derived from these relationships.

- Joint Enterprise – The sense of ownership and accountability derived from the negotiation and creation of an enterprise through engagement provides a space that can support increased self-efficacy through positive physiological states. The Women@SCS was specifically formed to provide this space for women in the CS program at CMU.
- Shared Repertoire – as members become proficient with the tools, adopt the language, and produce artifacts, the sense of accomplishment will increase self-efficacy. This in turn may help other minorities gain through vicarious experience.

Statement of the Problem

There is a lack of participation in CS by women and minorities as detailed in the previous section. It is also apparent by the previous statistics that there is even less participation by these groups in OSS. We should expect that any field would have a population that reflects the general population given no constraints. The projection for new jobs in computer and mathematics related fields is projected to be the sixth fastest growing field, with the projected creation of 778,300 jobs from 2010 to 2020 (Lockard & Wolf, 2012) At the same time the number of bachelor degrees being awarded in computer science was 54,763 in 2008, the most recent statistics available (NSF, 2011). If underrepresented groups continue to face barriers in the discipline of computer science we will fall short of filling the jobs that will be created as well as missing out on the skills and talents represented by these groups. The research outlined above gives us a partial view of constraints within CS in the solitary, independent, and competitive paradigm of learning that serves to exclude these underrepresented groups. This research also offers a way to mitigate this constraint with the use of collaborative environments for learning.

Diversity could, and should, be promoted under a social justice rationale, that this area of work and study should be equally accessible to all members of a society in order to not continue the reinforcement of unjust patterns of inclusion. However, it can also be argued that the continued exclusion of these groups denies the resulting industry access to a wide range of talents, as well as excluding the objectivity that is derived from the diverse life experiences that can be found within these excluded groups (Intemann, 2009). Research has shown that diverse groups have higher productivity (Bellini, Ottaviano, Pinelli, & Prarolo, 2008), are more creative (Madjar, 2005), solve problems more efficiently (Hong, Page, & Institute, 2001) and even improve the return on equity and total return to shareholders within a business model (Catalyst, 2004).

There is an absence of diversity within OSS and if these underrepresented groups are not part of the process they will fail to impact this increasingly relevant segment of society. OSS also misses out on the insights and creativity that diversity brings to the table. The homogeneity of the current technology industry may miss the needs of the underrepresented groups in developing new technological directions (Gürer & Camp, 2002) as well as risking future end user dissatisfaction. Research has shown that software that is assumed to be gender neutral in reality disproportionately affects female usage negatively causing females to adopt new features less often than males (Burnett et al., 2008) and further research found that a redesign of these features lowered the barriers for females without adversely affecting males (Grigoreanu et al., 2008).

Purpose of the Study

The purpose of the study was to explore using mentorship and collaborative group work, used to encourage a CoP through mutual engagement, joint enterprise and shared repertoire, as a bridge into an OSS community. I looked at the levels of engagement with group members and mentors as well as contributions to the community and the intent to continue working in OSS. This research focused primarily on the group and mentorship relationships and their effect on students. The works of Margolis and Fisher (2002) and Seymour and Hewitt (1997) demonstrate the importance of collaborative environments in the retention of women as well as racially diverse groups in computer science as well as science in general. These works were used as the basis of a pedagogical model, using CoP, to design a collegiate course, teaching students how to participate in an OSS project. This course was used to study the effects of this model on student participation, engagement, and contribution to an OSS project.

To help explore the effectiveness of the model measures of self-efficacy were taken. Bandura asserts that “people's level of motivation, affective states, and actions are based more on what they believe than on what is objectively true” (Bandura, 1997). Research has determined that low self-efficacy is a barrier to women and minorities in computer science (Scragg & Smith, 1998; Wilson, 2002) so the use of this measure is appropriate to help determine the effectiveness of this pedagogy.

Research Questions

This research is designed to answer the following questions:

1. How do students view the importance of group members and mentors and their effectiveness in promoting engagement in OSS?
2. Will the course design produce students who engage with the OSS community at a percentage greater than 10%?
3. Is there a change in self-efficacy from the beginning to end of the course?
4. Do students anticipate continued participation in OSS projects?

The insights from this research will be used to develop an academic curriculum to teach OSS development to include the participation of a more diverse population.

Importance of the Study

Statistics gathered by the US Department of Labor show that the technology field is growing rapidly with the projection of job openings exceeding the expected number of graduates in tech related majors (See Figure 1.2). Creating curriculum that is engaging and provides real world opportunities is not only vital to increasing the overall number of students in CS, but it is also necessary to increase the number of women and minorities in order to provide a diverse work force.

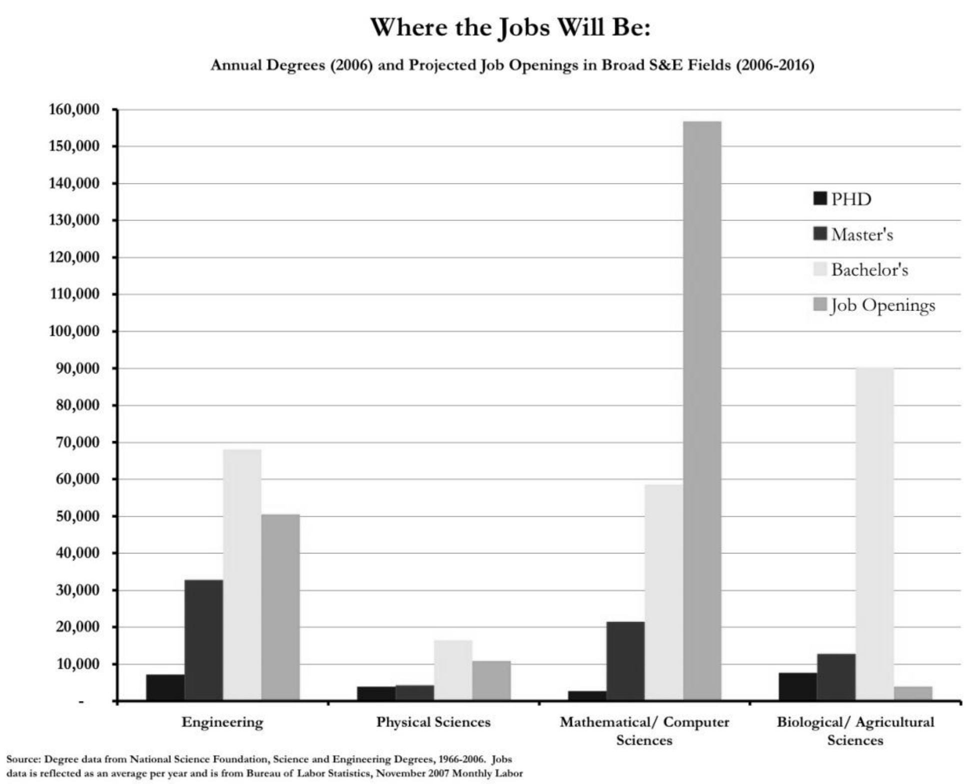

Figure 1.2: Degrees Awarded Compared to Future Demand

The lack of diversity in OSS development has begun to garner attention due to the exclusive culture of development and the use of this experience in hiring practices. There are communities involved in outreach programs, Ubuntu, Debian, Dreamwidth, and Python to name a few, who seek not only to attract diverse groups to their projects but to also ensure they are successful and stick around. It is problematic at best to retain these diverse participants when there are periodic and very public reports of sexist and abusive behavior in OSS communities, including pornography in presentations and aggressive flaming and ad hominem abuse on public mailing list and IRC channels (Geek Feminism, various).

Although sexist behavior is generally undesirable and needs to be addressed, there is but a small body of literature addressing the difficulties in recruiting women to OSS. Strategies that have proven successful within individual projects are; creating a welcoming environment, specifically being inclusive to all people (“Dreamwidth Diversity Statement,” n.d.), having clear documentation that shows how to set up a development environment as well as how to use the tools used by the project (Rustad, 2011), valuing non-coding contributions as much as coding contributions (Townsend et al., 2007), and providing mentors to ease the transition into the project (Schindler, 2009). How well these techniques translate across projects is unclear.

The lack of diversity in OSS is more than a mere concern about ensuring that there are enough developers to meet the needs of projects. For whatever reasons, underrepresented groups are being deprived of access to OSS communities and the skills derived from participating in them. Conversely a lack of diversity denies OSS communities access to diverse skill sets, experiences, and innovation. This has a tangible effect both for the individuals concerned as well as the OSS communities and the software they produce. This problem is becoming more critical as software companies begin to look for OSS experience in their hiring. According to David Heinemeyer Hansson “Open source is a golden gift to hiring process of technical people. It reduces the risk enormously by allowing you to sample candidates over a much longer period of time” (2005). Since women and minorities are vastly underrepresented in OSS this puts them at a distinct disadvantage in the job market.

As open source projects become increasingly important showcases of prospective employee skills, it becomes increasingly important to ensure all have the opportunity to make contributions, and to develop curriculum to teach students how to participate in OSS. This should leverage the research on encouraging women and underrepresented groups in CS to counter existing inequalities. The main focus of this course is on the inclusion of pedagogy shown to improve self-efficacy, as others have found a positive correlation between self-efficacy and retention (Fisher, Margolis, & Miller, 1997; Gürer & Camp, 2002; Margolis & Fisher, 2002; Roberts, Kassianidou, & Irani, 2002; Seymour & Hewitt, 1997). Although this research focuses primarily on women, treatments that increased self-efficacy in women were found to also increase self-efficacy in other minorities as well as the majority. The effects were, however, found to be more significant for minority groups (Smith, 2001).

Intemann (2009) proposed three rationales for diversity in research and it stands to reason that these rationales hold as support for diversity within technological innovation:

1. The Social Justice Rationale – Creating a bridge into open source development allows the underrepresented groups an opportunity to have equal access to resources, in this case job experience and skills that promote the attainment of lucrative careers. These resources are currently unjustly distributed to a privileged group, predominately white males, through the maintenance of a hostile environment that serves as a barrier to entry to those who lack this privilege.
2. The Talented Workforce Rationale – The groups who are not participating in OSS because of the barriers that have been erected represent more than half of our population. This erroneous exclusion may prevent some of the brightest minds from participating and in doing so robbing the field of much needed talent.
3. The Increased Objectivity Rationale – Finally it should be considered that the barriers that operate to exclude these groups also serves to deny the

technology field the innovation that could come from the diverse ideas, skills, and experiences.

These rationales can be applied to demonstrate how the lack of diversity in CS is detrimental to innovation. Margolis and Fisher (2001) argue the consequences of women not participating in the creation of information technologies with the following example:

> In the long run the greatest impact may be on the health of computing as a discipline and its influence on society. The near absence of women's voices at the drawing board has pervasive effects. Workplace systems are built around male cultural models, and entertainment software fulfills primarily male desires. In a particularly poignant example, some early voice-recognition systems were calibrated to typical male voices. As a result, women's voices were literally unheard. Similarly, some early video conferencing systems, in which the camera automatically focused on the speaker, ignored the participation of women. If women could not be heard, they could not be seen (Margolis & Fisher, 2002, pp. 2 – 3).

This example by Margolis and Fisher is just a glimpse into what happens when diversity is not included in the process of innovation. Lack of diversity in OSS is a problem, not only for the communities that do not benefit from the valuable input from these underrepresented group, but also for the innovation that is lost and for the end-users who are not served by technology.

Scope of the Study

This study focused exclusively on the Ubuntu project. Ubuntu was chosen for its efforts at community outreach focused on attracting diversity. Evidence that Ubuntu is actually working on attracting diversity, not just paying it lip service, can be found in their name, from the African concept of 'humanity towards others', in their diversity statement, detailing their commitment to “being a community that everyone feels good about joining”(“Diversity | Ubuntu,” n.d.), and code of conduct, which encourages all participants to be respectful and considerate (“Code of Conduct | Ubuntu,” n.d.). Additionally Ubuntu already had a mentorship program established providing mentorship within a large number of areas of contribution.

Mentorship within the study provided information on one aspect of developing a community of practice (CoP) model; I also looked for indicators that a CoP had formed within each group. Additionally I used student groups to foster a sense of community within the classroom, as well as teaching the tools of the project, having a member of the Ubuntu community virtually guest lecture, encouraging participation in Ubuntu events, and creating an atmosphere of community participation.

The goal of this study was to assess the effectiveness of this mentorship and community on participation, engagement, contribution, and potential retention.

Definition of Terms

For the purposes of this study the definition of diversity that was used as a framework is “the distribution of personal attributes among interdependent members of a work unit.” (Jackson, Joshi, & Erhardt, 2003, p. 802). Additionally, for OSS, the most widely accepted definition comes from the Open Source Initiative (OSI), which emphasizes that “open source doesn't just mean access to the source code” (“The Open Source Definition | Open Source Initiative,” n.d.). OSS projects must also meet other criteria in order to be considered open source. The software must be licensed in a manner that does not restrict free distribution and derived works must be distributed providing the same, or more liberal, license. Licensing of OSS must also not discriminate against any person, group of people or field of endeavor and must not restrict any software that may distributed along with the OSS (“The Open Source Definition | Open Source Initiative,” n.d.).

I looked for evidence of mutual engagement, joint enterprise, and shared repertoire, using indicators devised by Wenger, as evidence whether a CoP had developed within a group. I considered the CoP to be within a group rather than within the entire class. I also reviewed interactions with mentors and group meeting transcripts or reports to determine the level of engagement. Engagement was seen to be the interaction of group members who were sharing resources. This was seen at the group level, in group meeting forms or meeting transcripts, when one member of the group found an interesting tool, tutorial or received information from an outside source like a mentor or a member of the community on an IRC channel and then brought the information back to the group so that all members could benefit. Engagement was also considered from the community level of Ubuntu and evidenced by contributions by the students. It was often evident that these contributions were the culmination of the collective resources a group had gathered. In addition to the group members working together I also looked for evidence of engagement with their mentor or with other members of the Ubuntu project. Students were considered to be engaged with their mentors when they asked and answered questions and actively sought information on contributing to the project. Additionally students were considered to be engaged when they went to Ubuntu Local meetings in Salem or the Global Jam in order to ask questions that assisted them in contributing.

Delimitations and Limitations

Although I acknowledge the need for a change in the culture of OSS communities, this study limited itself to using mentorship to remove barriers to entry that often prevent engagement in OSS communities. Directly suggesting cultural changes was beyond the scope of this research. The significance of this study was to demonstrate a model for teaching participation in OSS that promoted participation by diverse groups. Developers will come and go in OSS projects creating a need for an incoming pool of OSS developers (Ibanez, 2010).

It stood to reason that a primary source for these developers could be higher education.

Chapter 2 : Review of the Literature

As stated in the introduction, there are five surveys (David et al., 2004; F/LOSS-POLS, 2006; Ghosh et al., 2002; Lakhani et al., 2002; Robles et al., 2001) that report the demographics of the OSS community as a whole. Although these surveys are flawed, they predominately addressed only English speaking projects and they drew from mostly a handful of large projects, they provide the best source of statistical information about the OSS community that exists. From these sources comes the broadest view we have of the community. The five surveys find that half of all OSS developers are from Europe and between 30% and 47% from the Americas. This leaves the rest of the world, and especially Asia with little representation (between 5% and 17%). By contrast, Europe accounts for 10.7% of the world's population, the Americas 12.9%, Africa 14.8%, Oceania 0.5%, and 60% from Asia. A consistent finding across surveys is low participation of women. This ranges from 1.1% to 2% and has remained consistent. While the categories and numbers differ slightly between surveys, the community appears to be divisible into thirds: A third of developers were pursuing or had completed graduate degrees, a third bachelor's degrees, and a third vocational or high school diplomas. This is different from the educational distribution in the US, where only 10.6% of the population has gone beyond a bachelor's degree, 30% have a bachelor's degree, and 59.4% have the equivalent of an associate's degree or less. These statistics paint a picture of little gender diversity and, based on geographic location of the developers, it appears as though there is little ethnic diversity.

The next logical question would be, does diversity matter? In reviewing the literature the importance of diversity is supported in many areas. Multicultural environments are positively correlated with productivity (Bellini et al., 2008; Ottaviano & Peri, 2006). Not only are diverse groups found to be more productive, Hong and Page (2001) found that a randomly selected group of “cognitively diverse” problem solvers not only located optimal solutions to difficult problem, but they collectively outperformed individual high ability problem solvers. Diversity within groups also promotes creativity by presenting new ideas and differing points of view (Madjar, 2005). Having a gender balanced upper management is also good for the bottom line. Companies with the highest percentage of women in senior management positions demonstrated a 35% higher Return on Equity and a 34% higher Total Return to Shareholders (Catalyst, 2004). The integration of diverse groups has also been shown to promote integration into new markets, but perhaps even more importantly, diverse work groups do this by rethinking and reconfiguring the project itself (Ely & Thomas, 2001). Although there is a great deal of research that suggests diversity is beneficial, it is not without cost. Research has shown that diversity within work groups can negatively affect communication, especially for non-collocated groups (Olson & Olson, 2000) and heterogeneous groups display an inability to agree on what to produce as well as policy making (Alesina, Baqir, &

Easterly, 1999). Additionally, the increasing the number of minority participants may create a backlash from the majority who feel threatened, resulting in sexual harassment, wage inequity and limited opportunities for promotion (Blalock, 1957; Kanter, 1977). These costs of diversity have, however, been mitigated by shared goals and common values (Chatman, Polzer, Barsade, & Neale, 1998; Jehn, 1997), training in tolerance (Kochan et al., 2003) and diversity being valued within the organization (Ely & Thomas, 2001).

Having examined the effects of diversity, providing positive outcomes for innovation, creativity, problem solving and profits when accompanied by shared goals and values, I use this lens to consider individual communities within OSS. There are examples of communities that have created visibly inclusive environments. Many communities have diversity statements outlining the intent to be inclusive regardless of race, ethnicity, sex, gender, or orientation. Many communities that encourage diversity in this manner have a greater participation of diverse groups. Ubuntu recently celebrated a gender milestone reporting 5% participation of women within the community (Bell, 2011), and Perl reports 4% community involvement. However, Dreamwidth reports an astounding 70% participation by women. Dreamwidth is considered an anomaly within OSS. I had the opportunity to hear a presentation on the story behind the success (Paolucci & Smith, 2010). Paolucci and Smith discussed the creation of Dreamwidth with diversity in mind, specifically using a programming language that was easy to learn, creating a welcoming environment that included a diversity statement (that is now widely adopted by other projects), and creating a clear, assisted point of entry into the project. I acknowledge that it is easier to create this environment from the outset rather than alter an existing project; however the assistance to newcomers is a salient point for established communities.

Projects do exist that promote diversity to a degree and, as pointed out with Dreamwidth, providing an assisted entry point lowers the barrier to entry. There are examples of mentorship projects within OSS, Google Summer of Code, Ruby Summer of Code, Debian Summer of Code, etc., However there is no data that has been reported on the success or failure of these programs. However anecdotal evidence suggests that these are popular programs and provide mentorship that lowers the barriers to entry into participating OSS projects. Research across projects is minimal; however there is a great deal of research within computer science education addressing recruitment of women that is strongly related to this issue. CS and OSS belong to the same constellation of CoPs allowing the use of the research on CoPs in CS to support their use in teaching OSS to promote diverse participation. One of the most widely recognized studies within CS on increasing the participation of women is a longitudinal study done by Margolis and Fisher (2002) at Carnegie Mellon University (CMU). The CMU research showed that girls need to be recognized and encouraged to participate in CS and that mentorship, both by peers and instructors, is especially important to women. Women come to college with a higher expectation of creating personal bonds with faculty and “failure to establish a personal relationship with faculty represents a major loss to women, and indeed, to all students whose high school teachers gave them considerable personal attention and who fostered their potential" (Seymour & Hewitt, 1997, p. 267). Additionally Seymour and Hewitt found that students of color are especially isolated, lacking peers, faculty role models, and mentors (1997, p. 320).

Mentorship within the framework of CoP provides newcomers with legitimate peripheral participation that allows them to begin to assimilate into the community (Lave & Wenger, 1991). By performing easier tasks, in the case of OSS this would be easy bug fixes and documentation, newcomers can observe the norms of the group. Mentors provide a filter to protect the newbies from becoming overwhelmed with the size of the project, and provide encouragement to promote continued participation. This relationship with a mentor is especially important to underrepresented groups (Seymour & Hewitt, 1997).

Self-Efficacy

The literature has shown that the gender gap in CS may be attributed to low self-efficacy in women, it has been reported time and again that girls have the belief that they do not perform as well as boys in CS (Gürer & Camp, 2002; Margolis & Fisher, 2002; Wilson, 2002). The fact that girls have skills equivalent to boys, yet feel inferior, and they have a higher attrition rate (Cohoon, 2001) supports the theory that self-efficacy plays a role in the high attrition rate of women in CS and information technology (IT). It follows that women placed in an

OSS environment face the same obstacles and thus have the same issues with low self-efficacy.

According to Bandura, self-efficacy has a direct influence on “how much effort people will expend and how long they will persist in the face of obstacles and aversive experiences” (Bandura, 1977). The following will present a detailed look at Bandura’s theory and how it relates to CS and OSS.

Bandura defines self-efficacy as “people’s judgments of their capabilities to organize and execute courses of action required to attain designated types of performance” (Bandura, 1986). He also asserts that “people's level of motivation, affective states, and actions are based more on what they believe than on what is objectively true” (Bandura, 1997). When students in grades 7 - 12 were given the Raven Progressive Matrices, an ability test that has been shown to be free of verbal and cultural bias, there was no difference in ability or performance found between boys and girls computing skills (Anderson, Klassen, Krohn, & Smith-Cunnien, 1982; Mandinach & Fisher, 1985). However, it has been reported time and again that girls have the belief that they do not perform as well as boys in CS (Gürer & Camp, 2002; Margolis & Fisher, 2002; Wilson, 2002). Bandura’s theory addresses self-efficacy with “expectations of personal efficacy are based on four major sources of information: performance accomplishments, vicarious experience, verbal persuasion, and physiological states.” (Bandura, 1977)

Bandura's first source of information on which self-efficacy is built is performance accomplishments; boys are more likely to tinker and explore computers as a hobby (Burnett et al., 2008; Margolis & Fisher, 2002). This continual tinkering leads to more variety in experiences, and more varied accomplishments.

The next set of experiences on which self-efficacy is based are vicarious experiences, or watching someone similar to oneself succeed, thus fostering the belief that success is possible. There are a large number of role models for men in CS (Wilson, 2002), and often the female role models are largely ignored (Townsend et al., 2007). Since there are a small number of women in OSS it is difficult for women to have participants like themselves taking part in various projects. These factors combine to erode self-efficacy that would come from vicarious experience.

In the area of verbal persuasion, studies have shown that women are not encouraged to pursue mathematics or science related fields, including CS. In OSS the lack of verbal persuasion is taken to a negative extreme with flaming, which is an insulting, or abusive reaction to a forum post or internet relay chat (IRC) communication. This behavior, often aimed at people new to the community, lowers self-efficacy. According to Herring (2004), flaming has more of a negative impact on women then it does on men.

The fourth source of information for self-efficacy is physiological states, or how a person interprets a physiological response to a given situation. In addition to flaming noted above there are also sexist jokes, pornographic presentations at conferences, harassment, and even death threats against women in the open source community (Robert, 2009). This creates a hostile environment for women and has a negative impact on self-efficacy.

Therefore it is a reasonable inference that there is a connection between OSS environments and practices that lead to low self-efficacy in women. Self-efficacy then provides a measure, but it does not provide a methodology for solving the problem. As the development of communities of practice (CoP) has been shown to raise self-efficacy I address this theory as the framework for my research in the next section.

Communities of Practice

There has been research that suggests that a sense of community factors into higher self-efficacy for both genders and that these results are more profound for underrepresented groups (Margolis & Fisher, 2002; Townsend et al., 2007). The following will outline the three characteristics that are crucial to a community of practice model and how that community can potentially raise self-efficacy. Following that I will show, based on the work of Wenger (1998), how both the discipline of CS and OSS can be seen as CoPs, and that these communities are in constellation with each other. As partner communities within the same constellation I can infer that the practices that create legitimate peripheral participation with in CS will also be conducive to legitimate peripheral participation in OSS, allowing for a trajectory into the community of developers in either Closed Source Software (CSS) or OSS.

Wenger defines communities of practice as “groups of people who share a concern or a passion for something they do and learn how to do it better as they interact regularly” (2005, p. 1). This is the foundational idea behind OSS, which operates under the philosophy that people working in groups, sharing code and mutually looking for and fixing bugs, creates an environment where the improvement of code takes place dramatically faster and more efficiently then solitary work (E. Raymond, 1999).

Wenger (1998, p. 72) describes “three dimensions of the relation by which practice is the source of coherence of a community”. These three dimensions are:

- Mutual Engagement – practice exists in the relationships between people, which are developed as they engage in practice, and the actions they take, whose meanings are negotiated with one another.
- Joint Enterprise – the enterprise the community is engaged with is defined by negotiation in mutual engagement. The enterprise is thus defined by the participants as it develops, in response to each action and negotiation.

- Shared Repertoire – as the enterprise is negotiated shared resources are developed. The negotiated enterprise thus produces agreed upon artifacts, and uses similar routines, words, tools, stories, symbols, actions and concepts that have been negotiated over the course of time in the community and have become part of its practice.

I extrapolate from this definition that a community of practice that supports full participation can improve self-efficacy in the following ways:

- Mutual Engagement – mutual negotiation of the community affects physiological states by providing a space for respect of different competencies and experiences. Verbal persuasion can be an element of the relationships that form as negotiated in the practice, and vicarious experience can be derived from these relationships as others' accomplishments support ones' own accomplishments.
- Joint Enterprise – The sense of ownership and accountability derived from the negotiation and creation of an enterprise through engagement provides a space that can support increased self-efficacy through positive physiological states.
- Shared Repertoire – as members become proficient with the tools, adopt the language as a shared medium to communicate and produce artifacts that meet or exceed the expectations of the community the sense of performance accomplishment will increase self-efficacy.

Wenger also provides a list of indicators that a community of practice has formed:

1. sustained mutual relationships – harmonious or conflictual,
2. shared ways of engaging in doing things,
3. the rapid flow of information and propagation of innovation,
4. the absence of introductory preambles, as if conversations and interactions were merely the continuation of an ongoing process,
5. very quick setup of a problem to be discussed,
6. substantial overlap in participants' descriptions of who belongs
7. knowing what others know, what they can do, and how they can contribute to an enterprise,
8. mutually defining identities,
9. the ability to assess the appropriateness of actions or products,
10. specific tools, representations, and other artifacts,
11. local lore, shared stories, inside jokes, knowing laughter, and
12. jargon and shortcuts to communication as well as the ease of producing new ones (Wenger, 1998, pp. 125 – 126).

This list of indicators will provide a concrete way to show if the classroom has formed a CoP. Additionally Raymond (2001) gives a description of computer programmers that provides a link to Wenger's list of indicators showing that a CoP exists among programmers. The "hacker culture" that surrounds programmers has special forms of communication dating from its inception in 1961 at MIT on the ARPAnet to modern day internet communication, however it is still common for this culture to use mediums like internet relay chat (IRC) as communication tools. The ARPAnet provided a medium for real-time communication and "brought hackers together in critical mass" (Raymond, 2001, p. 4). Hackers have their own languages which include, in written form, machine language and assembly language in addition to other programming paradigms including functional, declarative, imperative and object-oriented languages. Hackers also have a spoken language that differed from the norm, developed folklore, and had volumes of inside jokes and stories. A great deal of this aspect of the culture was originally captured in a cross-net collaboration called the Jargon File created in 1973 – 1975. The Jargon File was followed by the publication of *The Hacker's Dictionary* (Steele, 1984), and the revised edition *The New Hacker's Dictionary* (Raymond, 1996). This community of hackers is responsible for much of the innovation and technology that exists today. Many participated for the love of programming, the challenge of solving problems and bragging rights, and there were a few that made their fortunes. It is through Raymond's description that it is obvious that developers form a community of practice.

The OSS community that will be used in this research, Ubuntu, fits the model of a CoP that has been established. The CoP theory outlines a way for newcomers to become involved in the community by first participating in legitimate peripheral participation within an apprenticeship environment (Lave & Wenger, 1991). According to this theory the apprentice learns by participating in easier tasks while being around the old-timers while they work. In addition to observation the newcomer is often associated with a particular "master", or mentor, who shows them the ropes, introduces them to the community, and helps brings them up to speed. It is this mentorship that allows the newcomer to engage in the community, learn the repertoire that is shared, and eventual become a full participating member of the community.

Although CSS and OSS are separate communities Wenger's work provides a way of viewing these communities as related, or belonging to the same constellation, by considering the following criteria:

1. sharing historical roots
2. having related enterprises
3. serving a cause or belonging to an institution
4. facing similar conditions
5. having members in common
6. sharing artifacts
7. having geographical relations of proximity or interaction
8. having overlapping styles of discourses
9. competing for the same resources (Wenger, 1998).

Both CSS and OSS share the same educational foundation, and often have an overlap in culture, however they mainly differ on the ideology of the production of software. This difference in ideals is what separates the two communities. They do, however, the fit the criteria of belonging within the same constellation. As CS education can be seen as the apprenticeship into CSS programming, with professors engaged in the mentor role, it is reasonable to use the literature and research that have increased participation of women and minorities in CS as a logical basis for creating an educational model for providing apprenticeship with mentors into OSS programming.

Chapter 3 : Design and Methods

The purpose of this study was to examine a collaborative pedagogical model supporting contribution to an OSS project. The model was designed to improve student self-efficacy and increase participation with an OSS project among traditionally underrepresented students. Specifically I looked at how using collaboration to foster a CoP would affect the student perception of ease and comfort in joining an OSS project. The CoP was observed from two levels, the first was within assigned groups and the second was the project the students were attempting to engage with, the Ubuntu community.

Design of the study

This study looked at individual students, using questionnaires, writing assignments with open ended questions and observation, to determine if the model assisted in contribution to and engagement with the OSS community of Ubuntu. Questionnaires were used to collect quantitative data including demographics, beginning knowledge and goals. Self-efficacy measures were also used to collect quantitative data at the beginning and end point of the study. Additionally, writing assignments were given at the middle and end points. Teacher observation was also employed to gather qualitative data to provide a deeper understanding of the experiences of these students. Using the combination of self-efficacy measures, writing assignments and observation allowed me to address such questions as: How do students working collaboratively, using a CoP model, perceive their experiences? How does the level of collaboration affect engagement with and contributions to the OSS community? Is there a change in self-efficacy over time in the course? Is there any evidence of a connection between the previous questions and reported intent to continue outcomes?

Research by Ye and Kishida (2003) established OSS projects as meeting the definition of a CoP and provided a model that details the trajectory into a project (see Figure 3.1) by a potential developer. It is not necessary for all participants to reach the inner circle, it may be sufficient for many to stay in one of the outer circles. What is important in this model is to be able to visualize all of the members of the community within the layers of that community and to see where a trajectory can form. Using this model Ye and Kishda (Ye & Kishida, 2003) demonstrated that most OSS participants are motivated by the opportunity to learn and to improve their programming skills. Their study recommended the use of this model incorporating CoP in an education setting to teach participation in OSS.

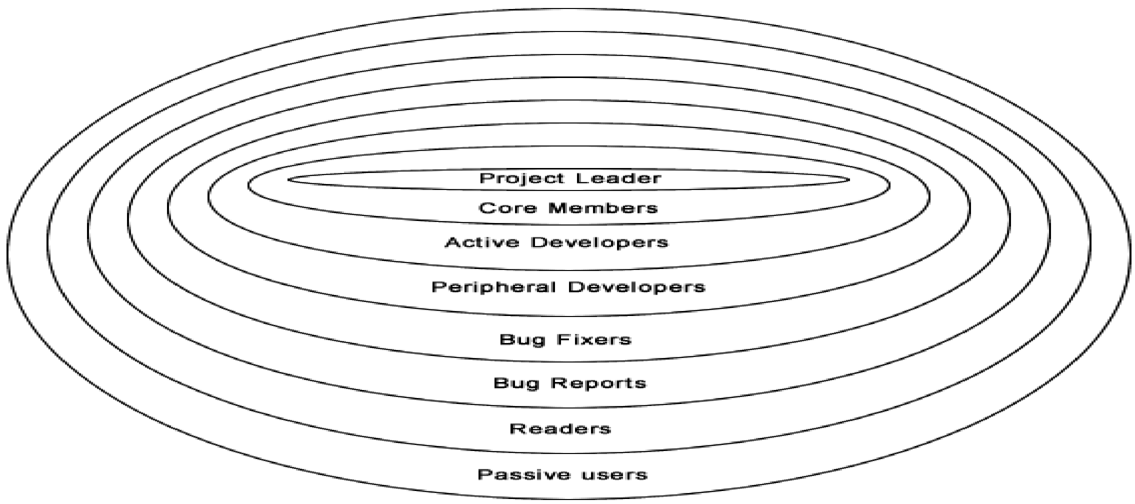

Figure 3.1: General Structure of an OSS Community (Ye & Kishida, 2003)

Using this model as a guide the course design started exposing students to and having them work within the environment of the Ubuntu operating system (O/S), the project chosen for course participation and contribution. The course then proceeded into bug reporting, triage, and replication as well as documentation as areas for student contribution. Mentors from within the Ubuntu project were provided to assist students in figuring out how to start the process and begin contributing. The insights from this research will be used to develop an academic curriculum to teach OSS development to include the participation of a more diverse population.

Choosing the Ubuntu Project

There are many ways to determine an OSS project’s desire to be inclusive. One was is to consider the steps they are taking to involve newcomers of all skill sets. I looked for whether they had a diversity statement, or code of conduct, and if it was inclusive. Another consideration was the size of the project. The project needed to be large enough to ensure students could find work that was of interest to them. This aspect of the course design was important to the engagement of students and also served to demonstrate that all contributions were valuable. Larger projects traditionally have more avenues to contributions and those avenues often include simpler tasks that allow newcomers ways to contribute while still observing the roles and rules of the community they are joining. In developing the course it was important that the project chosen also had mentors or a culture of mentorship. Mentors provide a bridge into a community, helping newcomers overcome the intimidation that is often a barrier to entry (Park, 2008; Rustad, 2011; Schanker & Goodman, 2003).

With my knowledge of the OSS community as a whole I considered the Dreamwidth project first. Dreamwidth is known within the OSS community for its openness and diverse programming population with 75% female developers, far above the 1.5% average for the rest of the OSS community. However, with the large variance from the norm, I considered the Dreamwidth project to be an anomaly and not a good representation of OSS for the purposes of this study.

The project I did choose was the Ubuntu project. Ubuntu demonstrates its commitment to increasing diversity in several ways. The community as a whole has both a diversity statement and a code of conduct. The diversity statement includes the conviction “Although this list cannot be exhaustive, we explicitly honor diversity in age, culture, ethnicity, genotype, gender identity or expression, language, national origin, neurotype, phenotype, political beliefs, profession, race, religion, sexual orientation, socio-economic status, subculture and technical ability (“Diversity | Ubuntu,” n.d.)”. Furthermore the code of conduct outlines the need to be considerate, respectful and collaborative. Providing further evidence of the intent to provide a culture of inclusivity Ubuntu has protocols specified to deal with grievances. These protocols serve to diminish the chance of public fighting, which can lead to flaming and ad hominem abuse. Additional evidence of their inclusivity came when the community celebrated the fact that they had increased their number of female contributors from 2.4% in 2006 to 5% in 2010(Krumbach, 2010). The fact that they track this metric shows the desire to affect change.

Ubuntu is a large project, documenting a little over 700 formal community members in late 2011 (Albisetti, 2011). In addition to the members who have completed the process for formal membership, Ubuntu considers thousands more in their community of users to have an active role in the community. These are the people who provide feedback and report bugs. Using a project of this size provided a large number of mentors as well as many points of entry into the community. Ubuntu considers all contributions to be important, not just programming or development. The list of ways to participate include development (writing packages or fixing bugs), designing the look and feel of the graphical user interface, creating and maintaining local user groups to assist participation locally, triaging and replicating bugs, producing documentation, providing support via mailing lists or IRCs, participating in testing, creating apps and brainstorming new ideas. Having many ways to enter the community gave my students a greater chance to find something within the community that interested them personally.

Instructor Prep for the Course

Prior to the beginning of the course I worked extensively within the Ubuntu environment, using the Ubuntu operating system (O/S), and with members of the Ubuntu community. I started by exploring the Ubuntu site. I bookmarked pages and wiki articles to be assigned as weekly readings to facilitate student’s goal of gathering resources. These readings were used to become familiar with various areas of the site without students becoming too overwhelmed by the sheer amount of information. I followed the documentation for each of the paths into the community mentioned above in order to be able to point my students in the right direction for whatever path they found interesting.

While I was reading documentation I also “lurked” on several IRC channels to get a feel for the communication between members. I was going to have my students on IRC so I wanted to be familiar with it first. Asking a question on an IRC channel is often quite intimidating for newcomers (Betts et al., n.d.) so I also asked questions on the two IRC channels I would be having my students start out in, #ubuntu-beginners and #ubuntu-us-or. The questions I asked were about finding mentors. In both cases I was greeted with enthusiasm and given a great deal of assistance.

I also had the opportunity to attend the Ubuntu Local Jam and Debian Bug Squashing Party in Portland, OR on December 4, 2011. This opportunity gave me the chance to research what tools they use in this situation, to participate in the community personally and to contribute my first patch. I found the people at this event to be welcoming and helpful. I also observed that the community uses the command line interface exclusively for working in the code base. From this experience I added a lab on learning how to use the command line interface to the course schedule.

Ubuntu uses Launchpad to manage the project. Launchpad includes code hosting, bug tracking, translation, feature blueprints and a community-based answer tracker. I created an account to work through the creation of tools students would use during the course. The first task was to create a Secure Shell (SSH) key pair via the command line within the Ubuntu O/S. SSH is a network protocol that creates a secure channel over an insecure network for the transmission of data. Once the key was created I uploaded the public key to my Launchpad account. Next I created a PGP key for encryption and decryption of email messages. This was necessary to sign the Ubuntu Code of Conduct. After I imported my PGP key to Launchpad it was necessary to decrypt an email using this key to validate the key with Launchpad. Using this PGP key I electronically signed the Code of Conduct, then uploaded it to Launchpad. All of the steps taken were achieved from the command line. This preparation formed the foundation of tools my students would use as part of the introduction to the Ubuntu community.

After working through all of the steps detailed above I had a student, Zach, who was not signed up for the class, assist me in validating the process. First I had him create the Launchpad account and the keys as well as sign the Code of Conduct. This helped me assess my instructions to for clarity and ease of use.

Next I had Zach work with the version control system used by Launchpad, Bazaar. I created a simple exercise where I uploaded a document and had Zach use his account to pull the document to his computer, change the document, then push it back to Launchpad using the command line interface. This process pointed out the areas where students may have problems and allowed me to trouble shoot before the actual assignment.

The final step in preparing this course was to create a bootable Ubuntu thumb drive that had persistence for the storage of individual student files. This drive contained the O/S with the pages of weekly reading bookmarked within the Firefox browser. This was used by students who did not want to create a dual boot with their own machine or who ran into difficulties with Oracle VM

VirtualBox. The weekly readings were also available on my faculty website for those using the alternative methods. Students were required to work in the Ubuntu O/S environment for the duration of the class.

Participants and Setting

The students

This course was created to introduce students to OSS programming. It was promoted to students who were finishing their sophomore level programming class (CS262), and data structures (CS260), as well as juniors and seniors who have already completed these courses. The completion of these prerequisites gave them the appropriate background in programming syntax and semantics to participate in an OSS project. I introduced the course during a session of these two courses as well as all upper division courses in the week prior to registration for the term in which I was teaching the OSS course.

During the promotion of the course I emphasized the importance of OSS participation in the job market, specifically that employers are beginning to look for OSS work within the resumes of applicants. I also discussed that OSS experience would provide them with an advantage over other applicants by allowing perspective employers to see their work within a functioning codebase. Lastly I talked to prospective students about OSS as an environment to gain real world skills that would provide them with a deeper understanding of software development within a large codebase rather than traditional small academic projects. Participants were then self-selected based on interest in OSS.

Table 3.1 shows the demographic makeup of the 30 students who participated in this course.

Gender		**Race**		**Age Group (1-4)**				**Length of time programming (in years)**				**Previous participation in OSS**	
Male	Female	White	Non-White	18–25	26–35	36–45	Over 45	0-1	2-4	4-8	Over 8	Yes	No
25	5	24	6	21	7	2	0	5	17	5	3	1	29

Table 3.1 Demographic Makeup of Students

The course

The course itself contained one common lecture session and two separate labs. (For a complete outline of the course and suggestions for replicating this study see Appendix 4)The lecture was held for one hour per week and was designed for students to share experiences as well as to administer surveys and self-efficacy testing. The students registered for both labs and each student was placed into labs alternating by sex to ensure equal distribution of underrepresented groups. The result was that although each student was self-selected based on the choice to register in the course, they were assigned to labs randomly by selecting names from a hat separated by gender. The lab that received assigned mentors first was randomly chosen by picking one group from the hat.

The purpose of the separate labs was to examine how the timing of the introduction of the mentor as a bridge into the community might affect student use of the mentor and the mentor's effect on engagement. This step examined if this variable warranted further study in future work. The data that was collected from the first group might have proven useful to the second group; however I did not look at the data collected from the first group during the course because there was still more to be learned from how the second group used their mentors. The examination of this portion of the course suggested an area of interest for further research.

The design of the course was deliberately created to assist in the formation of a CoP. The initial focus on learning the tools needed to participate in Ubuntu helped students begin to form relationships based on working toward a common goal. Students worked together to learn the different tools and students with previous knowledge shared what they knew in order to facilitate the process. Next the students read documentation to familiarize them with the Ubuntu site and to find their area of interest. The documentation of a project of the magnitude of Ubuntu can be daunting, therefore I gave the students a list of links to areas of interest and places to begin in order to prevent information overload. After reading the documentation on the main areas of contribution to Ubuntu, each student picked an area of interest. I created groups of students who shared an area of interest to work together for the remainder of the term.

The classroom atmosphere and researcher role

The classroom atmosphere was intentionally developed to be collaborative. The traditional paradigm of teacher as giver of knowledge and student as receiver of knowledge was traded for a paradigm of mutual knowledge seeking. Rather than answering questions directly I directed students to sources where they could find answers within the Ubuntu community and encouraged them to work together to find solutions. This design was intentionally created to facilitate contribution to the Ubuntu project using mentors and group members to assist in gathering resources and tools needed to contribute. The initial phase of the course, the first two weeks, introduced students to the Ubuntu O/S. Each student either created a virtual machine to use the Ubuntu or they were given the O/S on a flash drive that was created with Ubuntu installed with persistent memory allowing students to store data within the environment.

In addition to using the O/S students also spent the first two weeks becoming familiar with the tools used by the Ubuntu community. They started by creating an account in Launchpad, a project hosting site used by Ubuntu to host its source code. After creating an account students were required to create a ssh key and a pgp key. Secure Shell (ssh) is a network protocol that allows a network connection between two computers so that data can be exchanged in a secure environment. SSH keys were necessary if students wished to pull, push or create branches in Launchpad. From the command line students created the public and private key, and then transferred the public key to their Launchpad account. The Pretty Good Privacy (pgp) key was necessary to sign the Ubuntu Code of Conduct. After creating the key students used this key to decrypt a message from Launchpad as well as signing the Code of Conduct. All of this work was done from a command line because this is the tool used by Ubuntu developers and so it was considered necessary as part of the tool set.

The next step in learning tools used in OSS was to learn Bazaar, the version control system used by the Ubuntu project. Students participated in an exercise where I had created a project and they needed to pull the code, alter it in a prescribed way, and then push it back into the repository I had created. All of the students worked on this at the same time demonstrating why version control is important. They learned how to create a diff, a file that contains only the changes between the original file and the new file, which allowed them to upload just their changes, keeping the changes of their fellow classmates intact.

After the Launchpad environment was established students were required to learn about Internet Relay Chat (IRC). IRC is the primary form of real-time communication in most OSS projects and Ubuntu is no exception. Students read the Ubuntu documentation for IRC participants and we had a class discussion about the etiquette of using IRC. Ubuntu has an ever growing number of official

IRC channels numbering somewhere around 100, so students were given two channels to watch in the beginning. The first channel was #ubuntu-beginners, a channel specifically designed for beginners in Ubuntu to ask questions. Students were also referred to #ubuntu-us-or, the IRC channel for the local Ubuntu team. I had already participated in both of these channels and had found the participants to be friendly and helpful. Additionally one of the mentors for the course was often on the #ubuntu-us-or channel. In the beginning students lurked on the channels, observing the communication that took place. Lurking behavior is common for many people new to OSS either within an IRC channel or on mailing lists; however it is often the case that people do not progress beyond this stage (Kuechler, Gilbertson, & Jensen, 2012a). Knowing that lurkers often remain silent students were asked to pose a question to one of the two IRC channels during class. The responses to the questions were shared with the class in real time and students worked to determine how to best gather information from this source.

The introduction of tools during the first two weeks was designed to promote a shared repertoire or a shared way of accomplishing a goal. Providing students with access to these tools and practice using them promoted confidence and provided a source of self-efficacy in performance mastery. In addition to creating a pool of shared tools, students also worked together on engaging with the environment and had a joint goal in creating and using the elements of Launchpad. These exercises served to operationalize the formation of Mutual

Engagement, Joint Enterprise, and Shared Repertoire, the dimensions of a CoP as reported in detail in chapter four.

As a researcher I was both a participant in lab discussions as well as an observer of student interaction. I held the role of guiding students to using their mentors and group members as resources. Instead of using the model of individual students approaching me with questions that I answered for them, I would circulate through the lab and answer questions within groups. This process often served to involve other groups in the surrounding area in the discussions that ensued.

In addition to this participation I would also “eavesdrop” on conversations. This process of listening to students talking amongst themselves helped me to gain access into the process students were using to problem solve. This also allowed me to observe which groups were working together and which groups were having difficulties.

Serving in the role of teacher and researcher I purposefully did not look at the data collected until after the course was over. The goal was to avoid altering the mentor experience for the second lab based on the experience of the first lab.

Methodology

My research focused how the students felt about engaging with an OSS community using a course designed to help students overcome barriers, specifically those that significantly impact diverse groups. Because it was my intent to find out how a specific class felt about the process of joining an OSS project the paradigm of qualitative research using the methodology of case study was appropriate. I also chose to augment my qualitative data with quantitative data to aid in the description of what took place.

As a research methodology case study is used to look at a single case in depth. According to Stake a case study is “the study of the particularity and complexity of a single case, coming to understand its activity within important circumstances” (1995, p. xi). Case study used in research has a substantial history and has been used in many disciplines, including law, medicine and social science (Merriam, 1988). I specifically chose an instrumental case study, a delineation from the work of Stake (1995) for studies that are used to pursue an in depth look at a specific case, but the focus of the case study is to be consider an external interest. In my research the external interest can be seen as how the narrative garnered from the case can inform the development of the OSS course.

I chose the class to be the case I studied as it is a bounded system and I looked specifically at the activity patterns within this system. I was looking at how the students interacted with each other as well as with their mentors. It was these interactions that I considered the interesting part of the development of the course I was studying. Case studies have been especially useful for studying and evaluating educational programs (R. E. Stake, 1995). Educational settings and curriculum are in constant evolution as new pedagogic models are developed. The evaluation of these models historically looked specifically at outcomes that were measured quantitatively (R. Stake, 1967). This measure left out important information that could be gathered about the interactions within the class, between the students and with the teacher that could affect the outcomes of the curriculum. In the case of this research the case study approach is appropriate as I looked at interactions between students and with mentors as they operationalization the indicators of the presence or absence of formation of a CoP. It is within the rich descriptions of the interactions that I teased out the importance of the collaborative environment provided by the CoP model.

Data Sources

The methodology for this research used mixed methods to uncover a nuanced view of what was happening within the classroom and how the students perceived the Ubuntu community. Quantitative methods entailed the use of questionnaires and a self-efficacy instrument (see Appendix 2). Qualitative methods incorporated the collection of reflections, artifacts, logs, tests, and observation (see Appendix 3).

Questionnaires were used at the beginning of the course to gather demographic information including gender, age, year in school, experience programming, and experience in OSS. Students who had prior experience with an OSS project were asked what project(s) they had participated in. In addition to the demographic information the questionnaires asked about knowledge of OSS in general and any positive or negative feelings associated with the community as a whole. Students were further asked to provide information on what they expected to learn in the course and what skills they hoped to gain. Questionnaires administered in this fashion helped to develop a sense of the trajectory of each individual student. The questionnaire was reviewed for construct validity by Dr. Carlos Jensen and Leslie Hawthorn. Dr. Jensen is an Associate Professor of Computer Science at Oregon State University and focuses a significant amount of his research on diversity issues and OSS. Leslie Hawthorn has a background as the Outreach Manager at Oregon State University's Open Source Lab and as a Program Manager for Google's Open Source Team, where she managed the Google Summer of Code Program. The backgrounds of both of these individuals make them qualified to give feedback on the construct validity of the questionnaire.

The self-efficacy instrument used was adapted from a programming instrument developed by Ramalingam and Wiedenbeck (1998). The authors of the Computer Programming Self-Efficacy Scale reported an overall alpha reliability of .98 on the instrument for the first administration and .97 for the second administration. Because this study considered the effects of a programming environment on the self-efficacy of student, this is an appropriate instrument to use for data collection. Modifications of this instrument were made to adapt it to the programming language used at the research university. Originally the survey was written for the C++ language, and the research university teaches programming in Java. Both Java and C++ are Object Oriented Languages and, although each language was designed for different purposes, the languages are syntactically similar as Java was strongly influenced by C/C++. Additionally the focus of six questions was altered to reflect the community nature of OSS rather than the solitary nature of the work done in most programming courses. To validate these changes I communicated with one of the original survey authors, Dr. Susan Wiedenbeck, Professor, College of Information Science and Technology, Drexel University. Dr. Wiedenbeck gave me feedback reporting that the changes made to the survey were appropriate. She cautioned that the original survey length of 32 questions was daunting. Per Dr. Wiedenbeck's suggestion I shortened the survey to 15 questions (Wiedenbeck, 2011). In order to shorten the survey without sacrificing reliability I followed both the constructs used by the original authors (Ramalingam & Wiedenbeck, 1998) and a guide to the creation of self-efficacy surveys developed by Bandura (Bandura, 2006). The first step in shortening the list of survey questions was to look at the work of Bandura. In maintaining content validity I maintained the phrasing of the questions to focus on terms of "can do" rather than "will do". Self-efficacy is about a person's belief in their abilities, whether they can do something, rather than their intentions of accomplishing a task, or whether they will do something. Additionally I maintained the focus of the survey on programming because self-efficacy is situational, and I was looking at OSS development, so this was the appropriate measure. The survey was grounded in the theoretical framework of self-efficacy, so I also looked at the original study on the development of the model and reduced the number of questions following the format outlined in the research by considering the four categories developed, independence and persistence, complex programming tasks, self-regulation and simple programming tasks, and chose questions from each category in proportions equal in distribution to the original work. .

The Ubuntu Launchpad system allowed for the collection of artifacts that detailed the contributions of each student from their individual accounts. Ubuntu has a feature, called Karma, which shows what each contributor has added to the community and gives details of each contribution. Students printed out this information at the end of the term giving me the opportunity to see what each student was able to contribute.

Each student participated in logging two different aspects of the course. First they had to keep a log of their communication with their mentors. This communication typically took place via email or IRC channels. Students turned in a printout of these logs as part of their weekly work. Additionally students were all required to turn in a log of weekly group meetings. There was a form they could fill out detailing the date of the meeting, who attended, what form of communication was used, what was discussed, what problems were solved, and what problems remained to be solved. Alternatively students could turn in an IRC transcript of their meetings if they chose to.

Midterm and final reflections were written during the lecture part of the class. In addition to the written reflections, students answered questions testing their knowledge of the structure of the Ubuntu community. The formats of the tests were in keeping with the community spirit of the class and were conducted as group testing to test knowledge of the Ubuntu community. The reflection questions however, were answered individually. These reflection questions were designed as open ended to gather information on student's perceptions of their own experiences in the course as well as within the Ubuntu community. These questions were reviewed for construct validity by Dr. Jensen and Leslie Hawthorn, as were the items on the questionnaire above.

The final source of data came from teacher observation. As the teacher of the course I could observe both as a bystander as well as a participant in the discussions. Because two-thirds of the course was spent in a lab environment where students were focused on the project and contributing I had ample opportunity to observe their interactions. Field notes were kept on the observation made.

Data Collection

The initial phase of data collection took place on the first day of the course. Thirty one out of thirty two students registered for the class were present. Paperwork for the one student who was absent was completed before he participated in the lab section that week. Before any dissemination of information about OSS students were asked to fill out paperwork. This paper work included an informed consent form, a questionnaire, and a self-efficacy survey.

The students taking this course were predominately recruited in upper division CS courses. During the recruitment talk I gave a brief description of OSS and talked about the benefit of OSS participation resume building and getting a job after graduation. These students arrived knowing that the course was my research. By way of introduction to the informed consent, I talked about the course as being the basis of my dissertation research. I informed the students that they did not have to participate in the research and that participation would in no way affect their grade. Each student was given an informed consent letter with a form to sign acknowledging they had read the letter. All thirty one students agreed to participate and signed the form in class that day. The one student who was absent was given the same information about informed consent and my research and he also chose to sign the consent form.

After obtaining informed consent students were asked to fill out a questionnaire. The questionnaire asked for demographic information followed by information about previous experiences with programming in general and OSS specifically. Students were also asked to discuss what they wanted to get out of the class. Finally students were asked to fill out the self-efficacy survey. No information about the purpose of this survey was given so as not to skew the feedback.

All thirty two students completed the initial paperwork during the first week of the class. The confidentiality of the data was protected using a numeric code chosen by the student. Each student was asked to label all of his/her papers with a four digit code chosen by them. For ease of remembering most students chose to use the last four digits of their student ID. The only place where they listed both their names and their numeric code was on the questionnaire, which was stored in a location separate from the rest of the forms.

At the beginning of week six the mid-term was given. The focus of the midterm was to have groups navigate the Ubuntu community website and find specific information. After the group part of the midterm each student was asked to briefly discuss the three most important resources for engaging in the Ubuntu community up to that point in time.

In addition to the formal documents students were also responsible for weekly assignments. For the eight weeks of group work each group had to hold a weekly meeting and then turn in a group meeting form reporting what happened. An alternative to filling out the group meeting form was to conduct a meeting via an IRC channel and turn in the log from the meeting. In addition to the group form each student was required to turn in a log of their communication with their mentor during the three week time they had a mentor. Students either turned in a copy of the email exchanges they had with their mentor or they turned in an IRC log depending on their medium of communication.

The last self-efficacy survey was administered on the last day of the class. The final focused on student reflection. Each student was asked to write about their experiences in the course, their intent to continue specifically with Ubuntu or more generally any OSS project, and finally the students were asked to grade themselves and give reasons for the grade assigned.

The ongoing source of data collection was my observation of my students. I spent lab periods either in conversation with groups of students, listening to their ideas and how they planned to achieve their goals, or I would stand in different locations in the room and listen to how they were processing amongst themselves. I kept a journal of my observation over the course of the term.

Data Analysis

The qualitative data was analyzed using RQDA, an open source software product. RQDA is an extension module for the statistical analysis open source package R. The R package was used to consider the quantitative data provided by the self-efficacy surveys.

The process of analyzing data started with the creation of codes used as a lens to consider the data as it related to my research questions. I looked at each research question and considered what would support the presence of what I was looking for or demonstrate the absence of the constructs. I developed a list of codes related to my research questions a (Table 3.2). Additionally I coded all information provided by the one non-CS student as a special case.

5. How do students view the importance of group members and mentors and their effectiveness in promoting engagement in OSS? (Observation and written work)	BK_tutorial/UbuntuOregonMeeting Contributions Global_jam Mentor Mentor_late Non-Human_Help Barrier Frustration
1. Will the course design produce students who engage with the OSS community at a percentage greater than 10%? (Measured by number of students who contribute to the project, karma and final)	CoP_JointEnterprise CoP_MutualEngagement Cop_SharedRepertoire Engagement Barrier

	Frustration
2. Is there a change in self-efficacy from the beginning to end of the course? (Measured by self-efficacy survey and final)	SE_Performance SE_Physiological SE_Verbal SE_Vicarious Barrier Frustration
3. Do students anticipate continued participation in OSS projects? (Final)	Future_LackOfTime Future_Mentor Future_Non_Mentor Future_Plans_Other Future_Plans_Ubuntu Barrier Frustration

Table 3.2 Initial Code Relationships to Research Questions

During the first pass I looked at student input submitted in writing at the midterm and final points of the course. I took notes as I used the above codes to help in the revision of the coding scheme to facilitate a second pass through the data with codes that more accurately reflected the theoretical foundations of my research. For the construct of CoP Wenger provides a list of indicators that a community of practice has formed (Wenger, 1998).

1. sustained mutual relationships – harmonious or conflictual,
2. shared ways of engaging in doing things,
3. the rapid flow of information and propagation of innovation,
4. the absence of introductory preambles, as if conversations and interactions were merely the continuation of an ongoing process,
5. very quick setup of a problem to be discussed,
6. substantial overlap in participants' descriptions of who belongs
7. knowing what others know, what they can do, and how they can contribute to an enterprise,
8. mutually defining identities,
9. the ability to assess the appropriateness of actions or products,
10. specific tools, representations, and other artifacts,
11. local lore, shared stories, inside jokes, knowing laughter, and
12. jargon and shortcuts to communication as well as the ease of producing new ones (Wenger, 1998, pp. 125 – 126).

I created codes to look for evidence of these indicators within the assigned groups. Additionally I looked for evidence that a CoP had not formed within the assigned groups. I also looked for indicators of self-efficacy based on the work of Bandura:

1. Mastery experience (enactive attainment)
    - The most influential source of efficacy information because it is based on authentic mastery experience. Successes raise efficacy appraisals; failures lower them.
    - But the contextual filter is at work here--after "a strong sense of self-efficacy is developed through repeated successes, occasional failures are unlikely to have much effect on judgments of one's capabilities"
    - "People who are assured of their capabilities are more likely to look to situational factors, insufficient effort, or poor strategies as the causes"
    - "Failures that are overcome by determined effort can instill robust percepts of self-efficacy through experience that one can eventually master even the most difficult obstacles".
2. Vicarious experience
    - If others can do it . . . but by the same token, "observing that others perceived to be similarly competent fail despite high effort lowers observers' judgments of their own capabilities and undermines their efforts".

- Some factors make us more sensitive to vicarious influence
    1. Uncertainty about our own capability.
    2. Little prior experience.
    3. Criteria by which ability is evaluated - "Because most performances are evaluated in terms of social criteria, social comparative information figures prominently in self-efficacy appraisals"

3. Social persuasion (including verbal persuasions)
    - "Can contribute to successful performance if the heightened appraisal is within realistic bounds".
    - "However, the raising of unrealistic beliefs of personal competence only invites failures that will discredit the persuaders and will further undermine the recipient's perceived self-efficacy".
4. Physiological states
    - People "read their somatic arousal in stressful or taxing situations as ominous signs of vulnerability to dysfunction".
    - "Treatments that eliminate emotional arousal to subjective threats heighten perceived self-efficacy with corresponding improvements in performance" (Bandura, 1986, pp. 399 – 401).

Table 3.3 shows the indicators with the codes for each.

Indicators	Codes Positive	Codes Negative
sustained mutual relationships – harmonious or conflictual,	Relationships_+/-	Group_fell_apart
shared ways of engaging in doing things,	Shared_ways_of_engaging	
the rapid flow of information and propagation of innovation,	Rapid_info_flow	
the absence of introductory preambles, as if conversations and interactions were merely the	Ongoing_conversation	

Indicators	Codes Positive	Codes Negative
continuation of an ongoing process,		
very quick setup of a problem to be discussed,	Quick_disscussion_setup	
substantial overlap in participants' descriptions of who belongs		
knowing what others know, what they can do, and how they can contribute to an enterprise,	Who_knows_what	
mutually defining identities,		
the ability to assess the appropriateness of actions or products,		
specific tools, representations, and other artifacts,	Specific_tools	
local lore, shared stories, inside jokes, knowing laughter, and	Shared_humor	
jargon and shortcuts to communication as well as the ease of	Short_hand_speak	

Indicators	Codes Positive	Codes Negative
producing new ones		

Table 3.3 Codes for the Indicators a Community of Practice

<table>
<tr><th>Source of Self-Efficacy</th><th>Positive Code</th><th>Negative Code</th></tr>
<tr><td>Mastery experience</td><td>I_did_it</td><td rowspan="4">Could_not_understand
Did_not_do_enough
Frustration
I_could_not_do_it
Neg_self-efficacy
Ubuntu_negative_community</td></tr>
<tr><td>Vicarious experience</td><td>Someone_showed_me_how</td></tr>
<tr><td>Social persuasion (including verbal persuasions)</td><td>They_thought_I_could</td></tr>
<tr><td>Physiological states</td><td>I_had_the_support _of</td></tr>
</table>

Table 3.4 Codes for Sources of Self-Efficacy

The second set of codes was applied to the midterm and final reflection data as well as the notes and transcripts of group meetings each group turned in weekly. The second pass provided a more nuanced view of the data. After the second pass the codes and files were divided into categories and the files were assigned to specific cases to allow the analysis of the data from many viewpoints. File categories were designated as midterm and final. These categories are designated within the name of each file, the creation of file categories served to separate the data by category in order to consider data temporally.

Once the data was coded and broken down as outlined above, I read the mentor interactions students turned in and created memos on each student file to reflect the engagement level with their mentor. Engagement was determined by how often students communicated with mentors and how they followed up on suggestions given by their mentor. Additionally assignments from mentors were noted.

The quantitative data that was collected from the self-efficacy surveys was used to create box plots in order to look at the results of the surveys over the course of the term. With an n of 30 it was not possible to perform parametric statistical analysis; however it was of interest to look at the trends suggested by this data.

Finally the data was triangulated in the following way:

1. Mentorship was coded within the midterm and final reflection data which was triangulated against the mentor communication reports from each student.
2. The CoP/group interaction was coded within the midterm and final reflection data and triangulated against the group meeting logs.

3. Contributions to the community were considered from the reporting in the final reflection data which was triangulated against the Karma reports and Launchpad information.

Ethical Considerations

As a researcher/teacher I took specific measures to prevent the collection of data from contaminating the design of the course. The first step was not to look at the data collected to prevent information collected from the first lab with mentors from influencing how the mentors were assigned and administered to the second lab. Since mentorship was being considered from the perspective of student perception and was in part temporal in nature, it was important to not institute changes based on lessons learned from the first lab.

The second risk being a researcher/teacher is that my students may have been trying harder based on knowing that I was researching them and their relationship with mentors. This was done by design. I was striving for students to use their mentors and to engage with the community, telling them this was designed to emphasize the importance of these elements.

Chapter 4 : Research Findings

Introduction

Chapter four is divided into eight sections in order to fully explore my research questions. In section one the CoP framework is operationalized to provide the framework used to analyze the data in this study. Sections two through four will address the first and second research questions; 1.How do students view the importance of group members and mentors and their effectiveness in promoting engagement in OSS? and 2. Will the course design produce students who engage with the OSS community at a percentage greater than 10%? These two questions are answered together because the analysis of the data suggested that there was an association between students who valued their group members and engaged with their mentors and the level at which the student was able to contribute to the Ubuntu community. The fifth section looks at two unique cases that are excluded from the results section; however they provided a distinctive analysis of two different points of view. Section six addresses self-efficacy as considered in question three of the research questions; 3. Is there a change in self-efficacy from the beginning to end of the course? Section seven then looks at student intent to continue to provide an answer to question four; 4. Do students anticipate continued participation in OSS projects? And finally in the last section I summarize the results in this chapter.

In the sections two through four I considered the community aspects of the pedagogical paradigm provided by the framework for this study. I specifically looked at the answers to questions one and two as they are reflected in the three different categories, the groups that exhibited high levels of indicators of the development of a CoP (as discussed in Chapter 3) and high levels of engagement with their mentors, followed by groups with moderate levels, including the one group who had moderate levels of CoP indicators coupled with low engagement with their mentors, and finally the groups that had low or no indication of either. Analyzing the data from these three levels allowed me to operationalize these indicators of CoP while providing a more robust view of the commonalities and differences among groups that shared levels as well as the across the differing levels.

Mentor engagement was coded for communication with mentors to consider the amount of information a student was able to elicit in order to facilitate contributing to Ubuntu. The specific indicators used were:

- Were the questions asked just general how did you get involved questions?
- Were the questions asked detailed and focused on using a mentor to assist in finding ways to contribute?
- Did the mentor assign a task?
- If the mentor did assign a task, did the mentee follow through?
- How many times did the mentee communicate with their mentor?

I also analyzed the effect of the Global Jam and a virtual live demonstration of bug triage given by Benjamin Kerensa. The Ubuntu project has a release every six months. Approximately 6 weeks before the release Ubuntu holds a Global Jam in which groups of developers get together in locations around the world to participate in debugging the source code before the release. The Global Jam provided a place for students to go and be in the middle of the development of Ubuntu, working with developers from the project. Because Global Jam was on a Sunday students were given the information and a description of what it was about, but it was the student's choice whether to attend or not. Additionally Benjamin Kerensa, the team leader for Ubuntu Oregon, joined us virtually to give a hands-on presentation of bug triaging. Bug triaging includes:

- Responding to new bugs as they are filed.
- Ensuring that new bugs have all the necessary information.
- Assigning bugs to the proper package.
- Confirming bug reports by trying to reproduce them.
- Setting the importance of bugs' reports. *(Bug Control members Only)*
- Searching for and marking duplicates in the bug tracking system.
- Sending bugs to their upstream authors, when applicable.
- Cross-referencing bugs from other distributions.
- Expiring old bugs. ("BugSquad - Ubuntu Wiki," n.d.)

Bug triage, along with editing documentation, is often the place recommended for newcomers to start. Bug triage and documentation allow people who are new to the community to participate in meaningful ways that support Ubuntu. While they contribute they are afforded the opportunity to see how the packaging of the Ubuntu project works as well as being able to observe the way the community interacts. From the vantage point of triaging and documentation newcomers also get an overview that helps them hone in on areas of interest. This portion of the analysis pointed to the effectiveness of providing hands on walk-throughs. In the final part of sections two through four, I consider contributions to the Ubuntu community in light of these levels and look at the connections between these elements within each level.

Once I have reported the findings of research questions one and two I report the analysis of question three where I considered the presence or absence of a change in self-efficacy. I operationalize the coding scheme used to look at the qualitative data that addressed this issue as well as considering the quantitative results from the self-efficacy survey. I also consider the outcomes of the CoP and how it is associated with self-efficacy. This is followed by a discussion of the analysis of the data regarding student intent to continue contributing to Ubuntu or OSS in the future, as reported in the Final Reflection. The final section summarizes the results in this chapter addressing what worked, what didn't work, and the interactions of the elements all together.

Analysis

In the following three sections groups are separated into three categories, ranked high to low. All of the groups, with the exception of one, demonstrated similar levels of engagement with their group as they did their mentors. The group that had differing levels had a moderate level of CoP formation and low levels of mentor engagement. This group was included with the moderate category because they had outcomes that were similar to this category. Their case is noted during the discussion of the moderate category. The group ranking considered the number of meetings each group had, the course assignment was at least one a week, how focused the group was on finding ways to contribute, and how often group members brought outside resources to the group. Also considered was evidence of any attempts made by the group to engage in research between meetings. For example, one group had a contest to see who could triage the greatest number of bugs before their next meeting, showing a group effort to accomplish tasks that furthered their abilities to contribute to the Ubuntu project. This was contrasted with groups who met infrequently, less than once a week, and their Group Meeting Forms or logs reflected little to no discussion about the project and no evidence they sought out resources individually to bring back to the group.

The categories also look at engagement with mentors. Specifically I considered how often students communicated with their mentor and what was said in the exchange. A student who emailed their mentor and asked detailed questions about what steps they could to take to contribute to a specific area, for instance artwork, and had an ongoing conversation with their mentor was ranked high. However, a student was ranked low if they emailed only once or twice, asked vague questions or did not respond when their mentor asked specific questions in order to narrow down their focus and help them contribute. Moderate engagement was noted when students emailed less than once a week, but asked questions that would elicit some information on contributing, but not as in depth as the people who were rated highly.

The course was taught with students broken into two labs and each lab was broken into groups as described in Chapter 3. From each lab there were groups that fell into each of the categories described above. Within each of the following sections when I refer to the groups as individual groups within their lab time I will designate the nature of the lab first in the name, EMG for Early Mentor Group and LMG for Late Mentor Group, followed by the group number within the lab. For lab one there were five groups numbered 1 - 5 and lab two had four groups numbered 1 - 4. When referring to the categories represented by the three levels I will use High, Moderate, and Low respectively.

High Indication of CoP and Mentor Engagement

Group Characteristics

Of the total of nine groups, three groups developed high levels of indicators of CoP and mentor engagement, one group from lab one and two groups from the lab two. There were a total of twelve students in these groups, 40% of the total number of students in the course. The demographic breakdown, along with percentage of the total class, is shown in Table 4.1. It should be noted that one of the students included in this group was not officially assigned to any of the three groups, but was an unofficial member of EMG4. His own group did not have any indicators of the formation of a CoP and soon after groups formed he would “hangout” with EMG4 often participating with them in class as well as joining some, but not all of their meetings.

Gender		Race		Age Group (1-4)				Length of time programming (in years)			
Male	Female	White	Non-White	18–25	26–35	36–45	Over 45	0-1	2-4	4-8	Over 8
9 (36%)	3 (60%)	9 (37%)	3 (50%)	7 (33%)	4 (57%)	1 (50%)	0 (0%)	2 (40%)	7 (41%)	2 (40%)	1 (33%)

Table 4.1 Demographic Breakdown of the High Level Group

The conceptual framework of this research argued that a collaborative environment had a positive effect on the retention of women in CS education (McDowell et al., 2006) and furthermore using a CoP model supports one of the primary motivations for participation in OSS, that of learning new skills (Ye & Kishida, 2003). Using this framework to analyze the data drew attention to these three groups as having participated in meaningful ways that supported the formation of a CoP.

In considering the first indicator of mutual engagement (see Figure 4.2 on page 84), sustained mutual relationships, I looked at the number of group meetings for each group. The groups that fell in this high category all had group meetings at least once a week. Within the data contained in these reports and transcripts of group meetings, as well as the Midterm and Final Reflection, I looked for evidence of the other three indicators of mutual engagement, shared ways of doing things, rapid dissemination of information, and the sense of an ongoing conversation. I looked for students who shared specific ideas on how to they became familiar with an aspect of the project or a student who found a project that they shared with their group so everyone could contribute. In the following example Bobby (all names used are pseudonyms), from EMG4, brings a tutorial he found that is a resource for the programming language that is used in Ubuntu, Python, and shares it with his group so that they can use the resource also.

Bobby -	There's a guy on YouTube who does pretty good computer science tutorials, in case anybody's interested. His username's "thenewboston" and he's got a series of Python tutorials
Bobby -	I was going to take a look at those this week
Tom -	nice, I will look into that
Amelia -	awesome, thanks! I'll be looking at that too (EMG4 Group meeting IRC Log, 2/15/12).

LMG3 also gave an example of one of their group members bringing a project they found to the group so they could work on it together. In this particular example Niall found a posting in a mailing list asking for new people to contribute to a documentation project. He in turn brought it to his group members to pursue together. This project also brought this group in contact with members of the documentation team.

> As a group we updated the documentation to an add on application called Pidgen. Once I was able to understand the format in wiki it was rather elementary and was easy to get our small group of 4 people interacting and updating documentation (Niall, Final Reflection, Question 3).

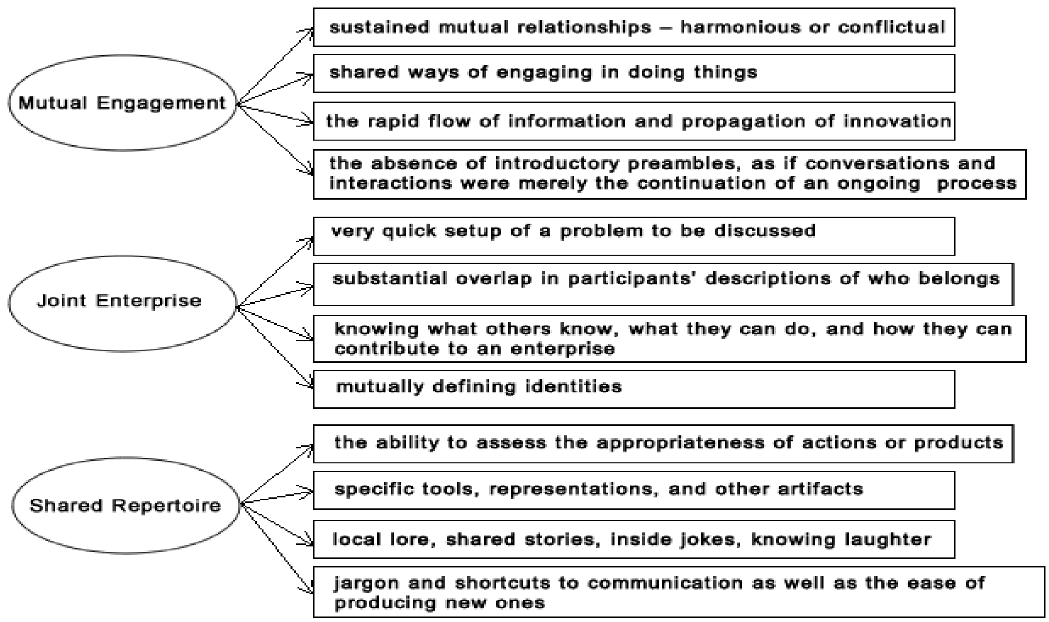

Figure 4.1 Indicators of the Formation of a CoP

I also considered a rapid flow of information. To find evidence of this indicator I looked for indications that one or more group members had gathered information and then disseminated it to the group within a short period of time. I was looking for signs that group members shared with each other in an effort to get everyone in the group engaged in and contributing to Ubuntu. LMG4 had this exchange in their Internet Relay Chat (IRC) meeting log shortly after Riley received information from his mentor. The time frame was established by comparing the date of this exchange, 8 March, 2012 and the date of the communication from Riley's mentor was 6 March, 2012. This shows a short space of time between when Riley got pointers on how to triage bugs and when he shared them with Harry.

Riley - I was also given some information from my mentor about how to go about fixing bug once they are triaged.
Harry - Oh really...
Riley - It's basically a way to find the easy to fix bugs.
Harry - Oh that's good
Harry - Yea I assumed that it would be. I'm excited though to see what you learned. Is that what you're working on now?
Riley - Nothing right now.
Harry - So look at this Bug #949117
Riley - That looks like an easy on to test.
Harry - ya I just did it and its correct ha-ha
Harry - the scroll bar blends in with the highlighted files
Riley - So then it's confirmed (LMG4 Group Meeting Log, 3/8/12).

EMG4 solved the following problem over IRC within their group, demonstrating their ability to find and share information in real time. In this conversation Josh found an issue that had been incorrectly reported as a bug. The report needs to be moved to the wish list, a place used to request features, but he did not know if he had the authority to make that change. Josh, Amelia and Brandon used one of the resources they had found, the Ubuntu Oregon channel, to ask for information. At the end of the exchange Josh looked at the #ubuntu-bugs IRC (the IRC channel devoted to people asking questions about bug triage) as was suggested. Amelia was still on #ubuntu-us-or (the Ubuntu Oregon channel) when Amelia saw information that Josh needed posted on the Oregon channel. She then told Josh to go to the Oregon channel for the information. This exchange of information happened in real time during a group meeting on EMG4s IRC channel and demonstrates a rapid flow of information.

Josh - ok I found it. It can be converted to wish list but I don't think we have that capability

Amelia -	I asked the Oregon group too Josh, see what they have to say.
Brandon -	like that Oregon bot that pulls bugs when you mentioned the bug number
Amelia -	Josh, Benjamin said "You might ask someone on #ubuntu-bugs to look at it who is on the control team so they can wish list it"
Amelia -	You're right, we probably don't have that capability
Josh -	yep I'm on that channel right now thank you
Amelia -	Josh, read the Oregon IRC, they have info for you.

The final indicator I looked at for evidence of mutual engagement was conversations that were ongoing. This element was more difficult to tease out of the data collected. There is a sense of ease and humor evident in the meeting transcripts. I also note that there is a focus on the task at hand that is evident in the group meeting reports from LMG3 and the IRC logs of the group meetings from EMG4. There is less evidence of this attribute in the data from LMG4. LMG3 demonstrates this focus on the task at hand in the following meeting report, which also shows a significant amount accomplished at this meeting.

> What was discussed:
> Plan of attack for the week, group is split between bug work and documentation updates for the Pidgin wiki page. Adam is still in work with his mentor to work on a Quicklist. Everyone seems comfortable with bug triage and documentation updates at this point. Plan is to have members triage a couple bugs this week and work on some documentation update to the Pidgin wiki page.
> Problems solved:
> Pidgin documentation updates completed by Adam, Bug work done by Niall, Rory, and Libby during the week.
> Problems remaining unsolved:
> Minor lull in work while awaiting release 12.04 in April. Team is working on minor bug issues that appear in the bug logs while we await release. With finals nearing shortly a slight lull in time to work on bugs is noted. But team members are confident in being able to continue the process even after the term completes. (LMG3 Group Meeting Form, 3/12/12)

LMG4 however had less focus and demonstrated more off topic conversation in their IRC logs of their group meetings and there was less evidence that both group members worked together to bring ideas to the table. For instance, in the following piece taken from an IRC log of a group meeting, they started by discussing the download of an ISO. The end of the previous meeting one week earlier ended with both students planning on downloading and working on testing this particular ISO.

Harry -	Hey hey
Riley -	Hello
Harry -	How's it going?
Riley -	Okay. And you?
Harry -	Not too bad
Riley -	Have you done any testing?
Harry -	I haven't downloaded the ISO. Working on that. You?
Riley -	I have downloaded the ISO, and may have found an error. The problem is not that I'm getting an error message, but that I'm getting 80 error messages.
Riley -	Makes it a bit hard to report.
Harry -	Yea not sure how you would go about that

This exchange happened later in the term and the students do not seem to stay focused on the practice in between meetings at this point. This would have been a good opportunity for them to contact a mentor or get on the Ubuntu IRC channels to see if they could get this question answered.

The second characteristic of the practice that is seen to bring a community together is joint enterprise. Mutual engagement showed the formation of relationships, the indicators of a joint enterprise look at how, through a history of interacting, the group shared an understanding of the enterprise. In this case their enterprise was pursuing ways to contribute to the Ubuntu project.

Teasing the evidence of joint enterprise from the data was a challenge. The indicators that suggested members' agreement on who belong and common definitions of identity were not explicitly stated. However there were nonverbal signs I recorded in field notes that pointed to evidence of these indicators, although it was not strong. I have reference to EMG4 taking on the title of "The Awesome Group". This label separated them from the other groups in lab, and established a sense of belonging in their group.

LMG3 showed cohesion in their group meeting reports by the joint nature of their participation in projects. This held true until the group was assigned mentors. After their introduction to mentors one of the members of LMG3 worked on an assignment separate from his group. Although they still met, Adam focused his time on the project from his mentor, removing him from the joint effort of contribution to Ubuntu. The remaining three members continued to work on the same segments of the Ubuntu project.

LMG4 did not show evidence that these indicators were present in their group. Although they acknowledged their group as a source of help in engagement, they did not have a strong cohesion, especially when compared to EMG4. LMG4 had no evidence of ways of belonging to their group.

I did code for the other two indicators of joint enterprise, the ability to quickly set up a focused discussion of a problem and knowing what each member knew. I found evidence of these indicators in all three groups. EMG4 started the meeting in the following excerpt by focusing right away on what they want to accomplish. They quickly honed in on ways to recreate a bug, part of the triage process. This example shows how quickly this group found a problem that one of them had, and then solved it with participation from all of the group members.

Tom -	So what do we want to talk about this meeting?
Bobby -	Let's talk about bug tracking. That's something I should probably know more about
Bobby -	Does anybody have any resources they want to share?
Bobby -	Like, outside of the Ubuntu webpage?
Amelia -	I haven't found out how to contribute anything yet.
Bobby -	I wonder if there are any good tutorials...
Tom -	I found a list of bug trackers on Launchpad
Amelia -	I bet trying to recreate a bug would be fun.
Tom -	https://bugs.launchpad.net/bugs/bugtrackers
Amelia -	Found this quote on ubuntu.com: On average, over 1,800 bugs are filed every week with more than 500,000 bugs filed so far. The Bug Squad is always in NEED of more help from the community!
Wendy -	That's an idea Amelia, I wonder how we would go about that.
Amelia -	Thanks Tom!
Amelia -	I have a feeling Launchpad is where to go (EMG4 Group Meeting IRC Log, 2/15/12)

Additionally EMG4 shows that they know who within their group to turn to for specific knowledge in the following two excerpts from the Midterm and Final Reflection:

> Groups have been great. I think that this has been very useful as we all have different ideas and see things in different ways, our group discussions are great as we contribute information and learn more than what we would individually. (Josh, Midterm Reflection)
>
> Also Amelia was a big help in learning to triage bug. If I had a question she generally knew the answer. (Wendy, Final Reflection, Question 2)

LMG3 exhibited group cohesion as well as the ability to discuss and solve problems within the timeframe of their meeting. This excerpt also shows that Libby is the person acknowledged for having specific knowledge about bug triaging that she gathered from the Global Jam. The group was able to assimilate her information allowing the rest of the group to benefit from her experience.

> Date of meeting: 05 March 2012
>
> Who attended: Adam, Rory, Libby, Niall
> Form of communication (e.g. Skype, phone, email, IM, IRC, in person): In-person
>
> What was discussed:
> As a group finalizing work on the documentation page of Pidgin (instant messaging program) As well as work to identify bugs throughout the week. Adam is still working on a Quicklist assignment from his mentor so has been doing this in regards to his work for the week. Libby attended the get together in Portland on Sunday for the bug squashing meeting. She was able to triage a few bugs there and is in the process of identifying more to work on this week. Group has a good working knowledge of triage process and documentation now and is more actively involved in working on harder assignments.
>
> Problems solved: Libby - various bugs triaged Rory and Niall - Pidgin documentation updates including updates to graphic files on the wiki page as well as cleanup to walkthrough tutorials. Adam has added a Quicklist for smplayer.

> Problems remaining unsolved:
> Bugs to work on and more documentation to clean up. (LMG3 Group Meeting Form, 3/5/12)

Libby also pointed out the importance of knowing who knows what, as well as having the group knowledge to turn to, in this quote from her Midterm Reflection:

> We work together to engage in Ubuntu, if someone finds something they think is useful/important they share it, which helps me get more information. I'd probably be lost without help from people. (Libby, Midterm Reflection)

Lastly LMG4 also showed indication of a joint enterprise. In the following excerpt from a meeting the two group members not only solved the problem of how to find bugs to triage, they also showed that they know where to go for help outside of their group to acquire knowledge they needed. Here they discussed going to a larger group within the local Ubuntu community to seek help:

Riley -	But getting back to my idea
Harry -	Oh true
Riley -	What we need to do it go to the Ubuntu hour in Salem and ask questions there.
Harry -	when does that start
Riley -	6 pm to 7 pm - Commons Coffee Salem Feb 16 (LMG4, Group Meeting Form, 2/16/12)

The third characteristic of a practice that assists in the cohesion of a community is shared repertoire, or strategies shared by the group to accomplish the goal of finding ways to contribute to Ubuntu. This element of the practice arose from this pursuit but it continued to be modified by continuing group interactions. As the group members formed a history they began to develop communal resources and tools. They also began to exhibit interactive signs of this history. I specifically looked for a shared sense of humor with inside jokes and an ease of communication that comes with this familiarity as well as evidence of resources and tools shared by all group members.

All students in both labs received a basic set of tools specific to the Ubuntu community, as discussed in chapter three, during the first two weeks of the course. In coding the data I looked for additional tools and resources the group developed beyond this initial toolset. In the following example LMG3 sets up tools to use for conducting meetings (Skype) and focuses on attending an Ubuntu community meeting being held on an IRC demonstrating the use of communal resources and shared tools.

> What was discussed:
> Worked on establishing solid communication link. Skype for all future meetings. Bug tracking and follow up for confirmation and assigning to a forward person to work on bugs. Meeting with IRC community and interfacing with the community. Next meeting scheduled for Tues. 21 Feb 2012 2pm or later.
>
> Problems solved:
> Channels open on IRC for information exchange. (LMG3 Group Meeting Form, 2/13/12)

Additionally there was evidence of how the development of tools effected participation by this group member of LMG3; “After I had learned and got comfortable with the avenues of communication everything definitely got easier” (Rory, LMG3, Final Reflection, Question 5).

EMG4 specifically addressed the aid of the toolset introduction from the beginning of the class as well as discussing other tools they found helpful. The following are examples of creating tools that allow the group to function together. “In the beginning this class helped show webpages and such to get accounts created (Amelia, EMG4, Final Reflection, Question 2) and “my instructor has taught me how to use the tools of the community and has given me a plethora of resources to learn from” (Wendy, EMG4, Midterm Reflection). This group also pointed out the importance of tools incorporated into their group “In order to engage in the community, I used IRC's to talk to classmates and other members. The Oregon room was most helpful.” (Amelia, Final Reflection, Question 2)

Shared sense of humor and ease of communication was apparent with all groups in this category, especially LMG4 and EMG4, as noted in my field notes. However I find it more striking to present examples of these attributes in their own words. Within a group meeting transcript for LMG4 the following interaction took place, showing a sense of humor that existed between group members. Here they discussed contributions about four weeks before the end of the term, the two group members were still struggling:

Riley -	So, what do you think we should do?
Harry -	Pray that our mentors can help
Riley -	I prefer to leave hoping that the fates can rescue us from catastrophe as a plan B personally.
Harry -	Ha-ha true (LMG4 Group Meeting IRC Transcript, 2/23/12)

EMG4 has the following exchange that demonstrates their humor and bantering communication style. This except also demonstrates the focus of the group on the goal of contributing to Ubuntu, but they have found a fun way of accomplishing this. Not only did they accomplish the task, but they are formed a bond, while also creating a story and a shared history.

Amelia	Certainly. I want to do this before Wednesday. Let's challenge each other? What do ya think? Like a race?
Amelia	Huh Rugby dude? You like competition right?
Amelia	Bobby, I bet you don't mind showing us up?
Tom	ha-ha yea
Bobby	I challenge everybody who is not currently on the IRC!!!
Amelia	Lol! What are the stakes if we don't achieve what we want?
Bobby	good idea though. Do you want to race to downloading code specifically, or just to contribute something
Bobby	?
Bobby	forgot to put the question mark
Amelia	probably contribute something or just jump a hurdle that is keeping us from progressing.
Bobby	if we don't achieve our goals, we have to do fifty push-ups
Amelia	I'd like to download something, you may want to focus on something else.
Amelia	Push-ups would be a challenge for me! I'm in.
Amelia	Tom can make a pie for pie day.
Tom	Sure sounds like a good plan
Amelia	hehe, jk Tom.
Amelia	Bobby, hmmmm, you can sing for the class :D

At this point it in analyzing the data I had seen the formation of a CoP within each group. Although I have shown that every element was not met by every group, I still found sufficient indications supporting the existence of CoPs in these groups. I also considered evidence of mentor engagement as part of the community support for contribution to Ubuntu. As noted above, all of these groups also showed strong engagement with their mentors. Within the analysis of the mentor data I also incorporated the effects of the Global Jam and the hands on demonstration by Benjamin Kerensa as all of these were potential sources of information on and guidance in contributing to Ubuntu from members of that community.

As I looked at the engagement with mentors, the Global Jam, and the Benjamin Kerensa Demo I looked explicitly for evidence of student interaction with Ubuntu members where specific questions were asked that led to information on how to contribute to a specific part of the Ubuntu project (e.g. documentation, testing, and bug triage) and where that information led to specific contributions. Below I report the results of having looked at statements made by students about their mentors as well as looking at the exchanges that took place between students and their mentors in the form of email. I also considered the perception of the students about the importance of the Global Jam and the live demo, specifically how these hands on experiences helped them contribute. The analysis of the data suggested that access to a mentor, the Global Jam and Benjamin’s live demo all served to assist the students in contributing to Ubuntu. As mentioned in chapter three the CoP that the groups form is within the larger practice of the Ubuntu community. This larger Ubuntu community shares the enterprise of creating and maintaining the Ubuntu operating system. The interactions considered below are examples of the contact the students had directly with this larger community. All of the students in this course were informed of the Global Jam. Five out of the seven students that participated were in the high CoP/ high mentor engagement group.

The group members of EMG4, including their unofficial member Brandon, all spoke of the value of their mentors and how they assisted in finding ways to contribute. At the mid-point of the term as well as at the end of the term, students were asked to write a reflection of their experience in the course and with Ubuntu. At the mid-point students were asked to share the three things that helped them most in becoming engaged with the Ubuntu project. At the end of the course they were prompted with questions, the ones addressed here asked what, specifically they did to become engaged and what their greatest source of help was in getting involved. These questions were designed to gather data on student's use of mentors and the Ubuntu community as well as the perception of the usefulness of these elements. The following is a reflection of EMG4 on these questions:

> Brandon [unofficial group member] was largely helpful in holding my hand and explaining the process about using the VirtualBox and Linux. However, getting plugged into Ubuntu itself, explaining expectations and defining terms, my mentor has been AWESOME! Having an experienced person that knows the ropes, can dumb things down, and willing to take time has made me feel more welcomed to ask questions, more confident in getting involved and more knowledgeable about what I'd like to so, as well as other options available. (Amelia, Midterm Reflection)
>
> My mentor Nathan - gives prompt, thorough, understandable, enthusiastic, accurate responses to all my questions and tells me where to look to get involved in the areas that I find interesting. (Bobby, Midterm Reflection)
>
> My mentor - this is another way to gain answers to questions I have.

> IRC channels and my mentor were the best source of help with getting engaged. After using them I started getting involved with bugs first and then tried to get into the artwork area. (Tom, Midterm and Final Reflection, Question 4)

> My mentor, my group members and the instructor herself. My mentor has been fantastic about pointing me in the right direction. He has assigned me tasks to complete and defined many terms I did not know. My group members have been very helpful, because often they have knowledge that I do not. Lastly, my instructor has taught me how to use the tools of the community and given me a plethora of resources to learn from.
> To engage in Ubuntu I began asking my mentor questions and figuring out small tasks I could do without getting overwhelmed. He showed me how to start editing wiki's and where to find ones that needed to be edited. (Wendy, Midterm and Final Reflection, Questions 2 and 4)

> I also got good information from my mentor. I got lots of details on information and requirement on how to report a bug and make changes to pages. (Josh, Final Reflection, Question 2)

> My mentor Benjamin was wonderful. Is quick to respond on IRC and made sure to mention local live events to attend. (Brandon, Final Reflection, Question 4)

Emails between these students and their mentors corroborate these reflections. Amelia’s emails with Vikram started with her asking broader introductory questions, however, by the second email she had honed in on asking about what terminology she needed to know, when due dates are within the community, and she also asked for specific links to learn how to contribute to artwork. Vikram responds with detailed information to Amelia’s questions. Amelia, Wendy and Josh all had Vikram as a mentor and Amelia cc’d Wendy and Josh to share the information she was gathering from Vikram with them, a sign of group cohesion. Wendy communicated separately with Vikram also asking for and receiving assignments from Vikram to help her start with documentation.

Bobby collected information from his mentor on languages he would need to know to develop desktop applications as well as links to help figure out how to contribute to bug triage. Josh had shorter communication with Vikram, but he was copied on the exchanges with Amelia. Tom gathered information on how to participate in artwork, what links to look at, the Launchpad artwork group and timelines effecting participation in the group. Finally Brandon hung out with his mentor on the IRC channel for the local Ubuntu Oregon team. Not only did Brandon get to interact with Benjamin in this manner, but he also got to "hang out" with other members of the Ubuntu community.

LMG3 engaged with their mentors and three of the four members discussed their mentors in their Final Reflections .The fourth member of the group does not specifically mention his mentor, but he posts links from an email that his mentor sent him in one of the LMG3 Group Meeting Forms. Additionally, reviewing his exchanges with his mentor showed engagement with him as evidenced by the specific questions asked and acknowledgement of the helpfulness of the mentor responses. LMG3 also picked up an assignment from the beginners mailing list which promoted interaction with the documentation team for Pidgin as they helped to rewrite the documentation for the Pidgin application.

One of the students in LMG3 was given an assignment to work on by his mentor. His mentor asked him to add Quicklists to the applications that he maintains. Quicklists are the lists that popup when a user right clicks on an application icon showing shortcuts specific to the application. This process required Adam to download the package code for the applications he was working on, and then add the Quicklist shortcuts. Once he had created the new file he created a diff file (a file that is just the difference between the original file and the changed file) and sent it to his mentor. He noted that this is one of the major sources of information for his eventual engagement in the Ubuntu project. When asked what his greatest source of help in becoming engaged he answered: "An assignment given to me by my mentor. It was helpful to me to have a clear assignment" (Adam, Final Reflection, Question 4). Emails between Adam and his mentor showed that the assignment promoted a discussion between them. There is evidence that questions were asked and answered which supported Adam's successful completion of the Quicklist tasks assigned to him.

Rory from LMG3 had several email exchanges with his mentor and his mentor helped him focus on bug fixes to help transition Rory into the project. Rory also asked for pointers to any additional documentation that he can work on beyond the Pidgin documentation he contributed to. It is also important to note that Rory, in the last exchange turned in for the class, talked to his mentor about how he could continue to contribute to Ubuntu after the class is over. In his Final Reflection Rory reported that under the guidance of his mentor he was able to find a way to successfully contribute to the Ubuntu project in two different areas, documentation and bug triage. This interaction was important to Rory as the following example of mentorship demonstrates the community that was formed around contribution.

> I contacted my mentor Vikram, for more guidance as far as what bugs to try and tackle, and what pages of the wiki needed the most updating. Our group also received an assignment from Benjamin, from Ubuntu Oregon, to fix up Pidgin's wiki page.
>
> (The greatest source of help in becoming engaged was) Probably reading the documentation at first, and then Vikram gave me some great direction after that. Having a mentor definitely helped, being able to ask direct questions and get in depth answers. (Rory Final Reflection, Questions 2 and 4)

The final member of LMG3, Libby, exchanged emails with her mentor, but she had a difficult time framing questions to her mentor that elicited information to guide her in contributing. In her first email she asked how long her mentor had been involved in Ubuntu and if it has been helpful. When her mentor responded answering those questions she wrote back asking the very generic question “Do you have any suggestions on what I could do?” referring to contributing (Libby, Personal Email, 3/3/12). In this same email she referenced her intent to attend the Global Jam the next day. The emails that followed focused on the Global Jam until her last two emails where in one she says “I've done some bug triage and documentation (for Pidgin). Not too sure on what else I should ask you...Do you have any other information/advice you think will help me?” (Libby, Personal Email, 3/13/12). Her mentor responded to this asking some pointed questions about what she was interested in and the last email she sent back she talked about still being unclear about what she wants to do.

> Well I really don't know what interests me still; I think I might like to try out developing where I could actually write software. I haven't looked into this much because I originally wanted to try bug triage and my class started to gear towards that anyways. But, developing is intimidating. (Libby, Personal Email, 19 March, 2012)

This email exchange demonstrated that, although Libby put effort into her communication with her mentor, she needed help in forming questions to ask to elicit the information necessary to help her contribute to the project. A positive outcome of Libby’s exchanges with her mentor was that she talked about going to the Global Jam and her mentor was also attending so they had the opportunity to meet in person.

Two members of EMG4, Amelia and Josh, along with Brandon, their unofficial group member, attended Global Jam. Analysis of the data suggested going to the Jam had a significant positive effect on all three of these students when examining the resources they found that supported their ability to contribute. When they arrived at the Jam, which was held at Puppet Labs in Portland, there were many members of the Ubuntu community set up to help debug the code for the upcoming release of Ubuntu 12.04, Precise Pangolin. People new to the Ubuntu community were placed with a veteran community member who walked them through bug triaging.

Amelia took notes while she was at the Global Jam on the specific steps to triage bugs (see Appendix 4). In her notes she recorded the steps of the hands on demonstration she received from the veteran Ubuntu developer. Her notes detailed how bugs get reported, how to view them in Launchpad, including how to find bugs that are easy to recreate in order to start triaging, and she outlined specifically what to do to triage any bug, including tips on making it simpler. She also took notes on how to report a bug. When she returned she presented what she learned to both labs, made her notes available, and gave encouragement to her group members, showing them how to triage bugs. Josh also took an active role within their group helping others to fill in the gaps in understanding how to triage bugs. Early on mentors provided the most significant sources of information for contribution; however, after the Global Jam, the hands on environment became the single most important source of information for both Amelia and Josh. Amelia's Final Reflection showed how the Global Jam increased her confidence by having someone with experience in Ubuntu define and explain the terminology used in the community and give a step by step demonstration of bug triaging. In addition to having the experience at the Global Jam, Amelia also remained in contact with the people she met there, continuing to use them as resources.

> But, the most helpful tool to get engage was going to the Global Jam in Portland. My mentor was a close second.
>
> The Global Jam was by far the greatest source. While there, other experienced members took time to show me things to do, explained tasks

> and terminology, and walked me through the first steps of BugSquad. They have also been just as supportive since meeting them.
>
> Getting help understanding the terminology and structure. After I had terms and processes explained to me, I was more confident that I could help, felt "able" to contribute, and had resources to turn to for help (those that explained this stuff to me). Understanding expectations of me was huge. At first, I was afraid I would do something "wrong". After experienced members clearly explained what to do and how, I wasn't afraid anymore. (Amelia, Final Reflection, Questions 2, 4, and 5)

In Josh's Final Reflection he discussed the Global Jam and noted that before this event he was unsure about contributing because he was afraid of making a mistake. After watching the step by step demonstration and being able to talk to experienced members of the community he felt confident to contribute triaging bugs.

> I also went to the global jam on Sunday and got lots of great information that allowed me to engage in Ubuntu
>
> The Global Jam in Portland was a big help. I was unsure on many things and I did not want to make a change that could possibly be wrong or not required. Listening and watching then go step by step on bug procedures reassured me that what I was doing was correct. (Josh, Final Reflection, Questions 2 and 4)

Libby from LMG3 also attended the Global Jam, and she spent some time with her mentor there. She gave credit to the Jam for providing her with the information she needed to contribute to the Ubuntu project by having experienced members of the community explain bug triage then show her how to contribute. When she came back from the Global Jam she also shared the information on how to triage bugs that she gathered from the demonstration for newcomers she attended with her group members to assist them in contributing also. In Libby's Final Reflection she reflected on how the Global Jam guided her in her contributions.

> I also attended the global jam in Portland at Free Geek which helped me engage in the Ubuntu community as well as contribute to Ubuntu.
>
> The greatest source of help in becoming engaged was the global jam. I attended because I received information and a lot of guidance from people who've been involved with Ubuntu. They explained in detail how to triage bugs and demonstrated several times. I suppose this would include IRC channels and emailing my mentor because they involve getting help from people who know Ubuntu well. (Libby, Final Reflection, Questions 2 and 4)

The final group of students in the high category, LMG4, also had one student attend the Global Jam and both students attended a weekly meeting of Ubuntu developers from the local Ubuntu team in Salem. Riley attended the Global Jam, along with Libby, Amelia, Josh, and Brandon, and had the opportunity to participate in the hands on demonstration of bug triaging. The group meeting transcripts showed that he used the information he gathered there to walk Harry through the process of testing a bug. The analysis of this data suggested that students who were in groups with high levels of CoP and that attended the Global Jam brought the information they gathered back to their group which benefitted the other members of the group. Riley reported in his Final Reflection that the hands-on demonstration was the most useful element of the Ubuntu community that helped him engage in contributing to the project.

> I went to the Ubuntu Global Jam, Ubuntu hour in Salem, spent time in various official Ubuntu IRC channels, emailed my mentor, and, of course, attended labs
>
> (The greatest source of help in becoming engaged) The experts at the Global Jam. When it comes to learning, nothing quite beats someone walking you through it. After that I would say it was the mentors, not as good as a getting help in person, but still helpful.
> I plan to continue triaging bugs while working my way up to development. My mentor gave me some info on this. I haven't looked at it yet. (Riley, Final Reflection, Questions 2, 4, and 7)

Harry demonstrates in this excerpt from his Final Reflection that the having a partner to work with and being shown by an experienced member of the Ubuntu community in the virtual live demo helped him to find a way to contribute through the community that had formed.

> What I thought was the greatest source of help was being paired up with another student, and also when an Ubuntu community member did a conference with the class to show how to in this case triage bugs to get us started. (Harry, Final Reflection, Question 4)

This section described the groups that had high indications that a CoP had formed. Additionally this section served to operationalize the indicators given by Wenger as signs that a CoP had in fact formed within these groups. Analysis of the data suggested that these groups formed strong communities for the purpose of contribution to the Ubuntu project. This section considered the first research question, how do students view the importance of group members and mentors and their effectiveness in promoting engagement in OSS? The analysis suggests that not only did these students find their group members and mentors important in seeking information about engagement and contributing, but these relationships supported a collaborative environment that encouraged successful contribution to the Ubuntu community. The next section looks specifically at the contributions made as they are associated with the levels of CoP formation and engagement with mentorship.

Contributions to Ubuntu

The goal of this course was to contribute to the Ubuntu project. The conceptual framework for this research supports the assertion that groups with a strong indication of the formation of a CoP and an active engagement with the community of Ubuntu, either through mentors, hands on demos or both, should have a high contribution level; that the collaborative pedagogy used in the design of this course promotes increased and deeper contributions to the Ubuntu project.

Data about contributions to Ubuntu were collect by the Karma feature on Launchpad. Launchpad is the platform where the bug triaging took place and as an incentive they have a system that keeps track of what individuals contribute to the community. Additionally documentation contributions were reported by the students and confirmed by looking at the history of edits made to the documentation. Contributions to the Ubuntu project were considered not just in quantity, but were also evaluated in terms of degree of effort required to complete the task. For example someone who tested and confirmed bugs put forth more effort than someone who changed the title of a bug. Although both were considered important in terms of contributing, effort was taken into consideration (See Table 4.2 for a breakdown of contributions by difficulty).

Simple	Edit a Bug Title
Moderately Simple	Report a Bug Revise a Bug
Moderately Difficult	Prioritize Bugs Write a description Comment on a Bug Asking for More input from the Reporter
Difficult	Patch Configuration Branch Merge Build Application Duplicate Bugs New Branch Quick Lists Test Cases

Table 4.2 Breakdown of Contribution Levels

In rating the contributions in this category I found that eleven out of twelve students ranked high on contributions. Contributions ranged from editing documentation, to testing and confirming bugs. Students also commented on bugs, asking for more information so the bugs could be triaged, reclassified bugs that were not really bugs, prioritized bugs, and reported bugs. Additionally there were students who patched a config file, proposed branch merges, registered new branches, and attempted to build an Ubuntu app. Figure 4.3 shows the contributions by this group by quantity and degree of difficulty.

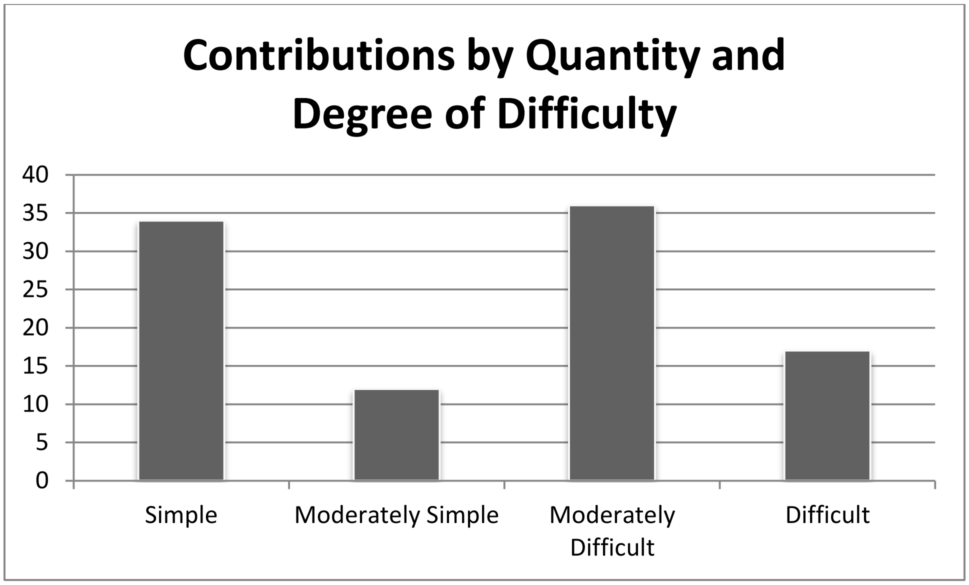

Figure 4.2: Contributions by High Category

Analysis of the data suggested that the students in this category made contributions to the Ubuntu project in a wide range of areas and with a variety of degrees of difficulty. It was also the case that individual students all had a range of difficulty within their contributions. When answering question three - will the course design produce students who engage with the OSS community at a percentage greater than 10%? –the collaborative model produced results that exceed this percentage. This category, high CoP and high mentor engagement, represents 40% of the students in the class and they all contributed at a high level in a variety of ways and to multiple areas.

Moderate Indication of CoP and Mentor Engagement

Group Characteristics

The category that represented the middle range of student contains five students who were rated as belonging to groups that had moderate indicators of a CoP and moderate engagement with mentors. In addition there was one group of four students, EMG3, who had moderate indicators of a CoP, but low engagement with their mentors. They were included in this group because their outcomes were in line with the other groups in this category. These nine students represent 30% of the total population of the class. The demographic breakdown of this category can be found in Table 4.2.

Gender		Race		Age Group (1-4)				Length of time programming (in years)			
Male	Female	White	Non-White	18–25	26–35	36–45	Over 45	0-1	2-4	4-8	Over 8
9 (36%)	0 (0%)	8 (33%)	1 (17%)	8 (38%)	1 (14%)	0 (0%)	0 (0%)	1 (20%)	6 (35%)	1 (20%)	1 (33%)

Table 4.3 Demographic Breakdown of the Moderate Group

One of these groups was from lab one and the other two groups were from lab two. It should be noted that LMG1 was originally a group with four members, but at week five of the term this group began to function as two separate groups. Two members of the original group are represented within this category, referred to as LMG1-1, and the other two members, referred to as LMG1-2, exhibited indications of belonging in the final (Low) group. Additionally the second group from lab two LMG2 originally had four group members, but one of their group members failed to show up to lab 60% of the time and only participated sporadically. This placed their fourth member in the final (Low) group while the remaining three members demonstrated traits that placed them in this category.

The groups that fell in the moderate range in terms of indicators of CoP had some evidence of the formation of relationships, but they either had a difficult time focusing in on how to contribute, or interaction between group members was sparse. When considering the aspect of “sustained mutual engagement” it was frequently the case that the groups met less than once a week. I looked at the Group Meeting Forms along with Midterm and Final Reflections in order to find evidence of the additional indicators of mutual engagement. Students demonstrate this relationship by sharing information they find or receive from others with the group.

EMG3 was the group that had the greatest indication of the formation of a CoP within this category. EMG3 met weekly and set up a way to share information using Dropbox and this was a successful way to share documents or links they found helpful. This not only supports the evidence of shared ways of engaging, but also supports rapid flow of information. They shared information by passing on information about applications that were helpful in testing ISOs, but there was little evidence that they solved problems or worked together with the information they shared. A good example of this can be seen in the following excerpt taken from a Group Meeting Form from EMG3 which shows a relationship exists between the members of this group evident in the sharing. However this exchange stops short of having group members use the information to contribute. In this example Cole tells his group members how to report a bug, but he stops short of walking them through it, engaging them in the solution, so the rest of the group is left with the problem remaining unsolved.

> Problems solved:
> Cole learned to bug report.
> Cole attempted to explain it to the rest of us;
> it pretty much makes sense. He can walk us through it if we need to.
> Problems remaining unsolved:
> Haven't actually reported a bug yet; plan to find a bug and attempt to report it ASAP. This should be relatively easy with Cole's expertise available to us (EMG3, Group Meeting Form, 3/14/12).

The other two groups in this category met fewer times and the meetings were often focused on other agendas. For example LMG2 submitted a Group Meeting Form that showed that they discussed that they wanted to become 1337, or elite, at VIM, a text editor that they did not find useful. There is no evidence of any serious attempt at trying to find ways to contribute to the project in this Group Meeting Report.

> What was discussed:
> We had a nice discussion about how vim is the most ridiculous IDE'ish text editor and find it insane that people dedicate huge sums of time to become 1377 Hackers at it. We also decided that there aren't any good videos on YouTube for explaining triaging.

> Problems solved:
> We both concluded that VIM text editor is insane, and the people that use it are crazy. So we are going to try to get really good at it too.
>
> Problems remaining unsolved:
> We need to find some good instructions on how to get 1337 at VIM. (LMG2, Group Meeting Form, 3/16/12)

LMG2 had a hard time finding a focus within Ubuntu. They also had evidence that a group member learned how to triage bugs, however the information was given verbally, rather than walking through an example leaving other members with no actual experience, as was seen with EMG3.

> What was discussed:
> Connor talked about what he learned at the Sunday Ubuntu meeting
>
> Problems solved:
> Helped us figure out what we should do to contribute to Ubuntu bug triage.
>
> Problems remaining unsolved:
> Still not experts in bug triaging. (LMG2, Group Meeting Form, 4/4/12)

This left Zayne still unclear on how to triage bugs. The following example from his Final Reflection demonstrates his uncertainty about how to contribute to bug triage, even though he was told how. This suggested that walking someone through the process, as groups in the previous category did, leads to greater contributions “I browsed through many other bugs that I could not contribute to because they were outside of my area of knowledge” (Zayne, Final Reflection, Question 3).

LMG1-1 had some evidence of a relationship between this two-member section of the group, however it appeared as though they met infrequently, four times during the term. It is apparent from their documentation that only one of the meetings took place outside the class. Alex and Caleb worked on writing music for a theme their group was creating for the Ubuntu community. In their Group Meeting Forms they discussed the music for the theme they were attempting to write, and shared information from their mentor who said it was possible to do. Each group meeting started where the last meeting left off, but there was no evidence that either member of the group gathered information between meetings. There was no real solution to problems by the end of the meeting so the same problems persisted throughout all of the meetings.

The last two indicators of CoP mutual engagement detail the rapid flow of information and the concept that conversations and interactions are part of an ongoing process. I was looking for evidence of a group member acquiring information then disseminating it to their fellow group members in an effort to assist everyone in the group with making contributions. I also looked for signs of a continuing conversation that focused primarily on contributing to an area of Ubuntu, the practice the groups were to be focused on.

EMG3 showed a definite progression from week to week, following similar threads as they brought more information to the table, such as sharing the applications they found to automate testing. This example demonstrates this group’s continuing conversation about testing and the tools they need as well as figuring out the theory of testing.

What was discussed:
Ubuntu testing/ automation. What programs are good tools to use:
-The Ubuntu Friendly project - community of sharing validation testing of specific computer models tor the latest version of Ubuntu. - Uses the Checkbox tool.
-Desktop testing - Automation built on LDTP (Linux Desktop Testing Project) for many versions of Linux. This project aims to create a set of processes and code to make writing automated test scripts easier and more reusable.
-Ubuntu ISO Testing aimed at validating the stability of ISO installations. - Uses the KVM and libvirt.

Problems solved:
Found other people's testing results.
We discovered the theory behind testing and uploading results with the automated testing software.

Problems remaining unsolved:
Still need to actually try to see if we can get some results. We don't understand Ubuntu ISO testing project. (EMG3 Group Meeting Form, 2/27/12 & 2/29/12)

What was discussed:
Automated testing using mago.ubuntu
Problems solved: We were able to run the automated tests; however we kept getting errors because we were missing certain tools. Austin was having problems getting his virtual machine working again after having to reinstall Ubuntu.

Problems remaining unsolved:
Getting the remaining tools installed so that the test succeeds. (EMG3 Group Meeting Form, 3/6/12)

LMG1-1 had a continued focus on writing the music for a theme. However, in a Final Reflection there was evidence of the frustration one member felt about not being able download and successfully install the Gnome shell that was recommended by their mentor, “I looked up tons of documentation on using Ubuntu, attempting to install things I needed to work through our art asset creation which failed miserably with tons of errors at every corner” (Caleb, Final

Reflection, Question 2). So this group was working toward a goal that they did not achieve, which produced a high level of frustration.

In early group meetings EMG2 showed signs of focusing on application development; however they did not come to meetings with new ideas. There was progress over the first three meetings about the type of app they want to make, but there is no suggestion that any of them figured out how to create the app and contribute it to Ubuntu. Analysis of the data showed that this group did not remain focused on the purpose of the class during meetings, contributing to Ubuntu, for example in one meeting they discussed tests they had in other classes, “We discussed that this week was going to be full of tests, so it's going to be hard to find the time to contribute. Our applications progress is going more slowly then we presumed. Version control is still proving to be a hassle” (LMG2, Group Meeting Form, 3/2/12).

The analysis of the data from these three groups suggested the formation of relationships; however they did not completely fulfill the criterion of sharing and disseminating information or of having an ongoing conversation that signifies mutual engagement. The next characteristic of practice that was analyzed was joint enterprise, or how the history they have formed in meeting together and sharing ideas further their abilities to contribute to Ubuntu. I looked for evidence of a sense of belonging or group cohesion within these three groups and then the two indicators that were coded for were the ability to quickly set up a focused discussion of a problem and knowing what each member knew.

Analysis of the data from EMG3 suggested that members found information that was useful, and they shared it with their group members. The group received benefit from that by being more productive as shown in the example given above where Cole shared information on bug reporting with the group. This demonstrated a sign of cohesion among his group. There is additional evidence that the group focused on Mago, a testing framework designed to create automated test scripts, to work on ISO testing together. These are signs that the group was working together on the practice, seeking a way to contribute. This group also showed that they knew who knew what in their group when they turned to Cole for his expertise in bug reporting in the group meeting on 3/14/12.

Exploration of the data from LMG1-1 and LMG2 did not show indicators of cohesion or a sense of belonging. Joint enterprise is indicated when group members are able to solve problems quickly together and know who to turn to in order to get information. LMG1-1, as mentioned above, had no evidence that suggested that they gathered resources separately to share with each other. This lack of resources made quick problem solving difficult and with the lack of resources it was difficult for these students to turn to each other for information. Analysis of the data from LMG2 suggested that even when a group member did have information, the other members of the group did not turn to them for demonstration. This was shown above in the excerpt from a group meeting on 4/4/12, when they talked about Connor's knowledge if bug triage, but did not apply this in action.

The third characteristic of the joint effort to find ways to contribute is the presence of a community way of getting the work done based on the history of working together. Groups had indicators of this when they shared tools, in addition to those taught at the beginning of the course, as well as the development of communal resources. As with the higher category in the previous section, I also looked for examples of this history in shared humor, inside jokes, and a sense of ease and familiarity in their communication.

EMG3 developed a tool set for their group, using Dropbox to share files and links and Mago for automated testing. This group looked at several tools for automated testing, The Ubuntu Friendly project, Desktop Testing, and Ubuntu ISO Testing, but made the decision that Mago was best suited for their group. Additionally Cole brought knowledge of bug triaging to the group to help the others contribute.

EMG3 had signs of shared humor demonstrated in an ongoing joke within this group. One of their group members was often a few minutes late for class. He would text a long story about rescuing abandoned kittens. As the term progressed the story grew and got more involved. It also got funnier as time went on. The following example answered the question “What grade do you believe you deserve in this class and why?” by referring to this inside joke.

> However I was occasionally late to class when I was performing public services for kittens. I believe that my contributions to the animal kingdom served for the greater good, and if there were a grade higher than an A+, it would be necessary for me to receive that grade. (Liam, Final Reflection, Question 9)

LMG2 shared resource with each other to facilitate their goal of creating an app. Together they researched and found out how to create a Personal Package Archive (PPA) within the Launchpad site as shown by this excerpt from this Group Meeting Form, “Well we solved the problem of getting started on the project! We set up a team Launchpad account and submitted some test documents as a test.” (LMG2, Group Meeting Form, 3/2/12). They also all used the utility Quickly, which contained a template for creating an application in order to simplify the process of creating and integrating an app into Ubuntu. They demonstrated some inside humor with the name they chose for their PPA, used in one Group Meeting Form as an ending – “Team pink Zombie kittens over and out. >'.'<” (LMG2, Group Meeting Form, 2/17/12).

Analysis for the data from LMG1-1 did not show any signs of either inside humor or shared tools. They attempted to create a new theme for Ubuntu and there is evidence that they worked on the look and feel of the theme, but there is no evidence that they actually found tools to work on the implementation. The two members of this section of LMG1-1 were focused on creating music for this theme and they asked their mentor if it was possible to create the theme that had music as an element. Their mentor replied that they could create a theme with music; however it is not evident that they sought advice on how to package this theme.

The analysis of the data for these groups suggested that they had some indications of the formation of CoPs, however it was not as substantial as the indicators within the first category, leading me to conclude that a weak CoP had formed in these groups.

As with the previous category, I reviewed how students communicated with their mentors. I specifically looked for signs that students communicated with these members of the Ubuntu community in ways that enabled students to contribute to Ubuntu. I considered communication from students to meet this criterion when they asked questions that were specific and focused on contributions. I analyzed the data gathered from the emails students exchanged with their mentors to assess how students communicated and what questions they asked. I also analyzed the data from Midterm and Final Reflections to look at what students perceived to be helpful in finding ways to contribute. I looked for the perception of usefulness, if present, of their mentors, as well as the Global Jam and the live demo by Benjamin Kerensa. There was one student in this category, from LMG2, who attended the Global jam.

EMG3 members each had one email exchange with their mentor. Two of these exchanges elicited instructions from two different mentors and links to help the students find a way to begin testing and triaging bugs. The emails from the mentors also included the invitation to continue contact if the students had any more questions, however members of this group did not turn to their mentors with further questions. One of these students did find this brief exchange supportive of contributing. In his Midterm Reflection he reported – “My mentor, he/she gave me some instructions that I could follow and re-assured me that I could do Ubuntu contributions very easily” (Liam, Midterm Reflection). However, by the final this student did not refer to his mentor as a significant source of help for engagement, but he did reference Benjamin Kerensa in this excerpt:

> There were several sources that helped me become engaged. I would say that the biggest ones were this:
> - My professor
> -Ben, the guy who showed us triaging in class
> -The websites, once I had the confidence to thoroughly navigate them (Ubuntu and Launchpad) (Liam, Final Reflection, Question 4)

The second student who received information in the form of links to helpful resources for testing from his mentor did not refer to him in either of the Reflections nor is he referenced in any of the Group Meeting Forms.

The other two members of EMG3 each had one exchange with their mentors, but they exchanged introductory information only, no specific questions were asked to assist contributions. Cole did send a second email to his mentor asking a specific question, “Hello, I was wondering what I would have to do in order to get involved with testing in the Ubuntu community?" (Cole, Personal Email, 2/8/12), but never heard back from his mentor. Austin, the second student referred to here, did not use his mentor as a source of information, but he did receive information that helped him contribute to bug triaging from the Live Demo as he points out here - "Another great resource was the class demonstration. When it came down to bug triaging I think that seeing someone else do it was the most helpful." (Austin, Final Reflection, Question 4).

Both of the students in LMG1-1 used the same mentor. They each exchanged two emails with him. Caleb shared the information he received from his mentor with his partner that his mentor believed they could create the theme they were working on - "okay, so the mentor dude told me that it WAS possible to make a theme." (LMG1-1, Group Meeting Form, 3/1/12). The questions Caleb asked his mentor were specific and elicited resources on two different types of themes, desktop themes or Gnome Shell themes, to pursue from his mentor. In the second email to his mentor there is confusion as to which type of theme Caleb was pursuing. There was a significant period of time that elapsed between the two emails, the first email exchange took place between 2/23/12 and 2/26/12 and the second email was sent 3/9/12 and received a reply on 3/12/12, during the last week of the term. There is no evidence of a follow up. Caleb wrote about this lack of follow through on his part, pointing out that he would have benefitted from communicating with his mentor more frequently. "Probably the thing that would have led to the greatest return, had it been pursued heavily, would have been the usage of the mentor that was provided.” (Caleb, Final Reflection, Question 4)

The second student in LMG1-1 had the same pattern of email exchange as his partner, the first correspondence took place on 2/24/12 and the second email was sent on 3/2/12, but a reply was not received until 3/16/12. Alex asked his mentor broader questions about the Ubuntu community in the first communication – “I was just wondering what your experiences are with the whole Ubuntu project and what you have contributed to the project.” (Alex, Personal Email, 2/24/12). This question elicited a response detailing his mentor’s roles with Ubuntu. However the first email Alex received back from his mentor ended with “There's probably a lot I'm missing out as to what I've contributed but if you ask me anything specific I can help you out.” (Alex, Personal Email, 2/24/12). In the next email, sent almost two weeks later, Alex acknowledged the resources sent to his group mate, but did not frame a specific question about contribution. Alex did point to his mentor, along with his group, as a source of help in finding ways to contribute in the Final Reflection – “Using my mentor and talking to others around me. I didn't know how to start my project but definitely got help in doing it.” (Alex Final Reflection, Question 4).

The final group in this category, LMG2, all exchanged two emails with their mentors and asked specific questions to elicit guidance and resources. These emails were all exchanged at the beginning of the time this group got their mentor assignments, but seemed to end after the first two weeks. Tyler emailed his mentor, received some resources that he followed up on, but when he sent his second email, asking a specific question about branching the LibreOffice project, he received no response. This is lack of response is mentioned as a barrier to contributing in Tyler's Final Reflection – "I also sent a few emails to my mentor, he stopped responding though" (Tyler, Final Reflection, Question 2). Similarly Zayne emailed his mentor and asked specific questions about using Quickly to develop an Ubuntu app. His mentor emailed back with an assignment to help him find information. Zayne responds to this email; however in this response he did not ask any further questions. There is no evidence that there was further communication. Zayne did express an issue with the timing of his introduction to a mentor, feeling that he would have accomplished more if he had access to him sooner– "I felt that the mentors were helpful; however I felt if I had access to one earlier I would have been able to accomplish more." (Zayne, Final Reflection, Question 4).

The final student in LMG2 exchanged two emails with his mentor, and also attended the Global Jam. Connor's first email to his mentor was explicit in looking for resources to help him contribute to bug triaging.

> We have been trying to understand bug triage as of late. As far as I understand the process of bug triage is choosing a bug that we feel we can possibly recreate. Attempting to recreate the bug and assigning the bug accordingly if it is found, possibly making a comment. This is what I feel that it is based on the documentation. Maybe you could better explain what the process is. (Connor, Personal Email, 2/24/12)

From this question Connor received a response fully detailing the BugSquad and the path of triaging bugs. Connor's mentor was Benjamin Kerensa so during this exchange Benjamin told him that he will be giving a demo to the class the following Monday. Connor replied that he was looking forward to the demo and that he was also going to the Global Jam. Connor expressed an experience similar to the other students who attended the Global Jam, that the hands on, step-by-step demonstration with an experienced community member gave him the skills to contribute.

> The Ubuntu Oregon was the most helpful thing, since the presenter walked us through every step of Bug triage. He even found some bugs on the spot and had us help him fix them.
>
> -went to Ubuntu Oregon meeting where I learned how to completely bug triage (Connor, Final Reflection, Questions 2 and 4)

Students in the groups in this category emailed their mentors, and some students asked questions designed to elicit specific information; however analysis of the data implied that they did not follow through on resources shared by their mentors. This analysis also suggested that these students did not fully use the time with their mentors, either ending communication well before the end of the term, or by not pursuing answers in a timely manner, letting too much time elapse before sending a second email.

This section considered the first research question, how do students view the importance of group members and mentors and their effectiveness in promoting engagement in OSS? Analysis of the data from this category suggested these students placed less emphasis on group members and mentors then the High Level group did, rather there was more focus on documentation, wikis and YouTube videos. The analysis indicated this category of students could not find a strong way to engage with one another or their mentors. The reason for this was unclear.

Contributions to Ubuntu

The objective of this class was to contribute to the Ubuntu project. All students had the same resources assigned from the curriculum. This included a mentor along with suggestions to participate in documentation and bug triage, which are considered two of the most accessible places to contribute while still being able to observe the community and how it works. The framework supports the assertion that the collaborative pedagogy used in the design of this course promotes increased and deeper contributions to the Ubuntu project in the students in the High Level category. From this assertion it is reasonable to claim that we would expect to find fewer contributions from this category where students had lower levels of collaboration.

Two students in this category scored moderate on contributions to the Ubuntu community while the remaining seven scored low. Data was collected in the same manner as it was for the students in the previous category and I used the same ranking of simple, moderately simple, moderately difficult and difficult to determine the degree of difficulty. I also followed the same procedure using the same breakdown of contributions into the given ranks. Figure 4.4 shows the contributions of this category by quantity and degree of difficulty.

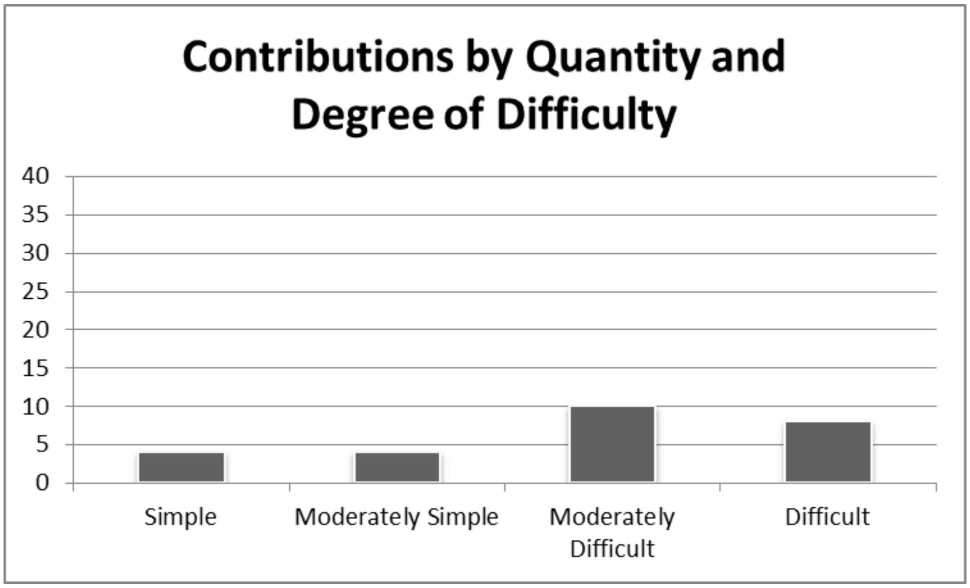

Figure 4.3: Contributions by the Moderate Category

The analysis of the data represented in this graph showed that the students in this category made fewer contributions then the students in category one. They did, however, maintain contributions over a range of difficulty levels. Addressing question three of this research, “Will the course design produce students who engage with the OSS community at a percentage greater than 10%?”, showed that all students did contribute, however only two made contributions that ranked above the low level.

Low Indication of CoP and Mentor Engagement

Group Characteristics

The final category contains the students who belonged to groups that had very low or non-existent evidence of the formation of a CoP. The category has eight students, representing 30% of the total number of student in the class. The demographic breakdown of these students can be found in Table 4.3.

Gender		Race		Age Group (1-4)				Length of time programming (in years)			
Male	Female	White	Non-White	18–25	26–35	36–45	Over 45	0-1	2-4	4-8	Over 8
7 (28%)	1 (20%)	7 (29%)	1 (17%)	5 (24%)	2 (29%)	1 (50%)	0 (0%)	1 (20%)	4 (23%)	2 (40%)	1 (33%)

Table 4.4 Demographic Breakdown of the Low Group

Included in this category are three students who missed a significant number of classes, one originally assigned to EMG1, one from EMG2 and one from LMG2. These students showed little to no involvement in their groups and in the case of the student from LMG2 showed no evidence of participation in any part of the course.

This category, based on the criterion used to rank all other student in the course, would also contain Charlotte, who took this course with no background in CS. She asked me at the beginning of the term if she could remain in the course and I told her she could. I felt that she may be able to provide a unique perspective as an outsider to CS. Given her unique situation her data is not included in the analysis of this category, nor is it in the demographic breakdown. This accounts for the columns that do not add up between category demographic tables. I discuss her case, along with one other student, in the next section. It should be noted that she was in the other half of LMG1, referred to as LMG1-2. The other member of LMG1-2 had no evidence of participating in any portion of this class, including the creation of the keys and signing of the code of conduct that took place the first two weeks of the course. The second member of LMG1-2 is included in the analysis of this category; however there is nothing that can be said about her case specifically because her lack of participation produced neither data nor artifacts.

In the analysis of the data from this category I first considered mutual engagement, looking at all four indicators; sustained mutual engagement (the number of times the group held meetings), shared ways of engaging (evidence of students sharing information they find or receive from others with the group), rapid flow of information (evidence of a group member acquiring information then disseminating it to their fellow group members in an effort to get everyone engaged), and ongoing conversations (a continuation of work extending from one group meeting to the next).

EMG1 originally had three group members, however Brandon ended up working with EMG4, as noted above in the High category. The remaining two members of the group showed no evidence of ever meeting; in fact the evidence is to the contrary. Dylan submitted email to me on two occasions reporting that he had emailed his group attempting to set up group meetings, but could get no response from them. This lack of participation from his group left Dylan with no sense of collaboration; he tried to navigate the Ubuntu project alone. This solitary attempt led Dylan to stop trying to contribute, as he pointed out in his Final Reflection:

> My group wouldn't meet and I had a hard time concentrating because I wanted so bad to be at home with my family. These things led me to kind of bow out of the class for a while. Though I showed up to class and lab, I did little to participate. I felt a little lost without the help of my group and I did not engage further with my group because I felt I was not liked. (Dylan, Final Reflection, Question 9)

There is one IRC chat log from the first week with groups where this group met with their mentor for the first time. Dylan participated in the conversation, asking questions and noting resources. During this one group meeting the final member of EMG1 only speaks once, to identify that he is present on the IRC channel.

EMG2 had two members. One member was a student who failed to show up for class a 50% of time. This group did meet three times, however there is no indication that they discussed resources or shared information. The first and second meetings were both about creating an app or a game for Ubuntu, but they did not discuss how to do this or where they could find information. In the third meeting they discussed bug triage to increase their karma, but it was not apparent what they were doing to achieve this goal. Although there was interaction between these two students I did not find indications that there was mutual engagement as operationalized by this study.

EMG5 turned in three transcripts of group meetings, but there was no discussion of resources or sharing of ideas on how to contribute. There was one instance where a group member tried to help by suggesting a tutorial for Python, but the suggestion was not acknowledged by the group member who was struggling with learning how to program in Python. Out of the transcripts of the three meetings, this exchange about Python was the only conversation that took place about the class itself. The rest of the conversations revolved around work for another class. The section about Python was brief and consisted mainly of one group member complaining that he could not figure out how to program in Python. This group did not show any indication that there was mutual engagement in pursuing contributions to Ubuntu.

In looking for the presence of a joint enterprise I sought evidence of cohesion or common ground. I analyzed the data for any evidence of groups in this category focusing on finding ways to contribute to Ubuntu and I also looked for any indication that these groups had found tools they shared, or even exchanged information about. It is obvious from the details above that EMG1 would have no indicators of either of the remaining characteristics because they had no interaction. Of the remaining two groups, EMG2 and EMG5, neither had indications of the formation of a cohesive group or of a shared repertoire.

Contact with mentors in this category was almost non-existent. There were four students who emailed their mentors. Two of them were from EMG2. The first student emailed his mentor and asked basic questions about how his mentor got involved in Ubuntu and about whether he considered participation a hobby or a job. His mentored replied at length, and ended by asking questions of Noah to determine how he could best direct him. In the next email Noah asked more superficial questions to which his mentor replied and then ended with "Hmmmm . . . I've asked you some questions in my previous email. Perhaps you could respond to those items as a means of getting started. Knowing that info will help me to help you." (Noah, Personal Email, 2/15/12). There was no further contact with his mentor. Noah did, however, attend the Global Jam. His reaction to the Jam mirrored the reaction of the other participants. He found the hands-on demonstration by the Ubuntu veteran to be helpful in furthering his ability to contribute – "Probably the Ubuntu Global Jam. Being shown literally a step by step hand holding session was insanely helpful." (Noah, Final Reflection, Question 4).

The second student from EMG2 exchanged two emails with his mentor. There was a long time lapse in-between, the first exchange on 2/8/12 and the second on 3/3/12. In the first email Toby asks a generic question about documents to become involved, but does not specify what he wants to become involved in. His mentor asks him to refine his question to be more specific. In the second email Toby asked a more specific question and received a detailed answer pointing him to resources for game development. There was no more email communication and there was also no evidence that Toby looked into the resources given to him by his mentor.

One student from EMG5 exchanged two emails with his mentor. The first email was a basic introduction that ended with him asking about his mentor's area of expertise. His mentor replied and asked the question "So what are you interested in?" The next email Jordan sent did not answer this question, but was an expression of frustration:

> Just following up. I don't have a great question to ask. I'm trying to get a feel for Python, but so far it is causing me to want to kill it. Can you give me a better suggestion than the standard tutorials which are too long and drawn out? (Jordan, Personal Email, 2/15/12)

There is no answer to this question from his mentor. Jordan perceived his mentor to be unhelpful.

> To be honest I mostly just tried to understand the programming language. I was told that Python was very intuitive programming language; however I found it to be confusing. Most of my problem is probably due to the fact that I am used to strongly written programming languages. My mentor was not very helpful when it came to answering questions so not much was accomplished on my part. Actually accomplished tasks where reading a ton of wiki's, some tutorials, and asking many questions. (Jordan, Final Reflection, Question 2)

Jordan also seemed to be frustrated with the Ubuntu community as a whole based on the evidence from his Final Reflection answering the question – What was the biggest hurdle to engagement?

> My biggest hurdle was and is getting past the elitist attitude that I keep running into. I am a noob at open source and I know that therefore it does

> not help when someone tells me to R.T.F.M. when I don't even know where to find it. I realize that this is not, or should not be the norm but it is my experience. (Jordan, Final Reflection, Question 5)

The final student in this category had no contact with his group members in EMG1, and he only exchanged one email with his mentor before he reportedly stalled in his attempts to participate. However Dylan was reenergized by the live demo by Benjamin Kerensa and this provided the motivation to contribute. In Dylan's own words when asked for the greatest source of help in becoming engaged, he answered "BENJAMIN KERENSA. Actually seeing bug triage in action was the best source of information possible". (Dylan, Final Reflection, Question 4). In Dylan's quote on page 49 he discussed the results of not having a group to lean on, but in the following quote he pointed out that having input from an experienced community member supported his ability to contribute. "After our conference with Ben, I got excited. Finally it all made sense and I wanted to participate. (Dylan, Final Reflection, Question 9)

In the analysis of the data from these students there was support for the assertion that participation with mentors by these students lacked understanding of the purpose of their mentor. Even when asked direct questions by their mentors to help them focus on ways to gather resources these students did not respond to the questions.

Contributions to Ubuntu

As with the previous two categories I present in this section a view of the contributions of this category. Each student's contributions were ranked in the same manner as the previous categories' contributions. It should be noted that the contributions from this category came from five of the eight students represented here. Of those five students Dylan, from EMG1, accounts for fifteen out of the total of twenty four contributions. Figure 4.5 show the quantity of contributions by degree of difficulty.

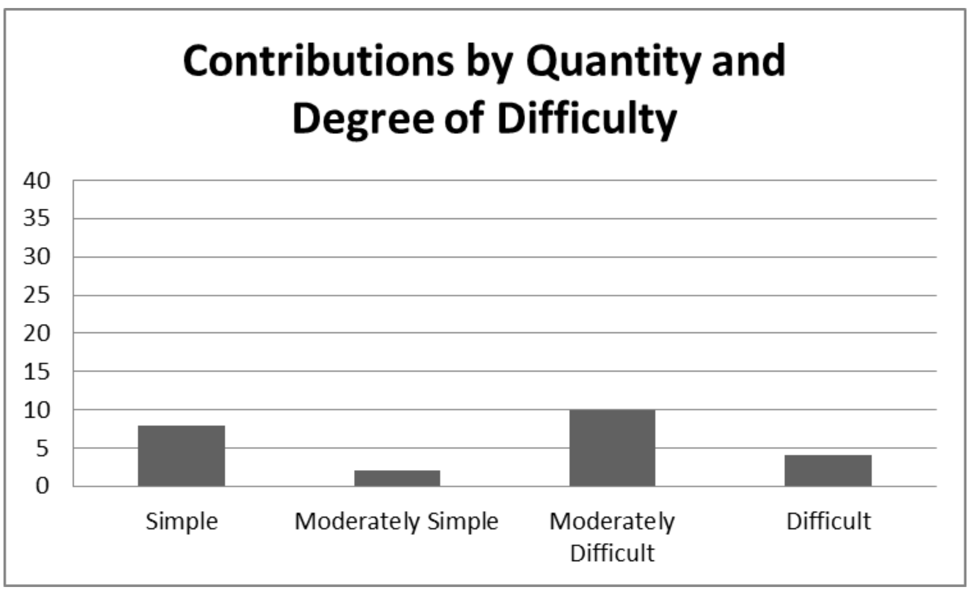

Figure 4.4: Contributions by Low Category

Analysis of the data from this category suggested that the lack of formation of a CoP and lack of engagement with a mentor inhibited contribution levels.

Two Unique Cases

There were two students in this class that were excluded from the analysis of the model. The first student addressed below is Charlotte, the non CS student mentioned in the Low level category. The second student, who is discussed after Charlotte in this section, signed the consent forms for inclusion in the study and filled out the questionnaire and the first self-efficacy survey, however he failed to complete the reflections or the post self-efficacy survey removing him from eligibility for this research. He does however present an interesting case that I address below.

The first day of class Charlotte asked if it would be ok for her to remain enrolled in the course because she had no CS background. I decided it would add an interesting view from someone outside the culture of computer science. Charlotte participated without hesitation. When other students were setting up their Launchpad accounts she worked alongside them. She used the command line in the first two weeks to complete the set up assignments, outlined in chapter three, which included creating an ssh key and a pgp key as well as completing the version control assignment. She worked collaboratively with her group to set up the development environment that was required of all students and they showed her how work in the environment we set up. Charlotte originally did not believe she could accomplish any of the tasks outlined, however, by the middle of the term analysis of the data suggested that she found ways to contribute to her group and she felt quite accomplished with the skills she had acquired. The Midterm Reflection asked students to list the three greatest sources of help in becoming engaged. Charlotte found that having others to work with was helpful and she valued the input she got from her group, LMG1, before they split.

> Peers and Prof - The first most useful thing that I have found to engage me with the Ubuntu community is my peers. Not being learned in CS I find just having them explain things to me or watching them go through steps has been incredibly helpful. (Charlotte, Midterm Reflection)

Charlotte spoke to me before the class ended and I asked her what her plans were. She said that she was really happy she took this class and, although she did not want to become a computer scientist and learn to program, she would definitely use open source software and she would report any bugs she came across because she felt she had learned how to do that in this class. She also said that she would be spreading the word about OSS to people she knows (Personal Conversation, Journal Entry, 3/16/12). This form of contribution is referred to as being an evangelist for OSS and is explicitly included as contributing to the Ubuntu community on their contribution page:

> Anyone can help shape and improve Ubuntu. This document will introduce you to the most common ways that you can contribute to Ubuntu: by using Ubuntu in your everyday life and recommending it to others, by helping other users, by translating programs or documents to your native language, by testing the software and reporting issues, by creating artwork or writing documentation, by fixing software issues, writing new software or keeping others' software up to date (“ContributeToUbuntu - Ubuntu Wiki,” n.d.).

When Charlotte was asked if she was to participate in another OSS community what would be the steps she would take to join, she felt confident she could navigate the community. She even came up with unique ideas to promote participation in OSS for people who may be intimidated. Analysis of her response suggested that she considered herself an insider in the OSS community so that she could provide support for people who may be intimidated or not know where to start.

> Pretty sure I would jump right in and again try to learn as much as possible. One big change I would for sure go also to some of the big OSS community meetings and converse with people and learn new things. I may also start a blog or site as a kind of invite/support group for people interested/passionate about OSS but may be intimidated/not know where to start. (Charlotte, Final Reflection, Question 6)

Examining the data provided by Charlotte's participation provided a unique view of the framework supporting the development of this course suggesting the collaborative environment is productive to students with no CS background. The goal of this course is to get students to contribute to an OSS community and this student with no CS background jumped in, got involved, and plans to continue contributing as an evangelist.

The second student that needs to be discussed in this section is Richard. Richard initially was assigned to EMG5. He failed to show up for class a 60% of the time and there is little evidence that he participated with his group. There is one example of his attendance of a group meeting where he tries to share a resource on Python programming with a fellow student, but the suggestion seems to be ignored.

As a solo participant in Ubuntu, Richard was highly successful. He had several contributions to the community some at a high degree of difficulty. Richard also had previous experience and fits in the demographic of the average developer in OSS. Richard is an example that there is definitely a place for students who prefer solo work and that for some it can be successful; however there is strong support in this research, and the framework that supports it, that a wider and more diverse group is served by an educational model that is collaborative in nature.

Self-Efficacy

The Midterm and Final Reflection data was coded for evidence of the sources of self-efficacy. Within this written work I looked for student reports of having performed a task, watching someone else demonstrate or successfully complete a task, having someone tell them they could complete a task, or evidence of the student perceiving their environment to be supportive –signs of positive sources of self-efficacy. I also looked for the converse of these sources; trying to perform a task and not succeeding, not having the experience of seeing the live demo, going to Global jam or participating with a group that someone demonstrated the skills to complete a task, not being in situations where someone could tell them they were capable (the case with high absenteeism), or evidence that the student found the environment in their group or the Ubuntu community negative, all negative sources of self-efficacy.

Once I had coded the data I looked at each student for signs of both positive and negative sources of self-efficacy. I recorded the number of instances reported in each category, performance/mastery, vicarious, verbal, and physiological, and balanced positives and negatives, so if someone reported a negative experience it was marked as a negative source, but if they reported that something happened that countered the negative experience with a positive one, the original experience was considered mitigated by the positive experience. I looked for students reporting a change that affected the way they viewed the Ubuntu project, for instance Tom said “It took me awhile to get involved because it felt like there was so much to do. But once I started it was easy to find ways of getting involved” (Tom, Final Reflection, question 2). Amelia said “At first, I was afraid I would do something "wrong". After experienced members clearly explained what to do and how, I wasn't afraid anymore” (Amelia, Final Reflection, question 5). The outcomes from this analysis are represented in Figure

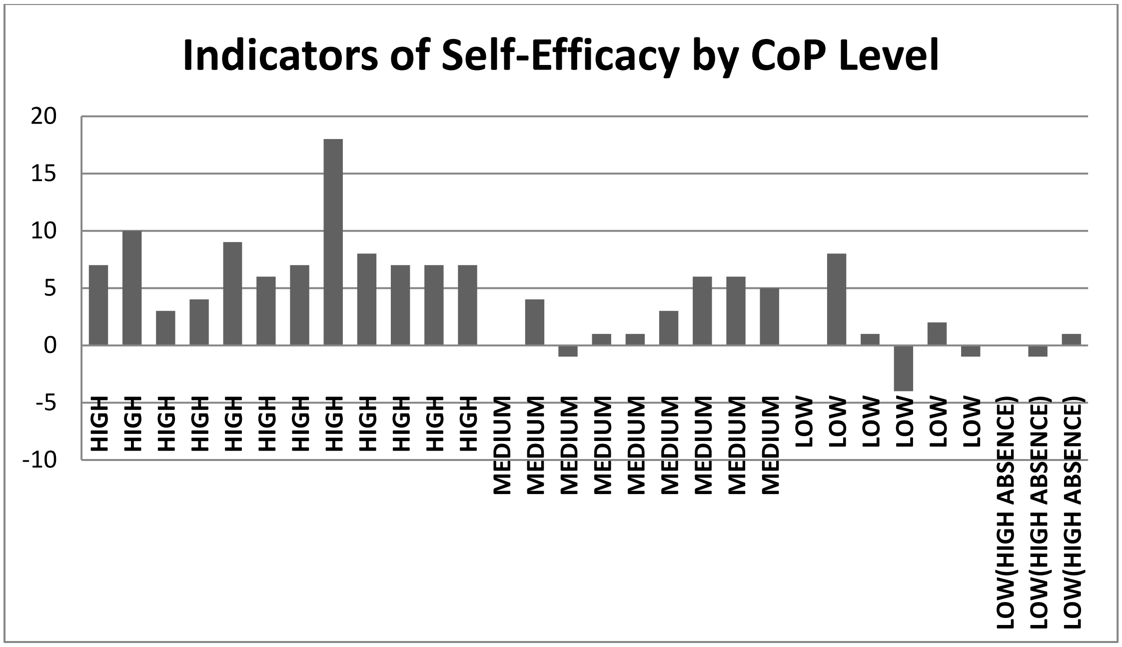

Figure 4.5: Indicators of Self-Efficacy by CoP Level

Analysis of the data suggested that there is an association between the groups that had high indicators of the formation of a CoP coupled with high mentor engagement also had higher instances of positive sources of self-efficacy. EMG4, the group who had the highest number of indicators of the formation of a CoP, and the highest evidence of mentor engagement also had the greatest number of reported positive sources of self-efficacy and were the only group who had no reported sources of negative self-efficacy. This group also reported that they had overcome perceived barriers. Josh reports his initial barrier - “I was unsure on many things and I did not want to make a change that could possibly be wrong or not required” (Josh, Final Reflection, question 4) followed by this change in his view point “Listening and watching them go step by step on bug procedures reassured me that what I was doing was correct." (Josh, Final Reflection, question 4).

When analyzing the data from the groups who scored the lowest on indicators of a CoP and mentor engagement I saw that these groups had less reported positive sources of self-efficacy and a greater number of reported sources of negative self-efficacy. Self-efficacy has a direct effect on how much and how long a person will persist in the face of obstacles (Bandura, 1977) so it is reasonable to say that the presence of negative sources of self- efficacy increases the likelihood that a student will have the perception that barriers are insurmountable and give up on trying to contribute sooner than a student who has positive sources of self-efficacy. The three groups who had the lowest levels of CoP and mentor interaction reported trying to accomplish tasks and repeatedly running into barriers. These three groups also reported barriers that they did not find ways to overcome. The group with the highest number of reported negative sources of self-efficacy, EMG5, also had the greatest number of reported barriers that were out of control of the student themselves. Jordan reported that "My biggest hurdle was and is getting past the elitist attitude that I keep running into. I am a noob at open source and I know that therefore it does not help when someone tells me to R.T.F.M. when I don't even know where to find it" (Jordan, Final Reflection, question 5).

The one group who had an interesting mix of reported sources of self-efficacy was EMG3. The group had moderate indicators of the formation of a CoP, but low mentor engagement. When looking at the reported sources for self-efficacy they all reported positive sources of self-efficacy, however three out of the four also reported negative sources of self-efficacy in the performance area. However, unlike the groups who did not exhibit indicators of a CoP, this group had no reported negative sources of self-efficacy within the physiological realm. This suggested that with the moderate formation of a CoP this group had fewer sources of stress then the groups who did not form a CoP and so did not have the perception of support by their group.

The coding and analysis of the Final Reflection was corroborated by the pre and post self-efficacy survey. Figure 4.6 shows a box plot of the change in self-efficacy by group number. This representation of the self-efficacy data mirrors the reported data; EMG5 shows a decrease in self-efficacy where EMG4 shows an increase. The big jump in self-efficacy in LMG1 is a reflection of Charlotte, the student with no CS background. When she took the pretest she had no source of self-efficacy, however, in line with her reports of figuring things out with the help of the instructor and her group, her self-efficacy jumped significantly.

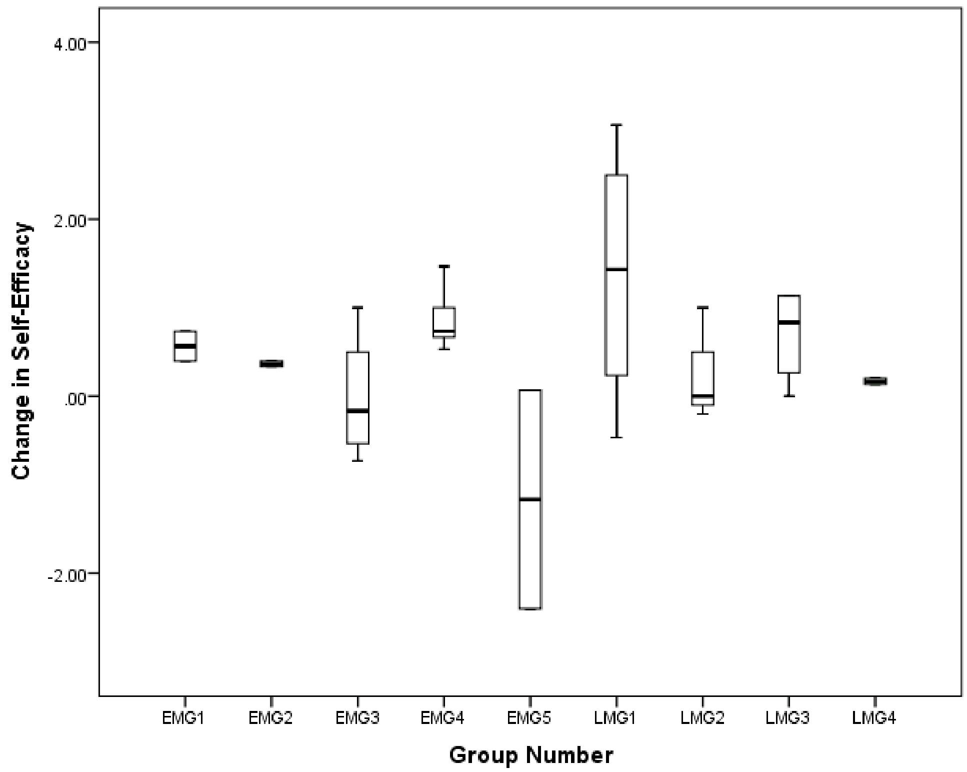

Figure 4.6: Boxplot of Change in Self-Efficacy Score Pre and Post

After looking at the data graphically I considered statistical testing. Given the size of the sample, n= 30, it was appropriate to perform non-parametric tests on the data to determine the significance of the variables studied. I first ran the Wilcoxon Signed Rank test to determine if there was a statistically significant difference between the pre and post score on the self-efficacy survey. The results of this test show I can reject the null hypothesis and state that there is a difference between the pre and post self-efficacy test results (Z = -2.998, *P* = .003). Looking at the descriptive statistics I can say that there was an increase in self-efficacy from the beginning of the term to the end of the term.

Next I looked for a difference between the independent groups to determine the significance of my binary variables of sex, and race using the Mann-Whitney U test and my non-binary variables of group, level of mentor engagement, and indicators of the formation of a CoP using the Kruskal-Wallis analysis of ranks. Of the variables the tests showed that only sex is significant ($p < .05$). The variable of mentor engagement was shown to be not significant ($p < .08$).

Summary of Self-Efficacy

In analyzing the data for self-efficacy I can say that there was a positive change in self-efficacy overall statistically within the class. What is still not clear is what affected that change in self-efficacy. Statistics did not support the qualitative data that points to the importance of the formation of a CoP, mentorship and the live, hands on demos. However, the sample sizes used in the statistical analysis were small making it difficult to draw statistical conclusions about the causes. It is possible that these variables did not go far enough in explaining the change in self-efficacy or that there were forces at work not measured by this model.

Intent to Continue

Finally I analyzed the data that was collected as part of the Final Reflection where students self-reported their intention to continue or not to continue participation in either Ubuntu or OSS in general. One of the goals of this class was to help create a pool of developers to support the OSS community. Although participation and exposure to OSS is valuable to the OSS community as a whole, continued participation would help create the developers that would be able to support projects like Ubuntu into the future, replacing the developers that leave the project.

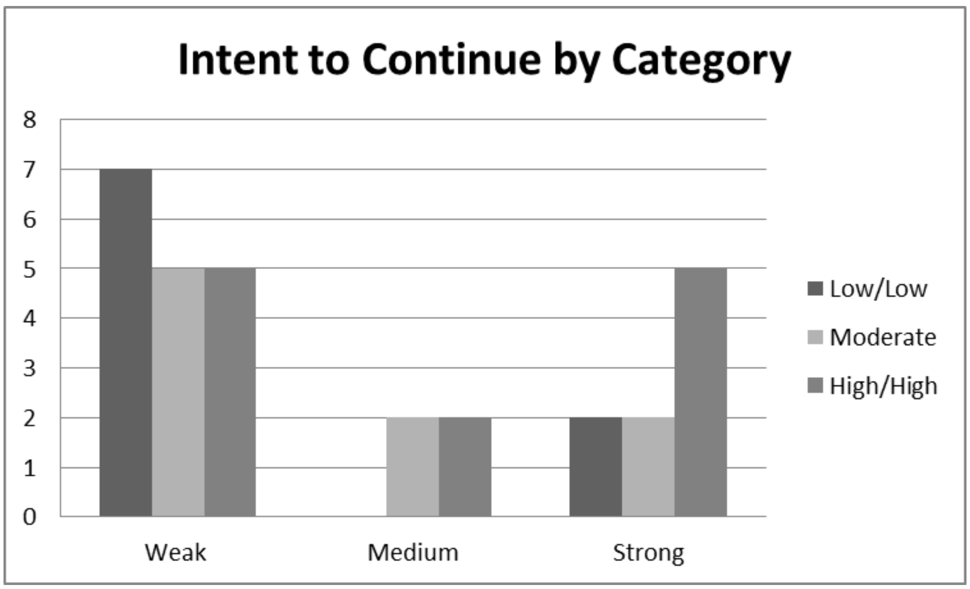

Figure 4.7: Intent to Continue Participating in OSS

Strong Intention to Continue

Out of 30 students 30% strongly indicated that they would continue in OSS. Out of the nine students that reported strong indications of continuing to participate, four had already found a new project and began to explore participating in that project. Another student in this category of strong intention of continuing is a web developer and, although he did not plan to continue with

Ubuntu he did express motivation that is common in OSS development, the desire to create or change an application that is used in the participants everyday life. In his words, “I am very interested in getting involved more in these sorts of applications to make improvements that will affect the ease of my everyday life” (Niall, Final Reflection, question 8). The four remaining students in this category planned to stay with Ubuntu. Two of these four students cited the importance of the community of Ubuntu in their desire to stay with the project. One of these students was Charlotte. Her plans to participate in OSS are given in the section Two Unique Cases above.

Analyses of the data pointed out that five out of nine of these students were in groups that had high levels of CoP indicators. This suggests that, although not all students who were in high CoP groups indicated that they would continue in OSS, the presence of a CoP was a factor for those intending to continue participating. In addition to the CoP factor there was also evidence that suggests that the number of years a student had programming also affects intent. 100% of the students with 4 or more years of programming and 40% of those with 3 years of programming strongly indicated the intent to continue. This suggests that maturity as a programmer may be a factor.

The number of students in the class that were female and/or non-white was small, five female and six non-white. Although the sample size is too small to say anything with statistical certainty, two of the female students, one who was non-white and one non-white male were in the category that ranked strong in intent to continue participation.

In the analysis of the data on intention to persist in OSS one case merited a closer look. A student who came from a group that had no indicators of a CoP and he had no communication with his mentor was rated as having a strong intention to continue, having already determined the OSS project he wanted to contribute to. He had one of the primary motivators that drive a large number of contributors to OSS, working with an application he already uses. He also fit the demographic of an OSS developer, so it is possible that he would have joined OSS development without the aid of a collaborative environment. He indicated in his Final Reflection that he has located a project that fits with his interests:

> I'd kind of like to get involved in a project called Mudlet. It's an Open Source telnet client for playing MUDS. The project is exceedingly small though and I hate their documentation (and its numerous gaps). I had a difficult time motivating myself to work on a large project with big areas I was unfamiliar with, so I was going to try contributing to a smaller project that I use extensively. (Parker, Final Reflection, Question 7)

Moderate Intention to Continue

Four students, 13% of the class, reported the intention to continue with OSS, listing Ubuntu specifically. One of these students wanted to finish the project they started during the term. The other three expressed a desire to continue, with two students saying they wanted to become more comfortable with Ubuntu before branching out.

What differentiated these students from the students coded as having a strong desire to continue was the wording they used. Three out of four of these students express reservations using phrases like “I am panning on staying involved in the community as long as I can.” (Tom, Final Reflection, question 7), “Honestly I think will still attempt to some degree” (Austin, Final Reflection, question 7), and “I probably will look into other OSS projects and see how I can participate, but for now I will hang out with Ubuntu until I feel a little more confident in working with open source” (Josh, Final Reflection, question 8).

Demographically this category had no females, one non-white student and less programming experience then the group that exhibited strong intent to continue. Two of the four students were in the High level category for CoP and mentor engagement and the other two were in the Moderate level.

Weak or No Intention to Continue

Seventeen out of thirty students had weak or no intention to continue with OSS. Of the seventeen students five are from groups that have high CoP and high mentor engagement scores and one student who had moderate CoP and low mentor engagement. Five of these students listed no time to participate as their reason for not continuing. Four out of five of these students who lacked time were graduating seniors going into the job market so it stands to reason that they would be concerned about time pressures in the near future.

Three students who were ranked as having weak or no intention of continuing with OSS come from LMG2, a group that scored at a mid-range for both indicators of CoP and mentor engagement. Analysis of the data from these students pointed to a lack of concrete statements about intent to continue participation in OSS causing them to be ranked as having weak intent. Two of the students say they might look into another OSS project and the third student says he is busy on another project.

The final group of students was seven students who came from groups that had few to no indicators of the formation of a CoP and little to no engagement with their mentors. Of these seven students six stated they had no intention of continuing to participate in Ubuntu or any other OSS project. The last student simply said he did not have the time.

Demographically this group had 60% of the women in the class and 57% of the non-white students, one student fell in both categories. This category also had 50% of the students who had one year of programming, 70% who had two years, and 40% who had three years. These percentages also supported the idea that maturity as a programmer may play a part in this decision.

The question I asked was “Do students anticipate continued participation in OSS projects?” Analysis of the data showed that 43% have a moderate to strong intention to do so. This is an encouraging sign suggesting that this course model is effective in promoting continued participation in OSS.

Summary

Analysis of the data from my research suggests that strong indicators of the formation of a CoP along with participation with mentors, participating in the Global Jam and attending the live demo supported engagement in the Ubuntu project. This is evidenced by the difference in contributions between the three categories (see Figure 4.8).

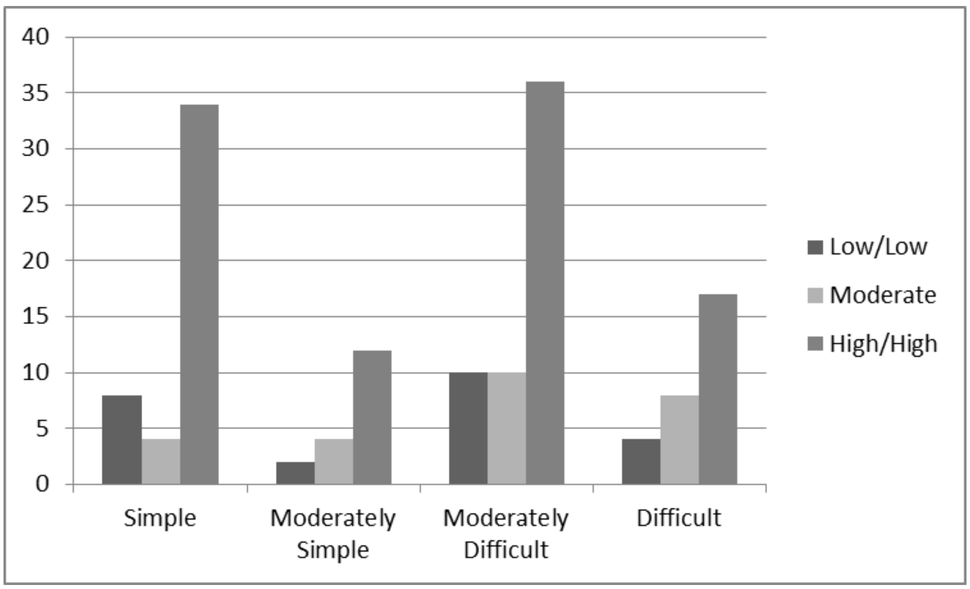

Figure 4.8: Comparison of Contributions by Category

Additionally analysis of the data supported the assertion that the course design would produce students who engage in the Ubuntu project at a percentage greater than 10%. In fact the results of this study showed that 25 out of 30 students, 83%, made some contribution to the project. Out of those 25 students 16, 53% of the total number of students, made a significant number of contributions ranging in difficulty from simple to difficult.

Statistical analysis showed there was an increase of self-efficacy overall. The small sample size in this study made it difficult to attribute the increase in self-efficacy to any of the attributes of the study. However, it is apparent from the box plot in Figure 4.6 that there is a decrease in self-efficacy in some groups. Although it is not possible to draw any conclusions from this decrease this does raise more questions. These questions point to the need for further research into the self-efficacy of students as it is associated with the collaborative design using a CoP.

Finally, 43% of the students who took this course show moderate to strong indication that they will continue participating in OSS. There were 5 students, 17% of the total student population, who specifically detailed either what they planned to do within the Ubuntu project or had already found other OSS projects that they were interested in.

Analysis of the data detailed in this section supported the use of this course design to assist in the creation of a much needed pool of OSS developers. This analysis suggested that this paradigm supports participation and contributions from underrepresented groups, as well as an overall success in contribution levels of a majority of the participants in the class.

Chapter 5 : Discussion

The first section of this chapter provides a brief over view of the problem statement and the purpose of this study. The second section is broken into three subsections in order to discuss the four questions asked by this research. The first two questions are addressed together as the analysis of the data revealed an association between the concepts represented in these questions. Each of these subsections begins with a summary of the main findings of each question, followed by a discussion of these results, their implications for OSS community, and recommendations for further research. This discussion is followed by a third section on the limitations of this research. The final section provides concluding remarks.

Introduction

The OSS community has a lack of diversity that is considered alarming to many who participate in this segment of software development and has been the focus of a great deal of discussion within the community (Byfield, 2009; Levesque & Wilson, 2004; Robert, 2009). It raises the question as to the characteristics of this community that results in this state of affairs. This lack of diversity is recognized as a detriment to overall innovation within the OSS community by denying projects the skills, experiences and viewpoint of those who are not represented. Additionally this lack of participation denies these underrepresented groups the benefits derived from participation, improved skills, work experience that is increasingly used as important showcases of prospective employee-skills, and the participation in the direction of the future of this technology.

OSS development is volunteer driven and, as such, has a high turnover. This high turnover demonstrates the need for a continual pool of developers to maintain the OSS community. This pool of developers has the potential to change the demographics of OSS by providing developers from an environment that encourages the participation of underrepresented groups.

One of the primary motivations for OSS participation and contribution is to learn new skills. Research has pointed to colleges as a place to provide an avenue for this participation, using a course supporting the formation of a CoP (Ye & Kishida, 2003) . The pedagogical model provided a collaborative environment to support participation in and contribution to an Open Source Software project by a diverse population of students. The revised model (see Figure 5.1) shows that the analysis of the data suggested a strong connection between this collaborative model and the students' level of contribution.

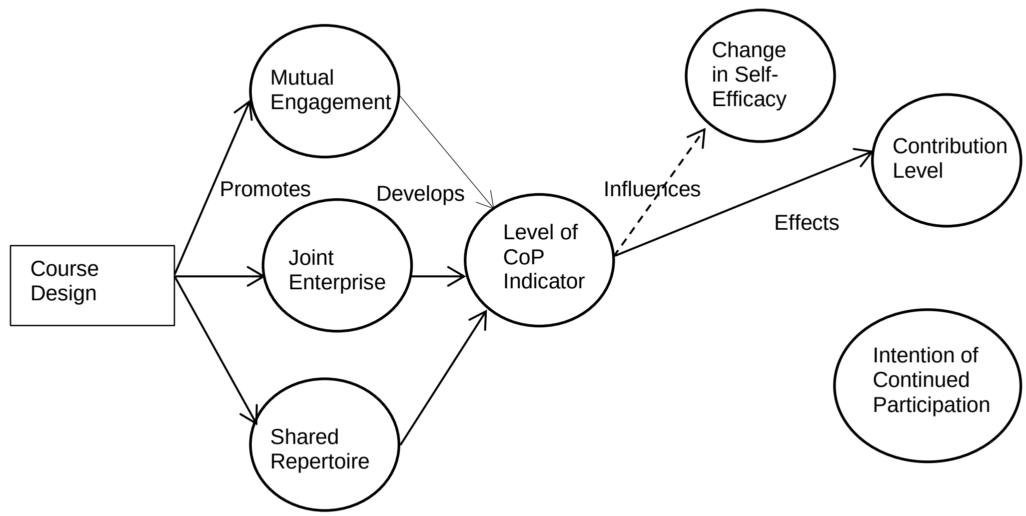

Figure 5.1 Revised Course Design Model

Most computer science courses are taught within a culture of competition and are solitary in nature. Students are discouraged from working together and assignments are to be completed alone. This research looked at a class designed to implement a collaborative culture based on fostering the formation of a CoP. Students worked in groups and solitary work was discouraged. Mentors were provided to assist in finding ways to contribute to the OSS community of Ubuntu as well. The research design was sensitive to student's perceptions of the importance of the elements of a CoP and mentorship in promoting contributions to the community. Additionally students' self-efficacy was measured to determine if there was a change over the course of the term and finally their future plans regarding continued OSS participation were reported.

Review and Discussion of the Main Conclusions of the Study

This study asked four questions in order to evaluate a pedagogical paradigm designed to teach students how to contribute to an OSS project. The study site was a CS class within a small university. The class had thirty students. The demographic makeup included five women, six non-white students and the age breakdown was twenty one students 18 - 25, seven 26 - 35, and two 36 - 45. Additionally the programming experience was broken down as 5 students had 0 - 1 years, seventeen 2 - 4 years, five 4 - 8 years and three over eight years. The data gathered to answer the first two questions was analyzed using a lens of collaborative learning, supported by the literature on pairs programming in CS education. The pairs programming literature demonstrates the benefit of collaboration for women and racial minority groups, providing an inclusive environment that improves the confidence and enjoyment of the students as well as providing a more successful foundation for future course work and improving retention (McDowell et al., 2006; Nagappan et al., 2003; Simon & Hanks, 2008). This environment was supported by the use of a CoP model to design the course, and the data was analyzed using indicators of a CoP that were operationalized within the context of the course.

Additional data was gathered to convey the presence or absence of a change in self-efficacy. The self-efficacy data was included in the design to assist in examining student perception of ability to contribute since contemporary environments are typically considered to be perceived as hostile. This hostility effects underrepresented groups as this hostility is more detrimental to these groups, eroding their self-confidence (Nafus, Leach, & Krieger, 2006) . Data was also gathered and analyzed to consider the intent of students to continue to participate in OSS after the course ended.

Questions One and Two – Community

How do students view the importance of group members and mentors and their effectiveness in promoting engagement in OSS?

Will the course design produce students who engage with the OSS community at a percentage greater than 10%?

The first two questions addressed by this research considered the effect of community on student perceptions of engagement in and contributions to the OSS project of Ubuntu.

Conclusions.

Community was analyzed from the level of classmates within groups as well as interaction with mentors at the level of the OSS community. The results of the analysis of the data gathered on community points to an association between the formation of a CoP and appropriate use of a mentor and contribution level. Groups with High levels of indicators of CoP and engagement with mentors rated the relationship with their group members and mentors among the top sources of help when seeking to contribute to Ubuntu. For a significant number of the students in the High category that went to the Global Jam, this experience also provided an important source of information as did the live demo. These relationships supported a collaborative environment that the analysis demonstrated figured significantly in the decision by these students to contribute to the Ubuntu community both in number of contributions and degree of difficulty. This evidence was further supported by the other two categories where the indicators of CoP and mentor engagement was less apparent, or in some cases did not exist at all. These students in the groups that provided lower to nonexistent levels of collaboration had significantly fewer contributions to Ubuntu. Additionally the literature in OSS motivation recommends the use of a CoP model, which was used to operationalize collaboration.

The analysis of students who made one or more contributions to the Ubuntu project showed that 83%, n= 25, of the students in this class made a contribution and 53%, n = 16, of the class made a significant number of contributions, as discussed below in the next section. This far exceeds the original assertion that the course design would produce a contribution rate of 10%.

Discussion and Implications

Volunteer developers dominate the OSS environment. This situation provides the basis for a high turnover that necessitates the need for a continuing pool of developers to replace those who are leaving. The demographics of OSS are mostly white males. As was presented in chapter one, this lack of diversity harms both the industry, denying it the skills, insights and creativity of the developers who are staying away, while at the same time denying those who stay away the benefits of increased skill, employability, and influence on the future of technology. While the sometimes hostile environment of OSS is especially uninviting to women and racial minorities, it also serves to create barriers for many white men who do not fit the “hacker model”, one of a lonely, obsessed, anti-social man.

To combat this attrition as well as rejection of potential contributors, especially those who bring diverse viewpoints, many OSS projects seek to evolve their projects to be more inclusive. These projects often focus on providing mentors to newcomers, monitor IRC channels and mailing lists to prevent hostile exchanges that adversely affect potential contributors, and create community outreach models to encourage newcomers. Although these are all favorable for helping to improve the culture of OSS, it is unclear if they are changing the number of people who actually contribute to OSS. One of the first steps in joining an OSS project is too subscribe to a mailing list. Research shows that 67.9% of women and 59.3% of men who join these mailing lists never post and of those who do post only 6.6% are women. These figures represent the percentages of newcomers who come to a project vs. those who actually contribute (Kuechler, Gilbertson, & Jensen, 2012b). Although there are no hard numbers that say how many of those who contribute to the mailing list go on to contribute to the project, it stands to reason that this number would be significantly lower.

So the question remains, how do we create a pool of contributors to OSS? This is answered, in part, by the results of this study. The positive number of students who contributed, 83%, and specifically those who had high levels of contribution, 53%, demonstrates this model has merit in creating a potential pool of OSS developers. As this research was driven by the underrepresentation of women and non-white students in OSS it is also important to note that 60% of the women, n= 3 and 67%, n = 4 of the non-white students contributed to the Ubuntu project at moderate to high levels. From a small sample it is possible to examine specific issues in depth, for example the details of how these underrepresented groups perceived the effectiveness of mentors and peers in helping to overcome barriers to contribution to Ubuntu. We argue that this outcome from the collaborative education model is an improvement over the current OSS model using solitary contributors. As OSS projects are non-colocated it is a reasonable assumption that a large number of those who are trying to join are doing so in isolation. The collaborative model of the classroom provides evidence that working within a group supports contribution, supporting the literature on collaborative work in computer science education.

In addition to the collaboration of the classroom environment, this research also suggests that mentorship is important; however it appears as though just providing a mentor was not sufficient for increasing participation. Students needed to know how to use this resource, and when they were unclear, they got little benefit. This information is not only important for the further development of this pedagogical model, but it also serves the OSS communities who are providing mentors with no structure. The outcomes of this research demonstrate that the use of mentorship should be more structured, providing mentees with a more concrete way to use their mentor, e.g., providing questions to help begin the process, have students work together to decide what questions to ask a mentor. Additionally mentors should be given information in advance to assist them in preparing assignments for mentees to further their abilities to contribute. The literature on cooperating teachers as mentors to student teachers shows that having an ongoing reflective dialogue provides a strong mentorship relationship (Stegman, 2007). Using this model is recommended as the framework for further research.

Students who did not interact with their mentors commonly cited documentation as their source of information to learn how to contribute. Currently documentation on large projects aimed at clarifying how to participate is exhaustive, providing all the information necessary, often at the cost of overwhelming newcomers. It was also apparent for students whose groups did not form a CoP that documentation, along with forums and YouTube videos were the resources they sought out to provide scaffolding for contributions. This, coupled with the value derived from the hands on demonstrations, suggests the need to explore the value of web-based tutorials. The ability to see someone actually walk through the process of bug triage was an important resource for students. This visual walk through served to erase the barriers to contributing almost completely in several instances, pointing to the value of walking through demonstrations of contributions, e.g. bug triaging or wiki editing in documentation, for the pedagogical model. The information from this element of the research also provides information for OSS projects attempting to attract new developers that, instead of just documentation, it is useful to newcomers to have video based tutorials on simple contributions.

The analysis of this model provides information for further research stemming from these two questions. First it will be important to consider the information elicited from the groups who formed a CoP and how a functional group was operationalized to provide information to increase the effectiveness of this resource across a wider number of students. The knowledge gathered from this study coupled with the research used to provide information on improving the compatibility of student partners in pairs programming assignments (Katira, Williams, & Osborne, 2005), provides a framework for additional research. A direction for new research might be utilizing compatibility models for pairs programming, pairing minority students, students with perceived similar skill levels, and students with different personalities, to facilitate the formation of groups within the classroom to assist in the development of a CoP within groups.

Further research into structured mentorship is also highlighted in this analysis. Providing structure for both mentor and mentee may increase contributions and effect student's overall experience with participation in an OSS project. The direction for new research that emerges from this study points to using the framework of reflective dialogue to promote a deeper experience for the mentees as well as providing pertinent information to the mentors to assist in guiding their mentees. It is also suggested as further research to consider strengthening the formation of group CoP by utilizing compatibility models from pairs programming as well as considering traits exhibited by the groups who formed strong CoPs in this study. Finally it is indicated that using hands on modeling of contributions is valuable to all students, especially those who belong to groups who failed to form a CoP.

Question Three – Self-Efficacy

Is there a change in self-efficacy from the beginning to end of the course? The third question of this research considered the effects of this pedagogical model on the self-efficacy of the students in the course.

Conclusions.

The analysis of the quantitative data showed that there was a positive increase in self-efficacy overall from the beginning of the term to the end. From the analysis of qualitative data we can get a somewhat finer grained understanding of self-efficacy that varied across groups. Students in groups that had a high indication of CoP reported higher numbers of incidents that that could be considered indicators of positive self-efficacy and groups that scored low on these indicators had few positive numbers of incidents and reported more incidents of sources of negative self-efficacy. Therefore effects of the instructional model on self-efficacy were mixed.

Discussion and Implications

Analysis of qualitative self-efficacy data considered alongside the outcomes from the analysis of the data on the formation of a CoP suggests that self-efficacy is in part associated with collaboration. This outcome is consistent with the literature suggesting that self-efficacy in women increases with interventions that provide mentorship and/or collaborative pedagogical paradigms.

Reexamining the characteristics of CoP and their influence on self-efficacy evidenced in the Women@SCS model from CMU (Frieze & Blum, 2002), discussed in Chapter 1, we can make the following comments from the results of this study.

- Mutual Engagement – In the case of Women @CSC mutual negotiation was focused on creating a culture in CS that was inviting to women. The female students at CMU formed a group that served to provide a place for other women in the program to come and get advice, participate in professional conferences and events, and find mentorship. These meetings served to allow female students to share experiences so they did not feel isolated, see that others like themselves were succeeding in the program, and to share stories about how they those who came before them succeeded.

Although this study offered multiple opportunities for elements of mutual engagement, the most prevalent source was students working together so they could share information. The sharing of information gave students more information to work with than they could have found doing all the research by themselves. This allowed students to assure others in their group that they could accomplish a task, "If I can do it you can". Additionally the sharing of information let students see how much they knew themselves.

- Joint Enterprise – Joint Enterprise is built on a shared history. The Women@SCS had an ongoing history that newcomers joined through the process of storytelling. During specific gatherings the senior students would share their experiences, giving advice on classes to take, information on teachers, and how to deal with issues that were specific to women in the program. These gatherings reduced the stress for incoming students and helped to mitigate the anxiety often felt by female students in male dominated majors.

  In the groups that formed a CoP in this study there was less evidence of stress from figuring out how to contribute to Ubuntu. This reduction in stress levels are demonstrated by an absence of indicators of negative sources of physiological self-efficacy. This lack of anxiety came from support from their partners and a sense of knowing they were all in it together. The success of the group promoted success of each individual. Groups with low to no indication of a CoP within their group reported higher numbers of sources of anxiety or angst.

- Shared Repertoire – as members became proficient with the tools, adopted the language of OSS, and contributed to the project, the sense of accomplishment increased self-efficacy. This provided the opportunity for other minorities to increase self-efficacy through vicarious experience.

  This held true in this study and also provided evidence of a history or bond that had formed between members of a group. The contributions to Ubuntu allowed students to gain confidence in their ability to perform resulting in a sense of mastery, one of the primary sources of self-efficacy.

The characteristics that demonstrated the formation of a CoP were operationalized with indicators taken from the literature on CoP (Wenger, 1998). Mutual enterprise and shared repertoire were the easiest characteristics to identify. The concept of joint enterprise as it was operationalized was difficult to tease from the data, however the history it encompasses was implied by the adoption of shared tools and a cohesion within groups. The data contained references to the importance of these groups for those who were in the High or Moderate category. These were also the groups that had data that supported higher reporting of positive sources of self-efficacy.

There have been many studies in the CS literature that point to interventions that use mentoring and the intentional development of community interaction (Blum, 2001; Cohoon, Gonsoulin, & Layman, 2004; Frieze, Hazzan, Blum, & Dias, 2006; Margolis & Fisher, 2002), as was done in this research model, to help change the culture of CS that excludes many women. More recently this research on inclusion has focused on a collaborative classroom environment providing positive sources of self-efficacy and creating a more enjoyable experience with the coursework for underrepresented groups. The results of the current research are not only consistent with this previous research, but also extend this framework to collegiate courses that promote OSS participation. The course structure in this study provides a model for an OSS course that involves participation in the OSS community and provides the basis for research to extend and elaborate the model.

Question Four – Future Plans

Do students anticipate continued participation in OSS projects?

The final question of this study looked at students' intent to continue contributing to OSS. This addresses the goal of this course to provide a pool of developers to the OSS community.

Conclusions.

A total of 43%, n = 13, of the students in the class reported a moderate to high degree of intent to continue contributing to OSS. Looking at underrepresented groups with the intent to continue there are 40%, n= 2, of the women in the class and 50%, n = 3 of the non-white students who reported a moderate to strong intent to continue with OSS development. It also appears from the analysis of the data that of the 8 students who had more than four years of programming experience 5 stated they would continue participating in OSS. There were no female students with over four years of programming experience so it should be noted that this could be a confounding factor.

Discussion and Implications

The discussion of questions one and two provided answers for increasing student contributions during the timeframe of the course. Question four provides an answer to whether analysis of this model supports the conclusion that students will continue contributing after the course ends, providing the pool of developers needed by the OSS community. This analysis suggests that overall students who are introduced to OSS through a course based on the research model do intend to continue. The rate of participation cited above from research looking at overall participation in OSS showed that 32.1% of women and 40.7% of men who join mailing lists post to the list. In comparison not only was the contribution level higher from this class, 83% or 25 students, 43% of students who were in this collaborative course expressed the intent to continue participation after the course ended. Within underrepresented groups 50%, four out of eight, of students who were female and/or nonwhite have an intention to continue, a higher percentage than the overall class.

The variable that emerged in the analysis of this question, length of experience programming, appeared to play a role in intent to continue contributing. This finding suggests future studies take this attribute into consideration. It may be important to have students obtain more maturity in the major in order to fully benefit from this type of pedagogical model.

Limitations

Researchers bring their own bias and beliefs to any research project. This bias has the potential to influence interpretations of the data analysis. As a researcher every precaution was taken to constrain these limitations. However there is always the frailty of human nature within the analysis of data. In this section I discuss the steps I took to ensure the validity and trustworthiness of my results.

The work of Padgett (1998) outlines six strategies to address threats to validity in a qualitative study; 1) prolonged involvement, 2) triangulation, 3) peer debriefing and support, 4) member checking, 5) negative case analysis and 6) audit trail. I employed four out of six of these strategies in order to mitigate potential bias.

This ten-week course included students who had been students of mine in the past. This prolonged exposure decreases the risk of respondent bias. These nineteen students in the current study had become accustom to the researcher's presence over time lessening the likelihood of reactivity and respondent bias, the presence of the researcher affecting the environment or the desire to respond to questions or situations in a manner that pleased the researcher. In addition to being the researcher, I was also the instructor of this course. I also assert that the ten-week course provided a sufficient time frame to be considered a prolonged exposure for the remaining eleven students.

Triangulation was used to lessen the effects of possible researcher bias. Although I rigorously worked to remain neutral by looking for disconfirming evidence I found it helpful to employ additional strategies to uphold the utmost integrity in my analysis. I used data triangulation in the analysis of data answering all four research questions. For question one I used student reflection writings to consider the effects of peer work groups within the class as well as mentors from within the Ubuntu community. These data sources were then triangulated with group meeting transcripts or forms and copies of email exchanges with mentors. Question two focused on student contributions and I looked at the self-reporting of students within their reflection writings triangulated against documentation in the form of Karma reports on Launchpad. Question three focused on a change in self-efficacy. Student reflections were coded for indications of sources of positive or negative self-efficacy. The results of this analysis were triangulated with the quantitative data gathered with the self-efficacy surveys. The final question on intent to continue participation was the only question that could not be triangulated as it addressed a future activity that was outside the scope of this study. In addition to the triangulation of the data I also made multiple passes through the data refining my coding scheme and considering the data from multiple angles with the addition of new codes.

The framework for this research addressed the effectiveness of the activities of this course to assist in the formation of a CoP. I was rigorous in the reporting of groups that did not attain this formation of CoP. Additionally I looked for and presented disconfirming evidence in the other levels of the formation of CoP pointing to the places where groups lack specific indicators. This view of the data during analysis led me to consider the groups who had high levels of indicators of CoP and how they were operationalized in order to provide the recommendation for future research to examine these factors closer. I also looked at the groups that had low levels to no indication of the formation of CoP and looked for the barriers to provide further instruction for future research.

Finally the use of audit trail served to make my analysis transparent. The notes I took in the field also served to remind me of bias I may have had to a specific instance allowing me to take care when analyzing the data that may have arisen from any specific incident.

The remaining two strategies, peer debriefing and support and member checking, proved to be impractical in this research. I worked on this research as both the researcher and the instructor. It was impractical in this situation to find a peer to use for debriefing or support as no one in the university where I conducted this study is involved in this area of research. Additionally the use of member checking proved to be out of the realm of possibilities because some of the students used for this study graduated. The remaining students proved to be predominately inaccessible due to the demands of their academic schedules. This lack of member checking means that students were not able to provide their perspective which might have differed from my own. Although this may have led to some bias, I attempted to mitigate this with triangulation and looking for disconfirming evidence to the extent these strategies could be used to lessen this bias.

Although every effort was taken to mitigate my bias on the interpretation of this research, it should be noted that my previous experience with my students had influence on students in this course. I had previous relationships with 90% of the students from previous courses. As an instructor I have an open door policy and it is often the case that students in my classes “hang out” in my office. It is this level of camaraderie that existed before this course that will not necessarily be a reproducible part of this paradigm.

It was my goal to provide a robust description of the course itself as well as the analysis of the data. I considered the factors that worked and those that did not. Additionally I used triangulation to ensure the accuracy in interpretation of student reflections. Every precaution was taken to prevent bias from seeping into the reporting of this research.

Conclusion

The tertiary educational system provides an ideal platform for providing a pool of developers to benefit the OSS community. The return benefit to the students is the acquisition of real world experience, gaining important skills, and having the opportunity to build work experience for getting a job after graduation. The development of the course in this study shows promise of providing the structure for a pedagogical paradigm designed to teach OSS participation. This course not only has the potential to affect the culture of OSS, but also may provide a new avenue for teaching CS. This course, based in a collaborative model that promotes inclusion of minorities, may prove to be an attractor course to CS for minorities. As this research adds to the literature on this collaborative paradigm it further supports a collaborative model for all CS courses. Within this pedagogical paradigm lies the potential to also provide a diverse pool to OSS. The changes promoted here hold the potential to change the landscape of the community of CS and OSS, changing the image of a computer scientist and an OSS developer.

Blum (2005), in a follow up of the pivotal longitudinal research of Margolis and Fisher, discuss critical mass of women in computer science. They define critical mass as being attained when “being ‘other’ is no longer a major defining or impeding quality: numbers alone are not necessarily the operative issue”. Once the women in the program became visible and effective in their participation, women felt comfortable in the program. They saw others that resembled them. The face of computer science changed at CMU. That is the change we strive for in OSS and this study can be used to provide one avenue for reaching this critical mass and changing the image of OSS.

Chapter one began by considering how image affects the participation of underrepresented groups in science, technology, engineering, and mathematics. As we are shaped by these images we also have the ability to reshape the images that represent these fields, as CMU demonstrated in their research of women in computer science. The pedagogical paradigm used in this research as the framework for a course to teach OSS holds promise to bring a more diverse pool of developers to the OSS community. This more diverse pool could alter the landscape of this community and in doing so change the image of what an OSS developer looks like. Perhaps someday when asked, students will draw pictures of computer scientists that look just like themselves.

References

Albisetti, M. (2011, November 3). Ubuntu members over time « Martin Albisetti's blog. *Martin Albisetti's Blog*. Retrieved from http://beuno.com.ar/archives/262

Alesina, A., Baqir, R., & Easterly, W. (1999). Public goods and ethnic divisions*. *Quarterly Journal of Economics*, *114*(4), 1243–1284.

Anderson, R. O., Klassen, D. L., Krohn, K. R., & Smith-Cunnien, P. (1982). Assessing computer literacy: Computer awareness and literacy: An empirical assessment. *Available from Minnesota Educational Computing Consortium*.

Bandura, A. (1977). Self-efficacy: toward a unifying theory of behavioral change. *Psychological review*, *84*(2), 191.

Bandura, A. (1986). Social foundations of thought and action. *Englewood Cliffs, NJ*.

Bandura, A. (1997). *Self-efficacy: The exercise of control*. Worth Publishers.

Bandura, A. (2006). Guide for constructing self-efficacy scales. *Self-efficacy beliefs of adolescents*, *5*(307-337). Retrieved from http://books.google.com/books?hl=en&lr=&id=Cj97DKKRE7AC&oi=fnd&pg=PA307&dq=GUIDE+FOR+CONSTRUCTING+SELF-EFFICACY+SCALES&ots=cF0TtWwPm4&sig=Pt_raYlKpwPwpfnYuppXNXCmth8

Bell, A. (2011). UbuntuMembers - Ubuntu Women. Retrieved December 6, 2011, from http://wiki.ubuntu-women.org/UbuntuMembers

Bellini, E., Ottaviano, G. I. P., Pinelli, D., & Prarolo, G. (2008). Cultural Diversity and Economic Performance: Evidence from European Regions. *HWWI Research Paper*, *3*, 14.

Benbow, C. P., & Stanley, J. C. (1980). Sex differences in mathematical ability: Fact or artifact? *Science*, *210*(4475), 1262–1264.

Betts, O., Hawthorn, Pico, A., Massey, B., Leto, J., Redman, J., … Tewari, M. (n.d.). Mentors Guide. *Floss Manuals*. Retrieved June 9, 2012, from http://en.flossmanuals.net/GSoCMentoring/index/

Beyer, S. (1999). The accuracy of academic gender stereotypes. *Sex Roles*, *40*(9), 787–813.

Beyer, S., Rynes, K., Perrault, J., Hay, K., & Haller, S. (2003). Gender differences in computer science students. *ACM SIGCSE Bulletin*, *35*(1), 49–53.

Blalock, H. M. (1957). Per cent non-white and discrimination in the South. *American Sociological Review*, *22*(6), 677–682.

Blum, L., & Frieze, C. (2005). As the culture of computing evolves, similarity can be the difference. *Frontiers*, *26*. Retrieved from http://www-2.cs.cmu.edu/~lblum/PAPERS/TheEvolvingCSCulture.pdf

Bonser, F. G. (1910). *The reasoning ability of children of the fourth, fith, and sixth school grades*. Teachers college, Columbia university. Retrieved from http://books.google.com/books?hl=en&lr=&id=fowVAAAAIAAJ&oi=fnd&pg=PA1&dq=The+Reasoning+Ability+of+Children+of+Fourth+Fifth,+and+Sixth+School+Grades&ots=AmCc5uyi-8&sig=rs3t41N3RVR1lut5fwsa-2keV4A

Brush, S. G. (1991). Women in science and engineering. *American Scientist*, *79*(5), 404–419.

BugSquad - Ubuntu Wiki. (n.d.). Retrieved October 25, 2012, from https://wiki.ubuntu.com/BugSquad

Burnett, M., Wiedenbeck, S., Grigoreanu, V., Subrahmaniyan, N., Beckwith, L., & Kissinger, C. (2008). Gender in end-user software engineering. In *Proceedings of the 4th International Workshop on End-user Software Engineering* (pp. 21–24).

Busch, T. (1995). Gender differences in self-efficacy and attitudes toward computers. *Journal of educational computing research*, *12*(2), 147–158.

Byfield, B. (2009, September 9). Sexism: Open Source Software’s Dirty Little Secret - Datamation. *Datamation*. Magazine. Retrieved June 5, 2012, from http://www.datamation.com/osrc/article.php/3838186/Sexism-Open-Source-Softwares-Dirty-Little-Secret.htm

Catalyst. (2004). The Bottom Line: Connecting Corporate Performance and Gender Diversity. Retrieved from http://www.catalyst.org/file/44/the%20bottom%20line%20connecting%20corporate%20performance%20and%20gender%20diversity.pdf

Chambers, D. W. (1983). Stereotypic images of the scientist: The draw-a-scientist test. *Science Education*, *67*(2), 255–265.

Chatman, J. A., Polzer, J. T., Barsade, S. G., & Neale, M. A. (1998). Being Different Yet Feeling Similar: The Influence of Demographic Composition and Organizational Culture on Work Processes and Outcomes. *Administrative Science Quarterly*, *43*(4), 749–780. doi:10.2307/2393615

Code of Conduct | Ubuntu. (n.d.). Retrieved May 17, 2012, from http://www.ubuntu.com/project/about-ubuntu/conduct

Cohoon, J. M. . (2001). Toward improving female retention in the computer science major. *Communications of the ACM*, *44*(5), 108–114.

ContributeToUbuntu - Ubuntu Wiki. (n.d.). Retrieved October 4, 2012, from https://wiki.ubuntu.com/ContributeToUbuntu

David, P. A., Waterman, A., & Arora, S. (2004). FLOSS-US: the free/libre/open source software survey for 2003. *Stanford Institute for Economic Policy Research, at http://www. stanford. edu/group/floss-us/, accessed*, *20*.

Deshpande, A., & Riehle, D. (2008). The total growth of open source. *Open Source Development, Communities and Quality*, 197–209.

Diversity | Ubuntu. (n.d.). Retrieved August 3, 2012, from http://www.ubuntu.com/project/about-ubuntu/diversity

Dreamwidth Diversity Statement. (n.d.). Retrieved May 17, 2012, from http://www.dreamwidth.org/legal/diversity

Eells, E. C., & Fox, C. S. (1932). Sex differences in mathematical achievement of junior college students. *Journal of educational psychology*, *23*(5), 381.

Ely, R. J., & Thomas, D. A. (2001). Cultural diversity at work: The effects of diversity perspectives on work group processes and outcomes. *Administrative Science Quarterly*, 229–273.

F/LOSS-POLS. (2006). *Free/Libre and Open Source Software: Policy Support*. March.

Fennema, E. (1974). Mathematics learning and the sexes: A review. *Journal for Research in Mathematics Education*, 126–139.

Finson, K. D., Beaver, J. B., & Cramond, B. L. (1995). Development and field test of a checklist for the Draw-A-Scientist Test. *School Science and Mathematics*, *95*(4), 195–205.

Fisher, A., Margolis, J., & Miller, F. (1997). Undergraduate women in computer science: experience, motivation and culture. *ACM SIGCSE Bulletin*, *29*(1), 106–110.

Frieze, C., & Blum, L. (2002). Building an effective computer science student organization: the Carnegie Mellon women@ SCS action plan. *ACM SIGCSE Bulletin*, *34*(2), 74–78.

Geek Feminism. (various). Timeline of incidents. *Geek Feminism Wiki*. Retrieved April 24, 2011, from http://geekfeminism.wikia.com/wiki/Timeline_of_incidents

Ghosh, R. A., Glott, R., Krieger, B., & Robles, G. (2002). *Free/libre and open source software: Survey and study*. Maastricht Economic Research Institute on Innovation and Technology, University of Maastricht, The Netherlands, June.

Grigoreanu, V., Cao, J., Kulesza, T., Bogart, C., Rector, K., Burnett, M., & Wiedenbeck, S. (2008). Can feature design reduce the gender gap in end-user software development environments? In *Visual Languages and Human-Centric Computing, 2008. VL/HCC 2008. IEEE Symposium On* (pp. 149–156). Retrieved from http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4639077

Gürer, D. (2002). Pioneering women in computer science. *ACM SIGCSE Bulletin*, *34*(2), 175–180.

Gürer, D., & Camp, T. (2002). An ACM-W literature review on women in computing. *ACM SIGCSE Bulletin*, *34*(2), 121–127.

Hackett, G., & Betz, N. E. (1981). A self-efficacy approach to the career development of women. *Journal of Vocational Behavior*, *18*(3), 326–339. doi:10.1016/0001-8791(81)90019-1

Hansson, D. H. (2005, September 7). Reduce the risk, hire from open source. *Loud Thinking*. Retrieved from http://www.loudthinking.com/arc/000505.html

Hong, L., Page, S. E., & Institute, S. F. (2001). Diversity and optimality.

Ibanez, L. (2010, February 17). The Kitware Blog - Opening Open Source. Retrieved September 25, 2011, from http://www.kitware.com/blog/home/post/13

Intemann, K. (2009). Why Diversity Matters: Understanding and Applying the Diversity Component of the National Science Foundation's Broader Impacts Criterion. *Social Epistemology*, *23*(3), 249–266.

Jackson, S. E., Joshi, A., & Erhardt, N. L. (2003). Recent Research on Team and Organizational Diversity: SWOT Analysis and Implications. *Journal of Management*, *29*(6), 801–830. doi:10.1016/S0149-2063_03_00080-1

Jacobs, J. E. (2005). Twenty-five years of research on gender and ethnic differences in math and science career choices: What have we learned? *New directions for child and adolescent development*, *2005*(110), 85–94.

Jacobs, Janis E., & Eccles, J. S. (1985). Gender Differences in Math Ability: The Impact of Media Reports on Parents. *Educational Researcher*, *14*(3), 20–25. doi:10.2307/1174575

Jehn, K. A. (1997). A Qualitative Analysis of Conflict Types and Dimensions in Organizational Groups. *Administrative Science Quarterly*, *42*(3), 530–557. doi:10.2307/2393737

Kanter, R. M. (1977). *Men and Women of the Corporation*. Basic Books.

Katira, N., Williams, L., & Osborne, J. (2005). Towards increasing the compatibility of student pair programmers. In *Software Engineering, 2005. ICSE 2005. Proceedings. 27th International Conference On* (pp. 625–626).

Knight, M., & Cunningham, C. (2004). Draw an Engineer Test (DAET): Development of a tool to investigate students' ideas about engineers and engineering. In *ASEE Annual Conference and Exposition*. Retrieved from http://gk12.poly.edu/amps-cbri/pdf/Draw%20an%20Engineer%20Test%20(DAET)%20-%20Development%20of%20a%20Tool%20to%20Investigate%20Students%E2%80%99%20Ideas%20about%20Engineers%20and%20Engineering.pdf

Kochan, T., Bezrukova, K., Ely, R., Jackson, S., Joshi, A., Jehn, K., … Thomas, D. (2003). The effects of diversity on business performance: Report of the diversity research network. *Human Resource Management*, *42*(1), 3–21.

Krumbach, E. (2010, November 2). It’s official, over 5% of Ubuntu Members are women! *Pleia2’s Blog*. Retrieved from http://princessleia.com/journal/?p=3651

Kuechler, V., Gilbertson, C., & Jensen, C. (2012a). Gender Differences in Early Free and Open Source Software Joining Process. *Open Source Systems: Long-Term Sustainability*, 78–93.

Kuechler, V., Gilbertson, C., & Jensen, C. (2012b). Gender Differences in Early Free and Open Source Software Joining Process. *Open Source Systems: Long-Term Sustainability*, 78–93.

Lakhani, K., Wolf, B., Bates, J., & DiBona, C. (2002). The boston consulting group hacker survey. *Boston, The Boston Consulting Group*.

Lave, J., & Wenger, E. (1991). *Situated learning: Legitimate peripheral participation*. Cambridge Univ Pr.

Levesque, M., & Wilson, G. (2004). WOMEN IN SOFTWARE-Open Source, Cold Shoulder-Proponents of open source software often describe their campaign as a great equalizer: Not only is it freely available, but. anyone who wants to help. *Software Development*, *12*(11), 40–45.

Lockard, C. B., & Wolf, M. (2012). Occupational employment projections to 2020. *Monthly Lab. Rev.*, *135*, 84.

Lund, F. H. (1932). Sex differences in type of educational mastery. *Journal of Educational Psychology*, *23*(5), 321–330. doi:10.1037/h0072575

Madjar, N. (2005). The Contributions of Different Groups of Individuals to Employees’ Creativity. *Advances in Developing Human Resources*, *7*(2), 182 –206. doi:10.1177/1523422305274525

Mandinach, E., & Fisher, C. W. (1985). Individual differences and acquisition of computer programming skill. *ACCCEL Report, Lawrence Hall of Science, University of California, Berkeley*.

Margolis, J. (2008). *Stuck in the shallow end: Education, race, and computing*. The MIT Press.

Margolis, J., & Fisher, A. (1997). Geek mythology and attracting undergraduate women to computer science. *Impacting Change Through Collaboration*. Retrieved from http://www.cs.cmu.edu/afs/cs.cmu.edu/project/gendergap/www/papers/wepan97.ps

Margolis, J., & Fisher, A. (2002). *Unlocking the clubhouse* (Vol. 262133989). MIT Press.

Martin, C. D. (2004). Draw a computer scientist. In *ACM SIGCSE Bulletin* (Vol. 36, pp. 11–12). Retrieved from http://dl.acm.org/citation.cfm?id=1041628

McDowell, C., Werner, L., Bullock, H. E., & Fernald, J. (2006). Pair programming improves student retention, confidence, and program quality. *Communications of the ACM*, *49*(8), 90–95. doi:10.1145/1145287.1145293

Mead, M., & Métraux, R. (1957). Image of the Scientist among High-School Students: A Pilot Study. *Science (New York, N.Y.)*, *126*(3270), 384–390. doi:10.1126/science.126.3270.384

Merriam, S. B. (1988). *Case study research in education: A qualitative approach.* Jossey-Bass. Retrieved from http://psycnet.apa.org/psycinfo/1989-97006-000

Miura, I. T. (1987). The relationship of computer self-efficacy expectations to computer interest and course enrollment in college. *Sex Roles*, *16*(5), 303–311.

Nafus, D., Leach, J., & Krieger, B. (2006). Free/Libre and Open Source Software: Policy Support FLOSSPOLS. *UCAM*.

Nagappan, N., Williams, L., Ferzli, M., Wiebe, E., Yang, K., Miller, C., & Balik, S. (2003). Improving the CS1 experience with pair programming. In *ACM SIGCSE Bulletin* (Vol. 35, pp. 359–362). Retrieved from http://dl.acm.org/citation.cfm?id=612006

Norman, D. (1983). Some observations on mentai modeis. *Mental models*, *7*. Retrieved from http://books.google.com/books?hl=en&lr=&id=QFI0SvbieOcC&oi=fnd&pg=PA7&dq=Some+observations+on+mental+models&ots=rTlUUw_BKJ&sig=YJRDRQPR9ZsqOfTbgEl13a-JfgA

NSF. (2009). *Women, Minorities, and Persons with Disabilities in Science and Engineering: 2009* (No. NSF 09-305). Arlington, VA: National Science Foundation, Division of Science Resources Statistics. Retrieved from http://www.nsf.gov/statistics/wmpd/

NSF. (2011). *Science and Engineering Degrees: 1966–2008.* (No. NSF 11-316). Arlington, VA: National Science Foundation, Division of Science Resources Statistics. Retrieved from http://www.nsf.gov/statistics/nsf11316/

Olson, G. M., & Olson, J. S. (2000). Distance matters. *Human-computer interaction*, *15*(2), 139–178.

Ottaviano, G. I. P., & Peri, G. (2006). The economic value of cultural diversity: evidence from US cities. *Journal of Economic Geography*, *6*(1), 9.

Padgett, D. K. (1998). Qualitative Methods in Social Work Research: Challenges and Rewards Sage. *Newbury Park, CA*.

Paolucci, D., & Smith, M. (2010, July 21). *Build Your Own Contributors (One Part at a Time)*. Presented at the OSCON 2010, The O'Reilly Open Source Convention - O'Reilly Conferences, Portland, OR. Retrieved from http://www.oscon.com/oscon2010/public/schedule/detail/13542

Park, Y. (2008). Supporting the learning process of open source novices: an evaluation of code and project history visualization tools.

Picker, S. H., & Berry, J. S. (2000). Investigating pupils' images of mathematicians. *Educational Studies in Mathematics*, *43*(1), 65–94.

Ramalingam, V., & Wiedenbeck, S. (1998). Development and validation of scores on a computer programming self-efficacy scale and group analyses of novice programmer self-efficacy. *Journal of Educational Computing Research*, *19*(4), 367–381.

Raymond, E. S. (1996). *The new hacker's dictionary*. The MIT Press.

Raymond, E. S. (2001). *The cathedral and the bazaar: musings on Linux and open source by an accidental revolutionary*. O'Reilly & Associates, Inc.

Robert, K. (2009, July 25). Standing out in the crowd: my OSCON keynote | Infotropism. Retrieved May 17, 2012, from http://infotrope.net/2009/07/25/standing-out-in-the-crowd-my-oscon-keynote/

Roberts, E. S., Kassianidou, M., & Irani, L. (2002). Encouraging women in computer science. *ACM SIGCSE Bulletin*, *34*(2), 88.

Robles, G., Scheider, H., Tretkowski, I., & Weber, N. (2001). Who is doing it? A research on libre software developers. *Research Paper, TU Berlin, August*.

Rustad, K. (2011). “ Suck It Up, Princess”: Outreach and Diversity in FOSS Communities.

Schanker, J., & Goodman, T. (2003). Open Source: The Movement’s Loss of Momentum.

Schindler, E. (2009, September 20). Mentoring in Open Source Communities: What Works? What Doesn’t? *IT World*. Retrieved September 17, 2011, from http://www.itworld.com/open-source/78271/mentoring-open-source-communities-what-works-what-doesnt?page=0,2

Schmerken, I. (2011, November). Open Source Technology - Capital Markets Outlook 2012 - Wall Street & Technology. *Wall Street and Technology*. Retrieved from http://www.wallstreetandtech.com/2012-outlook/open-source-technology

Scragg, G., & Smith, J. (1998). A study of barriers to women in undergraduate computer science. In *ACM SIGCSE Bulletin* (Vol. 30, pp. 82–86). Retrieved from http://dl.acm.org/citation.cfm?id=273167

Seymour, E., & Hewitt, N. M. (1997). *Talk about leaving: why undergraduates leave the sciences*. Boulder, CO: Westview Press.

Shirley, N. (2010, November 5). A hell of a time. *NerdChic.Net*. Blog. Retrieved April 24, 2011, from http://blog.nerdchic.net/archives/418/

Sierra, K. (2007, May 3). A Very Sad Day. *Creating Passionate Users*. Blog. Retrieved from http://web.archive.org/web/20070503012914/http://headrush.typepad.com/whathappened.html

Simon, B., & Hanks, B. (2008). First-year students’ impressions of pair programming in CS1. *Journal on Educational Resources in Computing (JERIC)*, *7*(4), 5.

Smith, S. M. (2001). The Four Sources of Influence on Computer Self-Efficacy. *Delta Pi Epsilon Journal*, *43*(1), 27–39.

Snyder, T. D., Hoffman, C. M., & Geddes, C. M. (1997). *Digest of Education Statistics, 1996* (No. NCES 96-133). U. S. Department of Education Office of Educational Research and Improvement. Retrieved from http://nces.ed.gov/pubs96/96133.pdf

Stake, R. (1967). The countenance of educational evaluation. *The Teachers College Record*, *68*(7), 523–540.

Stake, R. E. (1995). *The Art of Case Study Research*. SAGE.

Steele, G. L. (1984). *The Hacker's dictionary: a guide to the world of computer wizards*. Harper & Row.

Stegman, S. F. (2007). An exploration of reflective dialogue between student teachers in music and their cooperating teachers. *Journal of Research in Music Education*, *55*(1), 65–82.

Stroud, J. B., & Lindquist, E. F. (1942). Sex differences in achievement in the elementary and secondary schools. *Journal of Educational Psychology*, *33*(9), 657–667. doi:10.1037/h0057124

The gender factor in math: A new study says males may be naturally abler than females. (1980, December 15). *Time*, 57.

The Open Source Definition | Open Source Initiative. (n.d.). Retrieved April 27, 2012, from http://www.opensource.org/docs/osd

Townsend, G. C., Menzel, S., & Siek, K. A. (2007). Leveling the CS1 playing field. *ACM SIGCSE Bulletin*, *39*, 331. doi:10.1145/1227504.1227428

Trapasso, E., & Vujanic, A. (2010). *Accenture Newsroom: Investment in Open Source Software Set to Rise, Accenture Survey Finds*. Retrieved from http://newsroom.accenture.com/article_display.cfm?article_id=5045

VanRoekel, S., & Chopra, A. (2011, December 5). Data.gov Goes Global | The White House. *Open Gov Blog*. Retrieved from http://www.whitehouse.gov/blog/2011/12/05/datagov-goes-global

Vaughan-Nichols, S. (2009, September 15). Open-source software may unify the medical-records realm. *Computerworld*. Retrieved from http://www.computerworld.com/s/article/9137376/Open_source_software_may_unify_the_medical_records_realm

Walker, L. (2002, September 12). Whitehouse.gov switches to Drupal | The Archer Group. Retrieved from http://www.archer-group.com/2009/technology/whitehousegov-switches-to-drupal

Weber, S., & Mitchell, C. (1995). *That's Funny, You Don't Look Like a Teacher!: Interrogating Images and Identity in Popular Culture*. Psychology Press.

Wenger, E. (1998). *Communities of practice: Learning, meaning, and identity*. Cambridge Univ Pr.

White, R. T., & Gunstone, R. F. (1992). *Probing understanding*. Routledge.

Wiedenbeck, S. (2011, October 9). Self-Efficacy Instrument. Personal communication

Williams, D. A., & King, P. (1980, December 15). Do males have a math gene? *Newsweek*, 73.

Wilson, B. C. (2002). A study of factors promoting success in computer science including gender differences. *Computer Science Education*, *12*(1-2), 141–164.

Wurster, L., Igou, B., & Babat, Z. (2011). *Survey Analysis: Overview of Preferences and Practices in the Adoption and Usage of Open-Source Software | 1528219* (No. G00210068) (p. 29). Gartner Research. Retrieved from http://www.gartner.com/DisplayDocument?ref=clientFriendlyUrl&id=1528219

Ye, Y., & Kishida, K. (2003). Toward an understanding of the motivation of open source software developers. In *Software Engineering, 2003. Proceedings. 25th International Conference On* (pp. 419–429). Retrieved from http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1201220

Zeldin, A. L., & Pajares, F. (2000). Against the odds: Self-efficacy beliefs of women in mathematical, scientific, and technological careers. *American Educational Research Journal*, *37*(1), 215–246.

Appendices

Appendix 1

Research Course Outline

To replicate the study use the forms from Appendix One and Two. The questionnaire and first self-efficacy survey should be administered at the beginning of week one lecture. Mentors should be introduced in week three. Midterm reflection prompts should be given as a take home assignment before the midterm test. Final reflection prompts should be given as a take home assignment during week ten. There are forms for group meetings and mentor meetings, however it is recommended that students conduct meetings in an electronic form and turn in transcripts as this provides a more robust data source.

	Activity	**Homework**
Week One		
Lecture	Provide an introduction to Open Source and the course.	
Lab	WORK IN GROUPS Set up a dev environment on a 8GB flash drive using Universal USB installer to install Ubuntu Create a Launchpad Account - https://login.launchpad.net/8pvxWX7fnjdHgzmQ/+login	Read documents Code of Conduct - http://www.ubuntu.com/project/about-ubuntu/conduct Ubuntu Architecture - https://wiki.ubuntu.com/UbuntuArchitecture

	Create SSH and PGP keys Publish keys to Launchpad Join a mailing Create a file and filter the mail from the list into this file.	Using The Terminal - https://help.ubuntu.com/community/UsingTheTerminal Version Control by Example - http://www.ericsink.com/vcbe/html/index.html Easy Bazaar - https://help.ubuntu.com/community/EasyBazaar Beginners Guide to Bazaar - http://ubuntuforums.org/showthread.php?t=916132
Week Two		
Lecture	Ubuntu Community Lecture on the Ubuntu community, architecture and various ways to participate *(development, documentation, bug squad, artwork, testing and support). Group discussion of the Code of Conduct * Next time I do this I will assign the lecture material to students and have them research and present on the topics	
Lab	Sign the Ubuntu Code of Conduct. (This process allows students to use the pgp keys. All of this was done from the terminal) Join the Ubuntu beginner's mailing list.	Read documents Artwork - https://wiki.ubuntu.com/Artwork BugSquad -

	Sign onto the beginner's IRC channel and lurk http://webchat.freenode.net/ Choose an area of interest * For people who have never been in an OSS environment it was really important to go over the very basic stuff.	https://wiki.ubuntu.com/Bug-Squad Documentation Team - https://wiki.ubuntu.com/DocumentationTeam DocumentationTeam/SystemDocumentation - https://wiki.ubuntu.com/DocumentationTeam/SystemDocumenta-tion DocumentationTeam/Wiki - https://wiki.ubuntu.com/DocumentationTeam/Wiki UbuntuDevelopers - https://wiki.ubuntu.com/UbuntuDevelopers MOTU/GettingStarted – https://wiki.ubuntu.com/MOTU/GettingStarted Testing - http://mago.ubuntu.com/
Week Three		
Lecture	Learn IRC Etiquette Small group discussion on IRCs addressing fears, any positive feedback	

	from those who have already "talked"	
Lab	Look at documentation and wikis as a possible entry - form groups that have shared goals.(Putting students together in groups was extremely important for a lot of the students. They found it easier to find an in if they had others to work with)	Read documents Intro to Python - https://wiki.ubuntu.com/BeginnersTeam/FocusGroups/Development/Academy/IntroToPython Dev Beginnings - https://wiki.ubuntu.com/BeginnersTeam/FocusGroups/Development/Devbeginnings IRC Basics - http://www.ircbeginner.com/ircinfo/ircc-commands.html Also Find a document to help edit, explore bug triage, or find a test group to work with. Attend at least one group meeting
Week Four		
Lecture	Lecture about the Python language Demonstration on Wiki editing for documentation Continue to learn about the community structure from the weeks readings	
Lab	Working in groups find a way to	Attend at least one group

	contribute (it is not necessary to actually contribute, just to find entry points) Share with the group the information gathered from homework assignment Log onto the beginner's IRC channel and ask for suggestions on ways to help out.	meeting
Week Five		
Lecture	Share ideas on getting involved Hands on Demonstration of Bug Triage either by Ubuntu community member or student who has triaged bugs. Small group discussions assigning members from the same lab into small groups for discussion.	
Lab	Working in groups Triage at least two bugs per person in the group	Attend at least one group meeting
Week Six		
Lecture	Midterm Test over knowledge of community (The midterm ended up being a group project where they demonstrated the ability to navigate the Ubuntu site and find various things that would be important to contributing. I also used this as a data gathering vehicle asking for the 3 most useful things they had so far in contributing)	

Lab	Continue with levels of bug triage: -Responding to new bugs as they are filed. -Ensuring that new bugs have all the necessary information. -Assigning bugs to the proper package. -Confirming bug reports by trying to reproduce them. -Setting the importance of bugs' reports. *(Bug Control members Only)* -Searching for and marking duplicates in the bug tracking system. -Sending bugs to their upstream authors, when applicable. -Cross-referencing bugs from other distributions. -Expiring old bugs.	Attend at least one group meeting Read Open Advice – http://open-advice.org/
Week Seven		
Lecture	Share information – Discuss Open Advice to expand the discussion from Ubuntu to the OSS community Presentation on Mago Break into groups, but not the same groups as in lab, and share ideas on contribution.	
Lab	Contribute to Documentation	Attend at least one group meeting
Week Eight		
Lecture	Demonstration on ISO testing using Mago	
Lab	ISO testing	Attend at least one group meeting
Week Nine		

Lecture	Students present contributions	
Lab	Bug Triage, Documentation, ISO testing	Attend at least one group meeting
Week Ten		
Lecture	Students present contributions	
Lab	Bug Triage, Documentation, ISO testing	Attend two group meetings

Appendix 2

Self-Efficacy Survey

Self-efficacy survey – proposed subset of questions. Uses a 7-point Likert scale. Column one gives the proposed question and column two gives the original question. The changes were sent to Dr. Weidenbeck, one of the authors of the original instrument and she found the changes to be in line with the original survey.

1. I understand the object-oriented paradigm.	I understand the object-oriented paradigm. (same as originally survey)
2. I can make use of a pre-written function, given a clearly labeled declaration of the function.	I can make use of a pre-written function, given a clearly labeled declaration of the function. (same as originally survey)
3. I can make use of a class that is already defined, given a clearly labeled declaration of the class.	I can make use of a class that is already defined, given a clearly labeled declaration of the class. (same as originally survey)
4. I can debug (correct all the errors) in a long and complex program that I had written and make it work.	I can debug (correct all the errors) a long and complex program that I had written and make it work. (same as originally survey)
5. I could assess bug reports to determine if they contain enough information to be worked on.	
6. I could contribute to someone else's code if I had only the documentation for help.	I could complete a programming project if I had only the language reference manual for help.
7. I could write documentation for someone else's code.	
8. I could contribute to someone else's code if I could call someone for help if I got stuck.	I could complete a programming project if I could call someone for help if I got stuck.
9. I could contribute to someone else's code once someone else	I could complete a programming project if I could call someone for help

helped me get started.	if I got stuck.
10. I could contribute to someone else's code if I had just the built-in help facility for assistance.	I could complete a programming project if I had just the built-in help facility for assistance.
11. I could come up with a suitable strategy for a given programming project in a short time.	I could come up with a suitable strategy for a given programming project in a short time. (same as originally survey)
12. I could mentally trace through the execution of a long, complex, multi-file program given to me.	I could mentally trace through the execution of a long, complex, multi-file program given to me. (same as originally survey)
13. I could rewrite lengthy confusing portions of code to be more readable and clear.	I could rewrite lengthy confusing portions of code to be more readable and clear. (same as originally survey)
14. I can find ways of motivating myself to program, even if the problem area was of no interest to me.	I can find ways of motivating myself to program, even if the problem area was of no interest to me. (same as originally survey)
15. I could write a program that someone else could comprehend and add features to at a later date.	I could write a program that someone else could comprehend and add features to at a later date. (same as originally survey)

Questionaire

1. Name
2. Gender
    a. Male
    b. Female
    c. Other
3. Race
    a. White

   b. Asian/Pacific Islander
   c. Black
   d. Hispanic
   e. American Indian / Alaska Native
4. Age
   a. 18 – 25
   b. 26 – 35
   c. 36 – 45
   d. over 45
5. Current grade level
   a. Sophomore
   b. Junior
   c. Senior
   d. Other (please specify)
6. What programming languages have you used (circle all that apply)
   a. Java
   b. JavaScript
   c. C/C++/C#
   d. Python
   e. Ruby
   f. Other/s:____
7. How long have you been programming?

a. 0 – 1year
b. 2 - 4 years
c. 4 - 8 years
d. over 8 years

8. Have you ever participated in or attempted to participate in Open Source Software (OSS) Development?
   a. Yes
   b. No
   c. If yes, please list the project/s and describe your role and experiences.

9. Please discuss your knowledge of OSS and any positive or negative feelings or fears you may have associated with participation.

10. What do you expect to come away from this course knowing/ being able to do?

Appendix 3

Forms for Data Collection

Group Meeting Form (Alternate to Transcript)

Date of meeting:

Who attended:

Form of communication ((e.g. Skype, phone, email, IM, IRC, in person):

What was discussed:

Problems solved:

Problems remaining unsolved:

Midterm Reflection Prompt

List the three most useful things you have found to help you engage with the Ubuntu community (in order of importance with the most important listed first). Explain how they have helped you.

Final Reflection Prompts

Please answer the following questions in detail, writing a couple of paragraphs minimum for all questions (except 1).

1. What area of Ubuntu did you engage with?

2. What, specifically, did you do to engage?

3. What contributions did you make to the community?

4. What was your greatest source of help in becoming engaged?

5. What was the biggest hurdle to engagement?

6. If you were to participate in another OSS community what are the steps you would take to become engaged?

7. What plans, if any, do you have for continuing participation with Ubuntu?

8. What plans, if any, do you have for finding other OSS projects to participate in?

9. What grade do you believe you deserve in this class and why?

Appendix 4

Notes From the Global Jam

These are the notes taken at the Global Jam that were disseminated to students. They were also used by WINNIE to give a talk to both labs on bug triage.

I. How do bugs get into Launchpad?

   a. When app crashes-option to report

   b. Emailed to Ubuntu

   c. Created through Launchpad

   d. From command line type – Ubuntu-bug evince

      This will collect the bug report and send it to Launchpad.

II. To view bugs in Launchpad

   a. enter bugs.launchpad.net/Ubuntu/ in your web browser address bar.

   b. This will pull up a list of all reported bugs.

   c. Use the search bar to search by package name.

      i. Most apps have self named package names

1. i.e. Evince Document Reader's package name is Evince

ii. The library apps, such as LibreOffice are harder to find name.

d. Start with bugs on apps you are familiar with using.

i. Evince is nice because they are easy to recreate the bug.

III. Click on a bug, view the information.

a. If the title needs updating, click the pencil icon next to it, update it.

b. Next to the name and bug number is a sideways arrow, click.

i. This will open more information and allow you to update the status and importance.

c. AddTags icon: Use this to add what version of Ubuntu is affected by this bug.

i. Do this by recreating the bug and then posting comments of what version you used of Ubuntu and the app version.

ii. Tags are prefilled selection list. If not on the list, don't add.

d. Recreate the bug and post comments

i. Include the versions of Ubuntu and apps used. Include step by step information.

e. Not a bug? Is it a question?

i. On the right side of the screen, there is a link that says something like “Convert to a question.”

ii. Click this and fill out the information. More info the better.

f. Want all bugs for a package shortcut?

i. Launchpad.net/Ubuntu/+source/enter package name here

ii. Command line: apt-cache search package name here

g. Tips

i. Work on bugs for apps you know.

ii. Ask, is this a fixable bug? Can I help by recreating? Is this a question?

iii. Use the IRC’s!

1. Oregon room!
2. Beginner’s room
3. Bugs room!

h. Reporting a bug

   i. Include screen shots

   ii. Be sure the title matches the problem and clearly describes it.

   iii. Screen casts are also helpful

   iv. Include the package names and versions of all programs.

   v. Include step by step process that you did to find bug.