On the Impact of a Collaborative Pedagogy on African American Millennial Students in Software Engineering

Laurie Williams[1], Lucas Layman[1], Kelli M. Slaten[2], Sarah B. Berenson[2], Carolyn Seaman[3]
[1]*Department of Computer Science, z{lawilli3, lmlayma2}@ncsu.edu*
[2]*Department of Mathematics Education, {kmslaten, berenson}@ncsu.edu*
North Carolina State University, Raleigh, NC, USA
[3]*Department of Information Systems, cseaman@umbc.edu*
University of Maryland, Baltimore County, Baltimore, MD, USA

Abstract

Millennial students (those born after 1982), particularly African Americans and women, have demonstrated a propensity toward collaborative activities. We conducted a collective case study at North Carolina State University and North Carolina A&T to ascertain the role of collaboration and social interaction in attracting and retaining students in information technology. Responses from semi-structured interviews with 11 representative African American students in these classes were coded and analyzed. The responses from these minority students were used to evolve a social interaction model. The conjectures generated from the model suggest that pair programming and agile software methodologies effectively create a collaborative environment that is desirable to Millennial students, male and female, and, with the new evidence, minority and majority. Additionally, the African American Millennial students enjoy learning from their peers and believe that a collaborative environment better prepares them for the "real world."

1. Introduction

The Millennials gravitate toward group activity . . . gravitate toward activities that promote and reinforce social interaction . . . where students help each other. [27]

Research has suggested that African American, Hispanic, Asian and women students prefer and benefit from collaborative work environments that provide for social interactions [23, 24, 34]. Additionally, Millennial students (those born after 1982[1]) have demonstrated a propensity toward collaborative activities [27-29]. Yet, these students often have the perception that information technology (IT) careers overwhelmingly involve solitary activities. Educators reinforce this perception when insisting that students work on programming assignments alone, particularly in comparison with the "lab group" model common in many science courses. *Now, IT educators must create a collaborative, socially-engaging environment that appeals to the current generation of students and that paints a more realistic picture of the collaborative nature of professional IT careers.*

Between 2003 and present, we have conducted a longitudinal, collective case study to ascertain the effects of a collaborative pedagogy on student perceptions of IT careers. The study has been conducted with third and fourth year software engineering courses at North Carolina State University (NCSU); North Carolina A&T (NCAT), a historically black college/university; and Meredith College, a women's college. These students are at a critical juncture in deciding whether or not to actually pursue an IT career upon their impending graduation.

In this paper we present the latest phase of our longitudinal study. Previous phases of our study [5, 31] have focused primarily on women and non-ethnic minority[2] students. In this paper, we switch our focus and discuss the important social factors created by pair programming and collaboration as gathered from interviews with 11 African-American Millennial students at NCAT. The emergent themes among the African American Millennials are similar to those found in our prior phases of our study but also provide additional important insight into the support structures created by a collaborative development environment.

We present our findings in the context of the Social Interaction Model of Pair Programming (SIMPP v3), a conceptual framework for characterizing the driving forces through which pair programming and collaboration can help attract and retain students in IT. Previous versions of the SIMPP model were developed from interviews with seven ethnic majority and two ethnic minority students. We expand and further validate the

[1] Other names for the Millennial generation include: Generation Y, Net Generation, and Echo Boomers.

[2] In our studies, we consider ethnic minorities in computer science to be African American, Hispanic, and American-Indian students.

SIMPP model by adding and integrating the responses of the 11 African American Millennial students. We then use the SIMPP conceptual framework to make conjectures about the impact of the social interaction realized when students pair program.

The remainder of the paper is organized as follows: Section 2 contains background and related work; Section 3 describes our research methodology; Section 4 describes the original SIMPP conceptual framework; Section 5 discusses emergent themes among African American Millennial students; Section 6 presents the evolved SIMPP model based on responses from the 11 African American Millennial students in our study; and Section 7 presents our conjectures. Finally, we provide discussion and our conclusion in Section 8.

2. Background

In this section, we provide background and related work on the sociological preferences of the Millennial generation, underrepresented ethnic minorities, and women. We also provide related information on agile methodologies and pair programming.

2.1. Preferences of the Millennial generation

Both men and women thought the [IT] industry boring and not very appealing . . . could be due to the geeky image of the industry and the direct conflict between the values and desired lifestyles of the Millennials and the perceptions of the IT industry[3] *[35].*

Millennials' learning preferences tend toward teamwork and experiential learning, and their strengths include a collaborative style of working [27, 28]. They are accustomed to working in teams with people with whom they "click," thereby fostering the social aspects of work [29]. Millennials work best in office space set up physically to share ideas. They learn better through discovery than by being told, and they prefer learning through participation rather than by learning by being told what to do [28]. Results and actions are considered more important than the accumulation of facts [27, 28]. Educators are challenged to employ pedagogical techniques that appeal to the values and learning styles of Millennial students and, in the process, to create an educational environment that is more representative industry's collaborative environment.

2.2. Preferences of underrepresented minorities

Research has shown that the success rate of underrepresented minorities in science courses can be dramatically improved by shifting the learning paradigm from individual study to one that capitalizes on group processes, such as student work groups and student-student tutoring [26]. Treisman [34] performed an experiment with African American and Hispanic calculus students. He found that about 60% of minority students who had completed calculus at UC-Berkeley in the preceding decade received grades of D or F. Prior social norms had taught these minority students that they needed to study alone because working with others was cheating. In contrast the most successful Asian students formed "study-squads" to get through calculus, groups in which social status was increased by one's ability to help others.

Treisman told his students they were required to do peer checking of each other's work. Students also worked in collaborative groups during class. Only about 4% of the minority students completing Treisman's "workshop" calculus made a D or F, versus the 60% earlier [14].

2.3. Preferences of women

Geeks are taken to mean young men, working solo for long hours with just a terminal to communicate with . . . this image does not appeal to women [35].

Prior studies, including those of [1, 13, 23], indicate that female students are concerned about the insularity of working alone for long periods of time, as they perceive to be the case with computer science and IT careers. Additionally, women believe that a career in computing or software involves a lifestyle that is not well rounded or conducive to family life [13]. Margolis and Fisher cite instances where undergraduate women's confidence in their abilities to succeed in computer science declines as they feel they are spending long hours on assignments that fellow students appeared to finish in comparatively little time [24]. In theory, pair programming can help boost women's confidence in two ways: (1) when pairing, women become familiar with how long their fellow students actually work and how much they actually know (or do not know); and (2) together, the pair tends to more easily figure out an assignment and to finish it faster [24, 37, 38].

2.4. Agile software development and pair programming – a pedagogical intervention

Agile software development methodologies are a subset of iterative and evolutionary methods [17, 18] and are based on iterative enhancement [2] and opportunistic development processes [11]. Agile methods are highly collaborative in nature and thus may be an avenue for meeting the learning needs of Millenials. The Manifesto for Agile Software Development and the Principles Behind the Agile Manifesto [4, 12] document the strong

[3] Based upon a survey of 181 university business students (56% men, 44% women).

emphasis on interpersonal communication and cooperation within agile methodologies. For example, the Manifesto states that individuals and interactions are more important than processes and tools; and that customer collaboration is more important than contract negotiation. Social interaction is also emphasized in agile methodologies. Software developers sit in close proximity to their teammates, collaborate on all tasks, and make agreements face-to-face rather than through signed-off documents [8].

Pair programming is a practice of the Extreme Programming (XP) [3] methodology. Much of the research on pair programming in an academic environment has concentrated on establishing the efficacy of the practice for educating students. Studies conducted at NCSU [5, 6, 20, 22, 25, 31] have shown that pair programming creates an environment conducive to more advanced, active learning and social interaction, leading to students being less frustrated, more confident, and more interested in IT. Further, a study of undergraduate students at Pace University found a positive correlation between out-of-class collaboration and student achievement based on student projects and examination grades [15].

3. Methodology

In this section, we discuss the details of the software engineering course at NCAT and the student participants. The research design and method of inquiry are also addressed.

3.1. Course description

Two sections of a junior- and senior- level software engineering course at NCAT contributed data to this phase of the study[4]. The courses took place in the Spring 2005 and Spring 2006 semesters. There were approximately 20 students enrolled in the course each semester. Students attended three 50-minute lectures each week. Most weeks, one of the 50-minute lectures was replaced by a laboratory session. The laboratory for this course was set up for pair programming. The tables, arranged in a ring, have enough space for the computer equipment and accommodate the seating arrangements of the programming pair.

Students completed three solo assignments (two programming); seven pair assignments (including an analysis and design assignment); and one six-week term project in teams of three to four. Each solo assignment is completed in one week. The duration for paired assignments ranged from one week to 10 days. The shorter assignments involved the development of small projects where each assignment focusing on a different area of the software development process. These assignments were completed using an object-oriented development model in C++. Optionally, the XP methodology was utilized for the team project.

3.2. Research design

To examine multiple students' perspectives, we used a collective case study approach, looking for emergent themes in perceptions of computer science students. With collective case studies, several cases are studied within the same project to inquire into a particular phenomenon [32]. Collective case studies are based on the analysis of several individuals within a specific domain and the dynamics that emerge from that domain [10]. We define our domain as "social interaction" and analyze the dynamics resulting from instructional and programming approaches used by the students. These dynamics were examined from the point of view of the SIMPP (explained in Section 4) developed during our longitudinal study [5].

3.3. Method of inquiry

The primary sources of data were semi-structured interviews that were audio recorded and transcribed. Semi-structured interviews are conducted with a written protocol, but allow the interviewer to ask further questions based on the interviewees' responses [30]. Each interview lasted approximately 30 minutes. The transcriptions were independently coded by three researchers; the individual codings were then combined and compared by the third author. Coding is the process that categorizes qualitative data into different themes via three steps: open coding, axial coding, and selective coding [31, 33]. Open coding is the process of identifying the categories in the data and the properties of the different categories. Axial coding is used to connect the categories and find their interrelationships. In the last step, selective coding identifies one or two central categories and forms a conceptual framework, in our case the SIMPP, from which to generate conjectures.

3.4. Participants

Previously, the SIMPP had been developed and evolved through analysis of the input of nine Millennial students (seven majority/two minority) from NCSU and NCAT. Table 1 shows the progression of subjects in the creation of the SIMPP models. For this phase of the study, we wished to determine if the SIMPP model applied to African American students as well. Including the students in this phase of the study, the SIMPP has integrated interview responses from 20 representative students, including 11 women/nine men, and 13 minority/seven majority students. The focus of our longitudinal research project is the success and retention

[4] Information on the courses and students involved in building previous versions of the SIMPP model may be found in [5], [32].

of women and minorities in IT. Therefore, our aggregate sample of twenty students is approximately half female and half minority, which is not representative of the population distribution at large.

Table 1. SIMPP models and participants

SIMPP	# men	# women	# majority	# minority
v.1	0	3	1	2
v.2-v.3	4	2	6	0
v.3-R	5	6	0	11
Total	9	11	7	13

In this paper, we focus on the analysis of fifteen interviews with 11 African American students (six women/five men) from NCAT who were enrolled in the Spring 2005 and Spring 2006 software engineering courses. The students in this study were solicited randomly by the course instructor from all of the students enrolled in the courses. The choice of interviews used in our analysis was based on theoretical sampling [10]. Theoretical sampling is used in case studies where cases are chosen based on their likeliness to extend or reproduce an emergent theory. In our study, we are evolving the SIMPP. Theoretical sampling is first done with a homogenous group to develop a theory or model, which was done in the initial phase of study. Then the sampling is applied to a heterogeneous group to confirm or disconfirm the conditions of the previously-developed model.

The six female and five male participants from NCAT were interviewed about their experiences in the course concerning social interaction and group work. Of the female participants, Vanessa, Kayla, and Crystal (all names are pseudonyms) were interviewed twice, once at the beginning of the semester and then again at the end of the semester in Spring 2005. Teresa, Marie, and Sandra were interviewed once at the end of Spring 2006. Only one male participant, Shelton, was interviewed twice in Spring 2005. Two other participants, Nathan and Trevor, were both interviewed at the end of the semester in Spring 2005. Greg and Ryan were both interviewed at the end of Spring 2006.

4. Social Interaction Model of Pair Programming (SIMPP)

The conceptual framework for our study is based on situated cognition [5, 19], where students learn in collaborative apprenticeships that reflect real world practice; and on social constructivism discourse studies [36], where students' dialogues within themselves and with their peers contribute to their learning experiences. We examine their perceptions of their learning experience as students move beyond these initial peripheral apprenticeships toward more responsibilities of an IT worker.

The initial version of the SIMPP [5] was initially developed from a case study of three interviews with female students who took the software engineering course at NCSU in the Fall 2003 semester. The categories evolved from the initial version of the SIMPP model to the second version [31] as input from more students became available. The current version of the model, SIMPP v3, is similar to the second version with only nomenclature differences.

The SIMPP v3, shown in Figure 1, describes five interrelated components of social interaction in pair programming that emerged from past student interviews. Each component has a bi-directional relationship with *all* other components, that is, a change in one factor (for better or for worse) will perpetuate a change in the others. Yet, there is no ordering among the components – their relationship is circular. For example, students believe that project quality and confidence are intimately tied. As the quality of their project improves through collaboration, so does their confidence. Yet, because of the confidence in their solutions, they produce better future projects.

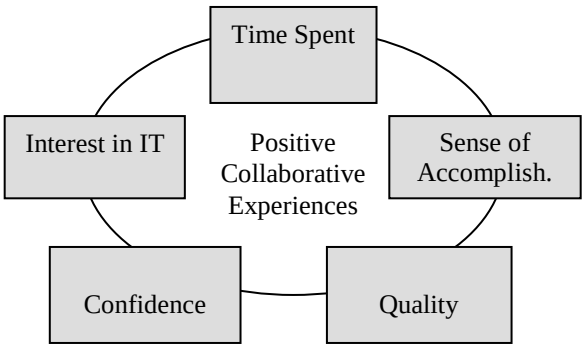

Figure 1. Social Interaction Model of Pair Programming v3 (SIMPP v3)

Students observed that errors in programming and rationale were corrected more quickly while working with others. Also, partners often had a solution to an arising problem. Both of these aids reduced the *time spent*. According to the students, the positive influence of social interaction on the time factor yielded a greater *sense of accomplishment* and also a *higher quality* product. Creating higher quality products in less time increased the students' *confidence* in their abilities. We propose that these four factors fostered or retained these students' *interest in IT* careers. By producing a higher quality product in less time, a student's confidence should increase, which in turn, should foster more interest in that particular activity, a concept supported by [1, 24].

In the remainder of this paper, we will examine the interviews of the 11 African American Millennial students

in our study to see if their responses support or contrast with the components of the SIMPP model. We will then expand and revise the model to incorporate this new information.

5. Emergent themes among African American Millennials

In this section, we explore four themes that emerged in interviews of the eleven African American Millennial students at NCAT. These themes were consistent with the previously-reported interviews, further validating the SIMPP model itself. Furthermore, these new interviews revealed additional important characteristics of the collaborative factors in the SIMPP.

5.1. Time and quality

Our previous findings suggest that time is an important factor in students' experiences with social interaction. Some students believe working together takes more time, but results in a better product, while others believe collaborating saves time over working alone by catching errors. The African American Millennials in our current study share similar ideas about time. On the one hand, collaborating with others saves time. Shelton says that working with others saves time on large projects:

… I guess it's best to work with another person because it might cut down the time that you really have to work… but when it comes to a major project you'll probably need some definite help. If I'm by myself I'll be intent on that one track until I get deeper into it and then it's wrong. Having someone else they're not necessarily standing over your shoulder but on the same page as you're on, you can share those ideas and perhaps have a better product.

Nathan's experience of working with his group together in the lab illustrates the importance of teamwork in solving problems quickly and efficiently:

…we can just kind of talk to each other while we're programming it, that way, eliminate some of the problems that we could have by each of us going out on our own and trying to make something up. I think that way there's more teamwork… I think working together in the lab has allowed us to catch our mistakes before we've actually thought we were done with something… I thought that was efficient, catching errors before we got too far.

For Shelton, saving time on large projects implies getting help which could result in a higher quality product. Marie would rather work with a partner on assignments because it saves her time:

I like it better than working by myself…Because I might understand it one way, and then the way I'm looking at it is the longer way to get the final answer, but then they have a shorter way, which is the easier way…it saves time…a lot less time.

On the other hand, collaboration can take more time than working alone. Coordinating schedules and ideas is a concern for Nathan:

I think one advantage of working individually is that since you have no one to disagree with, you'll probably get it done a little bit quicker, but since you have to coordinate your schedule and coordinate ideas and make sure that you agree on certain topics, certain issues, that I think it is more time consuming. But it's worth it in the end because you will have a better product from all the brain power going into it.

Vanessa feels that collaborating in the workplace is a time-saving practice where problems are solved faster:

I: How do you think that working in groups or with a partner - do you think that has a place in the IT work place?

V: Of course. Two heads are better than one. I mean if you're having a problem if you have more than one person working on it, more brains focused on this one problem it's going to get solved quicker.

As in our previous studies, we find that the relationships between time spent, higher efficiency, and higher quality are important factors to consider when examining students' perceptions of social interaction. Whether working together takes more time or less time, students feel that they end up with a better product overall and avoid more errors.

5.2. Peer learning

Our previous studies [5, 22] reveal the multifaceted category of higher quality resulting from collaborative experiences. Relating higher efficiency and higher quality, students in this and previous studies mentioned fewer errors, more ways of solving problems, including more elegant methods, and filling in gaps and missing pieces of ideas. Learning is again an important theme contributing to students' perceptions of higher quality.

Eliminating and avoiding problems while collaborating with their peers contribute to African American Millennial students' sense of accomplishment. Furthermore, being more efficient contributed to the learning experiences of these students. Vanessa prefers to work in a group over working solo:

…yes I'd rather work in a group. You get more done and you get to see how other people see things. Maybe learn some new things at the same time.

Similarly, Kayla feels that she learns more efficiently by working with others:

You can share each other's knowledge like sometimes it's better to learn from your peers than from the instructor because they might know how to relay it better. So when you're working in groups you're also learning.

Teresa also comments on her learning from her peers:

Something in particular, I guess one thing I'm kind of a little bit having trouble on, which I know will be on the final, is like state chart diagrams…I've been using my peers to find out exactly how to do it and make sure I'm doing it the correct way.

Shelton connects this idea of learning from others to his vision of the workplace:

Yes, I think it should be used in the workplace…Different ideas. One person might have one opinion and then somebody else will have one. And then I guess if you put it together there might be something new that one person couldn't have done all by themselves.

Learning from the different ability levels of his peers led Shelton to comment:

Everyone looks at something a little differently. I don't know too many people that truly write programs the same unless they're copying each other. So everyone looks at the problem or the approach to solving a problem differently and that can bring in different ideas good or bad and you get a learning experience.

Crystal appreciates the social aspect of learning from others with different ability levels:

Well nobody wants to do their work by themselves the entire time so I think the company of everyone is always nice… You know, see what you can learn from other people. That's definitely helpful. Because you know some of my weaknesses can be someone else's strength. So they'll kind of coincide and balance each other out.

Sandra comments on the importance of having others' perspectives:

…when you have pairs programming, you might think it's right, but you might have somebody else who's saying, oh, this isn't right, or maybe we need to take another approach. So, you have another set of eyes and another brain that could either help, you, or it could probably affect the grade. But, you have someone saying, well, maybe we should look at this differently. So you have somebody else who can make you think outside the box than what you're already thinking in.

Similarly, Kayla likes the quality of learning from communication with her peers instead of seeking out the teacher for help:

I'd say I prefer to work in a group just because you'll always have help if it's needed. If you're not clear on something, usually somebody in your group would be able to help you out with it without always running to the teacher. It's more peer-to-peer relationships going… You can share each other's knowledge like sometimes it's better to learn from your peers than from the instructor because they might know how to relay it better. So when you're working in groups you're also learning.

Ryan appreciates the importance of communication in working with others:

…the main thing in groups, just in general is how to listen…Working in groups, you listen and then you learn that way. Like me, personally, like I said, I'm a person who's an eye, ear and see kind of guy. So, if I'm in a group and somebody says, well, why don't we do it like this, and then they can back it up by showing us a picture or something, that's my ears and my eyes already seeing it and then hands-on, if you can make it, that's something I like to see. So, working in groups – the main thing, I think about working in groups is communication. If you all can communicate right, you pretty much can do any job.

In other words, learning from others contributes to higher overall sense of accomplishment and confidence by helping these students to see more efficient ways of approaching a problem.

5.3. Real Practice

Students in this study felt they were learning social and communication skills that would help them once they enter the workforce. That is, these students value the practical applications of social interaction.

Crystal comments on the development of communication skills:

And also like even if you knew a lot, if you were knowledgeable on it, it helps you as far as the communication to get it across. Like this is what it is. So I guess it's more of a helpful thing… I definitely think it's helpful because if you're not used to working with other people to get a project done then once you go off to work and you have to work with a lot of people that could mess you up in terms of how you're used to working. I guess

that's the big plus within the course. That it does teach you that you do have to work within a group to get stuff done.

Kayla also comments on learning communication skills and their importance in real practice. When asked what she had learned from collaborating in the course she commented:

I guess communication skills. They talk about it a lot like when you get in the corporate world, group work is also a necessity so that was just so you will know how to work in a group. Experience in working with groups.

Nathan took this idea further by adding that he learned social skills as well:

I guess I've been learning a lot from the two people I've been working with. Not just from a computer programming aspect, but also just social skills and just relating with people. So I'm learning things about life as well.

Students noticed that the closeness of the setting of their apprenticeship to workplace environments increased their competence, which in turn led to their increased interest in the IT field.

Nathan notes that interviewers ask him about his collaborative abilities:

Yeah, from the way I've been doing interviews and stuff, over the course of the semester, it seems like a lot of their focus has been on group work, how you feel about working in a group, or working in teams, have you ever worked in teams, what would you do if you don't agree with someone, stuff like that. So it seems like the next step in the industry will be working in a team in some way or some sort.

He continues with how his collaborative experiences in the software engineering course are helping him prepare for a career:

So I really feel that they've been beneficial to me, as far as getting a feel of how things will be done in the industry, because you won't be working by yourself on a particular assignment, you'll have other people working with you to accomplish the goal, as far as interacting with these people to accomplish this goal and being able to talk to other people and bring in different viewpoints in order to accomplish the task.

Shelton, on the other hand, explains how the course fits his vision of the workplace:

More group effort. A bunch of group work. It's not going to be like any individual work. I guess it gave me a chance to work with more people. Because I know that's how it's going to be in the real world when I go get a job.

Vanessa also values the relationship of her course experiences with her vision of the workplace:

I like the idea that we got to work in groups and we got to learn more about the software development life cycle. And we learned more things that we would actually be applying once we get a job. And that's what I liked about the course… Like how we get to work…like if we were to go out and actually get a job, most likely we'll be working on a software development team. And getting to work in groups and learning the life cycle and all those things I think it helped prepare us more for a real work experience.

Likewise, Ryan appreciates the relationship between his idea of the workplace and the Computer Science major:

I'd definitely say I like groups the most. Because most of the time when you get out now, in the real world, in a lot of organizations, you're going to work with groups…you definitely have to be able to do groups. As of now, with this kind of major, you don't do much by yourself. You think, but you implement with others.

These students valued their increased competence in social and communication skills. They see the need to learn beyond basic technology skills and are confident that their newly learned skills are practical.

6. Expanding the model

The additional data provided by the 11 interviews reported in this study supported the categories of the SIMPP and did not surface inconsistencies between these minority students and the prior majority students. The additional data, however, highlighted complexities within the categories of the SIMPP and allowed us to develop further relationships between categories. Thus, we expand our model to the new SIMPP v3-R (see Figure 2) where the R indicates that relationships are now shown. Relationships are denoted as ovals.

The issue of peer *learning* relates to the categories of higher productivity, higher quality and increased confidence. Students believed they learned more efficient ways of approaching problems, had fewer errors, and were able to fill gaps in their own knowledge when they worked with other students.

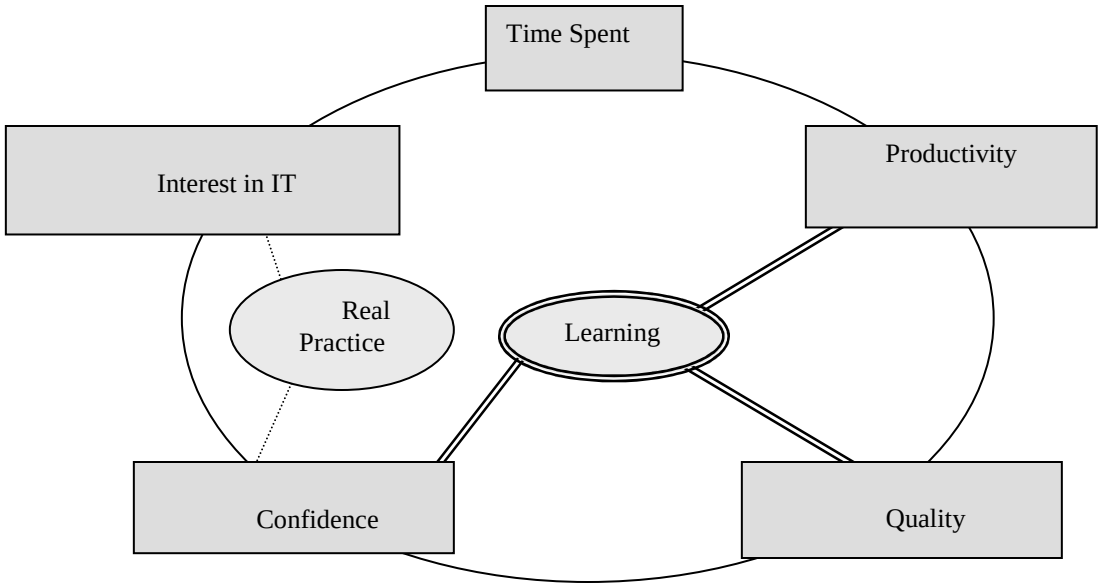

Figure 2. SIMPP v3-R

Furthermore, our new results implied that the students felt more confident and empowered to learn when they were able to ask their peers questions. The confidence also increased through the closeness between classroom practices and workplace practices, that is, through class practice being close to *real practice*. Students valued the connection of these collaborative practices to the workplace. This connection, as well as growth in the quality of their work and in the productivity, increased their interest in working in the IT field.

7. Conjectures on attracting and retaining African American Millennial Students in IT

Based upon our findings, we propose three conjectures about how social interaction can help attract and retain African American Millenial students in IT. The type of qualitative inquiry reported in this paper does not produce generalizable results as can be done with some quantitative analyses. However, qualitative research can draw implications, propose theories, or promote conjectures. Larger quantitative studies can then be designed to evaluate the generalizability of the conjectures. Initial quantitative validation of the SIMPP v2 model has been conducted [22, 31].

Conjecture 1: *Through social interaction in course activities, African American Millennial students change their perspective of IT as a solitary activity to IT as a collaborative activity.*

As stated in Section 2, women [1, 13, 23] and Millennials [35] generally perceive IT to be a discipline involving a great deal of solitary activity. Often, instructors of early courses in a computer science curriculum insist that students work alone. The first author conducted an informal email surve[5] of the SIGCSE[6] mailing list. Based upon 50 responses, the majority of educators felt that students needed to primarily work solo on their programming assignments in the first two years of study so they can learn the fundamentals of programming and develop their own skills. As such, the solo work requirement affirms the student view of "solitary work" in IT, potentially contributing to the retention problems of the computer science discipline. Conversely, a survey of 359 practitioners[5] indicates that software engineers spend almost 40% of their day working with others; the solitary work perception is not accurate.

Our interview data indicate that by the end of the software engineering course, the students change their perception. The students consistently verbalize their view that the collaborative work done in the class is representative of what they will do in industry, indicating a new perspective on IT as a collaborative activity.

Conjecture 2: *Social interaction improves African American Millennial students' interest in IT because it*

[5] Surveys conducted in preparation for keynote speech "Debunking the Geek Stereotype" presented at CSEET 2005 http://www.site.uottawa.ca/cseet2005/KeynotesPDF/LaurieWilliamsKeynoteCSEET2005.pdf

[6] ACM Special Interest Group on Computer Science Education; http://www.sigcse.org/

helps them to learn both technical and interpersonal skills, which in turn makes them feel more confident and better prepared for the "real world."

When students work in groups, they are continuously brainstorming and discussing alternatives. The act of self-explanation has been found to help organize thoughts and be beneficial in that it causes the speaker to better understand the subject at hand or to improve (and correct) his or her own imperfect mental model [7]. As such, learning occurs to both the speaker and the listener in the discussion. Students prefer to learn from each other over learning from a teacher [27, 28]. In the process of these discussions, students are learning more about communicating in the context of a software development team. From these increased technical and communication skills, students can be more confident that they will be more successful in the "real world."

Conjecture 3: *Social interaction improves African American Millennial students' interest in IT because it is a preferred work style when compared with working solo.*

Our data supports prior findings that Millennial students want to work together and learn together. Computer science students are often stereotyped as introverts. However, one of the classes in this study was comprised of about 70 students of which almost half were Myers Briggs [16, 21] extroverts and more had low introvert scores. In a five-year study of 23 co-educational computer science departments in Virginia, face-to-face interviews revealed that both men and women consider their classmates a necessary source of help [9]. Structured social interaction, such as is provided by pair programming and agile software development, allows students to work as they desire.

8. Summary

Studies indicate that Millennial students, particularly women and minorities, desire a collaborative, interpersonal environment. Increasingly, these students consider IT as boring and not appealing. In our longitudinal research study, we examine if and in what ways pair programming and collaboration can help to attract and retain students in IT disciplines. In the past two years, software engineering courses at NCSU and NCAT have incorporated a pedagogical intervention emphasizing collaboration, structured in the collaborative aspects of agile software development and its practice of pair programming.

We conducted a series of semi-structured interviews with students in these classes. Through analysis of these interviews, we have formulated a conceptual model we call the SIMPP. In this paper, we specifically discuss the themes and relationships that emerged from the interviews of 11 African American students. We use these themes to further develop the SIMPP into a more generalized model that reflects the views of men, women, ethnic minorities, and ethnic majorities. We further make conjectures of how pair programming and agile development-style collaboration impacts student views of IT. Our findings suggest that pair programming and agile software methodologies effectively create the collaborative environment desirable to Millennial students. Additionally, the students enjoy learning from their peers and feel a collaborative environment better prepares them for the "real world." Future research will evaluate and evolve these conjectures with additional students via additional qualitative and quantitative studies.

Acknowledgements

This material is based upon the work supported by the National Science Foundation under the Grant No, 00305917. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation. We would like to thank Sung Yoon, the instructor at NC A&T who taught the class and convinced his students to participate in the interviews. Additionally, we thank Stephen Czenszak for his assistance with coding the interview.

References

[1] American Association of University Women Education Foundation, *Educating Girls in the New Computer Age*, 2000, http://www.aauw.org/member_center/publications/TechSavvy/TechSavvy.pdf, accessed

[2] V. R. Basili and A. J. Turner, "Iterative Enhancement: A Practical Technique for Software Development," *IEEE Transactions on Software Engineering*, vol. 1, no. 4, 1975, pp. 266 - 270.

[3] K. Beck, *Extreme Programming Explained: Embrace Change*, Second ed. Reading, Mass.: Addison-Wesley, 2005.

[4] K. Beck, M. Beedle, A. van Bennekum, A. Cockburn, W. Cunningham, M. Fowler, J. Grenning, J. Highsmith, A. Hunt, R. Jeffries, J. Kern, B. Marick, R. C. Martin, S. Mellor, K. Schwaber, J. Sutherland, and D. Thomas, "The Agile Manifesto," http://www.agileAlliance.org, 2001 2001.

[5] S. B. Berenson, K. M. Slaten, L. Williams, and C.-w. Ho, "Voices of Women in a Software Engineering Course: Reflections on Collaboration " *ACM Journal on Educational Resources in Computing*, vol. 4, no. 1, 2004, pp. 1-18.

[6] S. B. Berenson, L. Williams, and K. M. Slaten, "Using Pair Programming and Agile Development Methods in a University Software Engineering Course to Develop a Model of Social Interactions," Crossing Cultures, Changing Lives Conference, Oxford, UK, 2005, pp. to appear.

[7] M. T. H. Chi, "Self-Explaining: The Dual Process of Generating Inference and Repairing Mental Models," in *Advances in Instructional Psychology*, R. Glaser, Ed. Mahwah, New Jersey: Lawrence Erlbaum Associates, 2000, pp. 161 - 238.

[8] A. Cockburn, *Agile Software Development*. Reading, Massachusetts: Addison Wesley Longman, 2001.
[9] J. M. Cohoon, "Toward improving female retention in the computer science major," *Communications of the ACM*, vol. 44, no. 5, 2001, pp. 108-114.
[10] J. W. Creswell, *Qualitative inquiry and research design: Choosing among five traditions.* Thousand Oaks, CA: Sage, 1998.
[11] B. Curtis, "Three Problems Overcome with Behavioral Models of the Software Development Process (Panel)," International Conference on Software Engineering, Pittsburgh, PA, 1989, pp. 398-399.
[12] M. Fowler and J. Highsmith, "The Agile Manifesto," in *Software Development*, August 2001, pp. 28-32.
[13] P. Freeman and W. Aspray, "The Supply of Information Technology Workers in the United State," Computing Research Association, Washington, DC, 1999.
[14] R. E. Fullilove and P. U. Treisman, "Mathematics achievement among African American undergraduates of the University of California, Berkeley: An evaluation of the Mathematics Workshop Program," *Journal of Negro Education*, vol. 59, no. 3, 1990, pp. 463-478.
[15] A. Joseph and M. Payne, "Group Dynamics and Collaborative Group Performance," Thirty-fourth SIGCSE Technical Symposium on Computer Science Education, Reno, NV, March 2003, pp. 368-371.
[16] D. Keirsey, *Please Understand Me II*. Del Mar, CA: Prometheus Nemesis Book Company, 1998.
[17] C. Larman, *Agile and Iterative Development: A Manager's Guide*. Boston: Addison Wesley, 2004.
[18] C. Larman and V. Basili, "A History of Iterative and Incremental Development," *IEEE Computer*, vol. 36, no. 6, June 2003, pp. 47-56.
[19] J. Lave and E. Wenger, *Situated Learning: Legitimate peripheral participation*. New York, NY: Cambridge University Press, 1991.
[20] L. Layman, "Changing Students' Perceptions: An Analysis of the Supplementary Benefits of Collaborative Software Development," 19th Conference on Software Engineering Education and Training (CSEE&T '06), Turtle Bay, Hawaii, 2006, pp. 156-166.
[21] L. Layman, T. Cornwell, and L. Williams, "Personality Types, Learning Styles, and an Agile Approach to Software Engineering Education," ACM Technical Symposium on Computer Science Education (SIGCSE '06), Houston, TX, 2006, pp. 428-432.
[22] L. Layman, L. Williams, J. Osborne, S. Berenson, K. Slaten, and M. Vouk, "How and Why Collaborative Software Development Impacts the Software Engineering Course," Frontiers in Education, Indianapolis, IN, 2005, pp. T4C 9-14.
[23] J. Margolis and A. F. (1997), "Geek Mythology and Attracting Undergraduate Women to Computer Science," Joint National Conference in Engineering Program Advocates Network and the National Association of Minority Engineering Program Administrators., 1997, pp.
[24] J. Margolis and A. Fisher, *Unlocking the Clubhouse: Women in Computing*. Cambridge, Massachusetts: The MIT Press, 2002.
[25] N. Nagappan, L. Williams, M. Ferzli, K. Yang, E. Wiebe, C. Miller, and S. Balik, "Improving the CS1 Experience with Pair Programming," ACM Special Interest Group Compuer Science Education (SIGCSE) 2003, Reno, 2003, pp. 359 - 362.
[26] C. E. Nelson, "Student Diversity Requires Different Approaches to College Teaching, Even in Math and Science," *American Behavioral Scientist*, vol. 40, no. 2, 1996, pp. 165-175.
[27] D. Oblinger, "Boomers, Gen-Xers, and Millennials: Understanding the New Students," *Educause Review*, vol. 38, no. 4, July/August 2003, pp. 37-47.
[28] D. Oblinger and J. Oblinger (Eds.), *Educating the Net Generation*. Boulder, CO: Educause, 2005.
[29] C. Raines, *Managing Millennials*, 2002, http://www.generationsatwork.com/articles/millenials.htm, accessed February 12, 2006.
[30] C. B. Seaman, "Qualitative Methods in Empirical Studies of Software Engineering," *IEEE Transactions on Software Engineering*, vol. 25, no. 4, 1999, pp. 557-572.
[31] K. M. Slaten, M. Droujkova, S. Berenson, L. Williams, and L. Layman, "Undergraduate Student Perceptions of Pair Programming and Agile Software Methodologies: Verifying a Model of Social Interaction," Agile 2005, Denver, CO, 2005, pp. 323-330.
[32] R. E. Stake, *The Art of Case Study Research*. Thousand Oaks, CA: Sage Publications, Inc., 1995.
[33] A. L. Strauss and J. M. Corbin, *Basics of Qualitative Research: Techniques and Procedures of Developing Grounded Theory*, Second ed. Thousand Oaks, CA: Sage Publications, 1998.
[34] U. Treisman, "Studying students studying calculus: A look at the lives of minority mathematics students in college.," *The College Mathematics Journal*, vol. 23, no. 5, 1992, pp. 362-372.
[35] E. Walker and L. O'Neill, "IT courses and the IT industry: Does the future rely on gender or generation?," Annual International Conference of the Higher Education Research and Development Society of Australasia (HERDSA), Perth, Australia., 2002, pp. 695-703.
[36] J. L. Wertsch and C. Toma, "Discourse and learning in the classroom: A sociocultural approach," in *Constructivism in Education*, L. Steffe and J. Gale, Eds. Mahwah, NJ: Lawrence Erlbaum, 1995, pp. 159-174.
[37] L. Williams and R. Kessler, *Pair Programming Illuminated*. Reading, Massachusetts: Addison Wesley, 2003.
[38] L. A. Williams, "The Collaborative Software Process," University of Utah Department of Computer Science, Salt Lake City, UT, PhD Dissertation, 2000.