The Relevance of Software Development Education for Students

Janet Liebenberg, Magda Huisman, Elsa Mentz

North-West University, South Africa

0481

The Asian Conference on Society, Education and Technology 2013

Official Conference Proceedings 2013

Abstract

It is widely acknowledged that there is a shortage of software developers, not only in South Africa, but also worldwide. Despite reports on a gap between industry needs and software education, the possible gap between students' needs and software education has not been explored in detail. Students want to take courses and be educated at university in courses and projects that clearly relate to their lives and their goals. This article reports on a quantitative study of 297 Computer Science students. All 12 factors identified from the data obtained from questionnaires indicated high reliability coefficients. The analysis suggests that there is a gap between students' needs and software development education, especially for certain groupings such as female students and students who rate their own academic performance as low. Software development education has more social relevance for students, but both personal and professional relevance is relatively lacking. We conclude that the identified groups of students should receive special attention to offer relevant software development education in their eyes and to meet the demand for software developers.

Keywords: Software development education; Software development students; Computing curricula; Software industry

1. INTRODUCTION

South Africa has a shortage of professionals with ICT skills (Harris, 2012), but this is a worldwide phenomenon (McAllister, 2012; Connolly, 2013). The Career Junction Index (CJI), which monitors online employment trends in South Africa, has published its 2013 findings on the Information Technology sector and found that software development (SD) is at the forefront of the country's significantly high IT demand (CareerJunction Index, 2013). South Africa therefore needs more software developers, but will students take courses if they do not find them relevant?

In promoting education, software developer Bill Gates (2007) focuses on his foundation's "3Rs" of "Rigour, Relevance and Relationships". The central pillar of relevance highlights the need for students to have courses and projects that clearly relate to their lives and their goals. At the same time, the SD industry expects students to be educated in courses and projects that are professionally relevant and that prepare them well for the workplace (Moreno et al., 2012). Numerous studies have investigated the relevance of SD education at university level, but have been mostly concerned with ways to make the education relevant to the needs of industry. Lethbridge (2000) analysed the relevance and depth of the knowledge that software professionals had received as part of their university education and identified a significant mismatch between software education and industry in terms of the knowledge needed by software engineers to meet the industry's requirements. Other researchers reported similar gaps (Kitchenham et al., 2005; Kim et al., 2006; Surakka, 2007; Lee & Han,2008;Aasheim et al., 2009; Moreno et al., 2012), and for Lethbridge et al. (2007) filling them is one of the most critical challenges for SD educators. In his article entitled “Higher education: who cares what the customer wants?” Reisman (2005) argues that studies in higher education often neglect to collect data from two of higher education's most important players—faculty and students. In the light of the shortage of software developers this study therefore aims to investigate the relevance of SD education in the eyes of students.

2. CONCEPTUAL AND THEORETICAL FRAMEWORK

In this section, we discuss the background and literature behind the main concepts of this study, namely the concept of relevance, the type of students currently in university classes, and the literature on software development education.

2.1 The concept of relevance

Relevance is broadly defined as closely connected or appropriate to the matter in hand; or having significant and demonstrable bearing on the matter at hand; or practical and especially social applicability (Oxford English Dictionary, 2013; Merriam-Webster Dictionary, 2013). Labaree (2008), who works in educational research, emphasises relevance as a function not only of person and purpose, but also of place and time. He argues that in this field the question “useful to whom and for what?” needs to be answered because a wide array of actors is involved, including teachers, students, parents, textbook publishers and curriculum developers.

Relevance in the educational context can be defined as the applicability of what is taught to the needs and interests of students and society (Holbrook, 2009).The process of instruction and learning is designed to make what is learnt relevant/current to the time so that it can be implemented in the social environment. As a result, the student then sees the learning as meaningful, timely, important and useful, and it builds on the intrinsic motivation of the student for self-concern, self-involvement, self-appreciation and self-development (Holbrook, 2009).

Holbrook (2003) suggests three relevance perspectives which are used in this paper to analyse relevance:

- Social relevance – the "useful in society" perspective, which is a perceived need for the society;
- Personal relevance – the "interest" perspective, which directly relates to concerns in the students' immediate environment or area of interest;
- Professional relevance – the "important for the course they are studying" perspective, which relates to the content of the curriculum that has to be interesting and useful to students.

2.2 The student in the software development class

Most students in current SD classes belong to the so-called Net generation, also known as the Millennial Generation or Generation Y. The Net generation (especially people born in the US and Canada from the early 1980s to the late 1990s) is characterised by students who may have never known life without the Internet (Cheese, 2008). Their early and omnipresent exposure to technology has defined their styles, their modes of communication, their learning preferences, their social choices, and their entertainment preferences (Saiedian, 2009). Numerous people analyse the main traits of different generations, but Hoover (2009) warns that it can be a strong form of stereotyping and that not all university students fit into one mould. For this reason, our study specifically investigated the range of students in the SD class.

A few studies have investigated the attraction to and retention of students in courses in computer science, and they have identified motivation, culture, pre-college experience, and confidence issues as contributing factors. Also, initial positive experience with computing, matching requirements of the discipline with perceived abilities, narrow perceptions on computing careers and career expectations were identified as key factors that influence students' decision, first, to pursue courses in computer science and, second, to study the field further (Klawe, 2001; Margolis & Fisher, 2002; Tillberg & Cohoon, 2005).

A closely related issue to the attraction and retention of students is the gender composition in computing classes, the male domination being a matter of great concern for SD educators (Margolis & Fisher, 2002; Tillberg & Cohoon, 2005; Blum & Frieze, 2005). Women have been found to enjoy using existing systems rather than developing new ones and they are attracted when they recognise computing as a form of communication, a means of creative self-expression, or as a path to a helping occupation. Moreover, women prefer a contextualised curriculum in which computing and technology in general are seen as tools for solving humanity's problems and enriching humanity's experiences (Tillberg & Cohoon, 2005; Shotick & Stephens, 2006). Men on the other hand have a greater technical appreciation of computers and they enjoy playing computer games (Carlson, 2006). Margolis and Fisher (2002) reveal that most men describe an early and persistent magnetic attraction between themselves and computers and the computer is the ultimate toy for them.

2.3 Software development education

Software and technical developments have been remarkable in the last few decades, and continue unabated (O'Grady, 2012). Not only is the dependence on software increasing, but the character of software production itself is changing – and with it the demands on software developers (Shaw *et al*., 2005; Shaw, 2000; Stankovic, 2009; Saiedian, 2009). This presents new challenges for the education of software developers (Shaw et al., 2005; Shaw, 2000).

Several studies suggest a gap between the knowledge and skills demanded by the industry and the knowledge and skills gained by graduates of university computing courses. The seminal study of Lethbridge (2000) in this field identified gaps between the education and training received and the knowledge required from the viewpoint of the software development industry. Lethbridge's survey dealt with professionals with industry experience and found gaps in: HCI/user interfaces, Real-time system design, Software cost estimation, Software metrics, Software reliability and Fault tolerance, and Requirements gathering and analysis. Kitchenham et al. (2005) ran a similar study as Lethbridge (2000), but with recent SE graduates and the results were quite different with gaps appearing to relate to Web-based programming, Project management, Configuration and release management, Multimedia, Security and cryptography, and Computer graphics. Both studies found that mathematical topics appear to be taught in more depth than required in the industry. Another similar study was performed by Surakka (2007) in the Finnish context, but he surveyed three role players namely software developers, professors and lecturers, and master students about the relevance of different matters. His results coincide with Lethbridge's (2000) and Kitchenham's (2005) results regarding the excessive importance attached to mathematics-related topics at university, and with Kitchenham's findings of the increased importance of Web-related subjects and skills in industry.

When Information Systems curricula were analysed from the perspective of the industry the following gaps in knowledge and skills were found: problem solving and project management skills, knowledge of business, IT business consultancy, security, end-user computing, soft skills related to core knowledge, knowledge related to leadership, and negotiation or giving presentations (Lee & Han, 2008; Kim et al., 2006; Moreno et al., 2012).

Education for software developers currently emphasises content inspired by closed-shop mainframe development. It is offered largely in traditional classroom formats. Software developers are now educated in much the same way as they have been for years. However, courses with a primary emphasis on current technology in which most of the knowledge will become obsolete as the technology does are a major challenge in the education of software developers. Pressures arising from the changing character of software and from external pressures on educational institutions will require changes in what we teach software developers and how we teach it (Shaw, 2000).

Lethbridge et al. (2007) argue that the majority of quality and budgetary issues with software have their root cause in human error or lack of skill. These in turn arise in large part from inadequate education. Therefore improving education should go a long way towards improving software and software practice.

Students need to see the relevance of teaching and learning, as it applies to them personally (their own lives, their career expectations, the wishes of their parents), or the relevance as it applies to society (wishes of the community, employers, the university) or as it applies to them professionally (the content / curriculum is meaningful) (Holbrook, 2003). The realities of the software industry for which the Net generation need to prepare have shifted away from those of the foundational beliefs and practices of many of their educators. The educators need to become familiar with the students' teaching and learning challenges and should investigate their distinctive qualities and personal preferences. Educators must identify the necessary ingredients for successful teaching and learning in order to improve teaching practices and course delivery methodologies (Saiedian, 2009).This study therefore investigates the relevance of software development education in the eyes of students.

3. RESEARCH METHOD

In this section, we discuss the demographics of the participants, the survey instrument and how the survey was conducted.

3.1 Research design and participants

This quantitative study was conducted at a university in South Africa and it aimed and was operationalised to investigate the relevance of SD education for students. Close to the end of the 2012 academic year 386 questionnaires were posted as an assignment on the e-learning system to students of the relevant SD classes. The number of usable responses received totaled 297, making for an overall response rate of 76.9%. The undergraduate students had a higher response rate (79.1%) than the graduates (56.8%).

Table 1. Profile of respondents (n=297)

		Number(%) of students
Gender	Male	222 (75%)
	Female	75 (25%)
Academic Year	1	145 (49%)
	2	76 (26%)
	3	55 (19%)
	4 (Hons)	21 (7%)
Self-rated academic performance	<= 59%	68 (25%)
	60% – 74%	158 (57%)
	>= 75%	50 (18%)

3.2 Data collection, instrument and analysis

A questionnaire with a pool of 57 items was developed by both writing new items and adapting items from available surveys, such as for instance ROSE (Schreiner & Sjøberg, 2004)

The first section of the questionnaire gathered information on the biographic data of the respondents as shown in Table 1. The questionnaire was further divided into four domains.

The first domain "Out of class" investigated personal relevance and had 12 items that gathered data on the students' out-of-class experiences such as using the internet and developing a software system. The participants were asked: "How often have you done this outside formal education?" with a five-point Likert response scale: Never / Once or twice / I don't know / Quite often / Very often.

The second domain "In class" investigated personal, social and professional relevance with 33 items and enquired on their perceptions of their SD classes, such as their enjoyment and interest in the classes.

The third domain "My career" had 12 items and investigated social relevance. It gathered data on their future career such as what is expected from a good software developer. The second and third domain were accompanied by a five-point Likert response scale from 1 (Strongly disagree) to 5 (Strongly agree).

Factor analysis was used to investigate the 57 items in more detail to reduce the variables into a smaller number of factors. Bartlett's sphericity test showed that the p-values were less than 0.001. This test result suggested that factor analysis was worth pursuing. The 297 responses were examined using principal components factor analysis as the extraction technique and the 57 attitude items yielded 12 interpretable factors. Factors were named according to their main context. A Cronbach's α coefficient was calculated for each of the 12 factors and were found as Table 2 shows, to be reliable (α ≥0.60).

Table 2. Reliability coefficients of factors*

Factor	Cronbach's alpha (α)
Basic computer use	.743
Advanced computer use	.605
In class_Learn	.876
In class_Perceptions	.745
In class_Attitudes	.853
In class_Importance	.724
In class_Teaching	.678
Career_Attitudes	.843
Career_Skills	.772
E-mail use	**
Internet use	**
Skype use	**

* See appendix A for the items in each factor ** Individual item

The 12 factors were further divided into three perspectives of Personal relevance, Professional relevance, and Social relevance using Holbrook (2003) as guideline (see 2.1). Table 4 shows the division of the factors. A Cronbach's α coefficient was also calculated for the 3 perspectives and were found as Table 3 shows, to be reliable (α ≥0.60).

Table 3. Reliability coefficients of perspectives

Perspective	Cronbach's alpha (α)
Personal relevance	0.652
Professional relevance	0.909
Social relevance	0.704

Basic analysis was done by calculating the mean values and standard deviation of each of the 12 factors, as well as those of the three relevance perspectives. The statistical tests used in our analysis varied as necessary to match the metric being analysed. Two groupings were identified based on gender and self-rated academic performance. The gender grouping was tested for significant differences between means in the different factors using a T-test and the academic performance grouping was tested with an ANOVA. Spearman's rank correlation analysis was also used to analyse relationships between the groupings and the factors. When the results of these interaction analyses are reported we will typically only discuss the significant interactions or primary effects. Unless noted otherwise, all statistical tests were performed with a significance level of $\alpha = 0.05$.

4. RESULTS AND DISCUSSION

In this section, we look at important data from each of the factors, as well as information that can be obtained by comparing answers to the different questions.

4.1 General results

Table 4 shows that the mean values of 10 of the 12 factors are relatively high. Advanced computer use is one of the factors showing a lower mean, which indicates that students don't have that much out-of-class experience with developing a software system for somebody, writing a computer program and building a device. Further analysis (see 4.2) indicated that gender played a significant role in the advanced computer use of students. The other factor showing a lower mean is Inclass_Perceptions indicating that students tend to have a perception that SD is a difficult subject area, that the volume of work is high and the instruction in the SD class is rigorous. They were anxious/stressed when doing practicals, they found SD hard to learn and they were not confident that they will obtain their degree. Further analysis (see 4.3) indicated that academic performance played a significant role in the Inclass_Perceptions of students.

It is not surprising that these IT students had high mean values for Basic computer use, E-mail use and Internet use. The use of Skype is lower, but the low access to computers (see Table 1) and the high cost and slow speed of the Internet in South-Africa (Muller, 2013) might explain that figure.

The students' views regarding attitudes and skills in a SD career were found to be relatively high.

Table 4. Basic analysis of 12 factors and division of relevance perspectives

Factor	Relevance perspective*	Mean**	Standard deviation
Basic computer use	Pers	4.2936	0.7059
E-mail use	Pers	4.5387	0.8811
Internet use	Pers	4.8615	0.4559
Skype use	Pers	3.3266	1.6330
Advanced computer use	Pers	2.6970	1.0680
In class_Perceptions	Pers	2.9439	0.8815
In class_Attitudes	Pers	3.7588	0.8102
In class_Importance	Soc	3.8894	0.6643
In class_Learn	Prof	4.0737	0.6923
In class_Teaching	Prof	3.4545	0.6479
Career_Attitudes	Soc	4.4876	0.5343
Career_Skills	Soc	4.0744	0.6719

* Pers-Personal; Prof-Professional; Soc-Social

** Likert style responses were ranked from 1 to 5 respectively

Table 5 shows SD education has of high social relevance to these students. SD education also has relatively high personal and professional relevance to them.

Table 5. Basic analysis of the 3 relevance perspectives

Relevance perspective	Mean	Std. Deviation
Personal_relevance	3.7894	.5568
Professional_relevance	3.6045	.7149
Social_relevance	4.1546	.5006

4.2 Gender

Gender differences were analysed with a T-Test and Table 6 shows significant differences in means between the male and female students in 3 of the 12 factors. There is a large practically significant difference between the advanced computer use of male and female students. The male students had significantly more out-of-class experience with developing a software system for somebody, writing a computer program and building a device.

There is a medium practically significant difference between the out-of-class basic computer use of male and female students. Again it was the male students who played computer games, opened devices to find out how they work, installed programs on a computer, read about computers in books or magazines, downloaded music from the internet and used a dictionary, encyclopedia, etc. on a computer significantly more than females. These findings concur with studies like Margolis and Fisher (2002) with men describing an early and persistent magnetic attraction between themselves and computers and the computer being the ultimate toy for them.

The other factor that shows a medium practically significant difference between male and female students is Inclass_Attitudes.The male students more than their female counterparts like the subject SD, would like to become software developers, their parents wish for them to become software developers, SD opened their eyes to new and exciting jobs, they enjoy working with computers and they find SD interesting.

Table 6. Gender differences in views on the relevance of SD education

Factors	Men (N=222)		Women (N=75)		Effect size	p
	Mean	SD	Mean	SD		
Basic computer use	4.4371	.5464	3.8689	.9256	0.61*	<0.001
Advanced computer use	2.9099	.9997	2.0667	1.0193	0.83**	<0.001
In class_Attitudes	3.9286	.6850	3.2516	.9390	0.72*	<0.001

* medium practically significant difference
** large practically significant difference

When the gender differences regarding the three relevance perspectives were analysed by using a T-Test, a medium practically significant difference ($d = 0.56$, $p<0.001$) between male and female students were found in terms of personal relevance. SD education has more personal relevance to the men because they use the computer more intensely and their Inclass_Attitudes show a significant difference.

4.3 Academic performance

Students were asked to rate their academic performance in their SD courses and they were divided in three groups as follows: <= 59% (n=68) ; 60% – 74% (n=158) ; >= 75% (n=62).

The results of an ANOVA in Table 7 indicates a practically significant difference between the <= 59% students and the >= 75% students in terms of their Inclass_Perceptions. The >= 75% students had a significantly more positive perception of the SD class. As can be expected, the <= 59% students had a perception that SD is a difficult subject area, that the volume of work is high and the instruction in the SD class is rigorous. They were anxious/stressed when doing practicals, they found SD hard to learn and they were not confident that they will obtain their degree.

Table 7. Differences between the three groupings based on self-rated academic performance

Factor		Effect size	p
In class_Perceptions	<= 59% *vs* 60% –74%	0.36	
	<= 59%*vs*>= 75%	1.12**	<0.001
	60% – 74%*vs*>= 75%	0.28	

** large practically significant difference

Furthermore, correlation techniques were used to analyse the correlation between the self-rated academic performance and all the factors. A Spearman rank correlation analysis was used to test the statistical significance of the associat

ion and found a medium practically significant relationship ($r = -.353$, $p<.001$) between the students' academic performance and their Inclass_Perceptions. The lower the students rated their academic performance the more negative perception they had of the SD class.

When the differences in self-rated academic performance with the three relevance perspectives as variables were analysed by using an ANOVA, no significant differences were found. Students' self-rated academic performance does not determine if SD education has more personal, social or professional relevance to them. Furthermore, when a Spearman rank correlation analysis was used to test the statistical significance of the association, no correlation between academic performance and the three relevance perspectives were found.

5. CONCLUSION AND RECOMMENDATIONS

SD education has social relevance for IT students, in other words they view it as useful to society. However, it has to a lesser extent personal and professional relevance for students. SD education does to a lesser extent relate to concerns in the students' immediate environment or area of interest and the content of the curriculum is less interesting and useful to them.

SD education has more personal relevance for male students than their female counterparts. They have significantly more out-of-class experience in the basic and advanced use of computers, they have a more positive attitude towards SD in class.

Students' self-rated academic performance does not influence their perspective on the personal, social or professional relevance of SD education. Students who rate their academic performance as high (>=75%) have a significantly more positive perception of the SD class than students that rate themselves as low (<=59%).

It can therefore be concluded that SD education has more social relevance for students, but personal relevance and professional relevance is relatively lacking. Certain groupings of students view SD education as more relevant than others. Male students, who rate their academic performance as high (>=75%) rate SD education as more relevant.

To improve the relevance of SD education in the eyes of students, effort should be made to improve personal and professional relevance for students. Furthermore, attention should be paid to female students and students who rate their academic performance as low. SD educators and SD curriculum developers should take cognisance of what makes SD education more relevant in the eyes of students to attract more students to SD classes – and retain them. This can hopefully result in meeting the demand for software developers.

REFERENCES

Aasheim, C., Li, L., & Williams, S. 2009. Knowledge and skill requirements for entry-level information technology workers: a comparison of industry and academia. *Journal of information systems education*, 20(3), 349-356.

Blum, L. & Frieze, C. 2005. In a more balanced computer science environment, similarity is the difference and computer science is the winner. *Computing research news*, 17(3), May.

CareerJunction Index. 2013. Executive Summary. Mar. http:// www.cji.co.za Date of access: 12 May. 2013.

Cheese, P. 2008. Netting the net generation. Businessweek.com. 13 Mar. http://www.businessweek.com/stories/2008-03-13/netting-the-net-generationbusinessweek-business-news-stock-market-and-financial-advice Date of access: 25 Jul. 2013.

Connolly, B. 2013. IT worker shortage continues as jobs remain unfilled. CIO. http://www.cio.com.au/article/454650/it_worker_shortage_continues_jobs_remain_unfilled/ Date of access: 5 Sep. 2013.

Gates, B. 2007. The 3r's solution. http://www.gatesfoundation.org/United-States/Education/ RelatedInfo/ 3Rs_Solution.htm Date of access: 14 Nov. 2011.

Harris, L. 2012. Mind the ICT skills gap. Brainstorm. http://www.brainstormmag.co.za/index.php?option=com_content&view=article&id=4699:mind-the-ict-skills-gap Date of access: 24 Jul. 2013.

Holbrook, J. 2003. Rethink science education. *Asia-Pacific forum on science learning and teaching*, 4(2):1-13.

Holbrook, J. 2009. Meeting challenges to sustainable development through science and technology education. *Science education international*. 20(1):44-59.

Hoover, E. 2009. The millennial muddle: How stereotyping students became a thriving industry and a bundle of contradictions. *The chronicle of higher education*. 11 Oct. http://chronicle.com/article/The-Millennial-Muddle-How/48772/ Date of access: 24 Jul. 2013.

Klawe, M. 2001. Refreshing the nerds. *Communications of the ACM*. 44(7):67-68.

Kim, Y., Hsu, J., & Stern, M. 2006. An update on the IS/IT skills gap. *Journal of information systems education*, 17(4):395.

Kitchenham, B., Budgen, D, Brereton, P. & Woodall, P. 2005. An investigation of software engineering curricula, *Journal of systems and software*, 74(3):325-335. http://dx.doi.org/10.1016/j.jss.2004.03.016.

Labaree, D. F. 2008. Comments on Bulterman-Bos: the dysfunctional pursuit of relevance in education research. *Educational researcher*, 37(7), 421-423.

Lee, C, & Han, H. 2008. Analysis of skills requirement for entry-level programmer/analysts in Fortune 500 corporations. *Journal of information systems education*, 19(1): 17.

Lethbridge, T.C. 2000. Priorities for the education and training of software engineers, *Journal of systems and software*, 53(1): 53-71.

Lethbridge, T., Diaz-Herrera, J., LeBlanc, R. & Thompson, J.B. 2007. Improving software practice through education: Challenges and future trends. In: 2007 Future of Software Engineering (FOSE '07). IEEE Computer Society, Washington, DC, USA, pp. 12-28. http://dx.doi.org/10.1109/FOSE.2007.13

Margolis, J., & Fisher, A. 2002. Unlocking the clubhouse: Women in computing. Cambridge, MA: MIT Press.

McAllister, N. 2012. Here's how to solve America's developer shortage. Infoworld. http://www.infoworld.com/d/application-development/heres-how-solve-americas-developer-shortage-185042 Date of access: 12 May 2013.

Merriam-Webster Dictionary. 2013. Relevance. www.merriam-webster.com/dictionary/relevance Date of access: 26 Aug. 2013.

Moreno, A., Sanchez-Segura, M., Medina-Dominguez, F. & Carvajal, L. 2012. Balancing software engineering education and industrial needs, *Journal of systems and software*, 85(7):1607-1620. http://dx.doi.org/10.1016/j.jss.2012.01.060.

Muller, R. 2013. SA's broadband speeds, prices benchmarked. Mybroadband. http://mybroadband.co.za/news/broadband/82313-sas-broadband-speeds-prices-benchmarked.html Date of access: 25 Jul. 2013

O'Grady, M. 2012. Practical problem-based learning in computing education. *ACM Transactions on computing education*. 12(3), Article 10, 16 pages. http://doi.acm.org/10.1145/2275597.2275599

Oxford English Dictionary. 2013. Relevance. http://oxforddictionaries.com/definition/english/relevant Date of access: 26 Aug 2013.

Reisman, S. 2005. Higher education: who cares what the customer wants? *IT Professional*, 7(6):62-64.

Saiedian, H. 2009. Software engineering challenges of the "Net" generation, *Journal of systems and software*, 82(4): 551-552, http://dx.doi.org/10.1016/j.jss.2009.01.020.

Schreiner, C. & Sjøberg, S. 2004. Sowing the seeds of ROSE. Background, rationale, questionnaire development and data collection for ROSE (The Relevance of Science Education)– a comparative study of students' views of science and science education. *Acta Didactica*, 4:1-120.

Shaw, M. 2000. Software engineering education: A roadmap. In: Proceedings of the conference on The future of software engineering (FOSE '00), ACM, pp. 371-380.

Shaw, M., Herbsleb, J., Ozkaya, I. & Root, D. 2005. Deciding what to design: Closing a gap in software engineering education. In: Inverardi, P & Jazayeri, M. (Eds.), Software engineering education in the modern age. ICSE 2005 Education Track, pp. 28 – 58.

Shotick, J. & Stephens, P.R. 2006. Gender inequities of self-efficacy on task-specific computer applications in business. *Journal of education for business*, 81(5):269–273.

Stankovic, N. 2009. Single development project. *Journal of systems and software*, 82(4):576-582.

Surakka, S. 2007. What subjects and skills are important for software developers? *Communications of the ACM*. 50:73–78.

Tillberg, H. & Cohoon, J. 2005. Attracting women to the CS major. *Frontiers: a journal of women studies*, 26(1):126-140.

APPENDIX A

Factor	Questionnaire items
Basic computer use	• played computer games • opened a device (radio, watch, computer, telephone, etc.) to find out how it works • installed a program on a computer • read about computers in books or magazines • downloaded music from the internet • used a dictionary, encyclopedia, etc. on a computer
E-mail use	• communicated with friends and family via e-mail
Internet use	• searched the internet for information
Skype use	• communicated with friends and family via skype
Advanced computer use	• developed a software system for somebody • written a computer program • built a device (robot, radio, computer, etc.)
In class_Learn	• In the software development modules we learn to collect and critically evaluate information • In the software development modules we learn to communicate effectively, both verbally and in writing • In the software development modules we learn to use science and technology effectively • In the software development modules students must be able to demonstrate an understanding of the world as a set of related systems by recognising that problem-solving contexts do not exist in isolation. • In the software development modules we learn to work with others as a member of a team or group • The software development modules require from students to organise and manage themselves and their activities effectively • In the software development modules we learn to identify and solve problems using critical and creative thinking • Some of the work in the course are carried out as projects
In class_Perceptions	• I am anxious/stressed when I do the practicals • Software development is a difficult subject area • The volume of work in the Software development modules is high • Software development is rather easy for me to learn • The instruction in the Software development class is rigorous • I'm confident that I will obtain my degree
In class_Attitudes	• I like Software development better than most other subjects • I would like to become a software developer • My parents wish for me to become a software developer • Software development has opened my eyes to new and exciting jobs • I enjoy working with computers • Software development is interesting
In class_Importance	• I learn something new every day in the Software development classes • I think that the Software development I learn will improve my career chances • Our country need more software developers • I think everybody should learn Software development • The things that I learn in Software development will be helpful in my everyday life • This course has high expectations for all students
In class_Teaching	• The same staff are involved in teaching and research • People from industry are brought into the classes • I know what the outcomes are for the Software development degree • I know what software developers do in the workplace now that I am busy with my studies. • I knew what software developers do in the workplace, before I started my studies • The instruction in the Software development class is relevant • Some of the lecturers have industrial experience
Career_Attitudes	• have a good attitude, including a willingness to listen and to take instructions • be prepared to work hard and to learn (a thirst for knowledge) • have a desire to make something of oneself through hard work and application, a desire to succeed (realistically ambitious) • good time-management skills • have a preparedness to take responsibility • have a reasonable level of general knowledge
Career_Skills	• a neat and tidy appearance • have modern leadership skills like self-confidence and a preparedness to lead by example • have the ability to relate well to and to build relationships with others (emotional intelligence) • have a good set of exam results • have respect for others • have at least some idea of what career direction one wish to take