Spring 2011

Gender Equality With Agile In Software Engineering

Deborah P. Martin
Regis University

Regis University
College for Professional Studies Graduate Programs
Final Project/Thesis

Disclaimer

A THESIS

SUBMITTED ON 30th DAY OF MAY, 2011

TO THE DEPARTMENT OF INFORMATION SYSTEMS

OF THE SCHOOL OF COMPUTER & INFORMATION SCIENCES

OF REGIS UNIVERSITY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS OF MASTER OF SCIENCE IN

COMPUTER INFORMATION TECHNOLOGY

BY

Deborah P. Martin

APPROVALS

Dolores Bilo, Thesis Advisor

Shari Plantz-Masters

Richard Blumenthal

Abstract

Women are under-represented in Information Technology careers in general and in the profession of software development in particular and their numbers are declining. Previous studies on this topic have investigated whether the reason for this is due to a difference in capability arising out of their gender, or whether the reason arises from social factors. A software development methodology called "Agile" has arisen in recent years which focus on collaboration, working software, and a sustainable workweek. Studies have shown that adopting Agile techniques in the classroom helps to retain women in computer information academic curricula and that adopting them in workplace teams improves communication across gender and ethnic diversity. This study invited women to speak for themselves as to whether they find Agile engineering techniques helpful in the workplace through the use of an online survey and follow-up interviews. The results revealed that women feel positively about these practices but have encountered some resistance to adopting them. Future studies discerning whether there are differences in the attitudes of men and women to these practices or whether there is a correlation between adoption of these practices and the numbers of women practicing as software developers in varying environments are logical future research topics to extend understanding in this area.

Acknowledgements

With gratitude to several women who helped me get this far: Dee Bilo, my Thesis Advisor, who provided both helpful suggestions and encouragement to keep me going, my sister, Amy, and my friend and music instructor, Tina, both of whom provided considerable emotional support and practical advice during some difficult times in my life. But most especially, I am grateful for my two daughters, and hope that they will be proud of this work. I also hope that the world in which they work will be more egalitarian than the one I grew up in.

Table of Contents

Abstract ii
Acknowledgements iii
Table of Contents iv
List of Figures vi
List of Tables vii
Chapter 1: Introduction 8
Women are underrepresented in the software programming profession and their numbers are declining. 8
Some postulated reasons why so few women end up writing code for a living. 2
Why female under-representation in software engineering is a problem 3
Agile Engineering Practices as Promoters of Gender Equality in Software Engineering 3
Chapter 2: Literature Review 4
Background on Agile Methodology 4
Lack of Diversity on Software Development Teams 7
Essentialist vs. Social Constructivist Theories 8
Computing with a Purpose 9
Conflicting Family Responsibilities 11
Hostile Environments 12
Women and Confidence 13
Diversity Equalization Using Agile practices 16
Gap in the Research 18
Chapter 3 – Methodology 19
Chapter 4 –Results 23
Attitudes and Experience with Pair Programming 23
Attitudes and Experience with Test Driven Development 27
Attitudes and Experience with Automated Build/Continuous Integration 30
Attitudes and Experience with the Planning Game 34
Some risks of the study 37
Chapter 5 – Conclusions 38
Interview Responses 40
References 42

List of Figures

Figure 1: Experience Spread of Respondents 23
Figure 2: Pair Programming and Quality 25
Figure 3: Pair Programming Skill Expansion 25
Figure 4: Pair Programming and Job Satisfaction 26
Figure 5: Pair Programming and Productivity 26
Figure 6: TDD and Productivity 28
Figure 7: TDD and Perception of Quality 29
Figure 8: TDD and Skill Expansion 29
Figure 9: TDD and Job Satisfaction 30
Figure 10: Continuous Integration and Productivity 32
Figure 11: Continuous Integration and Quality 32
Figure 12: Continuous Integration and Skill Expansion 33
Figure 13: Continuous Integration and Job Satisfaction 33
Figure 14: Planning Game and Productivity 35
Figure 15: Planning Game and Quality 35
Figure 16: Planning Game and Skill Expansion 36
Figure 17: Planning Game and Satisfaction 37

List of Tables

Table 1: Agile Methodologies 4
Table 2: Experience with Pair Programming 24
Table 3: Experience with TDD 27
Table 4: Experience with Continuous Integration 31
Table 5: Experience with The Planning Game 34
Table 6: Comparative Positive Responses 39

Chapter 1: Introduction

It is no secret that women are underrepresented in the profession of software development. Recent Bureau of Labor Statistics (BLS) surveys show that representation is even lower as programmers than as Information Technology (IT) workers in general (Bureau of Labor and Statistics, 2009). This inequality is proven in the literature (James, 2010). There are two preponderant theories: "Essentialist"—that women themselves stay away from this field either because of a cognitive or other built-in issue, or "Social Constructivism"—that women stay away because the culture is anathematic to them (Trouth, 2002). There is a real possibility that women are scared away by an "Alpha Male" culture characterized by a peculiar form of geeky machismo (Taft, 2009).

Women are underrepresented in the software programming profession and their numbers are declining.

Participation of women in the Information Technology (IT) workforce is declining while the participation of women in the workforce at large is increasing; this is becoming a cause for concern. In 2005, 56% of the professional workforce in the U.S. were female, but only 27% of professional IT workers were female (Wardell, Sawyer, Mittory, & Reagor, 2006). The percentage of women in the IT workforce declined from 40% in 1986 to 29% in 1999 (Tapia & Kvasny, 2004). Even more significantly, women are both more likely to leave the IT workforce and also more likely to not return (Quesenberry & Trauth, 2007).

Walk into any software development department and you will see few, if any, women in the role of developer. Survey the participants in Open Source on any project, and you will see few, if any, women (Levesque & Wilson, 2004).

Some postulated reasons why so few women end up writing code for a living.

Software engineering is perceived as requiring long hours, being done in isolation in the late hours of the night, and being done by men displaying a geeky sort of machismo. The movie, "The Social Network," exemplifies this sort of environment in a scene where young men are competing for a programming position in the new Facebook venture while having to drink shots of liquor. Needless to say, no women were portrayed as competing (Eisenberg & Garfield, 2010). Women generally prefer collaborative rather than solitary and competitive work (Berenson, Slaten, Williams, & Ho, 2004).

Nondiscrimination laws cannot change the fact that it is women who bear children, still perform the vast majority of household chores, and are also usually responsible for caring for aging parents and children. A maternity leave can be followed by a very steep learning curve returning to a profession where more happens in a week than used to happen in a lifetime. An exhausted mother trying to keep up with the demands of parenting and/or coping with the demands of frail, elderly parents can easily fall behind in the challenge to keep up with new techniques, languages, and methods. In a survey of IT workers completed in October by Technisource, women indicated clearly that flexibility was important to their career choice (Technisource, 2010).

A recent study by Dr. Heidi Halvorson found that confidence is a key factor in women attempting difficult goals; her theory is that young girls are more easily discouraged by difficult material while young boys are rewarded for effort in the face of difficulty. The logical outcome is that women believe that difficulty reflects their incompetence while men believe that difficulty is something to overcome (Halvorson, 2011).

Why female under-representation in software engineering is a problem

Women have a lot to offer the profession of software engineering. Women use software as much as men, so being involved in the design and coding is likely to make software more female-friendly. Women have historically contributed in significant ways to software engineering. Ada Lovelace is known as the first computer programmer, and Grace Hopper was instrumental in the development of the COBOL language (Lockheed, 1985).

The current domestic legal, political, and cultural climate expects businesses to establish gender equality in the workplace. It is common wisdom that it is cheaper to retain an existing employee than hire and train a new one, so businesses are wise to retain female employees.

Agile Engineering Practices as Promoters of Gender Equality in Software Engineering

This research explores whether Agile software development methodologies, with their collaborative nature and focus on a sustainable workweek (40 hours), are attractive to female software engineers as a means of keeping them in the workforce.

The following questions are germane to this research.

1. Do women prefer more collaborative Agile methodologies to the traditional Waterfall methodologies?
2. Do women feel that Agile methodologies enable them to be more productive, create higher quality software or learn new skills more quickly?
3. Do women think that Agile methodologies enable women to participate more fully on a software development team?

Chapter 2: Literature Review

Background on Agile Methodology

In the mid-1990's, the Chrysler Corporation was frustrated with a stalled project to rewrite its payroll system. The leadership of the company hired some experienced developers to take over, and these men decided to use a new and radical set of engineering practices to speed up development. These practices came to be called eXtreme Programming (XP). The payroll system was thus developed incrementally, although it never ultimately paid all of the employees as was its original intent due to a breakdown in communication between competing stakeholders (Herela, 2005). Initially named "lightweight" processes, they were compared to the more "heavyweight" processes such as Rational Unified Process (RUP) which are very plan-oriented and rely upon heavy planning, design, and up-front documentation. Eventually the term "Agile" was coined and this is how these related methodologies are now known. Agile processes now include such variants as Scrum, Crystal, Feature Driven Development, and Lean (Dybå & Dingsøyr, 2008). The Dybå et al. study reviewed empirical research on the effectiveness of Agile in terms of productivity, quality and job satisfaction and found there is some evidence that Agile methods contribute to each.

Table 1: Agile Methodologies

Type of process	Description
Crystal	A family of methods for co-located teams of differing sizes
Dynamic Software Development Method (DSDM)	Three phases in projects, nine principles including iterative and incremental development and a focus on people over processes
Feature Driven Development	Focuses on objects and models with iterative design
Lean Software Development	An adaptation of a production method employed by Toyota

Type of process	Description
	which emphasizes empowering the team, eliminating waste, amplifying learning, and deciding as late as possible
Scrum	A set of practices primarily relating to project management in situations where it is difficult to plan ahead
eXtreme Programming	A set of best practices around development incorporating such techniques as pair programming, testing, and collective ownership

In 2008, Dybå and Dingsøyr surveyed the known Agile methodologies at the time, noting and describing their peculiar and unique features. These are summarized in Table 1: Agile Methodologies, above. They point out that some of the methodologies such as Scrum are focused mostly on project management, while others such as eXtreme Programming are focused on software engineering practices so these various methodologies are not mutually exclusive.

In 2001, the Agile Manifesto was developed and signed by several of the methodology founders, including Alistair Cockburn, Martin Fowler, and Ward Cunningham (Beedle, et al., 2001). This document makes clear that people are favored over processes, and working functionality is favored over documentation. All of the Agile methodologies focus on iterative development and collaborative processes (Cockburn & Highsmith, 2001). The key phases of the Agile development lifecycle are Test, Code, Refactor, then Repeat (Jeffries, 1998). A guiding principle of Agile development is: “Do the simplest thing that could possibly work” (Jeffries, 1998).

Once considered radical, experimental, and marginal, Agile processes are now becoming more readily accepted and are nearly mainstream. Heavyweight processes are now becoming

iterative. Agile processes were designed to meet rapidly changing business needs, and the need to adapt to these changing requirements is clearly is a prevailing factor in today's economy (Boehm, 2005). In his article comparing and contrasting Agile and Waterfall (heavyweight and plan-oriented) practices, Boehm cross-analyzes the following factors: project size expressed as number of staff; mission criticality relating to effect of a system failure; personnel skills ranked on three levels of capability; project dynamism reflecting a percentage of requirements change per month; and organizational culture in terms of seeking to thrive on order vs. chaos. Boehm comes to the conclusion that the iterative nature of Agile methodologies suit them in particular to the highly volatile and changing nature of business. He still advocates the use of heavier-weight and plan-oriented practices when dealing with the larger projects, projects with high cost when systems fail, and large teams. He also suggests hybrid processes called "spiral" in some situations which incorporate some elements of both Agile and Plan-Driven project management. The analysis concludes that there are appropriate projects for each methodology and that no one methodology is good for every project.

Common, well-known Agile software engineering practices include Pair Programming, Test Driven Development, Continuous Integration and Automated Builds, and The Planning Game (Martin, 2005). In their comparison of various Agile methods, Abrahamsson, et al. define Test Driven Development (TDD) as the practice of writing a broken unit test in the same language as the functional code before ever writing functional code. After the test is written, code is implemented which makes the test pass. Novices to the practice are often under the mistaken impression that TDD is all about testing rather than about design and have trouble with the concept and practice. Pair Programming is the practice of having two programmers work together on the same code and design. This is usually done side-by-side, but an acceptable

alternative is for one programmer to write unit tests while the other programmer makes them pass. Continuous Integration, also sometimes called Automated Build is the practice of having a build server which monitors the code's Source Control for changes and when new code is checked in, a build followed by the running of the automated tests is triggered. Results are then reported to the entire team. These are engineering practices as opposed to project management practices (Abrahamsson, Warsta, Siponen, & Ronkainen, 2003). The Planning Game is a practice which bridges engineering and management: team members are expected to estimate using abstract story points, a level of effort required to implement a feature. The customer or business analyst must then prioritize them (Martin, 2005).

Lack of Diversity on Software Development Teams

Software development in the U.S. has traditionally been dominated by white males; although off-shoring and the immigration to the U.S. of individuals from other countries have changed this somewhat, this is still generally obvious to the naked eye. In 2005, 56% of the professional workforce in the U.S. were female, but only 27% of professional IT workers were female (Wardell, Sawyer, Mittory, & Reagor, 2006). Female participation in the IT workforce went from 40% in 1986 to 29% in 1999 (Tapia & Kvasny, 2004). In 2009, only 20.9% of computer software engineers were women, 22% of computer programmers were women, and 29.9% of computer and information system managers were women while women comprised 47.2% of the total workforce (Bureau of Labor and Statistics, 2009).

The same tendencies have also been observed in Europe. In her 2005 review of the efforts and recommendations of professional organizations in Germany, Schelhowe concludes that the field of computer science should be more self-critical in order to attract more women into this field (Schelhowe, 2005).

With Agile's focus on collaboration and iteration, it was not long before researchers began to investigate the potential of Agile practices in order to affect the diversity of the software development workforce.

Essentialist vs. Social Constructivist Theories

Essentialist theory concludes that women stay away from certain professional or academic fields of endeavor because they are by nature less capable or less attracted to that particular field while Social Constructivist theory presumes that women are equally competent in the field, but that some social factor prevents them from participating (Trouth, 2002). Trouth's research reveals that the core of Essentialist theories is the assumption that there is a fixed, unified, and opposed male and female nature. Social Constructivist theories, on the other hand derive from the assumption that there is a social context resulting in a definition of IT as a man's work rather than a woman's. Trouth conducted detailed interviews with individuals and exposed a wide array of individual differences in decision-making and coping mechanisms for women in the IT field.

The rarity of women in the Science, Engineering, Math, and Technology (STEM) disciplines have for some time been a cause for discussion and even concern. Some have postulated that women's cognition works differently than men and they are therefore attracted to and better at different disciplines (Ceci & Williams, 2007). This work is recent, but is already frequently cited. Ceci and Williams' book reviewed cognitive testing scores and revealed interesting and significant differences between men and women. Specifically, while averages are about the same, more men score at cognitive testing extremes (both high and low) than do women. An Essentialist interpretation of these scores deduces that this is because women have differing cognitive abilities than men and as a result are attracted to different disciplines. A

Social Constructivist interpretation of this research results in the conclusion that women are raised and socialized differently than are men and this is the reason for the different cognitive test scores. This book explores both of these theories with articles written by a number of different authors, each of whom presents reasons on one side or the other of the issue.

This book goes on to point out that the differences between males and females in test results are dwarfed by differences between national cultures where students in Japan, for instance, outperform students in the U.S. by margins considerably larger than the smaller differences between genders (Ceci & Williams, 2007). It also describes the concept of gender schemas which function to categorize people; women are perceived as more nurturing and collaborative than men. The book also reviews experimental research wherein survey respondents selected male candidates as more competent than women in the absence of measurable accomplishments. When presented with evidence that the women were more competent, survey respondents instead indicated that the women were less likeable.

Computing with a Purpose

Research has shown that women often enter technology-related fields for different reasons than men. Additionally, research has also demonstrated that collaborative teaching methodologies may enhance retention of women in computer science undergraduate academic programs. In a longitudinal study of women in the computer science at Carnegie Mellon University, professors at Carnegie Mellon found collaborative learning and mentoring approaches were correlated with an increase in the selection and persistence of women in the computer science department. They hypothesized that women in the U.S. do not grow up tinkering and playing with computers as do men and that they compare themselves unfavorably to these men who appear to already know everything and are whose comments can be very

discouraging to women. Several studies have shown that while North American men use computers as toys and for enjoyment early on in life, women on the other hand, use computers in order to accomplish tasks and get work done. Once introduced to computers, they view them as tools, but rarely with the same confidence that men have who started playing with them as boys (Lockheed, 1985) (Shashani, 1993) (Quesenberry & Trauth, 2007). Quesenberry and Trauth interviewed women in computing over a four-year period as to the alternative career anchors of technical versus management tracks and analyzed three dimensions: technical, managerial and geographical. This article concludes that a managerial choice eventually obviates the technical track, i.e., one must choose one or the other. The article further concludes that geographical choices are more transitory and driven by family needs.

The situation is different with international women who have other models of success than in North American culture (Margolis & Fisher, 2002). Margolis and Fisher explore the difference in cultural experience of men and women and how it relates to their confidence with computing. It also contains enlightening discussions of the reasons each gender uses a computer: men are more likely to use a computer to play while women are more likely to use a computer to get work done. The social phenomenon of "Geek Mythology" in the U.S. is presented, wherein the previously nerdy image of the male computer geek is glamorized and idolized. This book reviews social factors affecting participation of women in the computing industries, exploring the experiences of women from our own culture and women from other cultures as well. The movie, "The Social Network," (Eisenberg & Garfield, 2010) illuminates directly the social context described therein. Margolis and Fisher found that females' enthusiasm degraded over time due to corroded confidence, poor teaching techniques, negative comparisons to peers, and biased environments (Margolis & Fisher, 2002).

Conflicting Family Responsibilities

A career in programming inevitably requires long hours: hours spent either frantically striving to meet deadlines or spent learning new skills and technologies. Women are also the bearers of children, which unavoidably imposes certain physical limitations. Women who choose to have a family require leave for childbirth and recovery, and they are generally also still the primary caregivers of children. Once a woman has had to withdraw from the field for even a few weeks, a daunting uphill learning battle can easily thwart even the most aggressive of learners. Women are also usually the primary caregivers to elderly parents; as life expectancy increases, this generally entails many years of effort and expense. These common experiences lead to the conclusion that family and care giving considerations may be crucial in the recruitment and retention of women in the computer science workforce. In fact, studies have shown that a balance between work and life is critical in women's career decisions (Wardell, Sawyer, Mittory, & Reagor, 2006). This study analyzed statistics concerning the participation of women and men in IT, cross-referencing with other statistics such as education, amount of time commuting, and a many other details. Over a six-month period, extensive interviews were conducted of subjects under age 40 revealing that work-family balance is critical for women as compared to men and that this contributes to the under-representation of women in this field. In the Margolis and Fisher study discussed earlier, fully 20% of the female students questioned their interest in computer science because they felt they could not devote the "eat, sleep, and breathe" amount of time into it which they perceived their male counterparts as investing (Margolis & Fisher, 2002).

As mentioned earlier, Quesenberry's study found that for family reasons, women may select careers based upon geographical locations (Quesenberry & Trauth, 2007), although the family need may be transitory, such a choice affects the individual's career forever.

In a survey completed as recently as October, 2010 by Technisource, a research firm relying upon respondents from Monster.com, female IT workers indicated clearly that flexibility is still important in career choice; 16% of females as opposed to 12% of males indicated this is important to them (Technisource, 2010).

Hostile Environments

The Open Source community has touted itself as a bastion of freedom, many (but not all) of the programmers of Open Source are volunteers. Yet women participate here in very small numbers. In fact, it is estimated that only 1.5% of the participants are female (James, 2010). It is postulated that two reasons for the underrepresentation of women in the Open Source are that 1) the community is socially unfriendly to women and 2) women have less time than men for this type of volunteer occupation due to family responsibilities. It is also true that women do, in fact, participate in this community, but use masculine names in order to hide their gender to avoid any distraction relating to their being female (Levesque & Wilson, 2004). Levesque and Wilson surveyed speaker names for O'Reilly's annual Open Source conference as evidence that women do not participate in Open Source and postulated that the culture is unfriendly to women due to a peculiar geeky-macho mentality.

There have been anecdotal reports of environments actively hostile to women (Tapia A. H., 2003). In her 2003 case study, Tapia describes a shockingly hostile work environment where women were used as hiring enticements for men and concludes that a "gold rush" mentality at the time contributed to the situation. Women are not the only group which is under-represented,

both African-American and Hispanics are also underrepresented. In her literature study in 2004, she surveys issues relating to the retention of women and minorities and concludes that drop-out statistics are equally as or perhaps even more important than entry levels for minorities and that this may also reflect environments which are hostile to diversity (Tapia & Kvasny, 2004). "The Social Network" movie portrays a scene wherein several young male programmers are competing for an unpaid job with the nascent Facebook venture (Eisenberg & Garfield, 2010). This may or may not be an accurate portrayal of this particular situation, but it is entirely believable because of the peculiar geeky-macho culture known to exist in many programming environs. This kind of climate is diametrically opposed to how women are typically socialized to be collaborative rather than competitive in western countries. In their seminal work, already discussed above, Margolis and Fisher found that in many programming situations, women are easily intimidated by men who exhibit bravado and appear to already know everything and so the women become discouraged. In this work, they coined the phrase "Geek Mythology" relating to male attitudes to computing and also introduce the concept of computing with purpose relating to female attitudes (Margolis & Fisher, 2002).

Women and Confidence

Confidence is a key factor in tackling any challenge, including a career as a computer programmer. Studies have consistently shown that women do not have confidence in their own abilities in this area, are intimidated by male peers, and yet believe that women can and should be able to excel in this area (Lockheed, 1985) (Shashani, 1993) (Quesenberry & Trauth, 2007). The 1985 Lockheed study was one of the earliest empirical studies of how women and men view computers differently. The Shashani study surveyed 9th and 12th grade students in several Philadelphia, PA schools targeting attitudes towards interest in computers, stereotypes of

computer users, concepts of computers, confidence in one's own ability, and students' perception of parents' and teachers' attitudes towards computers. The boys showed significantly more interest in learning and enthusiasm in using computers. The gender difference in confidence in one's own computer ability was also quite sharp. There was also a positive correlation between encouragement from parents and teachers and the student's interest in computers. The Quesenberry study, described earlier, also concluded that women display less confidence with computers than do men.

Women typically assume that a difficulty with a task infers that they are not competent, while men assume that a difficult challenge is just that: they must simply work harder to overcome the challenge (Halvorson, 2011). In some research, this concept is referred to as self-theories. In their 2008 study of literature on this topic, Murphy and Thomas outline two diametrically opposite mindsets: fixed mindset versus growth mindset. Murphy and Thomas' research builds upon concepts introduced by a book entitled *Self-theories: their role in motivation, personality, and development* by C.S. Dweck (Dweck, 1999). A fixed mindset refers to a helpless response to a difficult and frustrating challenge while a growth mindset refers to a "mastery-oriented" response. These behaviors were observed by Dweck as a Professor of Psychology in the classroom; she conducted empirical studies and found that these attitudes were predictors of academic achievement. Murphy and Thomas relate how women often enter Computer Science classes with less previous experience than men and the results are feelings of inadequacy. Even after the women have overcome challenges and experienced success, they compare their level of effort unfavorably with men and thus diminish their own accomplishments (Murphy & Thomas, 2008). Murphy and Thomas point out classroom experiences wherein a more experienced student asks "pseudo-questions" in order to demonstrate their superiority and

the tendency of some instructors to favor such students. When combined with the sense of isolation as a result of actually being in the minority, women's confidence has been shown to shrink.

Studies have also shown that Pair Programming dramatically improves both individual confidence and software quality. In their 2000 study, Williams, Kessler, Cunningham and Ward measured number of man-hours required in Pair Programming versus individual programming and found that it took more man-hours of individual programming to achieve the same quality as was achieved in pairs. Ultimately, quality code was produced in about half the time with teams working in pairs. Furthermore, 90-96% of students reported via interviews and surveys that their confidence was improved as were their satisfaction with the work (Williams, Kessler, Cunningham, & Jeffries, 2000).

A case study observing upper-class female students in the Computer Science Department using Pair Programming in coursework showed women responded positively to collaborative practices and demonstrated increased confidence and improved retention in the classroom (Berenson, Slaten, Williams, & Ho, 2004). Out of three programming assignments, the junior and senior students were expected to work two of them in pairs and the third alone. Effort was made to ensure that no student had to work with a student with whom they felt incompatible, nor was any female alone in a group of males. The groups were also carefully balanced as to skill level. This study also established a direct relationship for women between product quality and confidence in their own skill level.

Murphy and Thomas point out that there is a contradiction between the studies demonstrating that collaboration in the classroom improving confidence and retention with the fact that students with a fixed mindset are less willing to accept help (Murphy & Thomas, 2008).

Their assumption is that students who drop out of academic programs in IT are afflicted with a fixed mindset and have given up in the face of a challenge. Another reason is possible, however. Accepting help when one has already demonstrated inability is hurtful to one's pride while planned collaboration guides students or team members to mastery-oriented thinking by removing the factor of pride and initiating cooperation at the beginning before difficulty is encountered.

Diversity Equalization Using Agile practices

With its focus on cooperation, collaboration, and the sustainable workweek, it seems likely that Agile methodology holds some promise for attracting and keeping women in the profession. The previously-mentioned longitudinal study at Carnegie Mellon University found that an increase in the selection and retention of women in the computer science department curricula followed the increased use of collaborative learning and mentoring approaches (Margolis & Fisher, 2002).

In another study, O. Hazzan et al. found that confidence was improved and that a good "fit" between how women work and Agile methods may exist among students (Hazzan, Frieze, Blum, & Dias, 2006). Hazzan et al.analyzed the cultural forces in the software development environment and compared differences between women raised in the U.S. by surveying confidence in undergraduate students associated with participation in Agile methodologies. Data from Arab and Jewish students in Israeli classes was also collected and a positive response to Agile methodologies was demonstrated there as well.

In her 2006 review of research on the benefits of Pair Programming in academic curricula, Williams found that research has shown a positive correlation between student retention and interaction in a course. She also found that Pair Programming broadened

participation in computing fields. Academic environments may discourage this kind of collaboration: some academic policy language insisting that students work alone and some instructors also resist the concept. She recommends earlier adoption of Pair Programming in academic curricula in order to improve retention in Computer Science programs (Williams, Debunking the Nerd Stereotype with Pair Programming, 2006).

Studies have shown that Agile engineering practices tend to equalize communications inside teams across gender and nationality (Hazzan & Dubinsky, 2006), (Hazzan O. , 2006). In one study, Hazzan et al. found that diversity is lacking in many software teams. Their study in an academic situation found that Agile methods tended to equalize communication between males and females and concluded that these methods enhance gender diversity, managerial diversity, and opinion diversity. Studies have also shown that Agile software engineering practices, noticeably Pair Programming, tend to improve retention of students generally in Computer Science curricula (Melnick & Maurer, 2003). In their 2003 study, Melnick and Maurer used mixed methods to interview and survey students on their attitudes to Agile practices. They asked such questions as what worked for them, what problems did they encounter, would they use Agile practices in the future, and asked about their impressions of test-driven development. They also asked how XP improved their learning. Their follow-up study in 2005 of survey and interview questions investigated student perception relating to a list of XP practices such as Pair Programming and Test Driven Development. In the 2005 study, they investigated a cross-section of students from five different levels of study at the University of Calgary and the Southern Alberta Institute of Technology. Students were introduced to Agile practices at the beginning of their courses; for many of them it was their first exposure to these practices. The students filled out anonymous online surveys and participated in interviews and

discussions. Students were queried on the following aspects relating to Agile: Collective Code Ownership, Continuous Integration, Coding Standards, On-site/On-call/Online Customer, Metaphor/Architecture, The Planning Game, Pair Programming, Refactoring, Unit Testing, Test-Driven Development, Short Releases, and "You Ain't Gonna Need it" (a/k/a Simple Design). The student responses were overwhelmingly positive: 78% believed Agile improved productivity, 76% believed it improved quality, and 65% would recommend the practices to an employer if they could (Melnik & Maurer, 2005). The survey responses were statistically significant. A slight negative correlation was found between the seniority of the students and their confidence in their estimation of effort in The Planning Game. A positive correlation was found with seniority and attitudes toward Test Driven Development. The authors concluded that this was understandable since increased experience reveals exactly how difficult accurate estimation can really be, while on the other hand, it takes developers some time to realize that Test Driven Development is not so much about testing as it is about design.

Gap in the Research

The above studies are suggestive that students generally and female students in particular will respond positively to Agile practices in the classroom resulting in increased retention within the Computer Science curricula. Studies also suggest that developers respond positively to Agile practices. But few studies exist which examine whether women in the profession of software engineering actually prefer or respond in a positive way to Agile practices such that it might help them to stay in the field. If Agile is more attractive as a methodology to women, this may provide one more reason to adopt it. This study proposes to begin to address this gap.

Chapter 3 – Methodology

Since there are many possible reasons and complexities relating to the paucity of women in the workforce as software developers, qualitative research was selected. Surveys were used to gather some initial data with follow-up interviews to zoom in on the possibilities and practices which women tend to favor; this has elements of grounded theory research (Willis, 2007).

Four core engineering practices characteristic of Agile methodology were focused upon in survey questions: These were selected as being both hallmark and key engineering techniques recognized in Agile practices. Arguably the best known of them is Pair Programming where two developers sit together to develop code. An acceptable alternative to this is for one developer to write unit tests while the other developer makes them pass. Test Driven Development is the practice of developing Unit tests which are broken before developing the code which makes them pass. These Unit tests are typically written in the same computer programming language as the code they are testing. Continuous Integration/Automated Build is the practice of having an automated process which polls source control for code changes and triggers a build, followed by running the Unit tests and then reporting the results. The Planning Game is the practice of collaborative estimation of effort using an abstract point system referring to the size of the effort (Abrahamsson, Warsta, Siponen, & Ronkainen, 2003).

There are many possible reasons why so few women work as software developers. The purpose of the survey was to determine how much exposure the women have had to these engineering techniques and whether it is perceived as having a positive effect on quality, productivity, skill expansion, and job satisfaction. The follow-up interviews probed in further depth as to whether the Agile practices are influential in any way in attracting or keeping women working in this type of work.

Interview research is associated with an approach referred to as the "Phenomenological approach," which assumes and depends upon the subjects' own perceptions (Leedy & Ormrod, 2005). The women's own perceptions are critical in the research process.

Survey analysis of experience with specific Agile engineering techniques.

Women who have experience in varying software methodologies were sought out using an online survey followed by interviews as to what they consider the strengths and weaknesses of each, their preferences, and the reasons for them. Groups such as "Women in Technology", "systers.org" and various Agile communities were used to publish a link to the online survey available to female software engineers who had the necessary experiences and were willing to participate. A similar non-gender survey was conducted by G. Melnik (Melnik & Maurer, 2005).

Respondents were asked to detail how many years' overall experience they have had as a developer. Then they were asked to specify how much experience they have had with Pair Programming, Test Driven Development/Unit Testing, Automated Build/Continuous Integration, and The Planning Game.

For each of those same practices, respondents were asked to rank on a scale of Strongly Disagree to Strongly Agree (with options of Neutral and Not Applicable) as to whether the technique resulted in higher productivity, in higher quality, higher potential for learning new skills, and also higher personal job satisfaction. Respondents were also permitted to skip any question. These categories were selected as being obviously important for software developers in terms of job security and retention.

The survey was constructed using an online survey service and published via a link. Respondents were able to respond to the questions at their convenience and in their own time.

Ultimately, 90 respondents accessed the link, and 70 respondents completed all the questions, the remaining 20 answering only some questions and skipping others.

Follow-up interviews

Those survey respondents indicating a willingness to participate in follow-up interviews were contacted and interviewed via phone.

If the survey responses indicated a positive outlook to the Agile engineering practices in question, follow-up survey questions included the following:

- What worked for you?
- If you could, would you recommend these practices to your employer? If so why?
- What problems have you encountered?
- How did these practices improve quality, productivity, learning, etc.?
- How important do you think these types of practices are?

If the survey responses indicated a neutral or less than positive outlook to the Agile engineering practices in question, follow-up survey questions included the following:

- What other than these has worked well for you?
- What practices would you recommend to your employer if you could?
- What problems have you encountered?
- Are there changes which can be made to these practices which might make you more inclined to feel they are valuable after all?

For women who have been in the field quite some time, the following question was asked:

- What factors have kept you working in this field?

- What factors have tended to discourage you and/or other women you have worked with as developers?

For women who are younger or less experienced:

- What do you need in order to stay working in this field?
- What might cause you to leave for other endeavors?

Six of the initial survey respondents completed a follow-up telephone interview.

Chapter 4 –Results

Women with many years of experience were unusually well-represented in the survey as is shown in Figure 1: Experience Spread of Respondents, below. Of the 80 women who responded to the question, 28 or 35% of them have been practicing developers for over 15 years. An additional 23 or 28.8% of them had been practicing for 7-15 years. Another 18 or 22.5% have been developing for 3-7 years. Ten women comprising 12.5% have been developing for 1-3 years. Only one woman had been developing for less than a year, this was 1.3% of the response. While not representative of the spread of experience for women actually in this field, the seniority of the women responding offered the opportunity to find out what women who are successful enough to obtain seniority really prefer.

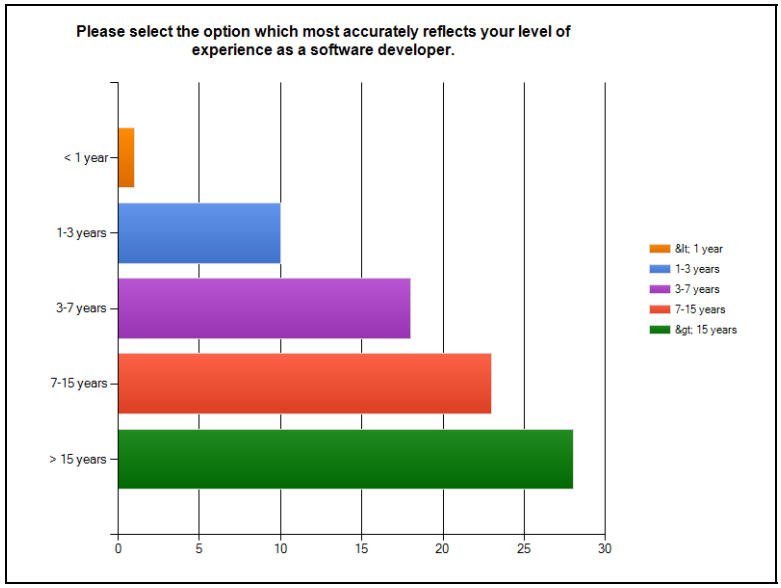

Figure 1: Experience Spread of Respondents

Attitudes and Experience with Pair Programming

Pair Programming is one of the most well-known Agile engineering practices. Yet as is shown in Table 2: Experience with Pair Programming, below, of the 70 women who responded, 28 of them had no experience with Pair Programming, only 13 had even a year of experience.

Another 11 had perhaps six months of experience. Six women skipped this question. Given the fact that so many respondents had many years experience as developers, this would indicates that although the practice is well-known, it is not commonly practiced.

Table 2: Experience with Pair Programming

How much experience have you had with Pair Programming?		
Answer Options	Response Percent	Response Count
None	37.3%	28
< 6 mos	26.7%	20
6 mos-1 yr	14.7%	11
1-4 yrs	12.0%	9
4-7 yrs	5.3%	4
> 7 yrs	4.0%	3
answered question		75
skipped question		15

Strikingly, of the 47 women who had any experience with this practice many of them agreed that this contributed to higher quality of software: 45.7% agreed and another 23.9% strongly agreed with this statement. Only two women disagreed either strongly or otherwise as can be seen in Figure 2: Pair Programming and Quality, below.

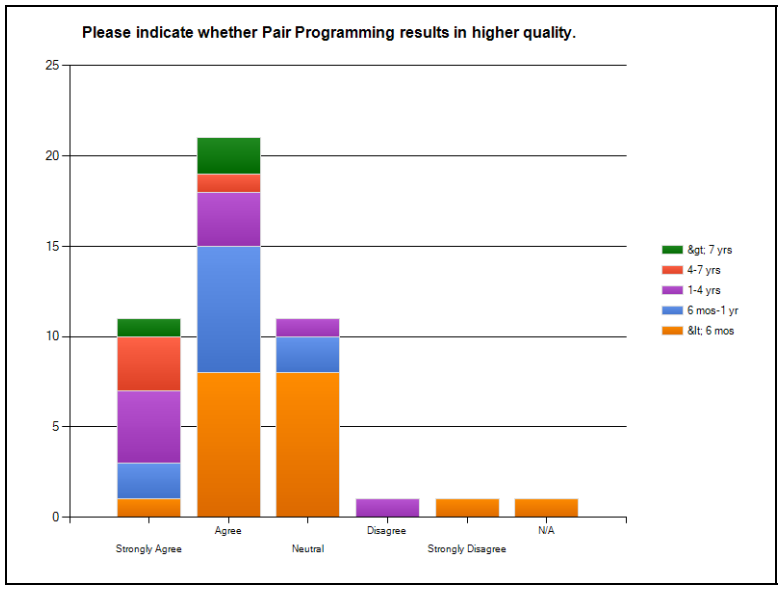

Figure 2: Pair Programming and Quality

They also obviously felt that this practice contributed to faster skill expansion: 23.9% of them strongly agreed and another 50% agreed. Figure 3: Pair Programming Skill Expansion, below also shows that only two women disagreed either strongly or otherwise.

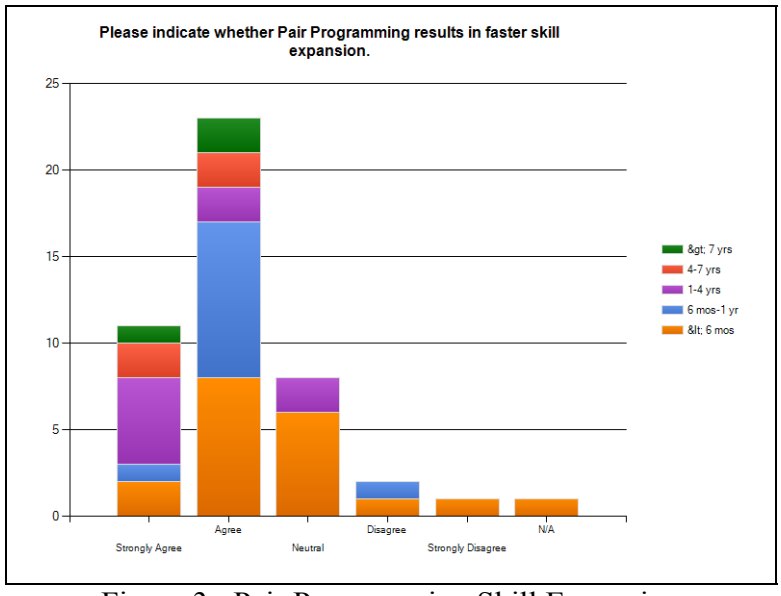

Figure 3: Pair Programming Skill Expansion

Its effect on job satisfaction was also positive, 30.4% agreed it contributed to job satisfaction, and another 13% strongly agreed with 37% feeling neutral. Eight women disagreed,

strongly or otherwise. These numbers are illustrated in Figure 4: Pair Programming and Job Satisfaction, below.

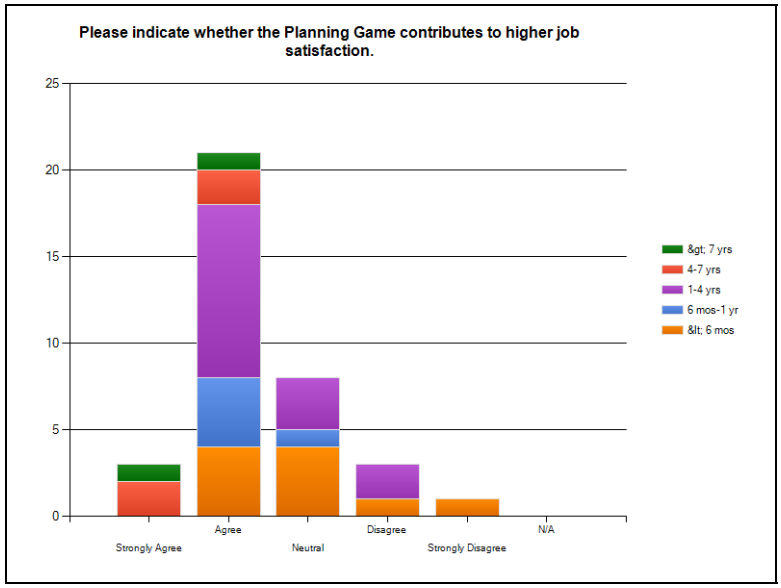

Figure 4: Pair Programming and Job Satisfaction

Its effect on productivity was positive too, as shown in Figure 5: Pair Programming and Productivity, below, but more neutral overall than the other aspects surveyed, 37% of the respondents so indicated while 28.3% agreed, and an additional 15.2% strongly agreed.

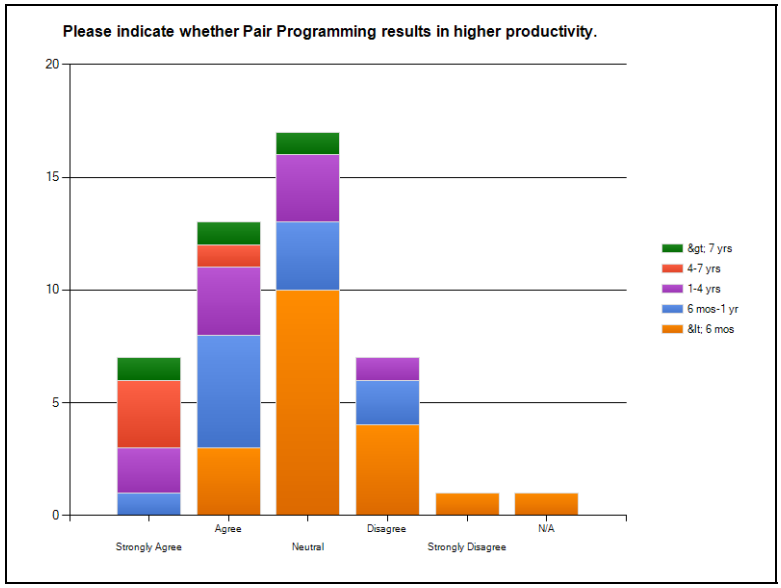

Figure 5: Pair Programming and Productivity

The follow-up interviews gave some insight as to the meaning of these responses. Respondents reported misapplication and misinterpretation of the practice to mean one programmer watches another one work. Often, management and sometimes even other programmers decide that the practice is wasteful of human resources, so there are hurdles to achieving common practice. A natural inclination is to think that having two people do the work traditionally done by just one is wasteful. One respondent who had considerable experience with this technique indicated that this practice at first is often uncomfortable to programmers who are not widely known for sociability but that they often come to appreciate it. Another interview respondent indicated she thought it helped to keep developers focused on the correct functionality. This practice has been shown to retain women in academic curricula (Melnick & Maurer, 2003) (Melnik & Maurer, 2005), but the responses of these women with a number of years of overall experience hints that the practice is not very common in the workplace.

Attitudes and Experience with Test Driven Development

Test Driven Development (TDD) is also a well-published Agile engineering practice. Of the 71 women who responded to this question, 20 or 28.6% had no experience with it, but 50 had at least some experience; 25 of them had at least a year. This would seem to indicate a slightly wider acceptance of this practice in the workplace. These numbers are displayed in Table 3: Experience with TDD, below.

Table 3: Experience with TDD

How much experience have you had with Test Driven Development?		
Answer Options	Response Percent	Response Count
None	29.6%	21
< 6 mos	19.7%	14

How much experience have you had with Test Driven Development?		
6 mos-1 yr	15.5%	11
1-4 yrs	18.3%	13
4-7 yrs	11.3%	8
> 7 yrs	5.6%	4
answered question		71
skipped question		19

As seen in Figure 6: TDD and Productivity, below, of the 50 women who had at least some experience with this practice, 19 agreed, and another eight strongly agreed that it contributed to productivity. Fourteen or 28.0% answered that it had no effect on productivity. Only six disagreed that it contributed to productivity.

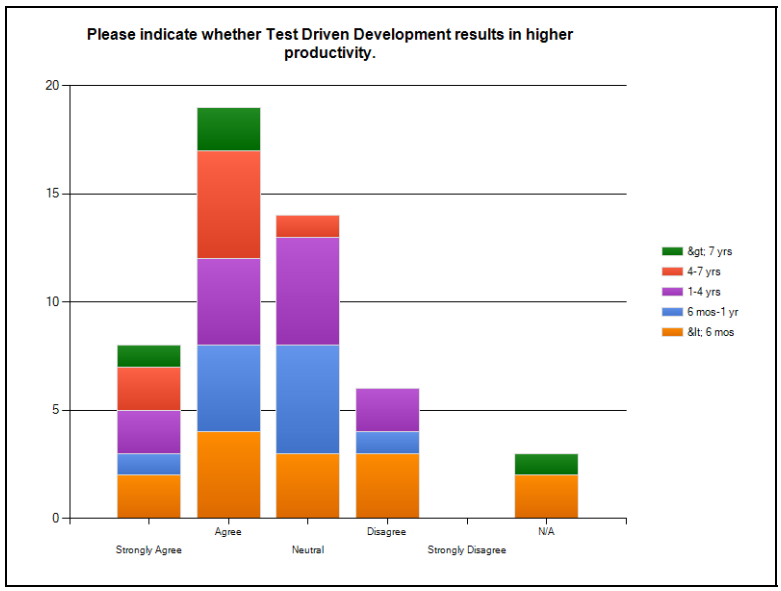

Figure 6: TDD and Productivity

An inspection of Figure 7: TDD and Perception of Quality, below demonstrates that the responses were striking concerning quality: 40% strongly agreed, and another 38% agreed that this practice contributes to quality. Only two disagreed.

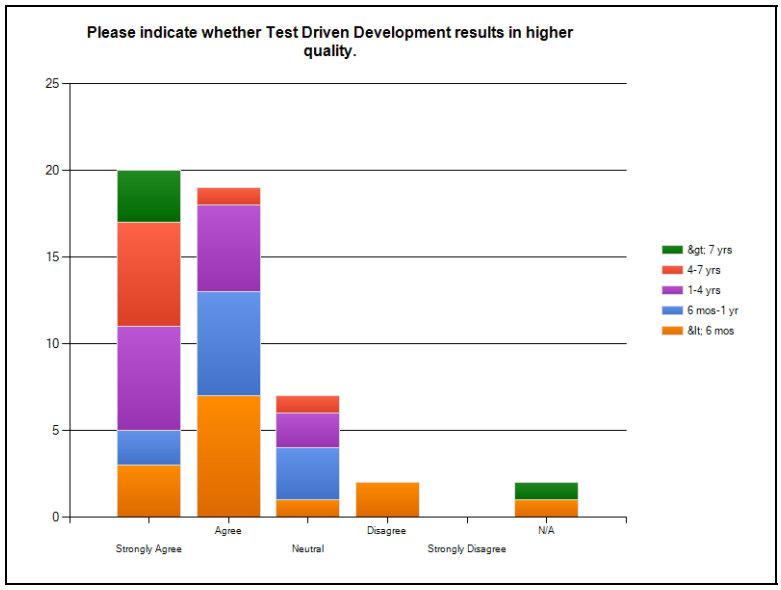

Figure 7: TDD and Perception of Quality

The responses were comparatively more neutral regarding whether TDD results in faster skill expansion; 48% of the women responding indicated that they felt neutral that TDD contributes to skill expansion. Figure 8: TDD and Skill Expansion, below, illustrates the responses for this category.

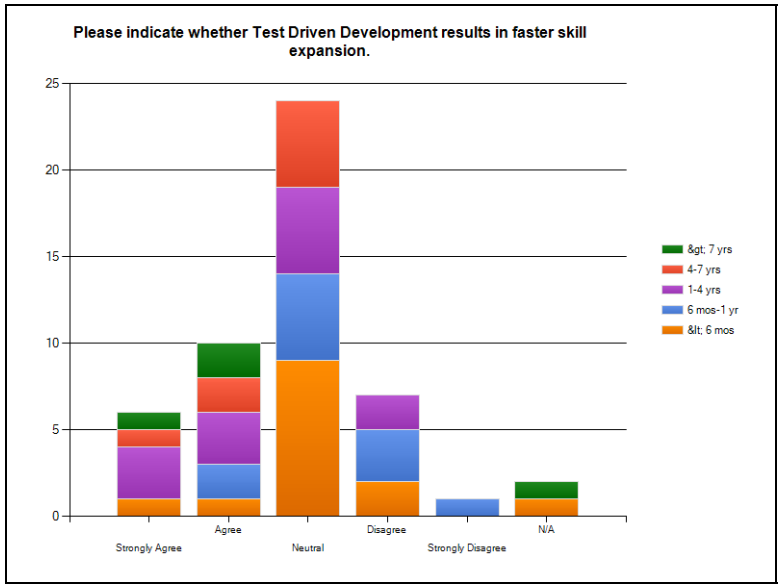

Figure 8: TDD and Skill Expansion

Women also appeared to feel that TDD contributed to their job satisfaction: 32% agreed, and another 16% strongly agreed with this statement; although 36% were neutral as shown in Figure 9: TDD and Job Satisfaction, below.

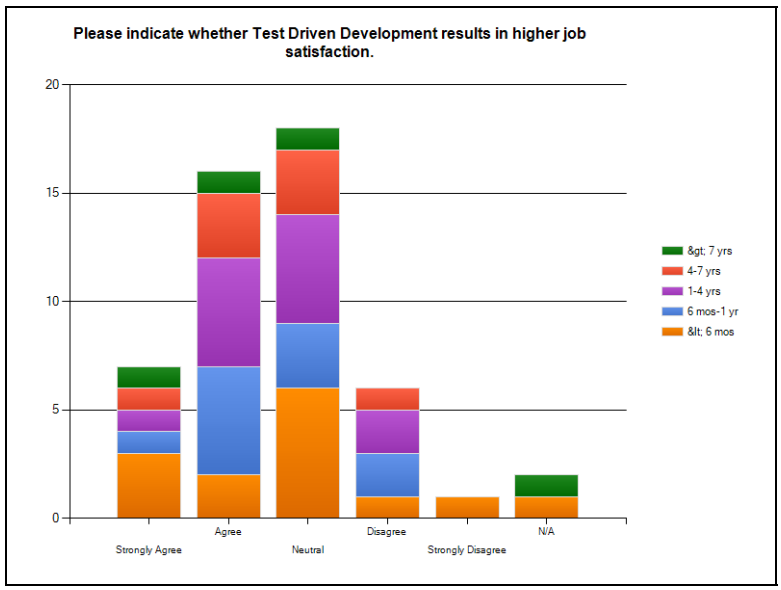

Figure 9: TDD and Job Satisfaction

Interview responses supported these reactions: TDD contributes to quality, and furthermore addresses the reputation of developers who are sometimes perceived as uncaring according to one respondent. Another respondent indicated that this practice keeps developers focused on the results and furthermore is an opportunity to get users involved if they are incorporated in designing the tests. Yet another respondent indicated that now she has incorporated Unit testing in her work she will never go back. Another respondent indicated similarly that initial acceptance and learning curve is tough, but once you do it you never want to go back.

Attitudes and Experience with Automated Build/Continuous Integration

Survey responses summarized in Table 4: Experience with Continuous Integration, below indicate that Automated Build/Continuous Integration practices is more widespread; 52 of

the 67 women who responded to this question had at least one year of experience, another nine had at least six months, and the remaining 18 had less than six months or none.

Table 4: Experience with Continuous Integration

How much experience have you had with Automated Build processes?		
Answer Options	Response Percent	Response Count
None	18.6%	13
< 6 mos	8.6%	6
6 mos-1 yr	12.9%	9
1-4 yrs	35.7%	25
4-7 yrs	14.3%	10
> 7 yrs	10.0%	7
answered question		70
skipped question		20

Of the women who had experience with this, Figure 10: Continuous Integration and Productivity, below shows that an overwhelming 44.6% agreed, and 35.7% strongly agreed that it contributed to productivity.

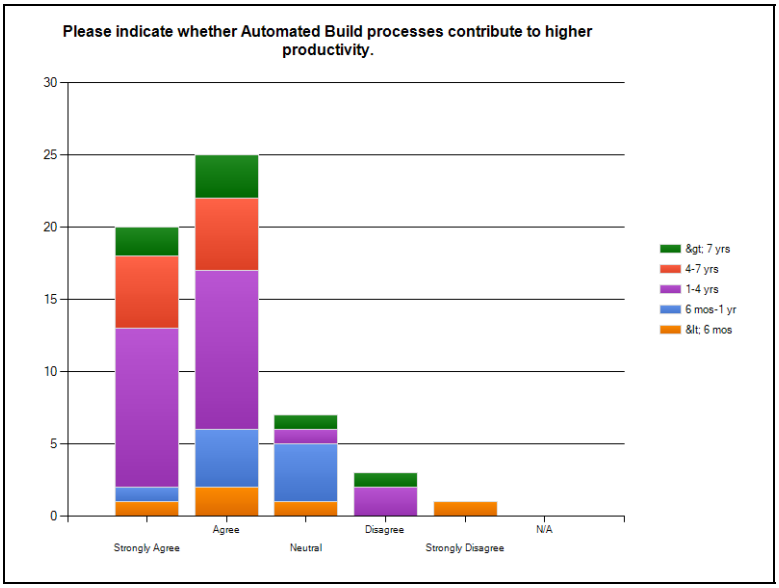

Figure 10: Continuous Integration and Productivity

It also is perceived as contributing to quality as seen in Figure 11: Continuous Integration and Quality, below: 38.6% of those with experience in it strongly agreed, and an additional 42.1% agreed while 17.5% were neutral.

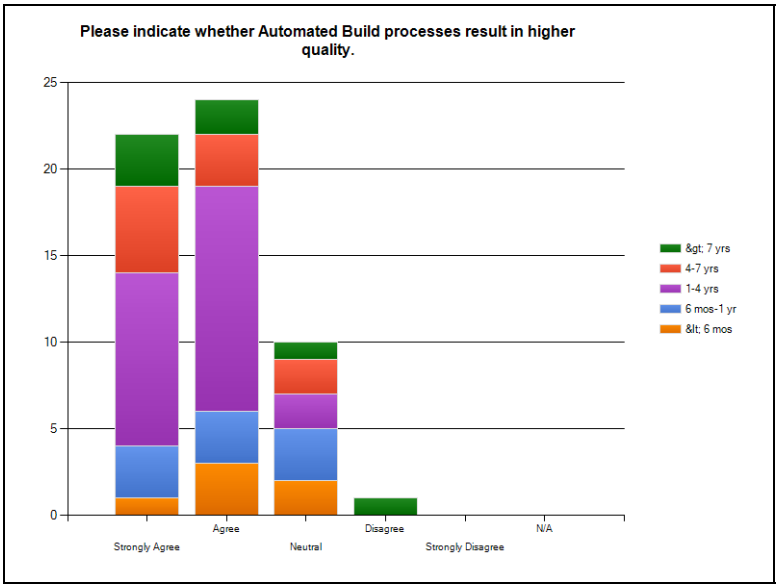

Figure 11: Continuous Integration and Quality

Figure 12: Continuous Integration and Skill Expansion and Figure 13: Continuous Integration and Job Satisfaction, below, show that the responses were more neutral regarding job satisfaction and skill expansion, 45.6% and 50% respectively for each question received a neutral

response from the women, although a 22.8% agreed, and 24.6% strongly agreed that this contributes to job satisfaction.

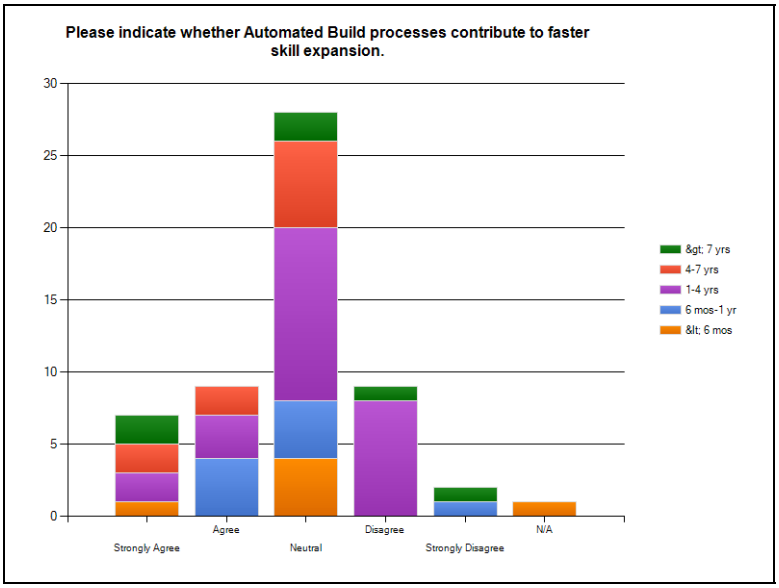

Figure 12: Continuous Integration and Skill Expansion

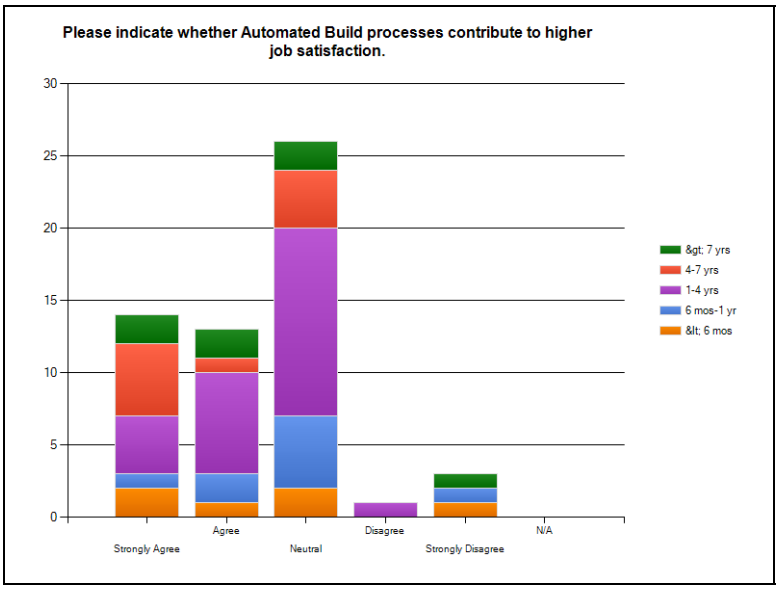

Figure 13: Continuous Integration and Job Satisfaction

The interview responses shed some light on the survey data. One respondent indicated that although Continuous Integration is sometimes a difficult practice to sell to management, it pays off quickly. Another indicated that this practice makes it easier to incorporate new team members and fosters collective ownership. Interestingly, although the survey responses

indicated a most favorable attitude to this practice, respondents mentioned TDD slightly more frequently than this practice in their interview responses (five versus two of the six who responded).

Attitudes and Experience with the Planning Game

The Planning Game enjoyed the dubious distinction of having the lowest adoption rate among respondents. This is the practice where developers, testers, and stakeholders meet in order to estimate the effort level of stories. The data suggesting lack of adoption can be seen in Table 5: Experience with The Planning Game, below; 32 of the women had no experience with this practice and another 10 had less than six months of experience with it. This left only 27 with any experience at all, and 20 having skipped the question.

Table 5: Experience with The Planning Game

How much experience have you had with the Planning Game?		
Answer Options	Response Percent	Response Count
None	47.1%	33
< 6 mos	14.3%	10
6 mos-1 yr	8.6%	6
1-4 yrs	21.4%	15
4-7 yrs	5.7%	4
> 7 yrs	2.9%	2
answered question		70
skipped question		20

Of those with experience in this practice, 19 agreed, and four strongly agreed that it contributes to productivity, although the majority of the respondents were either neutral or N/A

respondents. Notably, those with the most experience agreed the most that this contributes to productivity as can be seen in Figure 15: Planning Game and Quality, below.

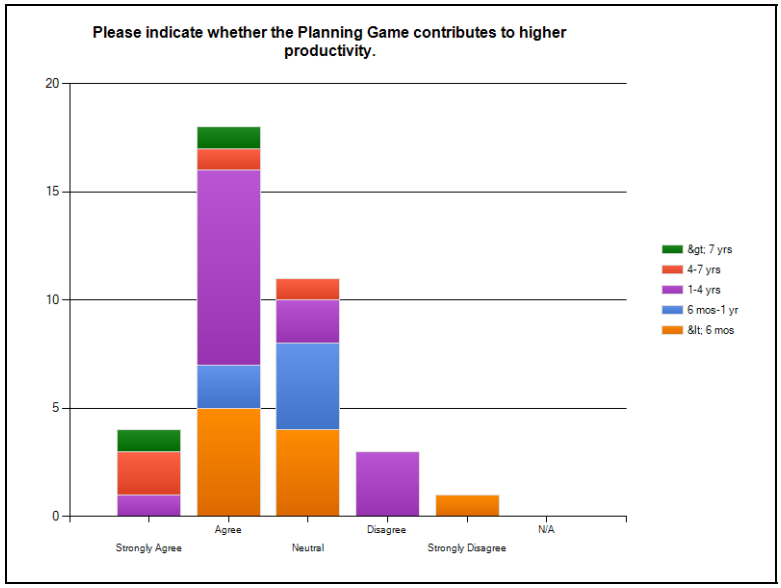

Figure 14: Planning Game and Productivity

When it came to assessing higher quality, the responses were also favorable, but distributed more evenly as to experience levels. Nineteen women agreed it contributed, and two more strongly agreed. A number also disagreed, some of them even strongly as can be seen in Figure 15: Planning Game and Quality, below.

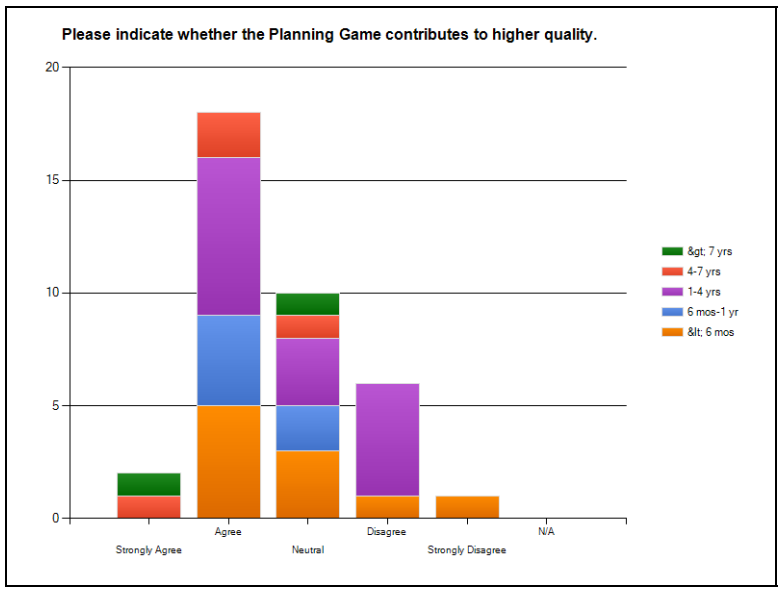

Figure 15: Planning Game and Quality

Women tended to feel more neutral concerning the Planning Game and skill expansion: ten of them agreed, nine of them disagreed, and 16 were neutral. Twenty-one of them, or 58.3%, however, felt that it contributes to job satisfaction, with another three or 8.3% agreeing strongly. Figure 16: Planning Game and Skill Expansion, below shows the answer spread of respondents who had experience with this practice.

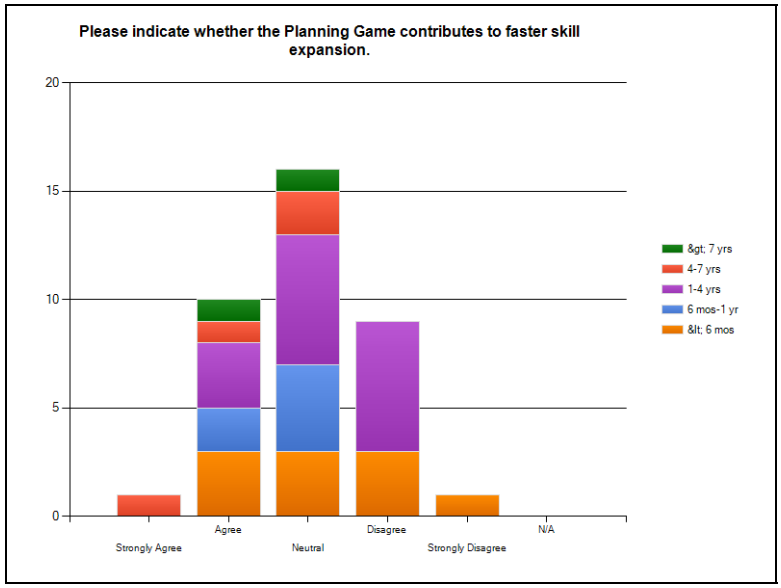

Figure 16: Planning Game and Skill Expansion

When it came to job satisfaction, women were clearly in agreement that the Planning Game contributes to this. Fifty-eight percent of the women agreed, and another 8.3% agreed strongly that The Planning game contributes to job satisfaction. Only 8.3% disagreed, and another 2.8% strongly disagreed while 22.2% were neutral as can be seen in Figure 17: Planning Game and Satisfaction, below.

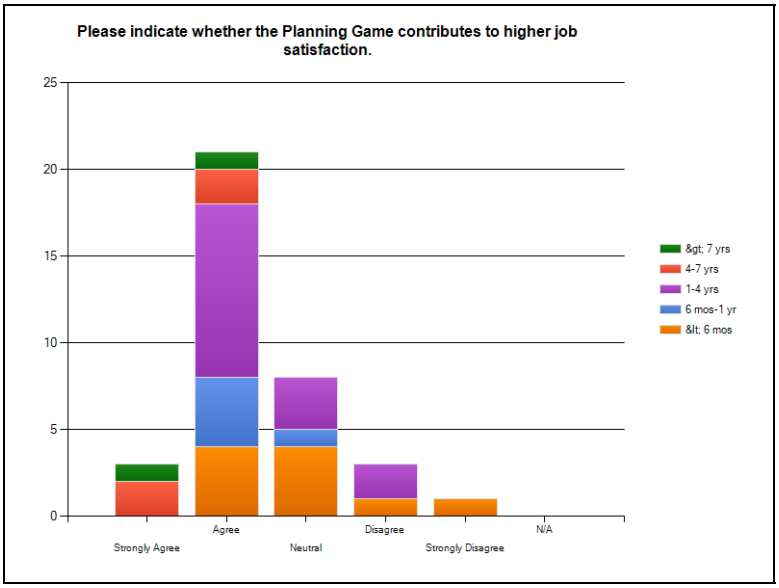

Figure 17: Planning Game and Satisfaction

Several interview respondents indicated that The Planning Game is a way to keep the stakeholders and business people engaged with developers so that progress makes sense. This practice centers upon negotiation and collaboration between stakeholders and developers to fit the highest priority stories which can be done into an iteration based upon the developers' estimation of the level of effort.

Some risks of the study

An unexpected number of respondents had a lot of experience as developers. See Table 2: Experience with Pair Programming, above. While this may not be representative of actual numbers of women currently working in this profession, it is likely to be helpful to indicate what sorts of attitudes keep women in this career long enough to become so experienced.

In addition, the women were all volunteers, responding to a request posted in various newsgroups, so it is possible that some self-selection by women who had strong opinions might skew the results.

Chapter 5 – Conclusions

This was a qualitative rather than a quantitative study; there was no expectation of any statistically significant results; however, both numeric and anecdotal data was gathered which strongly suggests that women respond positively to Agile engineering practices.

The practice of Pair Programming is one of the most widely-known Agile engineering practices, yet the reported experience of the responding women with Pair Programming suggests that it is not widely practiced. Of those who did, from their responses, it seems that this practice is perceived as contributing to quality, skill expansion, and job satisfaction, but less so (or rather more neutrally so) to productivity.

The practice of Test Driven Development (TDD) is also widely known, although the responses suggest that it is practiced only slightly more than Pair Programming. Unsurprisingly, the survey and interview responses all indicate that its biggest perceived benefits are to quality and also to job satisfaction. This suggests a possible relationship between quality work and job satisfaction for women.

The practice of Continuous Integration/Automated Build enjoyed a more widespread practice among the responding women. Unsurprisingly, respondents saw productivity and quality as its strengths more than skill expansion and job satisfaction.

The Planning Game affords considerable opportunity for collaboration with stakeholders. It does not directly affect how the code works, only how to estimate and prioritize which code will be worked on first. Women tended to feel its strengths were productivity, quality, and job satisfaction.

When it comes to comparing positive responses, the results are suggestive. In terms of only job satisfaction, all of the practices had a positive impact, the highest being Continuous Integration. Continuous Integration also had the highest positive impact relating to Productivity and Quality. It was Pair Programming, however, which had the highest impact on Skill Expansion.

Table 6: Comparative Positive Responses

Comparative Positive Responses				
Job Satisfaction	**Strongly Agree**	**Agree**	**Percent**	**Total Responding**
Pair Programming	7	14	28.77%	73
Test Driven Development	8	16	34.29%	70
Continuous Integration	14	13	**38.57%**	70
Planning Game	3	22	36.76%	68
Productivity				
Pair Programming	8	15	31.51%	73
Test Driven Development	10	20	42.25%	71
Continuous Integration	20	25	**65.22%**	69
Planning Game	4	19	33.33%	69
Quality				
Pair Programming	13	25	52.05%	73
Test Driven Development	23	20	60.56%	71
Continuous Integration	22	24	**65.71%**	70
Planning Game	2	19	30.43%	69
Skill Expansion				
Pair Programming	13	26	**53.42%**	73
Test Driven Development	7	10	23.94%	71
Continuous Integration	7	9	23.19%	69
Planning Game	1	10	15.94%	69

The survey responses were overall quite positive. The above table indicates that quality is perceived as a big benefit of Agile practices. Furthermore, Continuous Integration consistently obtained the highest rankings in terms of benefits among the four practices surveyed.

Interestingly, of the most senior respondents (those with greater than seven years of experience in the practice itself), the only negative impact was in the area of Continuous Integration where one woman with greater than seven years experience in the practice disagreed that Continuous Integration contributed to productivity, and one disagreed that it contributed to quality, and one disagreed, and one strongly disagreed that it contributed to skill expansion with one disagreeing that it contributed to job satisfaction. See Figure 13: Continuous Integration and Job Satisfaction, above.

Interview Responses

The interview responses to questions concerning what keeps women successful in this field, and what causes them to leave it often mentioned one of the following: isolation, collaboration, communication. The word “communicate” or “communication” was repeated some seven times by the six respondents. Some respondents also mentioned issues relating to work-personal life balance; still others, the type of product that they develop, and some the fact that they really liked their teammates. It is clear from the tone of the responses that collaboration and communication are important to these women. One respondent who indicated that her environment had considerable success attracting and keeping women directly attributed it to the adoption of Agile practices which helped to improve communication; she considered this to be a key factor. Another responded that Pair Programming set up a situation where multiple people

know any given area of code, resulting in making it easier to take off time when it was required, such as for a sick child.

When asked whether they would recommend these practices to their employers the women who participated in surveys invariably responded that yes, they would. Respondents reported, however, that there was resistance to these practices on the part of both management and sometimes other developers. One responded that in a mixed environment where some teams are Agile and others are not, communication between teams can be tough.

The responses to the survey and the interviews alike indicate that female software developers respond positively to these Agile engineering practices and therefore they are likely to be helpful in getting and keeping women in this field of endeavor. Some helpful future research would be to compare the responses of women to men in quantitative studies. Another line of research could be longitudinal studies comparing adoption rates of these practices with acquisition and retention rates of female software developers.

References

Abrahamsson, P., Warsta, J., Siponen, M. T., & Ronkainen, J. (2003). New Directions on Agile Methods: A Comparative Analysis. *International Conference On Software Engineering*, *25*, pp. 244-254.

Beedle, M., Bennekum, A. V., Cockburn, A., Cunningham, W., Fowler, M., Highsmith, J., et al. (2001). *Agile Manifesto*. Retrieved November 24, 2010, from Agile Manifesto: http://www.agilemanifesto.org/

Berenson, S. B., Slaten, K. M., Williams, L., & Ho, C. (2004). Voices of women in a software engineering course: reflections on collaboration. *Journal on Educational Resources in Computing (JERIC) , 4* (1. Retrieved May 17, 2008 from http://doi.acm.org.dml.regis.edu/10.1145/1060071.1060074).

Boehm, B. (2005). Some Future Trends and Implications for Systems and Software Engineering Processes. *Systems Engineering , 9* (1), 1.

Bureau of Labor and Statistics. (2009). *Labor Force Statistics.* Retrieved November 28, 2010, from United States Department of Labor Bureau of Labor and Statistics: ftp://ftp.bls.gov/pub/special.requests/lf/aat11.txt

Ceci, S. J., & Williams, W. M. (2007). *Why Aren't More Women in Science.* Washington, D.C.: American Psychological Association.

Cockburn, A., & Highsmith, J. (2001). Agile software development, the people factor. *Computer , 34* (11), 131-133.

Dweck, C. (1999). *Self-Theories: Their role in motivation, personality and development.* Philadelphia, PA: Taylor & Francis.

Dybå, T., & Dingsøyr, T. (2008). Empirical studies of agile software development: A systematic review. *Information and Software Technology , 50* (9-10), 833-859.

Fincher, D. (Director). (2010). *The Social Network* [Motion Picture]. Sony Pictures.

Halvorson, H. G. (2011, March 1). *The Trouble With Bright Girls.* Retrieved March 6, 2011, from The Huffington Post: http://www.huffingtonpost.com/heidi-grant-halvorson-phd/girls-confidence_b_828418.html

Hazzan, O. (2006). Diversity in Computing: A Means or a Target? *System Design Frontier Journal, Featured Column* .

Hazzan, O., & Dubinsky, Y. (2006). Can diversity in global software development be enhanced by agile software development? *Source International Conference on Software Engineering archive. Proceedings of the 2006 international workshop on Global software development for the practitioner table of contents.* (pp. 58-61.). Shanghai, China: ACM.

Hazzan, O., Frieze, C., Blum, L., & Dias, M. B. (2006). Culture and environment as determinants of women's participation in computing: revealing the "women-CS fit". *ACM SIGCSE Bulletin archive , 38* (1), 22-26.

Herela, H. (2005, April). *Case Study: The Chrysler Comprehensive Compensation System.* Retrieved January 19, 2008, from http://calla.ics.uci.edu/histories/ccc/

James, J. (2010, April 6). *IT gender gap: Where are the female programmers?* Retrieved November 28, 2010, from TechRepublic: http://blogs.techrepublic.com.com/programming-and-development/?p=2386

Jeffries, R. E. (1998). *Do the simplest thing that could possibly work.* Retrieved April 17, 2009, from xprogramming.com: http://www.xprogramming.com/Practices/PracSimplest.html

Leedy, P. D., & Ormrod, J. E. (2005). *Practical Research Planning and Design 8th Edition.* Upper Saddle River, New Jersey: Pearson Educaton, Inc.

Levesque, M., & Wilson, G. (2004, November 1). *Open Source, Cold Shoulder.* Retrieved November 26, 2010, from Dr Dobbs: http://www.drdobbs.com/architecture-and-design/184415216

Lockheed, M. E. (1985). Women, Girls and Computers: A first look at the evidence. *Sex Roles , 13* (3-4), pp. 115-122.

Margolis, J., & Fisher, A. (2002). *Unlocking the Clubhouse: Women in Computing.* Cambridge, Mass.: MIT Press.

Martin, R. (2005, August). *Agile Development Conference* . Denver, CO.

Melnick, G., & Maurer, F. (2003). Introducing Agile Methods in Learning Environments: Lessons Learned. *LECTURE NOTES IN COMPUTER SCIENCE* (pp. 172-184). Germany: Springer-verlag.

Melnik, G., & Maurer, F. (2005). A cross-program investigation of students' perceptions of agile methods. *Proceedings of the 27th international Conference on Software Engineering* (pp. 481-488). St. Louis, MO: ACM.

Murphy, L., & Thomas, L. (2008). Dangers of a fixed mindset: implications of self-theories research for computer science education. *ITiCSE '08: Proceedings of the 13th annual conference on Innovation and technology in computer science education* (pp. 271-275). ACM.

Quesenberry, J. L., & Trauth, E. M. (2007). What do women want?: an investigation of career anchors among women in the IT workforce. *Special Interest Group on Computer Personnel Research Annual Conference. Proceedings of the 2007 ACM SIGMIS CPR conference on Computer personnel doctoral consortium and research conference: The global information technology workforce* (pp. 122-127. Retrieved 5/12/2008 from http://doi.acm.org.dml.regis.edu/10.1145/1235000.1235030). St. Louis, MO: ACM.

Regis University. (1998, September 30). *§46.116 - Informed Consent Checklist - Basic and Additional Elements.* Retrieved June 8, 2008, from Regis University: http://www.hhs.gov/ohrp/humansubjects/assurance/consentckls.htm

Schelhowe, H. (2005). Gender questions and computing science. *CWIT '05: Proceedings of the international symposium on Women and ICT: creating global transformation.* New York, New York: ACM.

Shashani, L. (1993, March). Gender-based differences in attitudes toward computers. *Computers & Education , 20* (2), pp. 169-181.

Taft, D. K. (2009, April 30). *Do Alpha Male Geeks Scare Women Away from Programming?* Retrieved November 28, 2010, from eWeek.com: http://www.eweek.com/c/a/Application-Development/Do-Alpha-Male-Geeks-Scare-Women-Away-from-Programming-834170/

Tapia, A. H. (2003). Hostile_Work_Environment.com. *Special Interest Group on Computer Personnel Research Annual Conference archive Proceedings of the 2003 SIGMIS conference on Computer personnel research: Freedom in Philadelphia--leveraging differences and diversity in the IT workforce* (pp. 64-67). Philadelphia, Pennsylvania: ACM New York, NY, USA.

Tapia, A. H., & Kvasny, L. (2004). Recruitment is never enough: retention of women and minorities in the IT workplace. *Special Interest Group on Computer Personnel Research Annual Conference. Proceedings of the 2004 SIGMIS conference on Computer personnel research: Careers, culture, and ethics in a networked environment* (pp. 84-91. Retrieved May 17, 2008 from http://doi.acm.org.dml.regis.edu/10.1145/982372.982392). Tucson, AZ: ACM.

Technisource. (2010, October). *Women in IT Careers Survey.* Retrieved November 27, 2010, from Slideshare: http://www.slideshare.net/monsterww/women-in

Trouth, E. M. (2002). Odd girl out: an individual differences perspective on women in the IT profession. *Information Technology & People , 15* (2), 98-118.

Wardell, M., Sawyer, S., Mittory, J., & Reagor, S. (2006). Gender and IT professionals in the United States: a survey of college graduates. *Labour & Industry , 16* (3), 39-59. Retrieved May 17, 2008 from http://find.galegroup.com.dml.regis.edu/itx/start.do?prodId=AONE Gale Document Number:A153761989.

Williams, L. (2006, May). Debunking the Nerd Stereotype with Pair Programming. *Computer , 39* (5), pp. 83-85.

Williams, L., Kessler, R. R., Cunningham, W., & Jeffries, R. (2000). Strengthening the Case for Pair-Programming. *IEEE Software , 17* (4), 19-25.

Willis, J. W. (2007). *Foundations of Qualitative Research.* Thousand Oaks, CA, USA: Serge Publications, Inc.

Annotated Bibliography

Abrahamsson, P. Warsta, J. Siponen, M. T. Ronkainen, J., New Directions on Agile Methods: A Comparative Analysis 2003. In *International Conference On Software Engineering*, Vol. 25, pages 244-254, USA, ISSN 0270-5257 http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1201204

A good review of agile methodologies with explanations of differing types and emphasis on test driven and collaboration. Includes continuous integration, test driven development, as well as paired programming. Interestingly, Scrum does not require the adoption of any specific engineering practices which might explain some difficulties with it.

Beedle, M., Bennekum, A. V., Cockburn, A., Cunningham, W., Fowler, M., Highsmith, J., et al. (2001). *Agile Manifesto*. Retrieved November 24, 2010, from Agile Manifesto: http://www.agilemanifesto.org/

The Agile manifesto states a number of founding principles for the Agile methodology.

Berenson, S. B., Slaten, K. M., Williams, L., & Ho, C. (2004). Voices of women in a software engineering course: reflections on collaboration. *Journal on Educational Resources in Computing (JERIC)*, 4 (1). Retrieved May 17, 2008 from http://doi.acm.org.dml.regis.edu/ 10.1145/1060071.1060074.

This is a case study observing upper-class female students in the Computer Science Department using Pair Programming in coursework. The study finds correlation of collaboration, productivity, and confidence in female upperclassmen using Agile in the classroom. Detailed interviews are revealing as to individual thought processes, but only three individuals were interviewed.

Boehm, B. (2005). Some Future Trends and Implications for Systems and Software Engineering Processes. *Systems Engineering , 9* (1), 1.

An analysis of various Agile and Waterfall methodologies and their relative strengths for various software applications. This is a historical review and concludes that small-to-medium projects do better with Agile while Waterfall (or Plan-Driven) is better for larger projects with bigger teams and very high mission criticality.

Bureau of Labor and Statistics. (2009). *Labor Force Statistics.* Retrieved November 28, 2010, from United States Department of Labor Bureau of Labor and Statistics: ftp://ftp.bls.gov/pub/special.requests/lf/aat11.txt

Statistical data for the Department of Labor.

Ceci, S. J., & Williams, W. M. (2007). *Why Aren't More Women in Science*. Washington, D.C.: American Psychological Association.

This is a recent and yet already frequently reviewed and cited work. The research involves scoring on cognitive testing and reveals some significant differences between men and women. Specifically, while averages are about the same, more men score at extremes (both high and low) than do women. A series of papers discusses these findings from various viewpoints and with differing conclusions. Rather than a single conclusion, there are several contradictory conclusions, and the reader is left to make their own choice.

Cockburn, A., & Highsmith, J. (2001). Agile software development, the people factor. *Computer , 34* (11), 131-133.

A brief description of the importance of people in the Agile process.

Dweck, C. (1999). Self-Theories: Their role in motivation, personality and development. Philadelphia, PA: Taylor & Francis.

A book introducing the concepts of fixed mindset related to a helpless response to a challenging learning situation versus a malleable mindset related to a mastery-oriented response. Empirical studies demonstrated that these mindsets were predictors of academic achievement.

Dybå, T., & Dingsøyr, T. (2008). Empirical studies of agile software development: A systematic review. *Information and Software Technology , 50* (9-10), 833-859.

A broad overview of literature studying the effectiveness and types of Agile methodologies. This paper also reviews empirical studies of various methodologies.

Fincher, D. (Director). (2010). *The Social Network* [Motion Picture]. Sony Pictures.

A recent film depicting the birth and growth of the social web tool, "Facebook." It clearly depicts the cutthroat and strangely geeky-macho culture in computer programming.

Halvorson, H. G. (2011, March 1). *The Trouble With Bright Girls.* Retrieved March 6, 2011, from The Huffington Post: http://www.huffingtonpost.com/heidi-grant-halvorson-phd/girls-confidence_b_828418.html

This is an extremely lucid article concerning how women's confidence in their own abilities is fundamentally different from men's and how it affects their lives and livelihoods.

Hazzan, Orit, Diversity in Computing: A Means or a Target? *System Design Frontier Journal Featured Column*, http://www.cs.cmu.edu/~cfrieze/courses/OritHazzan_Diversity.pdf

An argument for why diversity might be valuable.

Hazzan, O., & Dubinsky, Y. (2006). Can diversity in global software development be enhanced by agile software development? *Source International Conference on Software Engineering archive. Proceedings of the 2006 international workshop on Global software development for the practitioner*. (pp. 58-61). Retrieved May 15, 2008 from http://doi.acm.org/10.1145/1138506.1138520. Shanghai, China: ACM.

This paper asserts that diversity is lacking in many software teams and reveals that Agile in an academic setting produces as much communication from females as males. It concludes that Agile methods enhance gender, managerial, and opinion diversity. There is some overlap with another 2006 article (below).

Hazzan, O., Frieze, C., Blum, L., & Dias, M. B. (2006). Culture and environment as determinants of women's participation in computing: revealing the "women-CS fit". *ACM SIGCSE Bulletin archive*, 38 (1), 22-26. Retrieved May 15, 2008 from http://doi.acm .org/10.1145/1124706.1121351 .

An analysis of the cultural forces in the software development environment and how they fit with women raised in the U.S., in particular how Agile methodologies are a fit for the female psychology. The research analyzed factors such as confidence and participation in Agile methodologies in undergraduate courses in the U.S. It also analyzed data from Arab and Jewish students in Israeli classes. The information and conclusions generated are clear and straightforward, but only students are studied, not working professionals.

Herela, H. (2005, April). *Case Study: The Chrysler Comprehensive Compensation System.* Retrieved January 19, 2008, from http://calla.ics.uci.edu/histories/ccc/

A case study of the C3 payroll system which presents a more critical viewpoint of Agile than many other published articles.

James, J. (2010, April 6). *IT gender gap: Where are the female programmers?* Retrieved November 28, 2010, from TechRepublic: http://blogs.techrepublic.com/programming-and-development/?p=2386

A TechRepublic blog article with recent statistics and theories on why there are not more women programmers.

Jeffries, R. E. (1998). *Do the simplest thing that could possibly work.* Retrieved April 17, 2009, from xprogramming.com: http://www.xprogramming.com/Practices/PracSimplest.html

An excerpt from the website of one of the founders of the Agile movement describing the Agile principle of simplicity.

Keuffel, W. (2005). Decline and Fall. *Software Development*, 13 (8), 64.

This is a very short article discussing the decline of enrollment of new students into Mathematics and Technical in the U.S. While not discussing specifically the causes of the drop of women in software engineering, its suggestion that curricula changes aimed more at software than hardware which will improve general enrollment numbers seems applicable.

Leedy, P. D., & Ormrod, J. E. (2005). Practical Research Planning and Design 8th Edition. Upper Saddle River, New Jersey: Pearson Education, Inc.

This is a book on basic research methods with some practical guidelines and checklists for good research. The usefulness of this book is in guidelines for a qualitative study and interviewing techniques.

Levesque, M., & Wilson, G. (2004, November 1). *Open Source, Cold Shoulder.* Retrieved November 26, 2010, from Dr Dobbs: http://www.drdobbs.com/architecture-and-design/184415216

An article analyzing the participation (or lack thereof) of women in the Open Source Software movement.

Lockheed, M. E. (1985). Women, Girls and Computers: A first look at the evidence. *Sex Roles , 13* (3-4), pp. 115-122.

A frequently-quoted early overview of women in computing postulating some reasons as to the inequality of numbers in the field

Margolis, J., & Fisher, A. (2002). *Unlocking the Clubhouse: Women in Computing*. Cambridge, Mass.: MIT Press.

This book covers cultural and social experiences of men versus women how it relates to their confidence with computing and contains interesting discussions of play (men) vs. computing "with purpose" (women). The social phenomenon of "Geek Mythology" in the US is presented, wherein recent events have glamorized the previously nerdy image of the male computer geek. This book is a cogent analysis with powerful and pertinent imagery of social factors affecting participation of women in the computing industries, including women from other cultures besides our own.

Martin, R. (2005, August). *Agile Development Conference* . Denver, CO.

A class at the 2005 Denver Agile Development conference describing several of the best-known Agile engineering practices.

Melnik, G. and Maurer, F. 2003. Introducing Agile Methods in Learning Environments: Lessons Learned. LECTURE NOTES IN COMPUTER SCIENCE. Springer-verlag, Germany, 172-184. ISSN 0302-9743. http://www.springerlink.com/index/8R0T3294KFWF4PBX.pdf.

Another study with mixed methods where students were interviewed and surveys taken including the following questions: − Did the students enjoy agile practices? − What worked for them? − What problems did they encounter? − Whether they would use agile practices in the future (if allowed) or not? − What were their impressions of the test-driven development? − How did XP improve their learning?

Melnik, G. and Maurer, F. 2005. A cross-program investigation of students' perceptions of agile methods. In *Proceedings of the 27th international Conference on Software Engineering* (St. Louis, MO, USA, May 15 - 21, 2005). ICSE '05. ACM, New York, NY, 481-488. DOI= http://doi.acm.org/10.1145/1062455.1062543

A listing of XP practices (paired programming, tdd) and student perception. Is likely to provide a list of obvious survey questions which can be asked.

Morales, A. W. (2004). From Boom to Boom. *Software Development*, 12 (9), 11.

Short non-academic description of the boom and bust nature of software development. The main interesting comment is on one husband who complained his wife was receiving software engineering offers based on her C average, when in fact, she was earning all A's.

Murphy, L., & Thomas, L. (2008). Dangers of a fixed mindset: implications of self-theories research for computer science education. *ITiCSE '08: Proceedings of the 13th annual conference on Innovation and technology in computer science education* (pp. 271-275). ACM.

A review of literature pertaining to the concept of self-theories directly pertaining to the confidence of women in STEM disciplines. The fixed mindset is compared and contrasted with the more malleable/mastery-oriented mindset and the responses of each type of individual to offers to help are compared.

Quesenberry, J. L., & Trauth, E. M. (2007). What do women want?: an investigation of career anchors among women in the IT workforce. *Special Interest Group on Computer Personnel Research Annual Conference. Proceedings of the 2007 ACM SIGMIS CPR conference on Computer personnel doctoral consortium and research conference: The global information technology workforce* (pp. 122-127). Retrieved 5/12/2008 from http://doi.acm.org.dml.regis.edu/10.1145/1235000.1235030. St. Louis, MO: ACM.

This is a survey of women in computing and the alternative career anchors of technical versus management tracks. Three dimensions are analyzed: technical, managerial and geographical. The article concludes that a managerial choice eventually obviates the technical and that geographical choices are transitory, driven by family needs. Interviews were conducted over a 4 year period, but this was not a longitudinal study, which would be the next logical step in this research.

Regis University. (1998, September 30). §46.116 - *Informed Consent Checklist - Basic and Additional Elements*. Retrieved June 8, 2008, from Regis University: http://www.hhs .gov/ohrp/humansubjects/assurance/consentckls.htm

This is a basic checklist of elements to include in an informed consent for Regis. This covers prerequisites for any study involving human subjects.

Schelhowe, H. (June 2005). Gender questions and computing science. *CWIT '05: Proceedings of the international symposium on Women and ICT: creating global transformation*. Bremen, Germany: ACM.

As in the U.S., Computer Science is a male-dominated field in Germany. A variety of quasi-professional groups in Germany and Europe relating to women in the computer information fields are surveyed, the conclusions relate to connecting the technical and social in order to attract more women into the field. The details are left unclear, and the article appears to stereotype women as users rather than makers.

Shashani, L. (1993, March). Gender-based differences in attitudes toward computers. *Computers & Education , 20* (2), pp. 169-181.

Another oft-quoted early study on the differing motivations and approaches between men and women to Computing fields

Taft, D. K. (2009, April 30). *Do Alpha Male Geeks Scare Women Away from Programming?* Retrieved November 28, 2010, from eWeek.com: http://www.eweek.com/c/a/ Application-Development/Do-Alpha-Male-Geeks-Scare-Women-Away-from-Programming-834170/

A commentary on a recent blog post by David Heinemeier Hansson, creator of the popular Ruby on Rails Web application development platform.

Tapia, A. H. (2003). Hostile_Work_Environment.com. *Special Interest Group on Computer Personnel Research Annual Conference archive Proceedings of the 2003 SIGMIS conference on Computer personnel research: Freedom in Philadelphia--leveraging differences and diversity in the IT workforce* (pp. 64-67). Philadelphia, Pennsylvania: ACM New York, NY, USA.

This is a case study of a small startup company which was born and died during the dotcom era which describes an extremely hostile work environment for women. It describes a truly appalling and unprofessional atmosphere. The article concludes that the gold rush mentality of the time contributed to the situation, which seems a quite reasonable conclusion.

Tapia, A. H., & Kvasny, L. (2004). Recruitment is never enough: retention of women and minorities in the IT workplace. Special Interest Group on Computer Personnel Research Annual Conference. *Proceedings of the 2004 SIGMIS conference on Computer personnel research: Careers, culture, and ethics in a networked environment* (pp. 84-91).

Retrieved May 17, 2008 from http://doi.acm.org.dml.regis.edu/10.1145/982372.982392. Tucson, AZ: ACM.

Women, Afro-American, and Hispanics are underrepresented in IT careers. This article surveys issues relating to the retention of women and minorities in Information Technology. It points out the dearth of management skills in technical professionals and that drop-out levels are as significant, or perhaps even more significant than initial levels at entry of these minorities. While presenting thought-provoking analysis, this is primarily a review of other literature and research rather than conducting primary research of its own.

Technisource. (2010, October). *Women in IT Careers Survey.* Retrieved November 27, 2010, from Slideshare: http://www.slideshare.net/monsterww/women-in

A monster.com survey, intriguing responses and age bracket spread.

Trouth, E. M. (2002). Odd girl out: an individual differences perspective on women in the IT profession. *Information Technology & People , 15* (2), 98-118.

An article explaining the two primary theories: essentialist vs social construction and research based upon detailed interviews.

Wardell, M., Sawyer, S., Mittory, J., & Reagor, S. (2006). Gender and IT professionals in the United States: a survey of college graduates. *Labour & Industry*, 16 (3), 39-59. Retrieved May 17, 2008 from http://find.galegroup.com.dml.regis.edu/itx/start.do?prodId=AONE Gale Document Number:A153761989.

Analysis and detail of statistics concerning the participation of women in the field of computing as compared to men alongside of other statistics such as education, amount of time commuting, and a wide of other details. Extensive interview study was conducted over a six-month period, and considerable detail is provided. Conclusions are highly suggestive as to the importance of family balance for women versus men and how this might contribute to the smaller numbers of women. The study only interviewed subjects under 40, however, so the workforce representation is skewed to a younger population. Future research exploring the perceived advantages of large corporate environments with formal family-friendly policies is indicated to confirm the family-friendly aspects of larger corporations as a factor in women's decisions.

Williams, L., Alexander, W. Berenson, S., Knight, V., Koster, B., Osborne1, J., Vouk, M., Sung Yoon, Collaboration through Agile Software Development Practices: A Means for Improvement in Quality and Retention of IT Workers 2006. In *The National Science Foundation's ITWF & ITR/EWF PRINCIPAL INVESTIGATOR CONFERENCE* (Raleigh, NC. April 4-6, 2006) North Carolina State University, 183-185 at http://agile.csc.ncsu.edu/itwf/proceedings_small.pdf#page=183

Another article demonstrating how agile practices benefitted women in academic curricula.

Williams, Laurie, Kessler, Robert R., Cunningham, Ward, and Jeffries, Ron, Strengthening the Case for Pair-Programming, *IEEE Software*, Vol. 17, No., 4, pp. 19-25, July/Aug 2000. January 2009 update: Retrieved May 21, 2011 from http://collaboration.csc.ncsu.edu /laurie/Papers/ieeeSoftware.PDF

A 1999 study of university students working in pairs with classic Pair Programming practice where quality and productivity were measured in terms of post-code-completion passing of test cases and amount of man-hours. This study found that the number of man-hours required for programmers to individually produce code of the same quality as that produced in pairs was actually higher. Using interviews and surveys, it also found that students felt considerably higher confidence in their work, found that they kept each other on task better, and 9096% stated that they found higher satisfaction working in pairs as opposed to working individually.

Williams, L. (2006, May). Debunking the Nerd Stereotype with Pair Programming. *Computer* , 39 (5), pp. 83-85.

A 2006 review of literature and studies on the benefits of Pair Programming and collaboration in the classroom.

Willis, J. W. (2007). *Foundations of Qualitative Research.* Thousand Oaks, CA, USA: Serge Publications, Inc.

Research methods book.

Wilson, G. V. (2007, May 21). *Why Aren't More Women in Computing?* Retrieved June 4, 2008, from Dr. Dobb's Portal: http://www.ddj.com/architect/199700375

This reviews the book, "Why Aren't more Women in Science." While not as authoritative as Journal articles, it debunks in plain language some current myths circulating concerning why there aren't more women in computing and references useful articles on the topic.

Wilson, M. L., & Wilson, G. (2004, November 1). *Open Source, Cold Shoulder*. Retrieved June 4, 2008, from Dr. Dobb's Portal: http://www.ddj.com/architect/184415216

Despite proclaiming itself as an equalizing movement and its use of Agile processes, the open source movement has even less women involvement than commercial software. A simple survey of speaker names for O'Reilly's annual Open Source Conference reveals this fact. This article theorizes that the antisocial geek culture is discriminatory, and that the use of Agile is less collaborative than in other circumstances. While not authoritative as some other more academic journal articles, this article adopts a useful critical view of the open source community practices.

Glossary

Automated Build: See Continuous Integration.

Computer Programmer: A person who write code in a computer programming language.

Continuous Integration: A server is responsible for polling source control for recently checked-in changes. It gets a copy of the latest relevant code and runs a build. If the compilation succeeds it then runs the existing set of unit tests and reports success or failure.

Pair Programming: The practice of two developers working together on code. Traditionally done side-by-side at a single workstation with one at the keyboard and the other providing input an acceptable alternative is for one programmer to write Unit tests while the other one writes the code which makes them pass.

Planning Game: Meetings where developers, testers and stakeholders meet to estimate the level of effort for implementing any given feature and prioritize it so that it can be scheduled into an iteration.

Software Developer: See Computer Programmer and Software Engineer.

Software Engineer: See Computer Programmer and Software Developer.

Test Driven Development (TDD): The practice of writing broken unit tests before implementing the code which makes them pass.

Unit Tests: These are written in the same language as the code itself using a unit testing framework and are run at will by developers while working and also as part of a continuous integration.