The Relevance of Software Development Education for Students

Janet Liebenberg, Magda Huisman, and Elsa Mentz

***Abstract*—Despite a widely-acknowledged shortage of software developers, and reports of a gap between industry needs and software education, the possible gap between students' needs and software development education has not been explored in detail. In their university education, students want to take courses and carry out projects that clearly relate to their lives and their goals. This paper reports on a quantitative study of 297 software development students. The analysis of the results suggests that software development education has a predominantly social relevance to students and also has moderate personal and professional relevance. The following approaches are recommended to improve students' views of the relevance of software development education: use various learning environments; pay special attention to female students, students who did not have IT as a school subject, and students who rate their own academic performance as low; update educators on the latest developments; design programs to appeal to students and to meet societal demands.**

***Index Terms*—Computer industry, computer science education, curriculum development, engineering students.**

I. Introduction

SOUTH AFRICA has a shortage of software developers, which is a worldwide phenomenon. To remedy this, more software development students are needed, but will students take courses they do not find relevant?

In promoting education, software developer Bill Gates [1] focuses on his foundation's "3 R's": Rigor, Relevance, and Relationships. The central pillar of relevance highlights the need for students to be exposed to courses and projects that clearly relate to their lives and their goals.

Numerous studies have investigated the relevance of education at university level, but these have predominantly been concerned with ways to make education relevant to the needs of industry. Lethbridge [2] analyzed the relevance and depth of the knowledge that software professionals had gained 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 industry's requirements. Other researchers reported similar gaps [3]–[9]; for Lethbridge [10], filling them is one of the most critical challenges for educators. In his paper "Higher Education: Who Cares What the Customer Wants?," Reisman [11] argues that studies in higher education often neglect to collect data from two of higher education's most important players—faculty and students. No study appears to have investigated if there is a gap between students' needs and their education, or whether software development (SD) education is relevant from the perspective of the students. In the light of the shortage of software developers, this study aimed to investigate the relevance of SD education to students.

A. Clarification of Terminology

For the purpose of this study, it is necessary to explain and clarify certain key terms.

Software development (SD)—The IEEE, ISO, and IEC define the software development process as the process by which user needs are translated into a software product [12]. For the purpose of this study, SD will refer to the orderly process of developing software products through successive phases.

SD classes will refer to the courses where the process of developing software is taught.

SD students will refer to the students taking the SD classes.

II. Conceptual Framework

A. Concept of Relevance

Relevance is broadly defined as being closely connected or appropriate to the matter in hand, or having practical and especially social applicability. Relevance in the educational context can be defined as the applicability of what is taught to the needs and interests of students and society [13].

Three relevance perspectives suggested by Holbrook [14] are used in this paper to analyze relevance:

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

B. Student in the Software Development Class

Most students in current university classes belong to the so-called Net generation, also known as Generation Y. The Net generation is characterized by students who may have never known life without the Internet. Their early and omnipresent exposure to technology has defined their styles, their modes of

Manuscript received July 10, 2014; revised September 15, 2014 and November 13, 2014; accepted December 04, 2014. Date of publication January 13, 2015; date of current version October 28, 2015. The work of J. Liebenberg was supported by the National Research Foundation (NRF).

J. Liebenberg and M. Huisman are with the Department of Computer Science and Information Systems, North-West University, Potchefstroom 2520, South Africa (e-mail: janet.liebenberg@nwu.ac.za; magda.huisman@nwu.ac.za).

E. Mentz is with the Faculty of Education Sciences, North-West University, Potchefstroom 2520, South Africa (e-mail: elsa.mentz@nwu.ac.za).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TE.2014.2381599

communication, their learning preferences, their social choices, and their entertainment preferences [15].

A few studies have investigated the attraction of students to, and their retention in, courses in computing disciplines; they identified motivation, culture, pre-college experience, and confidence issues as contributing factors. Also, initial positive experiences with computing, matching requirements of the discipline with perceived abilities, narrow perceptions of computing careers, and career expectations were identified as key factors that influence students' decisions to, first, pursue courses in the computing fields and, second, to study the field further [16], [17].

An issue closely related to the attraction and retention of students is the gender composition in computing classes; that these are predominantly male is a matter of great concern for educators [17]–[19]. Women have been found to enjoy using existing systems more than developing new ones, and are attracted when they recognize computing as a form of communication, a means of creative self-expression, or as a path to an occupation where they may be of assistance. Moreover, women prefer a contextualized curriculum in which computing and technology in general are perceived as tools for solving humanity's problems and enriching humanity's experiences [19], [20]. Many of the men, however, have a greater technical appreciation of computers and enjoy playing computer games [16]–[18]. Margolis and Fisher [18] reveal that most men describe an early and persistent magnetic attraction between themselves and computers—for them the computer is the ultimate toy.

C. *Software Development Education*

In the last few decades, remarkable strides have been made in software and technical development; these developments continue unabated. 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 [15], [21], [22]. The diversity of software applications requires adaptability in responding to client needs, and the diversity of clients and contexts requires the ability to discriminate among criteria for success. This presents new challenges for the education of software developers [21], [22].

Several studies suggest a gap between the knowledge and skills demanded by industry and the knowledge and skills gained by graduates of university computing courses. An analysis of computing curricula from an industrial perspective, in terms of technical and nontechnical knowledge and skills, revealed the following gaps: problem-solving and project management skills, knowledge of business, information technology (IT) business consultancy, security, end-user computing, soft skills related to core knowledge, knowledge related to leadership, and negotiation or giving presentations [3], [6], [8].

Lethbridge's [2] study identified gaps in technical knowledge between the education received and the knowledge required from the viewpoint of the SD industry. Lethbridge surveyed professionals with industry experience and found gaps in the following: HCI/user interfaces, real-time system design, software cost estimation, software metrics, software reliability and fault tolerance, and requirements gathering and analysis. Kitchenham [5] ran a similar study, but with software engineering (SE) graduates; 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 greater depth than required in industry. Another similar study was performed in the Finnish context by Surakka [7], who surveyed three sets of stakeholders (software developers, professors and lecturers, and Master's students) on the relevance of various matters. His results coincide with Lethbridge's [2] and Kitchenham's [5] on 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.

Computing departments often struggle to retain students because computing classes often start with programming. This can deter students who find programming difficult or uninteresting, but who would be very interested in working in areas such as information management, project management, or system management [23].

Education for software developers is offered largely in traditional classroom formats. Software developers are still educated much as they have been for years past. However, a major challenge in educating software developers is that in courses whose primary emphasis is current technology, as that technology becomes obsolete, so will most of the knowledge taught. The changing character of software and external pressures on educational institutions will require changes in what, and how, software developers are taught [22].

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 as it applies to society (wishes of the community, employers, the university), or as it applies to them professionally (the content/curriculum is meaningful and interesting) [14]. The realities of the software industry for which the students need to prepare have shifted away from those of the foundational beliefs and practices of many of their educators. Educators need to become familiar with the students' learning challenges and investigate their distinctive qualities and personal preferences. Educators must identify the necessary ingredients for successful teaching and learning to improve teaching practices and course delivery methodologies [15]. This study therefore investigates the relevance of software development education from the point of view of students.

III. Research Method

A. *Research Design and Participants*

In this quantitative study, conducted at a university in South Africa, 386 questionnaires were posted as an assignment on the university's e-learning system to students in the SD classes of four academic year groups. A total of 297 usable responses were received, an overall response rate of 76.9%. The participants included 276 B.Sc. undergraduate students and 21 students enrolled for the subsequent B.Sc. (Hons.) in computer science (CS) and information systems (IS). The undergraduate students had a higher response rate (79.1%) than the graduates (56.8%). Most of the undergraduate students follow the 3-year B.Sc. in IT and CS program in the Department of Computer Science and Information Systems. A subsequent 1-year B.Sc. Honor's degree in CS and IS is offered, which gives access to a

TABLE I
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%)
IT[1] as school subject	Yes	142 (48%)
	No	155 (52%)
Access to a computer since grade 1	Yes	170 (58%)
	No	123 (42%)

[1]*Information Technology (IT) is a subject that can be taken from Grade 10 to Grade 12 in South African schools, focusing primarily on programming skills.*

Master's degree in CS.[1] These programs are based on the Informatics Curriculum Framework of the International Federation for Information Processing (IFIP), so the results of this study could therefore also be of value to other university courses based on the IFIP framework. The IFIP framework refers to major widely accepted and widely implemented informatics curricula by professional organizations, such as the ACM, IEEE, AITP, and AIS.

Table I provides a summary of the biographic data.

B. Data Collection and Instrument

An initial list of questions was developed by both writing new items and adapting items from available surveys, such as ROSE [24], the SAQA's list of "Critical Cross-field Outcomes," and topics commonly found in computer science programs. Once the initial questions were generated, the instrument was sent to be refined by three industrial and two academic experts familiar with tertiary computing programs, who added additional topics. The experts' feedback served as the basis for correcting, refining, and enhancing the questions, resulting in a questionnaire with a pool of 123 items. The first section of the questionnaire gathered information on the biographic data of the respondents as shown in Table I. The remainder of the questionnaire was divided into four domains.

The first domain, "Out of class," investigated personal relevance, with 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, enquiring about their perceptions of their SD classes, such as their enjoyment and interest in the classes.

The third domain, "My career," had 12 items that investigated social relevance, gathering data on their notions of a future career, such as what they thought would be expected from a good software developer. The second and third domains used a five-point Likert response scale from 1 (Strongly disagree) to 5 (Strongly agree).

The last domain, "What I want," had 66 items to investigate professional relevance, testing their interest in curriculum topics such as specific programming languages or extreme programming. The participants were asked: "How interested are you in learning about the following?" on a five-point Likert response scale: Not interested/Slightly interested/Neutral/Quite interested/Very interested.

Factor analysis was applied to the 123 items to reduce the variables to a smaller number of factors. The 297 responses were examined using principal components factor analysis; the 123 attitude items yielded 20 interpretable factors, named according to their main context, and three single item variables. A Cronbach's α coefficient was calculated for each of the 20 factors and was found (see Table II) to be reliable $(\alpha \geq 0.60)$.

The 23 factors were further divided into three perspectives of Personal relevance, Professional relevance, and Social relevance using Holbrook [14] as a guideline. A Cronbach's α coefficient was also calculated for the three perspectives and was found (see Table III) to be reliable $(\alpha \geq 0.60)$. Table IV shows the division of the factors.

TABLE II
RELIABILITY COEFFICIENTS OF FACTORS

Factor	Cronbach's alpha (α)
Out_of_class_Basic_use	0.743
E-mail use	N/A
Internet use	N/A
Skype use	N/A
Out_of_class_Advanced_use	0.605
In_class_Learn	0.876
In_class_Perceptions	0.745
In_class_Attitudes	0.853
In_class_Importance	0.724
In_class_Teaching	0.678
Career_Attitudes	0.843
Career_Skills	0.772
Real-time and systems programming	0.935
Mathematics and statistics	0.902
Software management	0.918
General software design	0.907
Web and Mobile technologies and Games	0.934
Computer science theory	0.934
Specialized application techniques	0.926
Software engineering methods	0.916
Essential subsystem design	0.897
Computer hardware and other electrical and computer engineering	0.946
Alternative software engineering methods	0.880

**See http://tinyurl.com/SoftDevEd for the items in each factor*

TABLE III
RELIABILITY COEFFICIENTS OF PERSPECTIVES

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

C. Threats to Validity

The fact that all the survey respondents were from a single university can raise the question of the applicability of the

[1]See extracts of the programs from the yearbook of the Faculty of Natural Sciences at http://tinyurl.com/BScITandCS

TABLE IV
BASIC ANALYSIS OF 23 FACTORS AND DIVISION OF RELEVANCE PERSPECTIVES

Factor	Relevance perspective	Mean*	Standard deviation
Internet use	Personal	4.862	0.456
E-mail use	Personal	4.539	0.881
Career_Attitudes	Social	4.488	0.534
Out_of_class_Basic_use	Personal	4.294	0.706
In_class_Learn	Professional	4.074	0.692
Career_Skills	Social	4.074	0.672
In_class_Importance	Social	3.889	0.664
Web and Mobile technologies and Games	Professional	3.835	0.996
In_class_Attitudes	Personal	3.759	0.810
General software design	Professional	3.759	1.020
Specialized application techniques	Professional	3.757	0.969
Real-time and systems programming	Professional	3.720	1.075
Essential subsystem design	Professional	3.696	1.069
Software engineering methods	Professional	3.527	0.989
Computer hardware and other electrical and computer engineering	Professional	3.517	1.049
Software management	Professional	3.456	1.078
In_class_Teaching	Professional	3.455	0.648
Mathematics and statistics	Professional	3.363	1.125
Alternative software engineering methods	Professional	3.338	1.179
Skype use	Personal	3.327	1.633
Computer science theory	Professional	3.199	1.215
In_class_Perceptions	Personal	2.944	0.882
Out_of_class_Advanced_use	Personal	2.697	1.068

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

study. However, the university's curricula are based on the IFIP framework that is linked to major, widely accepted, and widely implemented curricular efforts and to high-quality bodies of knowledge undertaken by professional organizations, such as the ACM, IEEE, AITP, and AIS.

The participants were asked to rate their academic performance on a percentage scale, which could be viewed as a threat to validity, but according to Ackerman [25], individuals generally have accurate views of their relative standing on abilities and knowledge. Furthermore, the participants' actual academic performance could not be compared sensibly—the first-year students had only completed three SD courses, whereas the Honor's students had completed up to 21 SD courses by the time they completed the survey.

An instrument that measures the relevance of SD education for students could not be found. The researchers adjusted surveys for measuring relevance to the industry and surveys measuring relevance to school learners and sent their finding to industrial and academic experts familiar with tertiary computing programs to make provision for validity. However, additional validation is recommended. Future work can include a study to design a standardized instrument to specifically address the relevance of software development education for students.

D. Data Analysis

Basic analysis was done by calculating the mean values and standard deviation of each of the 23 factors, as well as those of the three relevance perspectives. The statistical tests used in the analysis varied as necessary to match the metric

TABLE V
BASIC ANALYSIS OF THE 3 RELEVANCE PERSPECTIVES

Relevance perspective	Mean	Std. Deviation
Personal_relevance	3.789	0.557
Professional_relevance	3.605	0.715
Social_relevance	4.155	0.501

being analyzed. Three groupings were identified based on gender, academic performance, and IT as school subject. All the groupings were tested for significant differences between means in the different factors using T-tests, except academic performance where an ANOVA test was used. Chi-square tests and Spearman's rank correlation analysis were also used to analyze relationships between the groupings and the factors. When the results of these interaction analyses are reported, only the significant interactions or primary effects will be discussed. Since a convenience sample was used instead of a random sample, the p-values will be reported for the sake of completeness but will not be interpreted.

IV. RESULTS AND DISCUSSION

A. General Results

Table IV shows that the mean values of 21 of the 23 factors are relatively high. Out_of_class_Advanced_use is the factor showing the lowest mean, which indicates that students have relatively little out-of-class experience with developing a software system for somebody, writing a computer program, and building a device. Further analysis indicated that gender played a significant role in the Out_of_class_Advanced_use of computers by students. The other factor showing a lower mean is In_class_Perceptions, indicating that students tend to have a perception that SD is a difficult subject area, that the volume of work is high, and that the instruction in an SD class is rigorous. In addition, some students were anxious/stressed when doing practicals; they found SD difficult to learn and were not confident that they would obtain their degree. Further analysis indicated that academic performance played a significant role in the In_class_Perceptions of students.

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

The factor In_class_Learn had a high mean with some of the items contributing to the high mean being the importance of "Project work," "Teamwork," and "Communication skills." The students therefore view the so-called "soft skills" as relevant in their education.

Students showed the most interest in the subject topics in the factor Web and Mobile technologies and Games. This is a significant finding in view of the findings of Kitchenham [5] and Surakka [7] of the increased importance of Web-related subjects and skills in industry, but even more significant is the interest of students in Mobile technologies and Games since these topics have not emerged in studies in industry.

Table V shows SD education has high social relevance to these students and also has moderately high personal and professional relevance.

TABLE VI
GENDER DIFFERENCES IN VIEWS ON THE RELEVANCE OF SD EDUCATION

Men (N=222) Mean	Men (N=222) SD	Women (N=75) Mean	Women (N=75) SD	Effect size	P
Out_of_class_Basic_use					
4.44	0.55	3.87	0.93	0.61*	<0.001
Out_of_class_Advanced_use					
2.91	0.99	2.07	1.02	0.83**	<0.001
In_class _Attitudes					
3.93	0.69	3.25	0.94	0.72*	<0.001
Real-time and systems programming					
3.91	0.93	3.14	1.26	0.61*	<0.001
Computer hardware and other electrical and computer engineering					
3.71	0.91	2.93	1.21	0.64*	<0.001
Alternative software engineering methods					
3.52	1.05	2.79	1.38	0.52*	<0.001

* *medium practically significant difference (d>=0.5)*
** *large practically significant difference (d>=0.8)*

TABLE VII
DIFFERENCES BASED ON SELF-RATED ACADEMIC PERFORMANCE

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

** *large practically significant difference (d>=0.8)*

B. Gender

Gender differences were analyzed with a T-test. Table VI shows significant differences in means between the male and female students in six of the 23 factors. There is a large practically significant difference in Out_of_class_Advanced_use and a medium practically significant difference in Out_of_class_Basic_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. In addition, it is the male students who play computer games, open devices to find out how they work, install programs on computers, read about computers in books or magazines, and download music from the Internet, significantly more than the female students. The other factor that shows a medium practically significant difference between male and female students is In_class_Attitudes. More than their female counterparts, the male students are interested in, like, and enjoy the SD courses; they and their parents want them to become software developers; and SD opened their eyes to new and exciting jobs. These differences in the genders' interests confirm the findings of numerous gender studies in computing.

With three factors showing a medium practically significant difference between male and female students' interest toward certain subject topics (Real-time and systems programming; Computer hardware and other electrical and computer engineering; Alternative software engineering methods), previous studies confirmed that male students exhibit a greater technical appreciation, and that female students prefer existing systems rather than alternative methods, such as extreme or real-time programming. Male students showed the least interest in Computer Science theory and the most interest in Web and Mobile technologies and Games topics. The female students were most interested in Mathematics and Statistics, which is in sharp contrast to the results of [3], [5], and [7] regarding the industry's view of the excessive importance attached to mathematics-related topics at university. The female students' interest in Mathematics might again be explained by their preference for existing systems since it is a familiar subject to them.

When the gender differences for the three relevance perspectives were analyzed by using a T-Test, a medium practically significant difference $(\mathrm{d} = 0.56, \mathrm{p} < 0.001)$ between male and female students was found in terms of personal relevance. The fact that SD education has more personal relevance to the male students can be explained by their more intense use of the computer and their more positive attitude toward their classes.

C. Academic Performance

The questionnaire listed the applicable SD courses in the program; students were asked to rate their overall academic performance in those courses on the scales: 75%–100% = between 8 and 10; 70%–74% = 7; 60%–69% = 6; 50%–59% = 5; Below 50% = 4. The results were then reported in three groups: $< = 59\% (\mathrm{n} = 68)$; 60%–74% $(\mathrm{n} = 158)$; $> = 75\% (\mathrm{n} = 62)$.

The results of an ANOVA in Table VII indicate a practically significant difference between the $< = 59\%$ students and the $> = 75\%$ students in terms of their In_class_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 difficult, the volume of work is high, and the instruction in the SD class is rigorous. They were anxious/stressed when doing practicals, found SD difficult to learn, and were not confident that they would obtain their degree.

A Spearman rank correlation analysis was used to analyze the correlation between the self-rated academic performance and all the factors and confirmed the results in Table VII with a medium practically significant relationship $(\mathrm{r} = -0.353, \mathrm{p} < 0.001)$ between the students' academic performance and their In_class_Perceptions. The lower the students rated their academic performance, the more negative the perception they had of the SD class.

When the differences in self-rated academic performance with the three relevance perspectives as variables were analyzed 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.

D. IT as a School Subject

IT refers to one of the subjects that can be chosen from grades 10 to 12 in some South African high schools. The school subject IT focuses primarily on programming, which means that some of the students enter the SD classes with 3 years of experience in programming. The school subject IT is, however, not an admission requirement for the B.Sc. degree in IT and CS program.

A T-test was used to determine if there were significant differences between the 142 students who had IT as a school subject and the 155 students who were only formally introduced to SD once they came to university. Table VIII shows that there is a high practically significant difference in their Out_of_class_Advanced_use, with the students who had IT showing significantly

TABLE VIII
DIFFERENCE IN VIEWS BETWEEN STUDENTS WHO TOOK IT AS SCHOOL SUBJECT AND THOSE WHO DID NOT

Factors	Effect size	P
Out_of_class_Advanced_use	0.95**	<0.001
In_class_Perceptions	0.54*	<0.001
In_class_Attitudes	0.62*	<0.001

* medium practically significant difference (d>=0.5)
** large practically significant difference (d>=0.8)

more SD interest and experience outside their formal education. There was a medium practically significant difference between the two groups of students in terms of their In_class_Perceptions and their In_class_Attitudes. Confidence and initial positive experience have been identified as contributing factors in the attraction and retention of students in computing courses [16], [17]. The students without IT as a school subject were less confident that they would obtain their degree. Since not all students can take IT at school, this result might be a compliment to the schools offering IT because they contribute to students' initial positive experience, interest, and confidence, and therefore the relevance of SD education for students.

A Crosstab was used to determine if there was an association between self-rated academic performance and students who had taken IT; contrary to what might have been expected, the result indicated no statistically significant association. In other words, having taken IT as a school subject did not influence a student's academic performance in SD at university level. The above findings therefore suggest that although having taken IT as school subject does not impact students' academic performance, the students who did not take IT still seem to lack confidence, and their perceptions of and attitudes toward their SD classes are less positive.

When a T-test was done to analyze the differences between the students who took IT as a school subject and those who did not, with the three relevance perspectives as variables, a medium practically significant difference $(d = 0.70, p < 0.001)$ in personal relevance was found between the two groups. SD education has more personal relevance to the group who had taken IT as a school subject, and it is mostly explained by their advanced use of the computer, their In_class_Perceptions, as well as In_class_Attitudes.

V. CONCLUSION AND RECOMMENDATIONS

Labaree [26], who works in educational research, emphasizes that, when considering relevance, the question "useful to whom and for what?" needs to be answered because there is a wide array of stakeholders. In this study, the focus was on the largest stakeholder group in SD education, the SD students, including the females, poor performers, and students who did not have IT as school subject.

SD education has high social relevance to students; in other words, they view it as useful to society. Students acknowledge the fact that more software developers are needed and what they learn in the SD classes could lead to improved career opportunities for them.

SD students find moderately high personal relevance in their education, therefore their education does relate to concerns in their immediate environment or area of interest.

SD education also has moderately high professional relevance to students, meaning that the content of the curriculum is relatively interesting and useful to them. Students are less interested in topics such as computer science theory and more interested in Web and mobile technologies and games.

Certain students view SD education as more relevant than others. Male students, students who had IT as a school subject, and students who rate their academic performance as high $(>= 75\%)$ rate SD education as more relevant.

Effort should be made to further improve the relevance of SD education for all students. The following is recommended for SD education.

Use Different Learning Environments: For computing departments to offer relevant SD education, attention should not only be paid to what students are taught, but also to how they are taught.

— *Game-based learning.* Since students show such a great interest in Game development, computing departments at universities could explore game-based learning environments to satisfy students' appetite for games.
— *Project-based learning.* The students in this study view project work as important, and therefore project-based learning can also be used.
— *Pair learning.* Since students view teamwork and communication skills as important, pair learning (the use of pair programming in education) can therefore be exploited. Pair learning has been found to boost self-confidence and can therefore aid in teaching students who did not have IT as a school subject [27]

Pay Special Attention to Certain Groups of Students:

— *Female students*. Since most female students prefer to develop software that solves humanity's problems and enriches humanity's experiences, the SD curriculum and especially practical assignments and projects must be contextualized to appeal to the female students so that they can find more personal relevance in SD education. In addition, pair learning has been found to benefit specifically female students [28].
— *Students who did not have IT as school subject*. Lecturers must design and offer a bridging course early in students' first year to provide students who did not have IT with an initial positive experience of SD. Furthermore, students often learn more from each other than from lecturers, and the students who had IT as a school subject could be appointed as facilitators in the practical labs and assist the less experienced, as well as the students performing poorly.
— *Students who rate their academic performance as low.* University courses will always have poor performers, but in the light of the shortage of software developers, educators should attempt to retain these students by using different learning environments, such as pair learning, which has been found to improve academic performance [29].

Put in Place Mechanisms to Ensure That Educators Stay Abreast of the Latest Developments in the SD Field: These mechanisms can include visits by lecturers to the software industry, guest lectures by industry experts, and joint university–industry projects.

Design Programs to Appeal to Students and to Meet Societal Demands: There is a great demand for software developers and students need to be informed from their early study years about the career possibilities and needs of the software industry. Curricular design must align the needs of students with the needs of industry. An emphasis on topics that do not interest students, such as computer science theory, can lead to a lower retention of students, but not emphasizing that topic could result in students being unprepared for the demands of the software industry. When students are aware of the specific needs of industry from their early study years, they might perceive the industry in a more positive light.

SD educators and curriculum developers should take cognizance of what makes SD education more relevant from the perspective of students in order to attract more students to SD classes—and to retain them. Enhancing the relevance of SD education is a profound and extremely complex pedagogical challenge that requires new education skills, methods, and approaches if the demand for software developers is to be met.

The paper focused on the needs of SD students; future research could include the needs of the SD industry to study the compatibility between student needs and industry needs.

Acknowledgment

Opinions expressed and conclusions arrived at are those of the author and are not necessarily to be attributed to the National Research Foundation (NRF).

References

[1] B. Gates, "The 3 R's solution," 2005 [Online]. Available: http://www.gatesfoundation.org/Media-Center/Speeches/2005/02/Bill-Gates-2005-National-Education-Summit

[2] T. C. Lethbridge, "Priorities for the education and training of software engineers," *J. Syst. Softw.*, vol. 53, no. 1, pp. 53–71, 2000.

[3] A. Moreno, M. Sanchez-Segura, F. Medina-Dominguez, and L. Carvajal, "Balancing software engineering education and industrial needs," *J. Syst. Softw.*, vol. 85, no. 7, pp. 1607–1620, 2012.

[4] S. Lee, S. Koh, D. Yen, and H. Tang, "Perception gaps between IS academics and IS practitioners: An exploratory study," *Inf. Manage.*, vol. 40, no. 1, pp. 51–61, 2002.

[5] B. Kitchenham, D. Budgen, P. Brereton, and P. Woodall, "An investigation of software engineering curricula," *J. Syst. Softw.*, vol. 74, no. 3, pp. 325–335, 2005.

[6] Y. Kim, J. Hsu, and M. Stern, "An update on the IS/IT skills gap," *J. Inf. Syst. Educ.*, vol. 17, no. 4, p. 395, 2006.

[7] S. Surakka, "What subjects and skills are important for software developers?," *Commun. ACM.*, vol. 50, pp. 73–78, 2007.

[8] C. Lee and H. Han, "Analysis of skills requirement for entry-level programmer/analysts in Fortune 500 corporations," *J. Inf. Syst. Educ.*, vol. 19, no. 1, pp. 17–27, 2008.

[9] A. Mohan, D. Merle, C. Jackson, J. Lannin, and S. Nair, "Professional skills in the engineering curriculum," *IEEE Trans. Educ.*, vol. 53, no. 4, pp. 562–571, Nov. 2010.

[10] T. C. Lethbridge, J. Diaz-Herrera, R. LeBlanc, and J. B. Thompson, "Improving software practice through education: Challenges and future trends," in *Proc. IEEE FOSE*, Washington, DC, 2007, pp. 8–28.

[11] S. Reisman, "Higher education: who cares what the customer wants?," *IT Professional*, vol. 7, no. 6, pp. 62–64, Nov.–Dec. 2005.

[12] ISO/IEC/IEEE, Geneva, Switzerland, "ISO/IEC/IEEE 24765, 2010, 3.2758: Systems and software engineering—Vocabulary,".

[13] J. Holbrook, "Meeting challenges to sustainable development through science and technology education," *Sci. Educ. Int.*, vol. 20, no. 1, pp. 44–59, 2009.

[14] J. Holbrook, "Rethink science education," *Asia-Pacific Forum Sci. Learning Teaching*, vol. 4, no. 2, pp. 1–13, 2003.

[15] H. Saiedian, "Software engineering challenges of the "Net" generation," *J. Syst. Softw.*, vol. 82, no. 4, pp. 551–552, 2009.

[16] M. Klawe, "Refreshing the nerds," *Commun. ACM*, vol. 44, no. 7, pp. 67–68, 2001.

[17] L. Blum and C. Frieze, "In a more balanced computer science environment, similarity is the difference and computer science is the winner," *Comput. Res. News*, vol. 17, no. 3, p. 2, May 2005.

[18] J. Margolis and A. Fisher, *Unlocking the Clubhouse: Women in Computing*. Cambridge, MA, USA: MIT Press, 2002.

[19] H. Tillberg and J. Cohoon, "Attracting women to the CS major," *Frontiers, J. Women Studies*, vol. 26, no. 1, pp. 126–140, 2005.

[20] J. Shotick and P. R. Stephens, "Gender inequities of self-efficacy on task-specific computer applications in business," *J. Educ. Bus.*, vol. 81, no. 5, pp. 269–273, 2006.

[21] M. Shaw, J. Herbsleb, I. Ozkaya, and D. Root, "Deciding what to design: Closing a gap in software engineering education," in *Proc. ICSE*, 2005, pp. 28–58.

[22] M. Shaw, "Software engineering education: A roadmap," in *Proc. FOSE*, 2000, pp. 371–380.

[23] M. Guzdial *et al.*, "Future of Computing Education Summit," Arlington, VA, USA, Jun. 2009 [Online]. Available: http://www.acm.org/education/future-of-computing-education-summit/FoCES_web.pdf

[24] C. Schreiner and S. Sjøberg, "Sowing the seeds of ROSE. Background, rationale, questionnaire development and data collection for ROSE (The Relevance of Science Education)," *Acta Didactica*, vol. 4, pp. 1–80, 2004.

[25] P. L. Ackerman, M. E. Beier, and K. R. Bowen, "What we really know about our abilities and our knowledge," *Pers. Indiv. Differ.*, vol. 33, no. 4, pp. 587–605, Sep. 2002.

[26] D. Labaree, "Comments on Bulterman-Bos: The dysfunctional pursuit of relevance in education research," *Educ. Res.*, vol. 37, no. 7, pp. 421–423, 2008.

[27] C. Bishop-Clark, J. Courte, and E. Howard, "Programming in pairs with Alice to improve confidence, enjoyment, and achievement," *J. Educ. Comput. Res.*, vol. 34, no. 2, pp. 213–228, 2006.

[28] L. Werner, B. Hanks, and C. McDowell, "Pair programming helps female computer science students," *J. Educ. Resour. Comput.*, vol. 4, no. 1, Mar. 2004, art 4.

[29] C. McDowell, L. Werner, H. E. Bullock, and J. Fernald, "The impact of pair programming on student performance, perception and persistence," in *Proc. ICSE*, Portland, OR, USA, 2003, pp. 602–607.

Janet Liebenberg received the Master's degree in computer science education from the North-West University, Potchefstroom, South Africa, in 2010, and is currently pursuing the Ph.D. degree in computer science education at the North-West University.

In 2010, she became a Lecturer with the Computer Science (CS) and Information Systems (IS) Department, North-West University. Her research interests include the teaching and learning of programming and minorities in CS.

Magda Huisman received the Ph.D. degree in computer science (CS) and information systems (IS) from the Potchefstroom University for Christian Higher Education, Potchefstroom, South Africa, in 2001.

She is a Professor with the Department of CS and IS, North-West University, Potchefstroom, South Africa. She has published in journals such as *Management Information Systems Quarterly*, *Information & Management*, and the *International Journal of Web & Semantic Technology*. Her research interests include the use and effectiveness of systems development methodologies and the diffusion of information technologies.

Elsa Mentz received the Ph.D. degree in computer science education from the Potchefstroom University for Christian Higher Education, Potchefstroom, South Africa, in 2001.

She is a Professor with the Faculty of Education Sciences, North-West University, Potchefstroom, South Africa. Her research mainly focuses on cooperative learning in computer science (CS) education as well as the fostering of self-directed learning skills in the CS classroom.

Prof. Mentz is a member of AERA and ACM.