Software Engineering Education –

Does gender matter in project results? – A Chilean Case Study

Maíra Marques
Computer Science Department
Universidad de Chile
Santiago, Chile
mmarques@dcc.uchile.cl

Abstract— Teaching software engineering in a practical course in academia is considered one of the best strategies to help students understand what they will face in industry. Teamwork, coordination, communication are soft skills that are demanded in real life, but how to improve these courses to be more effective and yield better results is not clear. Software engineering – as with most computer careers – is known to be short on women, but the participation of women improves diversity to student teams. Women may have personal characteristics that are different from the ones we normally see among male software engineering students. Do, mixed gender software teams have better project results than one-gender teams? After assisting software engineering courses for four years, I observed mixed gender teams to be better. With this idea in mind I performed a case study where I analyzed the behavior and the results of software projects over nine semesters. The obtained results show that mixed gender teams were more effective and coordinated.

Index Terms— Software engineering education, gender influence, effectiveness, teamwork

I. Introduction

Software engineering (SE) is a discipline oriented to applying a systematic, quantifiable approach to the development, operation, and maintenance of software [1]. As any engineering discipline, it is tightly linked to the industry; therefore, the software engineering curricula used in computer science programs need to prepare students for the world of industrial software development [2].

Fox and Patterson [3] pointed out that while new college graduates are good at coding and debugging, employers complain about other missing skills that they think are equally important (e.g. working with non-technical customers, working in teams, etc.). The usual faculty reaction to such criticism is that they are trying to teach principles.

It is well known that the industry needs people able to work in teams to develop and maintain software. However delivering young professionals that are proficient in technical and operative issues is a great challenge for educational institutions [4] [5] [6] [7]. Begel and Simon [8] pointed out that universities are striving to prepare students for the industrial scenario and to become lifelong learners, able to keep pace with the software engineering advances. Typically the universities tend to adequately teach the technical aspect of software engineering, but do not address operative issues such as managing projects/risks, defining/adjusting software processes, and performing teamwork.

Teams are the basis of software development organizations today, since the challenge and complexity of projects has made software projects unachievable for individuals. In this setting, professionals are not only required to have state-of-the-art knowledge and technical abilities. It is assumed that software will be built in cooperative and diverse teams, but the reality of software engineering teaching is sometimes different. Effective teamwork requires mastering specific abilities, such as leadership, coordination, conflict management and other skills. This implies that if higher education wants to meet the requirements of the students' future professional lives, it has to address the acquisition of such and have the technology to support the transference process. [9].

The participation of women in science, technology, engineering, mathematics programs and careers is disproportionately lower than that of men [10] [11]. Zeldin and Pajares [12] showed with a comparative study that social persuasion is one of the primary sources of self-efficacy beliefs for women in technology fields of study.

Women and men may have different goals when they participate in project-oriented courses; women's goals are more oriented towards teamwork and peer learning, than men [13]. In environments that are male-dominated, however, social learning strategies such as help seeking can be perceived as weakness [14], which may create situations that negatively influence the motivation, self-efficacy, and future engagement of students. Project-based learning presents opportunities for the pursuit of social learning goals, and evidence suggests that hands-on project activities contribute positively to the learning and satisfaction of both men and women [15].

Women are normally under-represented in IT educational programs and in the IT workforce [16]. But specifically in software engineering, there is not much reported on the gender difference. Nelson [17] stated in a recent article that diversity is important and that woman in computer science in general are

978-1-4799-8454-1/15/$31.00

not encouraged to pursue a computer science degree. He also speaks about some case studies that helped increase the participation of women. Diversity needs to be increased and stimulated in all areas because it results in diverse points of view that are needed to solve different problems. In Chile, our reality is not formally known; there is only anecdotal information about gender subjects, with few factual data reports.

At Universidad de Chile, we teach software engineering using a hands-on methodology where students have to develop their projects and deal with the client in their own way. Of course they are not just thrown to the lions (clients). The lions are known (inside clients) and they have full support of the instructor's team.

In order to identify if there is a difference among the results of software engineering projects in academia related to gender, two hypothesis were proposed: mixed gender teams are more coordinated (H1), the participation of women in software engineering teams makes teams work more effective (H2).

In this case study I will show that software engineering teams that have female participation as team members are more effective and have better odds of achieving positive results.

II. Background

The wide use of technology creates very complex situations that professionals in IT need to be able to manage [19]. Typically, most of the situations that engineers find in practice differ from those they found previously. Therefore, engineers must be able to use engineering concepts that they have learned during their education to create solutions for real-world problems [4]. However, it is well known that the software industry is usually unsatisfied with the level of real-world readiness possessed by new engineering graduates entering the workforce [4] [20] [21].

Parnas [22] defined three steps required to produce software engineering programs: (1) define the possible tasks that a software engineer is expected to perform, (2) define a body of knowledge required for the software engineer, then (3) transfer such knowledge as a training program. This definition clearly identifies the need for transferring the most important knowledge to the students and also the skills required to use it in future developments. This definition is aligned with the role that Denning states for software engineers [4]. These definitions suggest that we need to address two main challenges: (1) identify and transfer the key knowledge to the students, and also (2) encourage students to develop the skills required to apply such knowledge in real-world work scenarios. According to Parnas [20] [21] [22], these challenges should be addressed through hands-on experiences that expose students to some frequent problems before they become stranded on their own.

There are also other researchers that identified challenges on other aspects of the transference process: teaching software processes with traditional methods (expositive lectures) [23] [24], educational challenges [25] and features of the development process or evaluating the work performed by the students [2] [26].

According to the selectivity hypothesis [27], males process information in a heuristic manner – paying attention to cues that are highly available and salient in the context. Females process information in a comprehensive manner, attempting to assimilate all available possibilities. In our problem solving and project development, the selectivity hypothesis implies that males and females will respond differently to different insights, which should impact problem-solving decisions. These differences imply that the information, the environment presents and how the environment presents information will impact on decisions and on how females and males interact and contribute to a project.

In this work, coordination will be measured as a coordination self-efficacy measure, where we define self-efficacy as the strength of belief that one can complete a task. The role of self-efficacy is especially pronounced for women in engineering programs [28]. In the social cognitive theory model of self-efficacy proposed by Bandura [29], three factors that contribute to self-efficacy are identified: mastery experiences, vicarious experiences (i.e., identification with role models) and social persuasion (positive feedback from others). Mastery experiences, of course, are what educational experiences generally seek to specifically provide.

Project-based courses reflect authentic engineering experiences as they facilitate a sense of domain mastery and thus lead to an increase in engineering self- efficacy [30]. Self-efficacy is also closely related to motivation, as a student's confidence directly impacts his/her course of action [30, 31]; self-efficacy in engineering and related domains is a prerequisite for both choosing an engineering major and persisting in it [28, 20]. Finally, engineering self-confidence and self-efficacy can vary with demographics; they are notably lower for women than their male counterparts [32, 32]. There is evidence that men show increases in self-efficacy (across a range of constructs) after taking project-based courses, particularly those in which they build a physical prototype, but no similar evidence was found for women [3].

III. Related Work

Glass [34] points out that when he entered the workforce in the late 50' there were no shortage of women practitioner's, but in the middle 80's something changed and the rates of women entering the computer science field in general dropped out. And we are still suffering and seeing that. In South America, things seemed to be a little bit different. Although there is no official data to refer to, people that were working in the field back in the day when computer science began in Chile, report that there was a shortage of women since the beginning.

Some authors propose single-gender classes as one of solutions for this issue [35] [36] [37] [38]. They report that one-gender classes make males or females perform better on overall courses. But all these authors were talking about courses in general and not about project courses, where students have to deal with each other and have to face the reality of dealing with a client. The main idea of the software project courses is to simulate the reality of what they will face in real life when they start working in industry. In real life it is not feasible to choose to work just with one gender.

The Credit Suisse Research Institute [39] reports on an extensive analysis demonstrating the importance of having diversity on almost all company levels. In their investigation they even report that companies that have females on their board perform much better than companies that have only male board members

Burnett [40] reported an evaluation about the pluralism of gender related to spreadsheet use (feature-related confidence, playful exploration and usage in general). Her results show that there are differences in gender and that mixed genders completed each other in terms of accomplishment than one-gender teams.

IV. Experimentation Scenario

One of the last two software engineering courses at the computer science program at University of Chile is Software Engineering II (CC5401). It is a mandatory class for fifth year students (out of a six-year undergraduate program). For this study we monitored teams during nine semesters. The primary goal of this course is to teach students software engineering in a ¨hands-on¨ way, by resembling an industrial setting.

Students are put in teams of 4 to 7 students, according to their capabilities by a MBTI based tool, in use in the last five years [41]. Each student chooses the role they want to have on the team, and if more than one team member want to perform the same role, students have to agree among them.

The projects are real projects that have real clients and a problem that can be solved with information system software that should be addressed in the 15-week course time frame. The students must arrange weekly meetings with their clients and the course have four milestones where they have to deliver something tangible. In the first milestone, they must explain that they understood their client's problem, and show a mock-up solution of the software. In the second milestone they must show a demo of the software with the basic features that the client required. With the third milestone they must show the demo of the full software. And finally in the last milestone, they must deploy and deliver the software to the client. In each one of these milestones, the instructor evaluates student's presentations and the client evaluates the demos. After evaluations, students receive a grade between 1 and 7.

Teams are also supposed to use an iterative development lifecycle in each one of their milestones. And during the iterations they have to produce documents regarding user requirements, software requirements, design specifications, test cases and integrated tests.

In this context there are some challenges that are not minor, and they put some restrictions in our context such as:

- Type of student - 5th year students, neither used to and not prone to working in teams.
- Small projects - since the course has only 15 weeks, the projects need to fit this short time frame of development, and all the projects should have equivalent average size (around 100 function points).
- Real clients – clients are internal to the university, they normally do not expect much of the project, and they do not fully commit with it.
- Technology – most projects are brownfield ones, meaning that students do not have much of a choice in the technology that will be used in implementation of the solution.
- Type of project – all developed products can be classified as Web information systems, and typically the complexities are similar.
- Roles – each team has defined roles (project manager, analyst, designer, developer and tester) and, depending of the size of the team, there could be more than one team member with the same role.
- Product quality – to guarantee the quality of the product, students should use some of the software engineering best practices and provide quality verification.

Trying to help students minimize these challenges, students have to meet their monitor weekly, and the monitor is someone outside the evaluation team. Students are assured that anything they say to the monitor will not have repercussions on their grades or on instructor perceptions of their team. Monitors lead the meeting, asking the students questions according to the milestones. They also assist students with guidance in technical and social problems that they may have. Monitors do not tell teams what they should or should not do. They are not project managers, coaches or scrum masters.

In this work, nine semesters of the software engineering course are reported (from 2010 until 2014). In total we had 151 students and only 16 of them were women (10.60%), 27 teams were formed and just 10 of them had women's participation. And due to the lack of women it was not possible to set up a software engineering team of women.

Methodology

In order to assess the students' self-efficacy (coordination), a formal peer-review assessment was conducted. Such assessment have been used and evolved in this course over the last seven years. The assessment determines how well each person fits with the behavior expected from a teammate. The students were assured that all opinions would be anonymous and that there would be no formal grades given based on that peer- assessment process.

The instrument used to perform the peer-reviewing process considers eight items that should be rated. The rates were indicated using a five-point Likert scale (from 'strongly disagree' to 'strongly agree') [42]. The evaluated items were the following:

1. He/she assumes the project as a team effort, providing support in project tasks.
2. He/she is able to seek help when problems are found.
3. He/she fulfills his/her tasks properly, working transparently and generating the most value out of each working day.
4. He/she demonstrates initiative to achieve project success.
5. He/she shows a communicative attitude facilitating the teamwork.
6. He/she has maintained good communication with client, generating value to project execution.

7. He/she demonstrates interest in improving performance on the execution of his/her activities and role within the project.
8. He/she is able to admit to mistakes and accept criticism.

These questions are interpreted through five concepts: commitment (1, 2), communication (4), coordination (5), attitude (3, 7, 8), contribution (2,6). These five concepts are our reading and interpretation on what is relevant to teach software engineer. Given the nature of the collected data, a research approach inspired by the grounded theory was chosen to conduct this study. According to Glaser and Strauss [43] grounded theory is the discovery of theory from data systematically obtained from social research.

In order to analyze the effectiveness of the teams, and validate our second hypothesis (the participation of women in software engineering make teams work more effective) I took into consideration the final grades of software projects, considering mixed and non-mixed gender teams and the deployment and use of the software (successful project) as measures. Provided that the instructor, teaching assistant, course content and dynamics did not change during the observed semesters we can consider these teams comparable.

V. Results

Figure 1 shows women participation during the nine semesters being reported. Only sixteen female students enrolled in the software engineering course during the period being observed (approx. 10.6%).

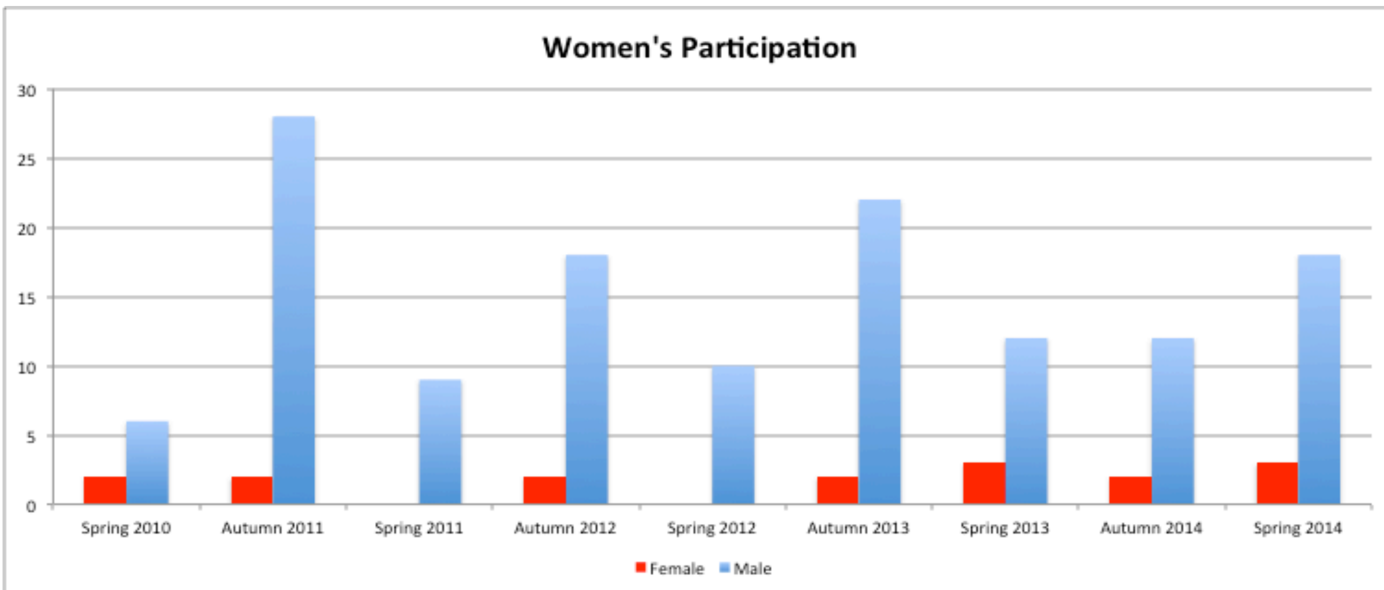

Figure 1 – Women's participation

A prior analysis of grades was performed with the students that were enrolled in these nine semesters to check if the female participants generally performed better then the males (in terms of grades). Only two female students were among the top students, and 15 male students that participated in this research were among the top students. We can say that the amount of top women participating in this sample is comparable with the amount of women participating in this sample.

In order to evaluate first hypothesis (teams that have women as team members are more coordinated) we used the results of the peer-reviewing process, as a way of self-efficacy perspective. Because they represent the teammates opinions, which probably are more accurate than the instructor, teaching assistant or monitoring opinion.

Figure 2 shows the obtained results for the peer evaluation and the concepts derived from the peer evaluation questions.

The results indicate that our hypothesis is aligned with the results that mixed gender teams have better coordination (in terms of self-assessment) than male only teams. The difference between mixed gender teams and male teams is 0.81; mixed gender teams are 20% higher in terms of self-perception than male teams. The other concepts have differences but they are not so significant as to be statistically valid (commitment 9.7%, communication 7.8%, attitude 9.6%, contribution 9.4%). The median of the concepts of the male teams was: commitment 5, communication 5, coordination 4, attitude 4, contribution 5 and collaboration 4; for the mixed gender teams the median values were: commitment 5, communication 5, coordination 5, attitude 5, contribution 5 and collaboration 5.

Looking in the roles performed by women in projects, there is no clear tendency; the women performed all the team roles that we have (project manager, tester, analyst, designer and developer); the only role that have just one female was the designer; all the other have proportional female participation. So, the women roles are not a relevant data to our results.

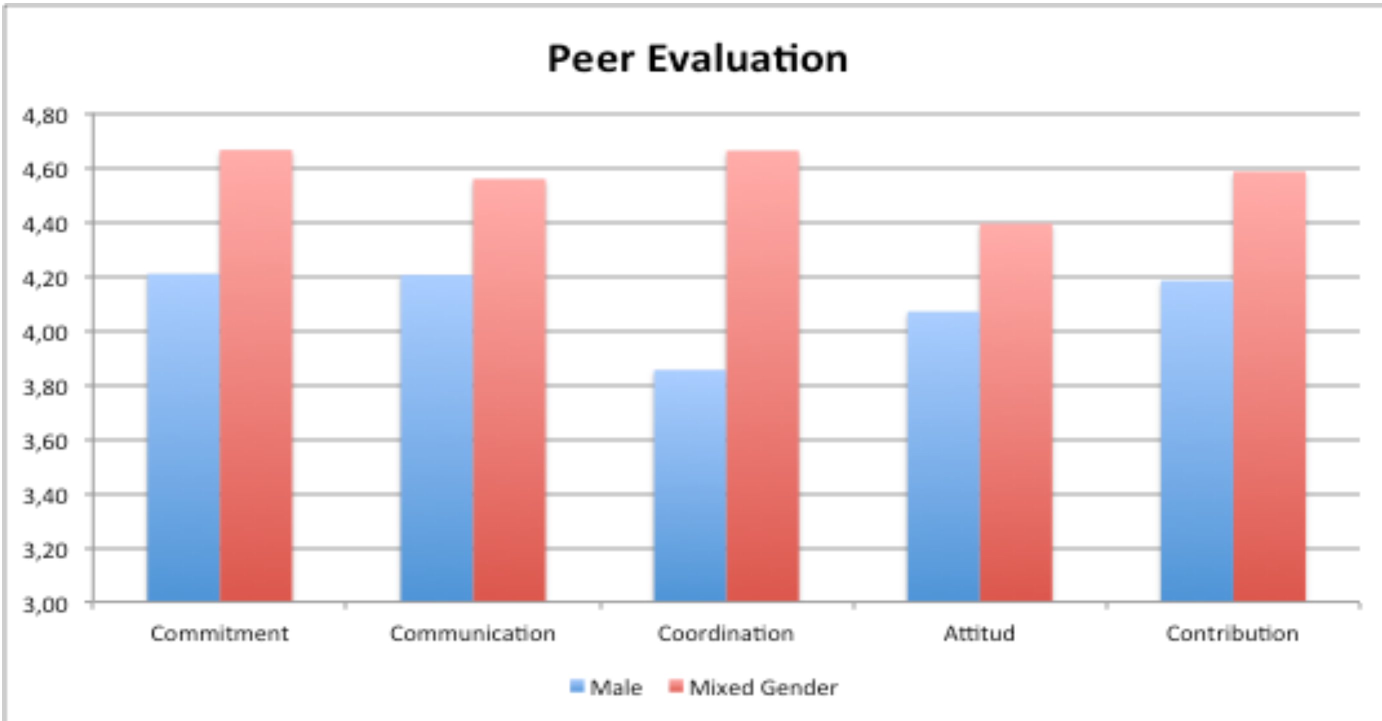

Figure 2 - Peer Evaluation

To analyze our second hypothesis (H2 - the participation of women in software engineering teams makes teams work more effective) the project grades and the software deployment were taken into consideration.

Figure 3 shows the project grades and the course final grades for male teams and mixed gender teams. The grades of mixed gender teams are 16% higher when we are talking about project grades and 13% higher when we are talking about final grades.

Figure 4 shows the project results. It also shows which projects were successful and which were used by the client (deployed) for the mixed gender teams. Figure 5 shows the same for the male teams.

The mixed gender teams have 10 projects. From these, 9 of them were deployed resulting in a success rate of mixed gender teams of 90%. The male teams have 17 projects and only 7 were implanted, so the success rate of male teams was 42%.

We conclude that the second hypothesis is aligned with our results, and that the participation of women on software engineering teams makes teams work more effective (H2).

In order to ensure that the data source used in this study is valid and reliable, the Cronbach's Alpha coefficient was calculated from the eight questions of the peer-review form, from the 138 respondents (total of 151 participants) and we had a Cronbach's Alpha of 0.82. The closer Cronbach's Alpha coefficient is to 1.0 the greater the internal consistency of the items is in the scale. According to Gliem and Gliem [44], a value of Cronbach's Alpha of 0.8 from Likert-type scales is good and is considered a reasonable goal.

Figure 3 - Project grades

Mixed Gender Teams									
Year	2010	2011		2012		2013		2014	
Semester	Spring	Autumn	Spring	Autumn	Spring	Autumn	Spring	Autumn	Spring
Team 2	✓								
Team 5		✗							
Team 9		✓							
Team 11				✓					
Team 16						✓			
Team 17						✓			
Team 19							✓		
Team 21							✓		
Team 24								✓	
Team 27									✓

Figure 4 - Project Deployment for Mixed Gender Teams

Male Teams									
Year	2010	2011		2012		2013		2014	
Semester	Spring	Autumn	Spring	Autumn	Spring	Autumn	Spring	Autumn	Spring
Team 1	✓								
Team 3		✗							
Team 4		✗							
Team 6		✓							
Team 7			✗						
Team 8			✗						
Team 10				✓					
Team 12				✓					
Team 13					✗				
Team 14					✗				
Team 15						✗			
Team 18						✓			
Team 20							✗		
Team 22								✗	
Team 23								✓	
Team 25									✗
Team 26									✓

Figure 5 - Project Deployment for Male Teams

VI. Threats to Validity

Analyzing the threats to validity of this case study, we found that the internal validity within the course is not a problem, since students may respond differently at different times. But, each time a student takes the course the project will be different. Cronbach's Alpha shows that there is consistency in the answers of students. The major threat regarding the conclusion validity is the quality of the data collected. The students are expected to answer the peer review assessment in the most truthful way, but sometimes students took the comrade behavior, and did not evaluate a colleague properly. Another threat is the project grades. To minimize this threat the grades are given by instructor, teaching assistant and client/user. The construct validity has one threat; the subjects are part of a course, where they are graded; this implies that the students may bias their data, as they believe that it will give them a better grade. To minimize this threat at the beginning of the course the instructor stated that the peer review was voluntary and anonymous.

VII. Conclusions

In this paper we have described a case study comparing the results of mixed gender teams and male teams in student software projects in academia. Two hypotheses were proposed: mixed gender teams are more coordinated and are more effective. The results show that in our context our hypotheses are aligned with the outcomes.

This paper has two main contributions: unfortunately, we found that female participation in software engineering (CS degree) is not so different from studies in other countries [45]. The paper also contributes to finding that the mixed gender teams can improve the results of software engineering teams in academia. This is reportedly happening in other areas. Mixed-gender teams offer more coordination and focus to student teams, help them focus their efforts on what really matters.

Mixed gender teams in software engineering education improves the results of software engineering teams in academia. Students learn to work on real-world projects with substantially better results.

The next step considers a more in-depth study on the gender subject, analyzing the recorded data from team meetings, an evaluation of which skills are increased in a mixed gender software teams in academia and why mixed genders perform better in terms of self-efficacy.

Acknowledgment

The work of Maíra Marques Samary was supported by the PhD Scholarship Program of Conicyt Chile (CONICYT-PCHA/Doctorado Nacional/2012-21120544). This work was also partially supported by the Project Fondef Idea, grant: IT13I20010.

References

[1] IEEE Standards Collection: Software Engineering. IEEE Standard 610.12-1990, IEEE 1993.

[2] Chen, J., H. Lu, L. An, and Y. Zhou. “Exploring Teaching Methods in Software Engineering Education”. Proceedings of the 4th International Conference on Computer Science & Education (ICSE). IEEE Press (2009): 1733-1738.

[3] Fox, A., and D. Patterson. “Crossing the Software Education Chasm”. Communications of the ACM, Vol. 55, no. 5 (2012): 44-49.

[4] Denning, P. J. “Educating a New Engineer”. Communications of the ACM, Vol. 35 no. 12 (1992): 82-97.

[5] Hilburn, T. B., and D. J. Bagert. “A Software Engineering Curriculum Model”. Proceedings of the 29th Annual Frontiers in Education Conference. IEEE Press. (1999): 12A4-6-11.

[6] Gorla, N., and Y. W. Lam. “Who Should Work with Whom? Building Effective Software Project Teams”. Communications of the ACM, Vol. 47, no. 6 (2004): 79-82.

[7] Wellington, C. A., T. Briggs, and C. D. Girard. “Examining Team Cohesion as an Effect of Software Engineering Methodology”. 2005 Proceedings of the Workshop on Human and Social Factors of Software Engineering. ACM Press. (2005): 1-5.

[8] Begel, A., and B. Simon. "Novice Software Developers, All Over Again." Proceedings of the 4 International Workshop on Computing Education Research. ACM Press (2008): 3-14.

[9] Rugarcia, A., R. M. Felder, D. R. Woods, and J. E. Stice. 2000. The Future of Engineering Education. A Vision for a New Century. Chemical Engineering Education 34, 1, 16-25.

[10] Baker, D. 2002. Where is gender and equity in science education? J. of Research in Science Education 39, 659-664.

[11] National Center for Educational Statistics. 2006. Digest of education statistics. Retrieved March 1, 2013 from http://nces.ed.gov/progrms/digest/

[12] Zeldin, A. L., Pajares, F. 2000. Against the odds: Self-efficacy beliefs of women in mathematical, scientific and technological careers. American Educational Research Journal, 37, 215-246.

[13] L. F. Denton and D. McKinney, "Affective factors and student achievement: a quantitative and qualitative study," 34th ASEE/IEEE Frontiers in Education Conference, pp. 20–23, 2004.

[14] C. M. Vogt, D. Hocevar, and L. S. Hagedorn, "A social cognitive construct validation: determining women's and men's success in engineering programs," The Journal of Higher Education, vol. 78, no. 3, pp. 337-364, 2007.

[15] Y. V. Zastavker, M. Ong, and L. Page, "Women in engineering: exploring the effects of project-based learning in a first-year undergraduate engineering program," 34th ASEE/IEEE Frontiers in Education Conference, pp. S3G-2 – S3G-6, 2006.

[16] [Chabrow, E. “Facts Behind the Decline of Women in IT,” CIO Insight, 2007. Retrieved March 23, 2007 from http://www.cioinsight.com/article2/0,1540,2106559,00.asp

[17] Nelson, Beryl. "The data on diversity." Communications of the ACM 57.11 (2014): 86-95.

[18] Marks, M. A., and F. J. Panzer. 2004. The Influence of Team Monitoring on Team Processes and Performance. Human Performance (Taylor & Francis) 17, 1, 25-41.

[19] Callahan, D. W. “Development of an Information Engineering and Management Program”. IEEE Transactions on Education, Vol. 46, no. 1 (2003): 111-114.

[20] Vaughn, R. B., and J. Carver. "The Importance of Experience with Industry in Software Engineering Education." Proceedings of the 19th Conference on Software Engineering Education and Training. IEEE Press. (2006): 19-24.

[21] Huang, L., and D. Port. "Relevance and Alignment of Real-Client Real-Project Courses via Technology Transfer." Proceedings of the 24th Software Engineering Education and Training (ICSE). IEEE Press. (2011): 189-198.

[22] Parnas, D. L. “Software Engineering Programs are not Computer Science Programs”. IEEE Software, Vol. 16, no. 6. (1999): 19-30.

[23] Oh, E., and A. van der Hoek. “Adapting Game Technology to Support Individual and Organizational Learning”. Proceedings of the 13th International Conference on Software Engineering and Knowledge Engineering. IEEE Press. (2001): 347-354.

[24] van der Duim, L., J. Anderson, and M. Sinnema. “Good Practices for Educational Software Engineering Projects”. Proceedings of the 29th International Conference on Software Engineering. IEEE Press. (2007): 698-707.

[25] Navarro, E. O., A. Baker, and A. van der Hoek. “Teaching Software Engineering Using Simulation Games”. Proceedings of the International Conference on Simulation in Education. IEEE Press, 2004.

[26] Chen, C. Y., and P. P. Chong. “Software Engineering Education: A Study on Conducting Collaborative Senior Project Development“. Journal of systems and Software, Vol. 84, no. 3 (2011): 479-491.

[27] Meyers-Levy, J. Gender differences in information processing: A selectivity interpretation. Cognitive and Affective Responses to Advertising, Lexington Books, 1989, 219–260.

[28] R. M. Marra, K. A. Rodgers, D. Shen, and B. Bogue. 2009. "Women engineering students and self-efficacy: a multi-year, multi-institution study of women engineering student self-efficacy," Journal of Engineering Education, Vol. 98, pp. 27-38.

[29] A. Bandura. 1997. Self-Efficacy: The Exercise of Control. New York, New York: W. H. Freeman and Co.

[30] M. K. Ponton, J. H. Edmister, L. S. Ukeiley, and J. M. Seiner. 2001. "Understanding the role of self-efficacy in engineering education," Journal of Engineering Education. Vol. 90, pp. 247-251.

[31] L. J. Sax. 2008. The Gender Gap in College: Maximizing the Developmental Potential of Women and Men San Francisco, CA: Jossey-Bass.

[32] D. Chachra and D. Kilgore, "Exploring gender and self-confidence in engineering students: a multi-method approach," In 116th ASEE Annual Conference and Exposition. 14-17 June 2009. Proceedings of the 116th American Society for Engineering Education Annual Conference.

[33] M. Besterfield-Sacre, M. Moreno, L. Shuman, J., and C. Atman, J. 2001. "Gender and ethnicity differences in freshmen engineering student attitudes: a cross-institutional study." Journal of Engineering Education, Vol. 90, pp. 477- 494.

[34] Glass, Robert L. "The Gender Gap: Is it a computing problem or simply a computer science problem?." IEEE Software 29.2 (2012): 0088.

[35] Dennis, A. R., Kinney, S. T., & Hung, Y.-T. C. 1999. “Gender differences in the effects of media richness.” Small Group Research, Vol 30 pp 405-437.

[36] Vashti Galpin. “Women in Computing Around the World.” SIGCSE Bulletin Vol. 34, No.2, 2002 June 94-100

[37] Adams, Joel. C., Bauer Vimala, and Shakuntala Baichoo, “An Expanding Pipeline: Gender in Mauritius,” ACM Press, SIGCSE 2003

[38] Zeid, Amir, and Rehab El-Bahey. "Impact of introducing single-gender classrooms in higher education on student achievement levels: A case study in software engineering courses in the GCC region." Frontiers in Education Conference (FIE), 2011. IEEE, 2011.

[39] https://www.credit-suisse.com/newsletter/doc/gender_diversity.pdf (accessed on October 24, 2014)

[40] Burnett, Margaret M., et al. "Gender pluralism in problem-solving software." Interacting with computers 23.5 (2011): 450-460.

[41] Silvestre, L., “Diseño de Equipos de Desarrollo de Software en Escenarios Universitarios”, Master Thesis, Universidad de Chile, 2012 (in Spanish).

[42] Likert, R.. “A Technique for the Measurement of Attitudes. “Archives of Psychology, 1932.

[43] B. G. Glaser, and A. L. Strauss. The Discovery of Grounded Theory: Strategies for Qualitative Research. New York: Aldine de Gruyter, 1967.

[44] J. A. Gliem, and R. R. Gliem. "Calculating, Interpreting and Reporting Cronbach’s Alpha Reliability Coefficient for Likert-Type Scales." In Proceedings of the Midwest Research to Practice Conference in Adult, Continuing, and Community Education, 2003.

[45] Quesenberry J. L., Trauth E. M., “The (dis)placement of women in the IT workforce: an investigation of individual career values and organizational interventions”, Information Systems Journal 2012, 457-473.