Essential Technical Competencies for Software Engineers: Perspectives from Sri Lankan Undergraduates

C.D. Manawadu Faculty of Information Science & Engineering, Management and Science University, Selangor, Malaysia Md Gapar Md Johar Faculty of Information Science & Engineering, Management and Science University, Selangor, Malaysia S.S.N. Perera Department of Mathematics, University of Colombo, Colombo, Sri Lanka

ABSTRACT

Software engineering complexities and challenges are multifold and does not only rise from technology advancements. Human factors related to software engineering is a key aspect which has been overlooked by many organizations and also received less research focus. The research which is a part of a longer study strives to build a competency framework targeting software engineers. Considering a future with more agility and product engineering this study is conducted through mixed mode methods taking Sri Lankan software engineering undergraduate students as a sample to determine the future outlook. The technical competencies identified would support undergraduates, academics and industry training providers with fresh insights from a research perspective. Further it will help undergraduates who intend to join the industry to sharpen their technical competencies.

General Terms

Software Engineering, Humans Aspects in Software Engineering, Technical Competencies.

Keywords

Software Engineering; Competency Framework; Sri Lanka, Sri Lankan Software, Software Industry, Undergraduates.

1. INTRODUCTION

A recent study of 5,400 large-scale software development projects reveled the maturity level of project success. 17 percent of the projects were below than average level and they threatened the company's existence. The study also revealed that 45 percent of the projects were over budget, 7 percent weren't on schedule, and 56 percent delivered less value than expected. [1]. One valid reason for software projects to fail is due to the poor quality in design, standards, methodologies and technologies which are used. However more than the quality, the lack of knowledge, capability and the talent of software engineers interacting with diverse technology spectrums and multi cultural teams also is a key factor for many of the projects to fail [2].

Excellent performance in the software engineering discipline is due to human factors is what research tells us. [3]. Even though software construction is an engineering process, creative software engineers who are human beings should exist to tackle the unpredictable and turbulent nature of it. Hence focusing on people and improving them will ensure they effectively perform well on a defined process and deliver successful projects. [4], [5], [6]. More precisely, Boehm points out that "After product size, people factors have the strongest influence in determining the amount of effort required to develop a software product "[7] and "Personnel attributes and Human Resource activities provide by far the largest source of opportunity for improving software development productivity" [8]. This indicates that software engineering boils down to technical competence and human factors. However, technical skills and soft skills don't receive the same degree of attention, especially by instructors of technical knowledge [9]. Hence it is important to build this engineering discipline on a competency based approach as similar to other practices and disciplines.

To overcome some of the key challenges, the agile software development methodology was introduced in the late nineties. Agile methods follow an iterative and incremental style of development where collaborative self-organizing teams dynamically adjust to changing customer requirements [10]. A research conducted within Sri Lanka identifies that a high percentage of Sri Lankan software development companies and software developers use agile methodologies to develop software [11].

Sri Lankan software development industry emerged in the late 1980's. According to the Sri Lanka Association of Software and Service Companies (SLASSCOM) there are more than 70 registered software development companies that supply products and services to offshore client firms [12]. Currently the workforce is around 34,000 spread across the country of which 49 percent of workers have a Bachelor's degree or a higher qualification in Information Technology or Computer Science and around 7123 working as software engineers [13].

The latest recognitions Sri Lanka received in the global arena comprise of been ranked 12th in the global locations trends report of IBM, been included into the top 25 global services location index according to A T Kearney, been included into the top 30 in terms of leading location for offshore services by Gartner. Receiving the outsourcing destination of the Year award from national outsourcing agency, UK in 2013 [13]. Having noted them, the expectation for the industry to earn three billion US\$ dollars by the year 2022. Currently there are earnings close to 500 million US\$ and it's the fifth largest export earner to the country [12].

The key trend in the future is that Sri Lanka is hoping to play a big role in the global centre of excellence. 2022 industry target is to be a high end software development market and make the human capital as its unique selling proposition (USP). The industry has identified product engineering to be the key value and revenue generator keeping a target to be among the top 3 globally. Customized software development is the next sought after segment and its target is to be among the top 3 regionally. Finally the support captive/outsourced software development, testing and maintenance markets to bring in substantial revenue and to make it the India + 1 choice naturally in the world. [12] Taking these targets into account, it is utmost essential to build the quality and performance of product engineering teams, whom comprise a substantial level of software engineers. Sri Lanka would have to transform its human capital to adapt and new practices and technologies and further encourage brighter and smarter individuals to join the industry in the years to come. Hence it is very vital to uplift the software engineering education among universities and also uplift training and development departments within the industry. [12]

Hence the purpose of this study is to establish the essential software engineering competencies which would define a successful software engineer within the Sri Lankan software development industry, focused on more agile and product development focus. This has been identified as a research gap and the main uniqueness of this study. Therefore this study would solve the following research questions related to software engineering;

- What are the most essential software engineering technical competencies for today's undergraduates
- What are the gaps between current and future expected level of competencies among undergraduate students
- Perception of training and inductions among undergraduates while entering industry

Hence it would bring many benefits for current software engineering undergraduates within the community to benchmark their technical competencies. Further it would support the industry and as well as academia to evaluate and uplift its course / training curriculums related to software engineering.

The next section of this research paper reviews the relevant literature in brief. It would be followed by the research methodology. Thereafter, the results and analysis section would present the findings in details. Finally the conclusions are presented based on the findings and recommendations and limitations are discussed.

2. BACKGROUND

A competency is defined as a capability or ability. It is a set of related but different sets of behavior organized around an underlying construct, which we call the "intent". The behaviors are alternate manifestations of the intent, as appropriate in various situations or times [14]. Vertically, competency framework is a tool to delineate job-level competencies and firm-level capabilities from the mission and strategy of the organisation. Horizontally, competency framework can be used for different purposes in human resource management (HRM), such as selection, appraisal, development and reward [15]. Therefore, a competencybased approach, on the one hand, guides to identify competency requirements of current and future human resource needs in alignment with organisational strategies, and on the other hand, it focuses on individual and team development plans to eliminate competency gaps in terms of those requested by a project, job role or enterprise strategy [16].

When we compare engineering with software engineering, both of these fields are distinct from each other. In engineering safety and economical aspects for their use is paramount as they are about building things and infrastructure. Safety in engineering is considered highly more than the usage [17]. Software engineering thus is different in nature from other engineering disciplines, due to both the intangible nature of software and to the discrete nature of software operation [17].

Hence differences software engineering and engineering are as follows; [17].

• Oppose to foundations been on natural sciences, computer science is the primary focus.

• Continuous mathematics is the focus and not discrete

• Concrete/physical artifacts are not considered and the concentration is on abstract/logical entities.

• There is no "manufacturing" phase in the traditional sense.

• Software "maintenance" primarily refers to continued development, or evolution, and not to conventional wear and tear.

In a recent study conducted, the researcher's purpose of building a competency framework was to support software engineers to develop and to advance in their work during their career with in the company. To face to the current challenge, software organizations needed better ways of developing the skills of their software engineers to solve complex mental tasks. The competencies involve the mobilization of practical and cognitive skills, creative skills, and other behavioral skills like attitudes, motivation and values [2].

In similar study conducted in Iran, focusing on career orientation, they have found that the career orientation can be segmented into three aspects, naming technical, managerial and entrepreneurial among the software engineers [3]. Among the technical competencies, updating knowledge and learning capability, problem solving, concentration and precision, domain specific knowledge, human relation, economical assessment. hardware competencies, organizational knowledge were identified. Among the managerial level competencies, leadership, organizational knowledge and maturity, risk taking, project management were identified. Finally among the entrepreneurial competencies, creativity and innovation, long term orientation and hardiness were identified. Hence out of this research it clearly indicates that technically oriented software engineers may need to continuously improve their technical competencies and emphasize their values for being marketable [3].

In another study conducted in Spain's software development industry in aims to identify technical competency levels relevant to Software Engineering, they found that software requirements, software design, software construction, software testing, software maintenance, configuration management, quality, software engineering management, software engineering tools and metrics, and the software engineering process stands out as most important competencies. However in their study they recommended to further pursue in identifying non-technical competencies as well, which would bring the role more holistic [18].

According to a study conducted by Raina & Pande related to Engineers in the IT sector within India, they focused more on the soft-skill competency area, there work clearly supports communication as a potential contributor in building effective interpersonal and harmonious working relationships. They claim that High levels of communication competence are important to organizations. The communication skill dimensions supported in their study are all critical thinking skills involving empathy, effective listening, adaptability and flexibility. High levels of affective, cognitive, and behavioral competence components are essential for establishing and developing strong relationships within organizational systems. [19].

A study on requirement analyst based on 64 interviews conducted among eight major North American and European financial services companies looked at addressing similar gaps in the industry, however among RA's. Ideally with this study they developed a competency model, which specifies 16 critical competencies. Out of the key competencies that they identified, 'Consulting others', 'Testing assumptions and investigating' and 'Explaining concepts and opinions' were the most frequently identified competencies. This indicates that for an effective analyst, close interaction and communication with customers is indeed crucial – but of equally importance is the critical questioning of the expressed needs. Hence the analyst role is somewhat similar to what the software engineers is performing and we could derive some of the competencies for our study as well [20].

Another study conducted among students on a Masters in Software Engineering identifies that for the future a mix of software skills aspects are essentials to improve quality among the software that are built. Hence these would be future aspects that software engineering should proceed on while maturing the industry [21]. Universities and companies therefore need to design training schemas to deal with the specifics of global software development, which are principally related to communication difficulties and time and cultural differences [22]. 7 intelligence modules to enhance the current curriculum mechanism in order to bring the levels of software education closure to Software Industry, was identified during a study and factors identified as intelligence were domain, life cycle, architecture, data and business, web, programming, application [23].

A study conducted through focus groups and interviews with software engineers and managers, have identified over 35 communication skills that comprise communication in the software engineering workplace, including a broad range of formal, interpersonal, professional, and team communication skills [24].

A research conducted in Australia found out that most of the issues in developing software engineering competencies at undergraduate level is due to the lack of an objective framework to guide student learning outcomes during projects. In this study they have developed an objective skillbased framework, focusing on managerial, engineering and personal skills. This kind of approach being adopted might be a solution for the lack of readiness of Software Engineering (SE) graduates for a professional career which is a frequent complaint raised by industry practitioners. The career succession of many new graduates is severely impacted due to the lack of all round skills pertaining to technology and soft skills. For example, some of the technically stronger graduates lack communication and behavioral skills together with management and the others the opposite. Hence the need of such framework and guidance is very much needed according to this research [25].

A study conducted in India to understand the expectations of the fresh graduates revealed that conceptual knowledge, problem solving, communication, self-learning, research aptitude has high expectations among the industry, further the industry also expected that students should be at least aware of various process models, standards such as ISO, SEI-CMM, Six Sigma and familiarity with various tools and techniques such as version control systems, visual modeling tools, defect management systems to name a few [26].

3. METHODOLOGY

To attain the objective of the research, the identification of essential software engineering competencies is the key. In formulating a list of software engineering competencies to be considered for the study, the approaches and strategies adopted by recent and similar software engineering competency research were inquired as described in the previous section of this paper. Hence, the literature on the formulation of a competencies list reveals that some create their lists concluding from previous research while some others have developed their own lists following a certain procedure.

The research carried out a four step approach to the methodology which was used to identify competencies that are essential for a successful software engineer within the software development industry as perceived by industry experts and software engineers in Sri Lanka.

First, a comprehensive list of technical competencies specific to software engineering was created that were taken from the competency literature. Second, in order to standardize the terminology of competency items, the technical competencies were summarized as they had been abstracted from various sources in the literature. This step helped to come up with the hypothesised list of competencies.

As the third step, industry experts attached to the Sri Lankan software development industry was consulted to rate and comment on the importance of each competency for software engineers from the hypothesized competency list presented. The interviews were carried out among 21 participants currently employed within the Sri Lankan software industry. The average experience among all these participants were around 9 working years and their roles spanned from Directors, Head of Delivery Units, Product and Project Managers, Architects and Technical Leads. It was decided to finalize and refine the competency list through the ratings and comments received.

Therefore, as the fourth and final step, seven software engineering technical competencies (As listed in Table 1) were identified and they were integrated into a questionnaire. This questionnaire was based on a self-evaluation method of personal competencies targeted at undergraduates who were targeting to join the software industry as software engineers and who has completed a minimum of two years of study in the undergraduate curriculum.

Competency	Software Engineering Importance
1. Programming	The ability of writing computer programs for multi platforms, devices and channels with ability to adapt with any programming language.
2. Computer Science	Ability to integrate the principles of computer science in order to produce tangible, physical artifacts

Table 1. Software Engineering Technical Competencies

3. Systems Analysis & Design	Ability to examines complicated industrial and business operations in order to find ways of improving or solving them systematically
4. Software Development Process	Ability to effectively use a software development process or life cycle which is a structure imposed on the development of a software product.
5. User Requirements	Ability to understand the expectations of the users of software and deliver them as expected
6. Software Tools Usage	Ability to use an array of software tools or build your own tools to bring productivity into tasks done in software engineering
7. Delivering Quality Code	Ability to deliver quality code adhering to best practices and principles with abstraction in mind and ensuring defects are not injected

The questionnaire was sent through emails, community groups, and discussion boards to a large pool of software engineers from the targeted sample. The data was collected over a period of four months from June, 2014 to October, 2014. The overall relevant response rate was 76% due to specific and technical nature of the survey. The questionnaire was open for all to access. We scanned data of 195 respondents, which were found more relevant and complete for evaluation of the results. SPSS was used to evaluate the respondents' results. The questionnaire was divided into four parts. It contained total 33 close ended questions to capture the relevant data. The first section, gathered demographic information of respondents, while the second section captured the current level of software engineering technical competencies possessed among the sample. The third section captured the future needed level of software engineering technical competencies that were needed to be possessed from the sample set of undergraduates who aspire to be software engineers. Section four was user to gather data on their perception on how future fresh software engineers could be trained and inducted into the position in a smooth manner.

Five-point Likert scale was used to establish the level of current competency expertise and the level of competency requirements for future success at software engineering level. In the current competency expertise level: the five-point Likert scale used to establish the level of competencies currently possessed and denoted as: 5, Very High; 4, High; 3, Average; 2, Low; and 1, Nothing at all. In the current expertise, a respondent estimates his/her competency on each of the 7 technical competencies.

Competency requirements for future success was denoted as: the five-point Likert scale used to establish the level of competency importance for future success is: 5, Very High; 4, High; 3, Average; 2, Low; and 1, No Need. In future requirement, a respondent makes a judgment on how important each of the 7 technical competencies will be for the future job performance (3 years to the future).

The list of 7 software engineering technical competencies rating both current expertise and future importance of each competency were incorporated into the online survey questionnaire for pilot testing. The pilot results were carried out among 15 respondents and was useful in fine tuning the survey questionnaire.

Descriptive statistics (mean and standard deviation) were used to analyse scores and to derive gaps at the software engineering level. Paired sample t-test was performed to explore differences between the current expertise level and future importance level in order to identify the most needed competencies. Factor analysis was used to derive component factors and for any data reduction. Cronbach's alpha values of each factor extracted and overall measure was greater than 0.7 which denoted that the questionnaire was reliable. Inter-item correlation, Kaiser-Meyer-Olkin(KMO) and factor loadings were generated for all sections of the data analysis to derive meaningful measures and analysis.

4. RESULTS AND ANALYSIS

Table 2 presents the demographic profile of the respondents. As it presents the majority of the sample were software engineers who were within the industry for less than one and a half years. Majority of companies are following scrum based agile methodologies.

Fable 2. Demogr	aphic Profile	of Respondents
-----------------	---------------	----------------

Variable	Frequency	Percent		
Undergraduate Specialization				
Computer Science	20	10.3%		
Information Technology	125	64.3%		
Software Engineering	45	23.7%		
Computing	5	2.6%		
Gender				
Male	125	64.1%		
Female	70	35.9%		
Marital status				
Single	195	100%		
Age				
20 to 23	115	59.0%		
24 to 27	70	35.9%		
31 and above	10	5.1%		
Undergraduate Internship				
Yes	135	69.2%		
No	60	30.8%		
Programming Certifications		76.000		
None	150	/6.9%		
Java Language	20	10.3%		
Microsoft .Net Language	25	12.8%		
Lifecycle Approach Used				

Scrum	5	2.6%
Waterfall	25	12.8%
XP	5	2.6%
Prototyping	115	59.0%
None	10	5.1%
Iterative	30	15.4
Rational Unified Process	5	2.6%

Table 3 presents the inter item correlation for the questions raised for software engineering's current competencies. In this construct there are 7 items. Those are the 7 competencies which were needed to be rated by the undergraduates. Each item is measured on a Likert scale of 1 to 5, where a response of 1 indicates disagreement while a response of 5 indicates agreement to the statement. The descriptive statics and the inter-item correlation value are given in table 3. Cronbach's Alpha was at 0.81.

Based on the mean values in table 3, there seem to be an average level of competency in all 7 statements on current competency ratings. Most participants has chosen that they are at average level currently and it shows signs that there is room for improvement. The highest correlation for each item with at least on other item in the construct is between 0.3 and 0.9. Hence all the 7 items do not correlate as required in the constructs. Generally all technical competencies have short comings from the current level of rating by the undergraduates. This ratings assure us that the current level within their scope is somewhat low. The Kaiser-Meyer-Olkin (KMO) value is .712, which is considered to be good. Two factors were extracted that explained 47.65 for first factor, 20.19% for second factor the variation in the 7 items. The smallest factor loading is 0.579. (> 0.5 norm). The mean of the 7 items was computed and saved as future competencies to be used in further analysis.

Table 3. Inter Item Correlation for Current Level of Software Engineering Technical Competencies

Item	Descriptive Statistics		Inter-item correlation						
	Mean	S	Q1	Q2	Q3	Q4	Q5	Q6	Q7
Q1	3.44	.673	1.000	.285	.162	.154	.048	.283	.515
Q2	3.56	.592	.285	1.000	.544	.334	.360	.533	.419
Q3	3.62	.838	.162	.544	1.000	.624	.735	.390	.266
Q4	3.64	.769	.154	.334	.624	1.000	.765	.204	.322
Q5	3.77	.734	.048	.360	.735	.765	1.000	.224	.354
Q6	3.36	.661	.283	.533	.390	.204	.224	1.000	.412
Q7	3.33	.694	.515	.419	.266	.322	.354	.412	1.000

This analysis clearly identifies the current level of competencies possessed by the software engineering undergraduate students in this sample. Clearly there is room for improvement within them and none of them have highlighted they do possess every competency listed. Since the items didn't correlated, the factor loading performed for the same data set, listed in table 4 shows us that clearly the competencies could be factored into 2 components. Component 1 was identified as the process bound as it contained all the technical related competencies related to more process and methodologies related and the other 2 components as the core technical level.

Table 4. Factor Loading for Current Level of Software Engineering Competencies

Competencies	Current Compete Clusters	Current Competency Clusters			
	Process Bound	Core Technical			
User Requirements	.960				
Software Development Process	.884				
Systems Analysis and Design	.839				
Programming		.849			
Delivering Quality Code		.772			
Exposure to Software Tools		.695			
Computer Science Theories		.579			

While analyzing the results of the future competencies which are of importance, in this construct there were 7 items as similar to previous analysis. Each item is measured on a Likert scale of 1 to 5, where a response of 1 indicates disagreement while a response of 5 indicates agreement to the statement. The descriptive statics and the inter-item correlation value are given in table 5. Based on the mean values in table 5, there seem to be a strong agreement in all 7 statements on future competency levels. Most participants has chosen that they need these 7 competencies very much in the future, to build a successful software engineering career. None of the respondents had thought that these competencies would not be useful to them in the future. We can clearly see that the mean values for these 7 items are higher in terms of the current competencies levels identified in the previous analysis. Hence we could solidly state that the competencies identified as essential for software engineering in the current context of Sri Lankan industry has been endorsed by this sample set of software engineering undergraduate students as as well. Cronbach's Alpha for this analysis was 0.93.

The highest correlation for each item with at least on other item in the construct is between 0.3 and 0.9. Hence all the 7 items correlates adequately in the construct. The Kaiser-Meyer-Olkin (KMO) value is .869, which is considered to be excellent. A single factor was extracted that explained 73.63% of the variation in the 14 items. The smallest factor loading is 0.502. (> 0.5: norm). The mean of the 7 items was computed and saved as future competencies to be used in further analysis.

Item	Descriptive Statistics		Inter-item correlation						
	Mean	S	Q1	Q2	Q3	Q4	Q5	Q6	Q7
Q1	4.21	.792	1.000	.502	.510	.595	.647	.672	.604
Q2	4.03	.699	.502	1.000	.753	.722	.745	.735	.640
Q3	4.21	.724	.510	.753	1.000	.793	.662	.645	.535
Q4	4.28	.751	.595	.722	.793	1.000	.791	.725	.805
Q5	4.18	.782	.647	.745	.662	.791	1.000	.813	.817
Q6	4.21	.792	.672	.735	.645	.725	.813	1.000	.757
Q7	4.28	.848	.604	.640	.535	.805	.817	.757	1.000

Table 5. Inter Item Correlation for Future Level of Software Engineering Competencies

Hence the first research question of the study could be satisfied by understanding the required competencies for a successful software engineer. With background theory and further ratings and comments from the industry panel, these competencies were cross validated with a sample of software engineering students in order to validate the research output thoroughly. To fulfill the second research question of the study, the competency gaps were measured. The size of the competency gap was determined by measuring the difference between the level of competency currently possessed and the level of of competency requirement software engineering competencies for future success. Each competency item was measured with the response of each participant. As a result we found three types of competency gaps, that emerged from the results: they were a negative gap (the present competency level was lower than required: value less than or equal to -0.51), an adjustment margin (the present competency level is lower but close to that required: between -0.5 and value 0.5) and a positive gap (the present competency level is higher than that required: value higher than 0.51). Though three different situations could be obtained from the subtraction, in this study focused on negative gaps and adjustment margin, since those are the needs. Table 6 presents the analysed data.

After the analysis all the technical competencies showed negative gaps. This analysis further strengthens our research question and overall objective that the current Sri Lankan software engineering practioners who are new in the industry needs adequate trainings, guidance and mentoring to be successful engineers. From sheer experience, if these software engineers are not properly performance appraised and given proper objectives, they would also fit into the mediocre level software engineering category, which is not expected. Hence this is an alarming note for training, human resource, senior level member of the software industry as well as the academic to re-focus on building the capabilities of the software engineers by giving them a holistic education and training.

Table 6 Software Engineering Competency Gaps

	Descriptive Statistics					t-Test (two-tailed)d			
Competencies	Current Expertise Mean (SD)	Future Importance Mean (SD)	Perceived Gap Mean (SD)	Type of Gap	t	df	Significanc e		
1. Programming	3.44 (.673)	4.21 (.792)	769	Ν	-13.975	194	.000		
2. Computer Science	3.56 (.592)	4.03 (.699)	462	Ν	-7.629	194	.000		
3. Systems Analysis & Design	3.62 (.838)	4.21 (.724)	590	Ν	-10.171	194	.000		
4. Software Development Process	3.64 (.769)	4.28 (.751)	641	Ν	-11.157	194	.000		
5. User Requirements	3.77 (.734)	4.18 (.782)	410	Ν	-6.813	194	.000		
6. Exposure to Software Tools	3.36 (.661)	4.21 (.792)	846	Ν	-15.321	194	.000		
7. Deliver Quality Code	3.33 (.694)	4.28 (.848)	949	Ν	-14.611	194	.000		

Table 8 Descriptive statistics – Entry Level Trainings

	Current Comp Clusters	Current Competency Clusters				
	Mean	SD	Skewness	Kurtosis		
Training on Programming Skills	4.36	.769	717	954		
Training on Architectural and Design Practices	4.15	.771	272	-1.270		

International Journal of Computer Applications (0975 – 8887) Volume 113 – No. 17, March 2015

Training on Software Tools used for Projects	4.26	.777	482	-1.189
Training on Company Process	4.28	.716	476	942
Training on Coding Standards and Best Practices	4.33	.655	473	710
Leadership Inductions	4.44	.711	860	553
More Inductions on Industry Practices	4.31	.607	272	622
1 - 2 weeks Period to adapt to the Industry	4.00	.786	-1.285	3.623
Individual Mentor been Assigned	3.97	.769	.044	-1.304
Specific Period of Class Room Training prior to Project Allocation	3.85	1.029	547	247
Soft Skills Training	3.97	.893	388	803
Session on Career Path at Company / Industry	4.23	.698	350	916

To explore on the third research question of this study, we asked 12 specific questions from the undergraduates about their opinion on entry to the software engineering industry and how they perceive the training and inductions. The 12 items that were questioned by the software engineering undergraduates could be viewed in table 8, which presents the descriptive statistics in a detail manner. Considering software engineering is a long oriented career rather than a job, giving morale level support from the very beginning is crucial. Hence the questions raised were focused on it and questioned the sample their perception of entering the industry. As per the results it is evident that the sample set of undergraduates strongly agree on most of the items that were raised as required inductions and trainings. Their perception would be that these kind of sessions would be available at entry and also would help them transition successfully. Taking one aspect as an example, configuration management as a function is sometimes not practiced or learned within the undergraduate course. Effective use of tools and practices for configuration management might be challenging at the start, hence getting the software engineers used to them is of importance. Coding standards, test driven development approach, continuous integration & deployment, best practices in object oriented languages and company processes might be somewhat hard to learn on the job, while performing some client or internal expectations. Hence this indicates that it is essential to identify skills, knowledge that are needed to be provided at a separate level. These would contribute to adding competencies at software development process, exposure to software tools and quality delivery of code.

Ideally with this analysis it gives a strong message to industry specific trainers and also the academia on the level of perception the undergraduates have. Also it gives another meaning that these undergraduates might not have much self confidence at this level. However these may not be possible within a couple of weeks, but it would be recommended to have these within a defined period of time.

5. CONCLUTION

The study focused the most essential software engineering technical competencies that are required for a software engineer. The study focused on the Sri Lankan software industry and gathered the analysis looking at the undergraduates who are aspiring to be future software engineers. The main aim was to identify the most essential software engineering competencies which would help software engineers to be successful in their careers. Through this research it is now evident that focusing on human aspects in software engineering is essential. Not only focusing on merely building processes, tools and techniques in this field, it is vital that software engineering performance is based and appraised on a competency focused approach.

The research showed that existing studies in a similar context are pretty much lesser. Even though the western world have fewer of these studies conducted, the Asian region has considerably lower. Hence it is important further to embark on research related to human aspects related to software engineering.

Since the study sampled against the Sri Lankan software industry, it is also evident that for Sri Lanka to achieve its goals and objects in 2022 as an industry unit, it would have to build higher level of graduates who are industry ready as well as able to perform at global competitive levels. The identified 7 technical competencies are focused on solidly building and guiding the careers of software engineers. Further it would help industry trainers and decision makers to base their selection, recruitment, performance appraisals and continuous development through the identified framework as it would be a good stepping stone. Further it would facilitate academia to refocus on their syllabi and engage with the industry to build effective software engineering curriculum which would eliminate the gaps at undergraduate level. These findings would not be easy to implement within a short period, however could be taken as guides and implemented in short, medium and long term levels to uplift the education.

Identified limitation of the study was that the research could adopted a qualitative method to analyze and identify the competencies at the software engineer level. However with a large sample, it was not possible to interview software engineering undergraduates and to limit any discrepancies, the study focused on interviewing a sample of industry experts to understand their perceptions of the needed competencies. Thus as future research, one could look at a more qualitative study at identifying the competencies within software engineers as well. Further as future work, the behavioral or soft skills aspects needed from software engineering students at undergraduate levels also could be analyzed to identify the current gaps and future expectations. With this type of study, we could clearly identify the perceptions of such undergraduates within the university level. Further identifying improvements at both academic level and even at industry level trainings and inductions would be analyzed with a study to gather very important points from a large population.

6. REFERENCES

- [1] Liu, S. (2013, October). Engineering the Success of Software Development. IT Pro IEEE, pp. 4 5.
- [2] Rivera-Ibarra, J. G., Rodríguez-Jacobo, J., & Fernández-Zepeda, J. (2010). Competency Framework for Software Engineers. 23rd IEEE Conference on Software Engineering Education and Training (pp. 33 - 40). IEEE Computer Society.
- [3] Alavi, S. B., Moteabbed, S., & Arasti, M. R. (2012). A qualitative investigation of career orientations of a sample of Iranian software engineers. Transactions D: Computer Science & Engineering and Electrical Engineering, 662 - 673.
- [4] Graziotin, D., Wang, X., & Abraham, P. (2014, August). Software Developers, Moods, Emotions, and Performance. IEEE SOFTWARE, pp. 24-27.
- [5] Pressman, R. (2014). Software Engineering: A Practitioner's Approach. New York: McGraw-Hill.
- [6] Ahimbisibwe, A., Cavana, R. M., & Daellenbach, U. (2015). A contingency fit model of critical success factors for software development projects A comparison of agile and traditional plan-based methodologies. Journal of Enterprise Information Management, 7 - 33.
- [7] Boehm, B. (1981). Software Engineering Economics. Englewood Cliffs: Prentice-Hall.
- [8] Boehm, B., Horowitz, B., Madachy, E., Reifer, R., & Clark, D. (2000). Software Cost Estimation with COCOMO II. Upper Saddle River: Prentice Hall.
- [9] Capretz, L. F. (2014, April). Bringing the Human Factor to Software Engineering. IEEE SOFTWARE, pp. 101 -103.
- [10] Misra, S., Kumar, V., Kumar, U., Fantazy, K., & Akhter, M. (2012). Agile software development practices: evolution, principles, and criticisms. International Journal of Quality & Reliability Management, 972 - 980.
- [11] Manawadu, C. D., Gapar, M. J., & Perera, S. S. N (2013). An Evaluation of Software Development Methodology Adoption by Software Developer in Sri Lanka. *International Journal of Computational Engineering Research*.
- [12] SLASSCOM. (2014). Sri Lankan IT/BPM Industry 2014 Review.
- [13] National ICT Workforce. (2013, November). National ICT Workforce - Survey Final Survey Report.
- [14] Boyatzis , R., & Ratti, F. (2009). Emotional, social and cognitive intelligence competencies distinguishing

effective Italian managers and leaders in a private company and cooperatives. Journal of Management Development, 821 - 838.

- [15] Boyatzis, R. E. (2009). Competencies as a behavioral approach to emotional intelligence. Journal of, 749 - 770.
- [16] Robert , J., & Boyatzis, R. (2012). Emotional and social intelligence competencies: cross cultural implications. Cross Cultural Management: An International Journal, 4 - 18.
- [17] SWEBOK. (2004). Guide to the Software Engineering Body of Knowledge - SWEBOK. Piscataway: IEEE Press.
- [18] Colomo-Palacios, R., Tovar-Caro, E., García-Crespo, Á., & Gómez-Berbís, J. M. (2010). Identifying Technical Competences of IT Professionals: The Case of Software Engineers. International Journal of Human Capital and Information Technology Professionals, 31 - 43.
- [19] Raina, R., & Pande, N. (2012). Communication Competence of Indian Engineers in IT & ITeS Sector. The Indian Journal of Industrial Relations, January
- [20] Klendauer, R., Berkovich, M., Gelvin, R., Leimeister, J., & Krcmar, H. (2012). Towards a competency model for requirements analysts. Information Systems Journal, 475 - 503.
- [21] Louise, I., Seidman, S., Pattinson, B., & Delaney, Y. (2011). Educating Software Engineers of the Future: Software Quality. CSEE&T (pp. 91 - 100). Honolulu: IEEE.
- [22] Monasor, M., Vizcaíno, A., Piattini, M., & Caballero, I. (2010). Preparing Students and Engineers for Global Software Development: A Systematic Review. International Conference on Global Software Engineering (pp. 177 - 186). IEEE.
- [23] Subrahmanyam, G. (2009). A dynamic framework for software engineering education curriculum to reduce the gap between the Software organizations and Software educational Institutions. Conference on Software Engineering Education and Training (pp. 248 - 254). IEEE.
- [24] Ruff, S., & Carter, M. (2009). Communication Learning Outcomes from Software Engineering Professionals: A Basis for Teaching Communication in the Engineering Curriculum. ASEE/IEEE Frontiers in Education Conference. San Antonio: IEEE.
- [25] Karunasekera, S., & Bedse, K. (2007). Preparing Software Engineering Graduates for an Industry Career. Conference on Software Engineering Education & Training. IEEE.
- [26] Garg, K., & Varma, V. (2008). Software Engineering Education in India: Issues and Challenges. Conference on Software Engineering Education and Training (pp. 110 - 117). IEEE.