Industry's perception of the relevance of software development education

J LIEBENBERG, [1]*M HUISMAN*[2] *AND E MENTZ*[3]

Abstract

It is widely acknowledged that there is a shortage of software developers, not only in South Africa, but also worldwide. Despite reports, in mostly quantitative studies, of a gap between the industry needs and software development education, the view the industry has of the new graduates and the problems, challenges and solutions in respect of software development education has not been explored in detail. This article reports on a mixed methods study of the relevance of software development education from the perspective of the industry. The analysis reveals some interesting views held by the industry, as well as by the different generations on their new recruits and the problems and challenges that are faced. The following solutions to the problems are suggested: teamwork; projects and experience; work-integrated-learning and mentoring; technical and soft skills; keeping up to date; career guidance; introspection; and generational awareness.

Keywords software development education, industry, computing curricula

1. Introduction

South Africa has a shortage of professionals with ICT skills (Harris, 2012) and this is a worldwide phenomenon (McAllister, 2012). The Career Junction Index (CJI), which monitors online employment trends in South Africa, found that the IT industry has the highest vacancy levels and that software development (SD) is at the forefront of the country's significantly high IT demand (CareerJunction Index, 2014). Not only is there a shortage of SD workers, but students from computing disciplines are also graduating with a lack of the skills that companies are searching for (Bateman, 2013). Regarding ICT skills in South Africa: "not all graduates are prepared for the working environment, and don't always fit in. This is a gap that needs addressing" (Mawson, 2010:1). The 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) but on the other hand, the mission and role of the university should also be considered. According to Reisman (2004), the mission of undergraduate and graduate programmes is educational, but that mission has steadily transformed into one of training. Reisman (2004) argues that the primary qualification of new employees should be their solid undergraduate education in which the curriculum focused on a common body of knowledge; a set of basic topics and principles. Employers can therefore build on that solid foundation and new employees can easily be trained by their

[1] . Corresponding author: Janet Liebenberg, Faculty of Natural Sciences, North-West University, Private Bag X6000, Potchefstroom, 2520, South Africa; *Tel*: +27-18-2992536, janet.liebenberg@nwu.ac.za

[2] . Prof Magda Huisman, Faculty of Natural Sciences, North-West University, Private Bag X6000, Potchefstroom, 2520, South Africa, Tel: +27-18-2992537, magda.huisman@nwu.ac.za

[3] . Prof Elsa Mentz, Faculty of Education Sciences, North-West University, Private Bag X6000, Potchefstroom, 2520, South Africa, *Tel*: +27-18-2994780, elsa.mentz@nwu.ac.za

employer for entry-level project work using particular (vendor) products. Several researchers have studied the knowledge and skills requirements for IT professionals through quantitative analysis but no mixed methods study could be found that studied the view the industry has of the new graduates and the problems and challenges connected to SD education. In addition, the solutions to the problems and challenges surrounding SD education have been discussed in papers with a view to the future but have not been researched in detail.

In view of the rapid pace at which technology is changing and in the light of the shortage of software developers, the present study investigated the relevance of software development education as it pertains to the industry. The research questions were:

- What is the industry's view of its new graduates?
- Is there a gap between software development education and the workplace from the perspective of the industry?
- What are the problems and challenges surrounding software development education and what are the solutions to these problems and challenges?

The results of this study could help SD educators and curriculum developers to form a picture of the recent graduates from the perspective of the industry in order to relate the suggestions of the SD industry for relevant SD education to the academic preparation of future software developers. Furthermore, the results can give an indication of the responsibility and contribution of the software industry in respect of the education of new recruits. The SD students, especially those nearing the end of their studies, could use the results of this study to prepare for the expectations of their future employers. It can therefore result in more relevant software development education with regard to the industry and it might contribute to meeting the demand for software developers.

1.1 Terminology

The ACM, IEEE Computer Society and AIS have joined forces to regularly create and update curriculum guidelines for undergraduate degree programmes in the following computing disciplines - computer science, computer engineering, information systems, information technology and software engineering. However, the dynamic nature of computing makes it difficult and potentially misleading to define the computing disciplines and related terminology (Guzdial *et al.*, 2009; Gruner, 2014).

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

- *Software development (SD)* - the ISO (International Organization for Standardization) and the IEC (International Electrotechnical Commission) define *developer* as an individual or organisation that performs development activities (including requirements analysis, design, testing through acceptance) during the system or software life cycle process (ISO/IEC, 25000, 2014). For the purpose of this study *Software development* will refer to the process of developing software through successive phases in an orderly way.
- *Industry* – the ISO and IEC define *software manufacturer* as a group of people or organisation that develops software, typically for distribution and use by other people or organisations (ISO/IEC, 19770-2, 2009). For the purpose of this study, *Industry* will refer to software manufacturers, as well as organisations where software is not the organisation's main product but software is developed for use within the organisation.

1.2 Transdisciplinary research

This article reports on the transdisciplinary field of research by transcending the disciplinary borders of Education and Computing (including the Computing disciplines of Computer Science, Information Systems, Software Engineering and Information Technology). This study explored industry's perception of the relevance of software development education and therefore ties in very well with the properties of transdisciplinarity outlined by Kroeze and van Zyl (2014) namely its broad, evolutionary scope, its focus on societal complexity, its holism, its practical relevance and its tangible impacts. True to transdisciplinarity, the study focused on practical relevance of software development education in society (the software industry) and aimed at finding practical solutions to the problems and challenges faced by the software industry. Furthermore, this study attempted to address the call of Madni (2007) for transdisciplinary research and transdisciplinary education whereby transdisciplinary education will flow from this transdisciplinary research since the identified problems and solutions can be included in curricula in order to bridge knowledge gaps.

2. Literature review

In this section, the background and literature regarding the new graduate in the workplace, the education new graduates received and the challenges will be discussed.

2.1 The new graduate in the workplace

Most new graduates 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 people who may have never known life without the Internet (Oblinger and Oblinger, 2005). 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). In the workplace, Millennials want flexible work schedules and they do not like traditional office rules and hierarchies. They want continuous performance feedback and career advice from managers and they think that managers could learn from their young employees. What could be misinterpreted as the "self-importance" attitude of Millennials is actually an optimistic sense of having many new ideas and a desire to not only contribute their ideas, but also to have their skills utilised by managers. Millennials want to wear jeans to work and especially in IT companies, the norm is to wear casual clothes, for example jeans, sneakers, flip-flops and sweatshirts – (Facebook CEO Mark Zuckerberg's famous hoodie and former Apple CEO Steve Jobs' black turtleneck with jeans are good examples) (Schawbel, 2012). The older generations, on the other hand, are more prone to believing in the importance of maintaining a standard professional look in the workplace. It seems as if Millennials also prefer casual attire because they do not separate their personal and professional lives in the same way that the older generations do.

However, not all today's students can be described as the Net generation, since not all students had and still have the benefit of state-of-the-art ubiquitous technology and the latter group of students exhibit a high degree of technological diversity. They may have information literacy characteristics and IT skills quite different from the typical Net generation. Higher education comprises a highly diverse and growing student body possessing a core set of technology based skills, but beyond this core there is a diverse range of skills across the student population (Lorenzo *et al.*, 2007; Jones *et al.*, 2010; Kennedy *et al.*, 2008).

Much is written about the so-called "generational gap". Young professionals or Millennials are joining Generation X (born 1960-1980) and Boomers (born 1940-1960) in modern organisations. The three generations are inherently different - they approach work, work/life balance, accountability, delegation, loyalty, authority, motivation and reward systems differently (Macon and Artley, 2009). However, Augustine (2001) states that a popular misconception exists that the workforce consists of generational tribes engaged in rivalries and conflicts with each other. Augustine (2001) admits that there might be differences, misunderstandings and tensions among workers from different generations, but the division and conflict are often exaggerated.

Another misconception is that Millennials want to change jobs frequently (Murray, 2015). Millennials actually value job security more highly than previous generations and they also value job enjoyment and are inclined to leave a job they do not enjoy. There is also a misconception that money does not matter to Millennials. A high-paying job is near the bottom of their list of work priorities - but the same applies to other generations, in nearly equal numbers (Murray, 2015). In their study Soni *et al.* (2011) determined the differences in work commitment of software professionals from the X-generation and Y-generation (Millennials). The study examined generational differences for the five types of work commitment - work involvement, job involvement, work group commitment, organisational commitment and professional commitment. Organisational commitment and professional commitment in turn had three components each - affective commitment, continuance commitment and normative commitment. Soni *et al.* (2011) found that the two generations differed significantly on only three of the nine factors examined. Continuance commitment to the profession is significantly higher for Generation X than for Generation Y, meaning that Generation X feels obliged to stay in the software profession. The Generation Y group of employees has higher job involvement and their continuance commitment to the organisation is significantly higher than that of the Generation X group.

In higher education a popular notion exists to describe students as 'digital natives' and lecturers as 'digital immigrants'. Students' immersion in digital technologies creates in them a radically different approach to learning and it is expected from lecturers to adapt their methods to students' new way of learning in order to remain relevant. However, Bayne and Ross (2011) call for a more cautiously critical and nuanced understanding of the effects of new technologies on the practices and subject positions of learners and teachers in higher education.

2.2 The education of new graduates

According to Spicer (2011), any business needs from its incoming recruits "Critical Cross-field Outcomes" - the skills and abilities that the South African Qualifications Authority (SAQA, 2000) requires to be achieved in all their registered qualifications. The SAQA (2000) lists seven outcomes, namely problem solving and creative thinking; being able to work in a team; the ability to manage oneself and one's activities, being able to critically evaluate information; good written and verbal communication; an effective use of science and technology; and being 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. These outcomes describe what can also be called soft skills or non-technical skills. 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. Lethbridge (2000) analysed the relevance and depth of specific core topics 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 and 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.

When Information Systems curricula were analysed from the perspective of the industry the following gaps in knowledge and skills were identified: 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 and Han, 2008; Kim *et al.*, 2006; Moreno *et al.*, 2012). Aasheim *et al.* (2009) compared the perceptions academics have of the importance of various skills for entry-level IT workers with the view that IT managers have. They found that IT managers place more importance on issues related to hardware concepts, operating systems, leadership skills or entrepreneurial traits than academia. However, both groups ranked broader categories of skills - interpersonal skills, personal skills, technical skills, organisational skills and work experience - in the same order of importance. The research of Bullen *et al.* (2009) examined workforce trends in IT-provider companies and encountered problems in the following areas: graduates who are not trained in areas that the marketplace is seeking; thin pipeline for specific technical skills; increasing pressure to source IT capability; and lag in university responsiveness to the needs of the marketplace. To effectively fill this gap between the knowledge and skills demanded by the industry and the knowledge and skills gained by graduates of university computing courses, it would be necessary, on the one hand, to guarantee that the educational programmes provide the knowledge required for the job profiles suggested by industry and on the other hand, to ensure that this knowledge is taught in a manner enabling future professionals to correctly tackle the problems that they will face during their professional career (Loftus *et al.*, 2011).

Industry's view on the relevance of SD education is determined not only by the knowledge and skills (technical, non-technical and business) acquired by graduates at university but also by their awareness of the proven techniques, challenges and expectations of their roles in employment. Gallagher *et al.* (2010) investigated the skills considered critical when hiring entry-level employees and report that while both technical and non-technical skills are important, the skills most critical are non-technical skills, such as project management, business-domain knowledge and relationship skills. IS professionals' competencies change as they gain experience in the workplace and they are required to have higher levels of technical skill in early stages and higher levels of non-technical skill in the later stages of their careers (Lee *et al.,* 2001). Plice and Reinig (2007) found that emphasising technical topics at the expense of business content may provide short-term benefits in transitioning to the workforce, but it might inhibit career advancement as graduates assume greater managerial responsibilities. Emphasising communication and teamwork skills while maintaining the existing curriculum balance between business and technical content is indicated as an appropriate strategy to align the IS curriculum with the needs of industry.

Hassan (2008) warns that as a result of the increasing breadth of content introduced into the IS field, undergraduates may become multidisciplinary illiterates. The multidisciplinary nature of the content produces graduates as generalists instead of specialists. Hassan (2008) further warns that without a clear body of knowledge built within a succinct codified framework for students, the IS field will continue to struggle in recruiting new members into the field.

Software and technical developments have made remarkable strides in the last few decades. 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; Saiedian, 2009). The diversity of software applications requires adaptability in responding to client needs and the diversity of clients and contexts requires the ability to discriminate between criteria for success. This presents new challenges for the education of software developers (Shaw *et al.*, 2005; Shaw, 2000).

2.3 Challenges

The challenges facing SD education have not been researched in empirical studies in detail but the British Computer Society (McGettrick *et al.*, 2004), Shaw (2000), Lethbridge *et al.* (2007) and Mead (2009) reported on some of the challenges and aspirations they foresee for computing education and software engineering education.

The perception of computing. Computing departments experienced a decline in enrolment that has occurred since 2000, with bright minds being attracted to other disciplines. Low enrolments result in low numbers of post-graduate students, which in turn will restrict the capacity for research in the field. A positive public image of computing must be promoted whereby good students are attracted and the public gains respect for the field and the professionals who practise within it (McGettrick *et al.*, 2004; Lethbridge *et al.*, 2007).

The dimensions of the field. There are five different undergraduate degree programmes in computing covering computer science, information systems, software engineering, computer engineering and information technology. Knowledge about specialties in terms of how practitioners work, the distinct roles in software development and what their different educational needs are, is needed. This will allow for better-tailored programmes, simpler models of computing and a smaller core resulting in the provision of appropriate education for each student (Lethbridge *et al.*, 2007; Shaw, 2000; McGettrick *et al.*, 2004).

Real-world industrial practices. A large collective effort should be launched to establish two-way knowledge and skill exchanges with companies from more industrial sectors and with more professionals within these companies in order to communicate these real-world industrial practices more effectively to students (Lethbridge *et al.*, 2007; Mead, 2009).

Evidence based education. Computing education should ensure that everything in the design and delivery of SD education must become increasingly based on carefully-evaluated evidence from industrial practice. Computing education models must be examined with workforce needs in mind (Lethbridge *et al.,* 2007).

Educator background. Just as students and curricula have to stay updated in the rapidly changing world of technology, similarly the educators should not stay behind. Educational institutions should ensure that educators have the necessary background (Lethbridge *et al.*, 2007).

Curriculum standards that are forward-looking. In an evolving field such as SD, the challenge is a fast response to changes in technology and designing curriculum guidelines for the future, while supporting current practice (Lethbridge *et al.*, 2007; Shaw, 2000).

Education for existing practitioners. In addition to educating new students, the knowledge levels of the existing workforce need to be raised by providing an infrastructure to provide support and guidance on a career-long basis for practitioners (Lethbridge *et al.*, 2007; Shaw, 2000; McGettrick *et al.*, 2004).

Information technology in the classroom. Educators in the software field can do better at exploiting technology to support the learning process itself (Shaw, 2000).

Educational research in computing. The prestige and quality of educational research in computing need to be raised (Lethbridge *et al.*, 2007). Educational research is often not recognised as "legitimate" technical research and Mead (2009) challenges educators to continue to push for recognition of educational research as a legitimate field of research.

Variety of delivery mechanisms. Educators need to adjust from the traditional lecture format because distance delivery via a variety of mechanisms is here to stay (Mead, 2009; Shaw, 2000; McGettrick *et al.*, 2004).

The literature was explored regarding new graduates, the possible gap between software development education and the workplace and the anticipated challenges faced with new graduates' education. As these topics were not explored in detail, this study therefore investigates the relevance of software development education from the point of view of the industry.

3. Research method

In this section the terminology will first be explained and then the research design, the demographics of the participants, as well as the data collection and analysis are discussed.

3.1 Research design and participants

A mixed methods approach was used to conduct the research. Tashakkori and Creswell (2007:4) describe mixed methods as:

> *Research in which the investigator collects and analyses data, integrates the findings, and draws inferences using both qualitative and quantitative approaches or methods in a single study or programme of inquiry.*

Mixed methods research can aid in the development of rich insights into phenomena of interest that cannot be fully understood using only a quantitative or a qualitative method. Mixed methods research will often provide the most balanced, informative, complete and useful research results (Venkatesh *et al.*, 2013; Johnson *et al.*, 2007). Creswell and Clark (2007) suggested four major types of mixed methods design: (1) triangulation; (2) embedded; (3) explanatory; and (4) exploratory. In this study the type of mixed methods research was explanatory, as the objective of the qualitative investigation was to supplement the quantitative investigation and to better understand and explain the observations of the quantitative investigation. This mixed method study was conducted in South Africa and for the quantitative part of the study a survey was used and the qualitative data was acquired through the comments by survey respondents.

In 2013's last quarter a convenience sample of 995 professional software developers in South Africa was taken. The respondents were members of the following groups of the professional networks LinkedIn and MyBroadband: Software and Web Developers in South Africa, SA Developer.NET and C# Developers/Architects. They were contacted via e-mail and requested to complete the anonymous online survey. Some of the respondents indicated that they sent the link of the survey to their colleagues for completion. In addition, five managers at software houses were contacted and they sent the link of the survey to the software developers in their company. The number of usable responses received totalled 214, which indicates a response rate of around 21%.

Table 1: Profile of respondents (n=214)

		Number(%)respondents
Gender	Male	196 (92%)
	Female	18 (8%)
Age category	18-24	25 (12%)
	25-29	64 (30%)
	30-39	94 (44%)
	40-49	28 (13%)
	50-59	3 (1%)
	>=60	0
Education	Matric	10 (4.5%)
	Certification	22 (10%)
	National Diploma	30 (14%)
	IT degree(s)	104 (49%)
	BSc/BCom	38 (18%)
	Engineering degree	10 (4.5%)
Level of education	Matric	10 (4.5%)
	Certification/Diploma	52 (24%)
	B-degree	70 (33%)
	Hons	54 (25%)
	Masters	28 (13%)
	PhD	1 (0.5%)
Work experience (in years)	0-4	63 (29.5%)
	5-9	62 (29%)
	10-14	51 (24%)
	15-19	22 (10%)
	20-29	13 (6%)
	30-39	3 (1.5%)
	>=40	0 (0%)
Years at current employer	0-2	123 (57%)
	3-4	47 (22%)
	5-9	26 (12%)
	10-19	13 (6%)
	20-29	4 (2%)
	30-39	0 (0%)
	40+	1 (0.5%)
Involved in hiring of new graduates?	Yes	99 (46%)
	No	115 (54%)

Table 1 provides a summary of the biographic data. The gender profile is a concern but not surprising with only 8% of the respondents being female and the results obtained from the females can therefore not be generalised. The age profile indicates that 42% of the respondents are young, Generation Y software developers, 57% are Generation X software developers and a mere 1% are Boomers. In terms of the respondents' education 49% of them are in possession of an IT degree or degrees, with another 22.5% having related degrees. It is not uncommon in software development to find people with few qualifications (4.5% of the respondents). There are software developers who taught themselves to program and they provide proof of their knowledge, skills and experience to employers in the software industry through their work. The level of the education of the respondents was quite high with 38.5% of them possessing a post-graduate qualification and even one with a PhD. The work experience of respondents indicates that 70.5% of them have more than five years' work experience and 46% of them are involved in the hiring of new graduates. A common trend is for software developers to not stay long in one position or workplace and this trend also surfaced in this study with only 43% of the respondents working more than two years at their current employer.

3.2 Data collection, instrument and analysis

An initial list of questions was developed by writing both new items and adapting items from available surveys, such as ROSE (Schreiner and Sjøberg, 2004) and the SAQA's list of "Critical Cross-field Outcomes". Once the initial questions were generated, they were sent to industrial and academic experts to refine the instrument. Feedback from this pilot study served as the basis for correcting, refining and enhancing the questions and it resulted in a questionnaire with a pool of 30 items. 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 two domains. The first domain, "Educational background of new recruits", had 16 items and enquired on what respondents view as important in SD classes, such as industry experience of lecturers. The second domain, "Career", gathered data on SD careers, such as what is expected from a good software developer. The first and second domains were accompanied by a five-point Likert response scale from 1 (Strongly disagree) to 5 (Strongly agree).

Factor analysis was used to investigate the 30 items in more detail to reduce the variables into a smaller number of factors. The 214 responses were examined using principal components factor analysis as the extraction technique and Oblimin with Kaiser Normalization as the rotation method. After analysis of the results of the factor analysis, the researchers felt that some items were of lesser importance in answering the research questions, which resulted in five items being omitted. Of the remaining 25 attitude items, 10 items were being handled as single research variables and the remaining 15 items yielded four interpretable factors. Factors were named according to their main context. A Cronbach's α coefficient was calculated for each of the four factors and was found, as Table 2 shows, to be reliable (α ≥0.60).

Qualitative data was gathered by means of an open-ended question at the end of the questionnaire, asking for further comments on the education of software developers and 77 respondents added comments. Furthermore, there were 21 respondents who felt so strongly about the topic that they gave up their anonymity and e-mailed the researchers with more comments and suggestions.

Table 2: Factors* (with reliability coefficients) and items

Factor	Cronbach's alpha (α)
Critical outcomes required in software development modules.	0.718
Positive attitude towards colleagues/management.	0.793
Positive attitude towards tasks/work.	0.666
Emotional/social skills required.	0.700
Item	
Neat and tidy appearance required.	
Good set of exam results required.	
New recruits have the knowledge and skills to immediately contribute to the development of software.	
Students know what software developers do in the workplace.	
New recruits have to be trained/sent for training before they can contribute to the development of software.	
Our country needs more software developers.	
People from industry should be brought into software development classes.	
Lecturers should have industry experience.	
In the software development modules students should learn to use science and technology effectively.	
Projects play an important role in the education of students.	

** See Appendix A for the items in each factor*

Basic analysis of quantitative data was done by calculating the mean values and standard deviation of each of the 14 variables. Various statistical tests were used in the analysis to match the metric being analysed. Three groupings were identified based on gender, age and education. All the groupings were tested for significant differences between means in the different variables using T-tests and ANOVA tests. Chi-square tests and Spearman's rank correlation analysis were also used to analyse relationships between the groupings and the variables. When the results of the interaction analysis are reported, only the significant interactions or primary effects will typically be discussed. Since a convenience sample instead of a random sample was used, the p-values will be reported for the sake of completeness but will not be interpreted (Steyn *et al.*, 1999).

For the analysis of the qualitative data the ATLAS.ti 7.1.4 computer program was used and the data was analysed as follows:

The data of the open-ended question in the online survey and the data from the e-mails were assigned to a single hermeneutic unit;

The relevant information was separated from the irrelevant information;

The relevant information was broken into a number of text segments;

The emerging themes were coded in a process of inductive categorisation and the text segments were linked to the coded themes;

The codes were grouped into code families;

Networks were drawn that reflect the meaning of the respondents' views; and

The networks were used to develop an overall description of SD professionals' views of SD education.

The code families and themes that emerged from the analysis were:

> *New graduates: adaptation to the workplace; weaknesses; new graduates vs. unqualified recruits; passion; thirst for knowledge.*
>
> *The possible gap: true; false.*
>
> *The problems and challenges: who is to blame?; lacking skills and knowledge; rapid change.*
>
> *The solutions: teamwork; projects; experience; work-integrated-learning (WIL); mentoring; tips for teaching.*

Since the product of qualitative research is richly descriptive (Merriam, 2009), the participant comments were analysed, interpreted and presented in the form of quotes.

4. Results and discussion

In this section, important data for each of the concepts are considered, as well as the qualitative data that provides a rich description of the information obtained.

4.1 The new graduate

From Table 3 it is clear that the respondents felt very strongly that new recruits must have a positive attitude towards colleagues and management, as well as a positive attitude towards tasks and work. Respondents also felt quite strongly about the emotional and social skills expected from new recruits when they enter the workplace. The respondents did not feel that a neat and tidy appearance is such an important requirement and respondents had a relatively neutral feeling about the standard of the examination results of students but the deviation was quite large.

Table 3: Basic analysis of new graduate factors and items

(I)tem / (F)actor	Mean*	Standard deviation
Positive attitude towards colleagues/management (F)	4.44	0.562
Positive attitude towards tasks/work (F)	4.55	0.458
Emotional/social skills required (F)	3.75	0.720
Neat and tidy appearance required (I)	2.79	1.111
Good set of examination results required (I)	3.03	1.041

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

Gender differences were analysed with a T-Test and the female respondents felt more strongly ($d = 0.47$, $p<0.05$) about a neat and tidy appearance required in the workplace. In addition, the T-Test showed that the female respondents felt more strongly ($d = 0.48$, $p<0.05$) that good examination results are important.

Generational differences were analysed with an ANOVA and the results in Table 4 indicate that the older software developers (Generation X) differed from the young software developers (Generation Y) in terms of new recruits' appearance. Surprisingly enough, it was the older generations that felt that a neat and tidy appearance is not such an important requirement in the workplace. This finding contradicts the view of Schawbel (2012) that the older generations believe that it is important to maintain a standard professional look in the workplace.

There is a medium practically significant difference between the 18 to 24 age group and the software developers over 40 in terms of their views on academic performance. The young software developers view a good set of examination results as significantly more important than the older respondents. This might be explained by the experience of Generation X where they had seen that it is not necessarily the academic giants of university that turned out to be the best and most innovative employees.

In terms of emotional and social skills required, Table 4 shows that there is also a medium practically significant difference between the 18 to 24 age group and the software developers over 40. The fact that the Generation X software developers view emotional and social skills as significantly less important than the young respondents might be explained by one of the three items in the factor, namely: "*To be a good software developer you have to have modern leadership skills, such as self-confidence and a preparedness to lead by example*". Schawbel (2012) pointed out that Millennials' attitude could be misinterpreted as "self-importance" by the older generations and in the qualitative data this problem with the new recruits' attitude emerged as well.

Table 4: Differences between the age groups

(F)actor / (I)tem	Age groups	Effect size	p
Neat and tidy appearance required (I)	18-24 *vs* 25-29	0.53*	0.078
	18-24 *vs* 30-39	0.48	
	18-24 *vs* 40+	0.62*	
Good set of examination results required (I)	18-24 *vs* 25-29	0.41	0.114
	18-24 *vs* 30-39	0.23	
	18-24 *vs* 40+	0.57*	
Emotional/social skills required (F)	18-24 *vs* 25-29	0.25	0.061
	18-24 *vs* 30-39	0.46	
	18-24 *vs* 40+	0.54*	

** Medium practically significant difference*

A Spearman rank correlation analysis found a medium practically significant relationship (r = .416, p<0.001) between the respondents' views of a developer's appearance and his/her examination results. Therefore, respondents who viewed a neat and tidy appearance to be important, also felt strongly about good examination results for software developer students.

The qualitative data revealed the following regarding the new graduates:

Adaptation to the workplace

New recruits adapt quickly to the workplace and they learn fast.

> *I have several varsity graduates with BSc Comp Sc qualifications who graduated in the last 2 to 3 years working for me and all of them are very intelligent young people who learn fast and have proven to be valuable members of my team. Having said that, however, that was NOT the case when they first arrived.*

> *Devs need the ability to learn quickly and adapt to constraints.*

Weaknesses

Respondents pointed out the weaknesses of new recruits. A great number of respondents pointed out that new recruits have theoretical but not applied knowledge and they lack problem-solving skills.

> *Lots of graduates/junior developers have excellent theory but have absolutely no clue how to implement it.*

> *Graduates seem to lack the analytical, logical and problem-solving abilities required*

Respondents felt that new recruits arrive in the workplace with a laid-back attitude and are emotionally immature. Respondents experience a lack of respect from the new recruits and a lot of attitude. This finding can probably be explained by the "generational gap" because as Schawbel (2012) pointed out, Millennials' attitude could be misinterpreted as "self-importance", but Millennials see no reason for a strict hierarchy and most Millennials want to contribute, believing that everyone's ideas should be heard.

> *They have a YOLO (You Only Live Once) attitude - I guess we were all young and impetuous once, but we need emotionally mature and stable professionals.*

> *Starters seem to have a lot of attitude and little respect. This makes it difficult to make them productive members of a team.*

Another weakness from the perspective of the software developers is that new recruits apply for and end up in a career that does not match their personality type.

> *Wrong career choice for their personality. Software developers are usually DISC "C" personalities (INTx in Myers Briggs). Project Managers and Architects are usually D/C - i.e. "Left Brainers". Business Analysts need more people / emotional skills.*

New graduates vs. unqualified recruits

The respondents, however, felt that new recruits without a degree had more weaknesses that probably cannot be rectified.

> *Unfortunately for the most time I have noticed a lack of architectural growth from degreeless developers in the long term, 3 years+. Their work often shows signs of dirty coding and lack of interface design skills. Lack of following standards and doing things the best way is often a sign of poor background. They also seem to project a type of unwarranted arrogance.*

Passion

Respondents commented that the best graduates are passionate about software development and the employers find that passion is lacking in some of the new recruits.

> *The best graduates (and thus the hardest to get) are the ones who unquestionably speak with an inbuilt passion for development.*
>
> *What ultimately stands out for me when I hire new people is their passion for what they do and hunger to learn more and constantly improve, as well as high quality work.*

Thirst for knowledge

In view of the rapid pace at which technology is changing, the respondents emphasised the fact that new recruits must have a desire to try out and learn new things.

> *Hunger for knowledge and the work ethic to see it through are essential – I'm probably learning more every day now after 30 years than I did at university.*
>
> *Software developers require a natural thirst to solve problems through their skills, as well as a natural want to learn new things.*
>
> *Hard work, thirst for knowledge and learn as many technologies as possible (be dynamic not conservative).*

4.2 The possible gap between software development education and the workplace

Table 5 shows that the respondents felt quite strongly about the shortage of software developers in South Africa and they strongly agreed that further training is required for new recruits. On the other hand, the mean values of 2 variables are relatively low with the respondents feeling that new recruits cannot immediately contribute to the development of software and that students do not have a clear idea of what software developers do in the workplace.

Table 5: Basic analysis of gap items

Item	Mean*	Standard deviation
Our country needs more software developers.	4.32	0.890
New recruits have to be trained/sent for training before they can contribute to the development of software.	4.01	0.836
New recruits have the knowledge and skills to immediately contribute to the development of software.	2.35	1.063
Students know what software developers do in the workplace.	2.70	1.037

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

Differences based on work experience were analysed with a T-Test and a medium practically significant difference ($d = 0.50$, $p<0.001$) between the respondents with less than 10 years' experience and the rest of the respondents was found, in terms of the new recruits' knowledge and skills, to immediately contribute to the development of software. The respondents with less than 10 years of work experience had a significantly stronger opinion that new recruits can immediately contribute, whereas the more experienced software developers felt that new recruits do not have the knowledge and skills to immediately contribute to software development.

From the above it is not surprising that the results of an ANOVA in Table 6 indicates that the older the software developers were, the more they differed with the young software developers in terms of the knowledge and skills of new recruits. The older respondents felt that new recruits lack the knowledge and skills to immediately contribute to the development of software.

Table 6: Differences between the age groups

Item	**Age groups**	**Effect size**	**P**
New recruits have the knowledge and skills to immediately contribute to the development of software.	18-24 *vs* 25-29	0.43	<0.05
	18-24 *vs* 30-39	0.61*	
	18-24 *vs* 40+	0.77*	

* *Medium practically significant difference*

Furthermore, correlation techniques were used to analyse the correlation between the variables. A Spearman rank correlation analysis found a medium practically significant relationship ($r = 0.376$, $p<0.001$) between the respondents' views of the new recruits' knowledge and skills entering the workplace and students' knowledge of what software developers do in the workplace.

When conducting mixed methods research, a researcher may find contradictory conclusions from the quantitative and qualitative strands, but these contradicting findings are valuable in that they enrich understanding of a phenomenon but also open new avenues for future research (Venkatesh *et al.*, 2013). In the reported quantitative results above it showed that the respondents felt that there is a gap between SD education and industry needs and some of the qualitative results re-affirmed that notion. However, the comments of some respondents contradicted the idea of a gap between industry needs and SD education.

True - there is a gap

When looking at the qualitative data, some respondents felt strongly that there is a gap between SD education and industry needs. Words, such as "*huge gap*", "*big gap*", "*definitely a gap*", "*significant gap*", "*massive gap*" were used.

> *In my experience there is a massive gap between university grad knowledge and real-world industry needs.*
>
> *I do think there is distinct disconnect between what students are being taught at varsity and what they actually need to know to work in a proper software development house and be useful and productive.*
>
> *A piece of paper obtained by a student in IT in no way proves the skills and ability of that student. A huge misconception. It's not a good enough test of a good developer. Most of my best devs haven't finished their degrees…*
>
> *Industry vs. Training doesn't weigh up!*
>
> *Some respondents spoke of shock and horror at the graduates' knowledge and skills.*
>
> *We brought on 3 graduates on a 3-month internship. I was shocked at how poor their actual skills were. Some of them knew a lot of theory but no practical application.*
>
> *Last year we brought on 3 graduates for an internship. I was horrified at how little they knew of real software development.*

False – there is not really a gap

However, not all respondents felt negative about the knowledge and skills of graduates and they think that graduates have a broad basic knowledge, that it is not difficult to train them and they are a great asset in the long term.

> *Computer science degrees often don't prepare an individual to go out there and start developing systems from scratch, but it gives them a long term advantage, a broad knowledge and understanding of how they should learn development and what they should try to avoid.*
>
> *Training them to be productive real-world developers has never been a particularly difficult or time-consuming task and in general, if they have earned a BSc Comp Sc, it does tend to indicate that they are of a higher calibre than most candidates and will prove to be a better hire in the long run.*
>
> *I have noticed that university gives developers a foundation of broad computer science knowledge, that without a degree very few individuals have. A degree also shows commitment to something that will take a while to complete and shows a character that is often necessary in long-term self-development of the individual.*

From the above it is clear that the respondents had opposing views regarding the gap between the new graduates' education and their workplace. These opposing views most probably originate from the respondents' views on the mission of the university. If they view the mission as educational as Reisman (2004) also pointed out, then they do not think there is a big gap, but if they view the university's mission as one of training graduates to immediately be productive software developers, then they experience a big gap.

4.3 The challenges in software development education

Analysis of the qualitative data highlighted a number of challenges and problems that the software developers in the industry foresee. "*We have a big problem in South Africa when it comes to software writing skills*".

Who is to blame?

Some respondents laid the blame for the problems with new software developers at the door of the university.

> *I really feel that the tertiary education system is failing them horribly.*
>
> *I have been attempting to convince XYZ University to produce more able software developers, but nothing is changing.*

Other respondents blamed the software industry for the problems with new software developers.

> *No problem with the education. What falls flat is what happens with IT companies after graduation.*
>
> *There is a shortage of developers, but companies do not want to invest in graduates as they are afraid they will leave for better salary, etc. However, those same companies go to India, etc. for developers and further add to the shortage of developers. All a vicious circle, sadly.*

Lacking skills and knowledge

Respondents listed a number of skills that are lacking: "no basic troubleshooting and investigation skills"; "No working experience at all"; "a lot of theory but no practical application"; "No teamwork experience"; "Too informal processes"; and "they have no concept of how real projects are managed, how projects are planned and timings are estimated".

One of the respondents echoed the claims of Gallagher *et al.* (2010), Lee and Han (2008) and Plice and Reinig (2007), namely that the main problem is that new recruits lack business knowledge.

> *I think it very much depends on the industry you go into. In my personal experience – I have gone into investment banking (still working in IT though). I felt a big gap when I entered the working world purely because I wasn't prepared from a business knowledge perspective."*

Rapid change

Respondents emphasised the continuing and rapid changes in the software industry and that the knowledge the graduates acquired at university is often outdated. Some respondents felt that students should rather learn foundational principles and theories at university and others felt that university courses should regularly be updated.

> *The world is moving ever faster – what worked over the last 10 years won't necessarily work in the future*
>
> *...they also tend to do things in ways that are now considered old-fashioned and outdated and often had to be taught to do things differently to what they had learnt at university.*
>
> *Practical software development skills change rapidly so tertiary institutions should focus on general theory*
>
> *According to quickly changing software industry, varsity courses need to be revised every 3-4 years – to keep knowledge up to date.*

The words of a software development manager effectively summarises the above opinions: "The general take away for me, when dealing with varsity graduates is not that they are being taught badly or incorrectly at varsity, but rather that what they are being taught always seems to be slightly out of date and behind current trends and technologies. On the plus side, I don't think that their degrees can be considered easy or of little value by any means. They clearly worked very hard to earn those degrees and they do seem to be taught a lot of solid foundational principles that can be built on very easily. The market has a severe lack of solid theoretical computer science training."

4.4 The solutions to problems in software development education

All the variables listed in Table 7 have high means indicating that software developers viewed these variables as important solutions to the problems in educating software developers. Respondents felt that experts from industry should be used to teach students and that the lecturers should in turn gain experience in the workplace. Respondents also strongly agreed that the education of software developers requires what can be referred to as non-technical skills which in South Africa are known as critical outcomes.

Table 7: Basic analysis of solutions

(I)tem / (F)actor	Mean*	Standard deviation
People from industry should be brought into software development classes (I)	4.37	0.769
Lecturers should have industry experience (I)	4.43	0.752
In the software development modules students should learn to use science and technology effectively (I)	4.20	0.707
Projects play an important role in the education of students (I)	4.32	0.734
Critical outcomes required in software development modules (F)	4.40	0.488

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

The qualitative data revealed some interesting and useful suggestions as solutions to the problems in educating software developers.

Teamwork

Respondents felt strongly that the university and the IT industry should work together in creating up to date curricula.

> *I think it's of vital importance in the software industry, more important than any other industry, that academic institutions consult with the private sector to learn how best to equip students for their first real-world position as a software engineer.*
>
> *Industry professionals NEED to be brought in for consultation on the tool chain they make daily use of in their development role.*

The above comments confirm the challenges Lethbridge *et al.* (2007) and Mead (2009) anticipated.

Projects

Respondents felt that students should receive more challenging practical assignments and projects during their years of study.

> *More practicals! Lots of graduates/junior developers have excellent theory but have absolutely no clue how to implement it.*
>
> *Practical SDLC experience in a team setting (systems development projects) is a crucial part of the learning process.*
>
> *They need to do more stressed-for-time projects. The projects must not be just about stuff they do know. It must challenge them.*

Experience

Respondents stressed the fact that the software industry is looking for people with experience and not necessarily a degree - *"someone with a degree AND experience is the holy grail of HR managers"* - and a great deal of emphasis is placed on practical and real-life experience.

> *Give them the practical experience they need and projects related to current real-life environments*
>
> *Universities should bridge this gap by integrating more experience into the curriculum.*

Work-integrated-learning (WIL)

Several respondents suggested that work-integrated-learning in the form of a vacation job or internship could be a solution for effective software development education.

> *It should work the same as an engineering degree where the student needs a year of actually working for someone before the degree is awarded.*
>
> *Software Developers should have vacation jobs in relevant companies where they can get corporate experience in development.*
>
> *Important to reward future software devs if they do well - a holiday internship at a dev house would be the best reward.*

Mentoring

The education of software developers is not only the duty of the university, but the industry can play its part through a mentorship programme.

> *Graduates need to be mentored and coached in their first year of employment. This is lacking extensively!*

Tips for teaching

Respondents commented on what lecturers should focus on in the education of software developers, including soft skills and technical skills. Respondents confirmed the aspirations of Lethbridge *et al.* (2007) and Shaw (2000) that an important factor is for the university to keep up to date within the rapid changing technology environment.

> *The education of software developers should emphasise soft skills as much as technical skills.*
>
> *Lecturers should be able to teach more up to date skills to students, and open their eyes to possible exposures to methodologies.*
>
> *My kids will MOST LIKELY also work in this industry and I hope that they will be taught the correct languages to make them appropriately skilled for the job market.*

One respondent pointed to the fact that software developers rarely start from scratch when writing a program and students should therefore be taught to utilise all the resources at their disposal.

> *...show them how to troubleshoot a problem properly and leverage the internet, Google, online resources, etc. properly to their advantage.*

A respondent wraps up some of the sentiments unveiled above: "The most useful thing that varsities could probably do would be to try and make the material they are teaching the students more current and more in line with what is actually going on out there. I realise that it is incredibly difficult when you are trying to plan a syllabus ahead of time and the industry changes at the pace that ours does, but if you want to improve the marketability of students straight out of varsity, that's what will do it."

5. Conclusion and recommendations

The first research question in this study was: What is the industry's view of its new graduates? The software industry views its new recruits as fast adapters to the workplace who learn fast while several new recruits have theoretical but not applied knowledge. The SD industry views emotional and social skills as important, although some new recruits are found to be emotionally immature and disrespectful. New recruits are observed as having a "self-importance" attitude, especially in the case of Generation X. The software developers also experience that new recruits enter the workplace with their personality type not matching their career choice. However, most new graduates have a passion and thirst for knowledge, whereas unqualified recruits have weaknesses that probably cannot be rectified. New recruits' positive attitude towards colleagues and management, as well as towards tasks and work is important as far as the industry is concerned. New recruits' appearance is not perceived as too important, especially by Generation X, but women view a neat and tidy appearance as more important than men. Software developers feel relatively neutral about students' exam results, but women and Millennials view good examination results as important.

The second research question posed was: Is there a gap between software development education and the workplace from the perspective of the industry? Some people in the SD industry feel strongly that there is a gap between SD education and industry needs and that new recruits in the SD workplace are relatively unprepared to immediately start to work on projects, resulting in a need for training before they can make a constructive contribution. Some recruits do not know what the work of a software developer entails until they enter the workplace. Generation X software developers do not have confidence in the abilities of new recruits. However, some people in the SD industry think that graduates have a broad basic knowledge, that it is not difficult to train them and that they are a great asset in the long term. These opposite views can probably be explained by people's views on the mission of universities as either an educational institution or a vocational training institute.

The third and last research question was: What are the problems and challenges surrounding software development education and what are the solutions to these problems and challenges? The software industry confirms that it is challenged by a shortage of software developers. The university, as well as the software industry are to be blamed for the problems and challenges surrounding software development education. The problems and challenges identified from the perspective of the SD industry are the following. Graduates often lack business knowledge and they lack experience in teamwork, troubleshooting, investigation skills and general practical experience on real-life projects. The rapid pace at which technology is changing often causes the knowledge graduates acquire at university to be outdated.

The following solutions to the problems and challenges are recommended in order to offer relevant software development education as far as the industry is concerned:

- *Teamwork.* The university and the IT industry should work together in creating up-to-date curricula. People from industry should be brought into software development classes and lecturers should acquire industry experience.
- *Projects and experience.* Practical assignments and projects play an important role in the education of students and real-life and practical experience must be included in students' education.
- *Work-integrated-learning and mentoring.* The software industry can be helpful in accommodating students to gain experience in the workplace during their studies and mentoring them once they have entered the workplace.
- *Technical and soft skills.* Universities must examine their curricula to ensure that not only technical, but also soft skills are included.

- *Up to date.* Universities must attempt to keep the curriculum abreast of the rapid change in technology. The challenge is a fast response to changes in technology and designing curriculum guidelines for the future, while supporting current practice.
- *Career guidance.* Universities, in conjunction with the industry, should provide career guidance to students, so that they do not apply for and end up in a job that clashes with their personality type.
- *Introspection.* The mission of the university is not that of a vocational training institute but rather a higher education institute. Both the software industry and the university must do introspection to establish where they are failing to contribute to the offering of relevant software development education.
- *Generational awareness.* It is clear from the study that the older software developers or Generation X have a different, more negative view of new recruits. SD students, as well as new recruits or Millennials must be aware of the values and beliefs of the older generations, especially since most of them will start work under a Generation Xer. On the other hand the older software developers must be aware of the traits, values and beliefs of the Millennials in order to get the most out of their new recruits.

This study contributes to the sparse empirical literature on the perspective of the SD industry of their new recruits, the problems and challenges that are faced and the solutions to the problems as far as the SD industry is concerned. In South Africa some universities are significantly higher ranked than others and the same quality of education is not offered in all universities and university departments. Some of the highly ranked universities might argue that the above recommendations are already implemented by them, but it is clear from this study that the SD industry perceives a gap between SD education and the workplace and consequently all universities should take note in order to further improve the quality and relevance of SD education. SD educators and curriculum developers must take the expectations and suggestions of the SD industry in the academic preparation of future software developers into account. The SD industry can ascertain which contributions towards the education of new recruits can be made by the industry. With reference to this study, the SD students can observe the perspective of the SD industry of their new recruits and therefore be better prepared for the expectations of their future employers. If each of the role players in SD education does his/her bit, it can result in more relevant software development education.

References

Aasheim, C., Li, L. and 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.

Augustine, R.K. 2001. Thanks kiddo! A survival guide for professional generation Xers. *Organization Development Journal*, 19(2):19-26.

Bateman, K. 2013. The irony of an unemployment problem and an IT skills shortage within the IT industry, *ComputerWeekly.com,* Oct. http://www.computerweekly.com/blogs/itworks/2013/10/the-irony-of-an-unemployment-p.html Date of access: 30 Jul. 2014.

Bayne, S. and Ross, J. 2011. 'Digital Native' and 'Digital Immigrant' Discourses. (*In* Land, R. and Bayne, S., *eds*. Digital difference: Perspectives on online learning. Rotterdam: Sense Publishers, p. 159-169).

Bullen, C., Abraham, T., Gallagher, K., Simon, J.C. and Zwieg, P. 2009. IT workforce trends: Implications for curriculum and hiring. *Communications of the Association for Information Systems,* 24:129-140.

CareerJunction Index. 2014. Executive Summary, Jul. http://www.careerjunction.co.za/cji/exec-pdf/CJI_Executive_Summary.pdf Date of access: 30 Jul. 2014.

Creswell, J.W. and Clark, V.L.P. 2007. *Designing and conducting mixed methods research*. Thousand Oaks, CA: Sage Publications.

Gallagher, K.P., Kaiser, K.M., Simon, J.C., Beath, C.M. and Goles, T. 2010. The requisite variety of skills for IT professionals. *Communications of the ACM,* 53(6):144-148.

Gruner, S. 2014. On the historical semantics of the notion of software architecture. *TD The Journal for Transdisciplinary Research in Southern Africa*, 10(1):37-66.

Guzdial, M., Prey, J., Topi, H., Urban, J., Cassel. L. and Schneider, D. 2009. Future of Computing Education Summit, June 25-26, 2009. http://www.acm.org/education/future-of-computing-education-summit/ FoCES_web.pdf Date of access: 30 Jul. 2014.

Harris, L. 2012. Mind the ICT skills gap. *Brainstorm,* Sep. 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.

Hassan, N.R. 2008. Courting multidisciplinary illiteracy. (*In* Proceedings of the Americas Conference on Information Systems (AMCIS 2008). Paper 245.) http://aisel.aisnet.org/amcis2008/245 Date of access: 11 Mar. 2015.

ISO/IEC 25000, 2014: Systems and software engineering - Systems and software Quality Requirements and Evaluation (SQuaRE) - Guide to SQuaRE. ISO/IEC, Geneva, Switzerland.

ISO/IEC 19770-2, 2009: Information technology - Software asset management - Part 2: Software identification tag. ISO/IEC, Geneva, Switzerland.

Johnson, R.B., Onwuegbuzie, A.J. and Turner, L.A. 2007. Toward a definition of mixed methods research. *Journal of Mixed Methods Research*, 1(2):112-133.

Jones, C., Ramanau, R., Cross, S. and Healing, G. 2010. Net generation or Digital Natives: Is there a distinct new generation entering university? *Computers & Education*, 54(3):722–732.

Kennedy, G., Judd, T.S., Churchward, A., Gray, K. and Krause, K. 2008. First year students' experiences with technology: Are they really digital natives? 'Questioning the net generation: A collaborative project in Australian higher education', *Australasian Journal of Educational Technology*, 24(1):108-122.

Kim, Y., Hsu, J. and Stern, M. 2006. An update on the IS/IT skills gap. *Journal of Information Systems Education*, 17(4):395-402.

Kitchenham, B., Budgen, D., Brereton, P. and Woodall, P. 2005. An investigation of software engineering curricula. *Journal of Systems And Software*, 74(3):325-335.

Kroeze, J.H. and van Zyl, I. 2014. Transdisciplinarity in Information Systems: Extended Reflections. (*In* Proceedings of the Americas Conference on Information Systems AMCIS 2014, Savannah. p. 1-10).

Lee, C. and Han, H. 2008. Analysis of skills requirement for entry-level programmer/analysts in Fortune 500 corporations. *Journal of Information Systems Education*, 19(1):17-27.

Lee, S., Yen, D., Koh, S. and Havelka, D. 2001. Evolution of IS professionals' competency: an exploratory study. *Journal of Computer Information Systems*, 41(4):21-30.

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. and Thompson, J.B. 2007. Improving software practice through education: Challenges and future trends. (*In* 2007 Future of Software Engineering. IEEE Computer Society, Washington, DC. p. 12-28).

Loftus, C., Thomas, L. and Zander, C. 2011. Can graduating students design: Revisited. (*In* Proceedings of the 42nd ACM Technical Symposium on Computer Science Education (SIGCSE '11), ACM, New York, NY. p.105-110).

Lorenzo, G., Oblinger, D. and Dziuban, C. 2007. How choice, co-creation, and culture are changing what it means to be Net savvy. *Educause Quarterly*, 30(1):6-12.

Macon, M. and Artley, J.B. 2009. Can't we all just get along? A review of the challenges and opportunities in a multigenerational workforce. *International Journal of Business Research*, 9(6):90-94.

Madni, A.M. 2007. Transdisciplinarity: Reaching beyond disciplines to find connections. *Journal of Integrated Design and Process Science*, 11(1):1-11.

Mawson, N. 2010. ICT skills shortage to cost SA. *ITWeb,* http://www.itweb.co.za/index.php?option=com_content&view=article&id=29992 Date of access: 24 Jul. 2014.

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.

McGettrick, A., Boyle, R., Ibbett, R., Lloyd, J., Lovegrove, G. and Mander, K. 2004. *Grand challenges in computing education*. British Computer Society, Swindon, United Kingdom.

Mead, N. 2009. Software engineering education: How far we've come and how far we have to go. *Journal of Systems and Software*, 82(4):571-575.

Merriam, S.B. 2009. *Qualitative research. A guide to design and implementation,* Jossey-Bass, San Francisco, CA.

Moreno, A., Sanchez-Segura, M., Medina-Dominguez, F. and Carvajal, L. 2012. Balancing software engineering education and industrial needs. *Journal of Systems and Software*, 85(7):1607-1620.

Murray, A. 2015. What millennials do and don't want from their employers. *Fortune.Com*, http://fortune.com/2015/03/05/millenials-best-companies/ Business Source Premier. Date of access: 30 Mar. 2015.

Oblinger, D. and Oblinger, J. 2005. Educating the net generation. Louisville, CO: Educause. http://www.educause.edu/ir/library/pdf/pub7101.pdf Date of access: 30 Mar. 2015.

Plice, R.K. and Reinig, B.A. 2007. Aligning the information systems curriculum with the needs of industry and graduates. *Journal of Computer Information Systems*, 48(1):22.

Reisman, S. 2004. Higher education's role in job training. *IT Professional*, 6(1):6-7.

Saiedian, H. 2009. Software engineering challenges of the "Net" generation. *Journal of Systems and Software*, 82(4):551-552.

SAQA (South African Qualifications Authority). 2000. The National Qualifications Framework and Curriculum Development. http://www.saqa.org.za/docs/pol/2000/curriculum_dev.pdf Date of access: 16 Nov. 2011.

Schawbel, D. 2012. Millennials vs. Baby Boomers: Who would you rather hire? *Time,* Mar. http://business.time.com/2012/03/29/millennials-vs-baby-boomers-who-would-you-rather-hire/ Date of access: 31 Oct. 2013.

Schreiner, C. and 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, ACM, New York, NY. p. 371-380).

Shaw, M., Herbsleb, J. and Ozkaya, I. 2005. Deciding what to design: Closing a gap in software engineering education. (*In* Proceedings of the 27th International Conference on Software Engineering (ICSE '05). ACM, New York, NY. p. 607-608).

Soni, S., Upadhyaya, M. and Kautish, P. 2011. Generational differences in work commitment of software professionals: Myth or reality? *Abhigyan*, 28(4):30-42.

Spicer, M. 2011. What does business expect from its incoming recruits? Do they meet our expectations? (*In* The International Conference of the International Association for Cognitive Education in Southern Africa (IACESA), Keynote Address). http://www.businessleadership.org.za/gup/filez/Speeches_IACESA_Conference_17_February_2011.pdf Date of access: 31 Oct. 2013.

Steyn, A.G.W., Smit, C.F., du Toit, S.H.C. and Strasheim, C. 1999. Modern statistics in practice. Pretoria: Van Schaik.

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

Tashakkori, A. and Creswell, J.W. 2007. The new era of mixed methods (editorial). *Journal of Mixed Methods Research*, 1:3-7.

Venkatesh, V., Brown, S.A. and Bala, H. 2013. Bridging the qualitative-quantitative divide: guidelines for conducting mixed methods research in information systems. *MIS Quarterly*, 37(1):21-54.

Appendix A

Factor	Questionnaire items
Critical outcomes required in software development modules.	In the software development modules students should learn to work with others as a member of a team or group.
	The software development modules should require from students to organize and manage themselves effectively.
	In the software development modules students should learn to collect and critically evaluate information.
	In the software development modules students should learn to communicate effectively, both verbally and in writing.
	To be a good software developer you have to
Positive attitude towards colleagues/management.	have a willingness to listen
	have a willingness to take instructions
	have respect for others
	have a desire to succeed (realistically ambitious)
Positive attitude towards tasks/work.	be prepared to work hard
	be prepared to learn (a thirst for knowledge)
	have good time-management skills
	have a preparedness to take responsibility
Emotional/social skills required.	have modern leadership skills, such as 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 reasonable level of general knowledge