MEASURING THE SOFTWARE ENGINEERING COMPETENCIES: A SRI LANKAN SOFTWARE INDUSTRY PERPECTIVE

C.D. Manawadu[1], Md Gapar Md Johar[2]

Faculty of Information Science & Engineering, Management and Science University, MALAYSIA.

[1]dilhanmanawadu@gmail.com, [2]gapar@msu.edu.my

ABSTRACT

Sri Lanka an emerging nation among the global software product engineering industry strives towards nurturing the competencies of present and future software engineering practitioners and strategizes in becoming one of the top nations in the industry. Building people with the right mix of product engineering mindset and agile practice is the key challenge for Sri Lanka as well. The purpose and aim of the research is to determine the future landscape of the software product engineering industry within Sri Lanka and identify the competencies essential for software engineering practitioners. The research adopts a mixed mode study, using Grounded Theory and quantitative techniques for data gathering, analyzing and interpretations. Findings presents a significant gap within the present competencies of the software engineering practitioners and future software engineering practitioners who are currently studying as undergraduates. The solutions to overcome these challenges are within immediate and long term plans which mainly needs to be addressed by improving the practices related to software product engineering. The findings of this study would benefit the current and future software engineering students, software engineering practitioners and academics of the tertiary education system.

Keywords: Software Engineering; Competencies; Sri Lanka, Sri Lankan Software Product Engineering, Software Industry.

INTRODUCTION

Software industry evolution starts from the 1940's with the use of computing applications used in solving scientific, military and business problems. With the advent of the personal computer, the software usage has seen a drastic change and today its part and parcel of every human being's life (Ghazi, Moreno, & Peters, 2014). According to Gartner in the year 2013, the worldwide earned revenue from software totaled $407.3 billion which is a 4.8 percent increase from 2012 revenue of $388.5 billion (Pettey, 2014).

In today's world software engineering processes favor and focuses on the above mentioned facts by practicing agile software development methods which emerged in the late 1990s (Larman & Basili, 2003). A hallmark of agile software development is its focus on people and social interactions. Agile teams are meant to be democratic teams where all members are considered peers at the same level. (Martin, 2002). Agile methods follow an iterative and incremental style of development where collaborative self-organizing teams dynamically adjust to changing customer requirements (Misra, Kumar, Kumar, Fantazy, & Akhter, 2012). A research conducted within Sri Lanka identifies that most Sri Lankan software development organizations and developers in the current context use agile methodologies for the software development. (Manawadu, Md Johar, & Perera, 2013).

Exploring into the Sri Lankan context of software development, the ICT industry in Sri Lanka was established about one and a half to two decades ago, with a large number of small to medium enterprises emerging. According to the Sri Lanka Association of Software and

Leena and Luna International, Chikusei, Japan.
(株) リナアンドルナインターナショナル, 筑西市,日本
ISSN: 2186-8492, ISSN: 2186-8484 Print
www.ajssh. leena-luna.co.jp

Service Companies (SLASSCOM) there are more than 70 registered software development companies that supply products and services to offshore client firms. (SLASSCOM, 2014). Currently the workforce is around 34,000 of which 49 per cent have a Bachelor's degree or a higher qualification in IT and around 7123 working as software engineers (National ICT Workforce, 2013). 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 (SLASSCOM, 2014).

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. 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. (SLASSCOM, 2014)

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 engineerin:

- What are the most essential software engineer competencies?
- Find out the existing gap between current and future expected competencies
- How the identified gaps in competencies be eliminated in the short term?

Hence it would bring many benefits for current software engineering practioners as well as undergraduates within the community to benchmark their 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. In the final section, conclusions on the findings are drawn, the implications of the findings are discussed, and research areas for further inquiry and understanding are presented.

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. (Boyatzis & Ratti, 2009). Over the last 40 years the concept of competency has evolved into a flexible framework for the selection, assessment and development of human talent in organizations all over the world. (Robert & Boyatzis, 2012). The concept of competency-

based human resources has gone from a new technique to a common practice in the 36 years since David McClelland (1973) first proposed them as a critical differentiator of performance. The drive for effectiveness in organizations fuels the quest for understanding the talent and capability of the people that create or determine effectiveness. Of the many ways to address this need, competency research and applications arrived in 1970 and spread. It built upon earlier work on skills, abilities, and cognitive intelligence (Campbell et al.,1970).

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 (Sparrow 1995; Hondeghem and Vandermeulen 2000; Schippmann et al. 2000; Becker, Huselid and Ulrich 2001). Therefore, a competency-based 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 (Draganidis and Mentzas 2006).

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 is thought twice prior to usage (SWEBOK, 2004). However software engineering emerged forty years ago during a NATO scientific conference (Naur & Randell, 1968) in a response to the realization that large software systems were frequently delivered late – if delivered at all – were over budget with feature deficiencies, and had suspect quality. Software engineering thus is different in character from other engineering disciplines, due to both the intangible nature of software and to the discrete nature of software operation (SWEBOK, 2004).

METHODOLOGY

To achieve the purpose of the study, the identification of relevant software engineering competencies is the foundation. In generating the list of software engineering competencies to be considered for the study, the methodologies adopted by recent competency research were studied as described in the previous section of this paper. Hence, the literature on the generation of list of competencies reveals that some build their lists drawing from previous research while some others have developed their own lists following a certain procedure.

For the current research, a four step methodology 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 competencies specific to software engineering and behavioral attributes related to software development was created that were taken from the competency literature. Second, in order to standardize the terminology of competency items, the technical and soft-skill competencies were summarized as they had been abstracted from various sources in the literature. Grounded theory concepts were used to develop these competencies (Glaser, 1992). The identified competencies were coded and common set of competencies were identified. Analyzing the data sources and generating core concepts from what are disparate sources helped refine them. Grounded Theory methods of constant comparison and memoing are employed to ensure a rigorous data consolidation technique. This step helped to come up with the hypothesised list of competencies.

Leena and Luna International, Chikusei, Japan.
(株) リナアンドルナインターナショナル, 筑西市,日本
ISSN: 2186-8492, ISSN: 2186-8484 Print
www.ajssh. leena-luna.co.jp

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, 14 software engineering 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 fresh graduates who joined the software industry recently and who had total post-graduation experience of less than one and a half years.

Table 1. Software Engineering Competencies

Competency
1. Programming
2. Computer Science
3. Systems Analysis & Design
4. Software Development Process
5. User Requirements
6. Software Tools Usage
7. Delivering Quality Code
8. Problem Solving
9. Innovativeness
10. Quick Learner
11. Can do Attitude
12. Communication
13. Research
14. Team Work

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 68% due to specific and technical nature of the survey. The questionnaire was open for all to access. We scanned data of 108 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 31 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 competencies possessed among the sample. The third section captured the future needed level of software engineering competencies that was needed to be possessed from the sample set of 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 14 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 14 competencies will be for the future job performance (3 years to the future).

The list of 14 software engineering 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 24 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.

RESULTS AND ANALYSIS

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. Related to the current competency assessment, most participants have chosen that they are at average level currently and it shows signs that there is room for improvement. Can Do Attitude, Communication Skills, and Team Work has a tendency to be higher at even this level. This can be due to these software engineers being within the industry for few years, as well as completed graduation. This analysis clearly identifies the current level of competencies possessed by the software engineers in this sample. Clearly there is room for improvement within them and none of them have highlighted they do possess every competency listed.

While analyzing the results of the future competencies which are of importance, in this construct there were 14 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. There seem to be a strong agreement in all 14 competencies on future competency levels. Most participants has chosen that they need these 14 competencies very much in the future, to build a successful software engineering career.

Leena and Luna International, Chikusei, Japan.
(株) リナアンドルナインターナショナル, 筑西市,日本
ISSN: 2186-8492, ISSN: 2186-8484 Print
www.ajssh. leena-luna.co.jp

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 14 competencies 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 engineers as well.

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 engineer 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 established by measuring the difference between the level of competency currently possessed and the level of competency requirement for future success.

Table 2. Software Engineering Competency Gaps

Competencies	*Descriptive Statistics*				*t-Test (two-tailed)d*		
	Current Expertise Mean (SD)	*Future Importance Mean (SD)*	*Perceived Gap Mean (SD)*	*Type of Gap*	*t*	*df*	*Significance*
1. Programming	3.85(.807)	4.59(.684)	-.741	N	-7.636	107	.000
2. Computer Science	3.33(.773)	4.41((.737)	-1.074	N	-15.512	107	.000
3. Systems Analysis & Design	3.41(.786)	4.48(.742)	-1.074	N	-13.664	107	.000
4. Software Development Process	3.59(.684)	4.56(.688)	-.963	N	-13.381	107	.000
5. User Requirements	3.67(.723)	4.44(.740)	-.778	N	-8.781	107	.000
6. Exposure to Software Tools	3.67(.865)	4.56(.688)	-.889	N	-8.628	107	.000
7. Deliver Quality Code	3.78(.631)	4.63(.620)	-.852	N	-9.464	107	.000
8. Problem Solving	3.81(.614)	4.70(.600)	-.889	N	-12.475	107	.000
9. Innovativeness	3.78(.688)	4.67(.611)	-.889	N	-10.510	107	.000
10. Quick Learner	3.93(.720)	4.63(.620)	-.704	N	-9.567	107	.000
11. Can do Attitude	4.15(.653)	4.63(.620)	-.481	AM	-7.897	107	.000
12. Communications Skills	3.93(.720)	4.52(.690)	-.593	N	-8.359	107	.000
13. Research Skills	4.07(.770)	4.59(.627)	-.519	N	-6.120	107	.000
14. Team Work	4.04(.842)	4.70(.659)	-.667	N	-10.344	107	.000

The gaps were calculated for each respondent for each competency item. Three types of competency gaps were identified: negative gap (the present competency level is lower than required: value less than or equal to - 0.51), adjustment margin (the present competency level

is lower but close to that required: between -0.5 and value 0.5) and 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 2 presents the analysed data.

After the analysis all the competencies showed negative gaps except one competency which was "Can do Attitude". The implication here would be that usually the software engineers would possess such a can do attitude from the start of their career journey as it is something that would have already inspired them to join this challenging and rewarding industry. However still it is within the adjustment margin (AM), which means still a gap exist with a considerable value. 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 engineer. 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.

CONCLUTION

The study focused on software engineering competencies. 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 14 competencies are focused on technical as well and behavioral aspects which would solidly build and guide 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 engineers 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 perception levels 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.

Leena and Luna International, Chikusei, Japan.
(株) リナアンドルナインターナショナル, 筑西市,日本
ISSN: 2186-8492, ISSN: 2186-8484 Print
www.ajssh. leena-luna.co.jp

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.

REFERENCES

[1]. Boyatzis, R. E. (2009). Competencies as a behavioral approach to emotional intelligence. *Journal of*, 749 - 770.

[2]. Ghazi, P., Moreno, A. M., & Peters, L. (2014, February). Looking for the Holy Grail of Software Development. *IEEE Software*, pp. 93 - 95.

[3]. Glaser, B. (1992). *Basics of Grounded Theory Analysis.* California: Sociology Press.

[4]. Larman, C., & Basili, V. (2003). Iterative and Incremental Development: A Brief History. *Computer*, 47-56.

[5]. Manawadu, C., Md Johar, M., & Perera, S. (2013). An Evaluation of Software Development Methodology Adoption by Software Developer in Sri Lanka. *International Journal of Computational Engineering Research*.

[6]. Martin, R. (2002). *Agile Software Development: Principles, Patterns, and Practices.* Pearson Education.

[7]. 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.

[8]. National ICT Workforce. (2013, November). National ICT Workforce - Survey Final Survey Report.

[9]. Pettey, C. (2014, March 31). *Gartner Says Worldwide Software Market Grew 4.8 Percent in 2013.* Retrieved from Gartner : http://www.gartner.com/newsroom/id/2696317

[10]. Ramasubbu, N. (2014). Governing Software Process Improvements in Globally Distributed Product Development. *IEEE Transactions on Software Engineering*, 235 - 251.

[11]. Robert, J., & Boyatzis, R. (2012). Emotional and social intelligence competencies: cross cultural implications. *Cross Cultural Management: An International Journal*, 4 - 18.

[12]. Slasscom. (2014). Sri Lankan IT/BPM Industry 2014 Review.

[13]. 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.

[14]. Swebok. (2004). *Guide to the Software Engineering Body of Knowledge - SWEBOK.* Piscataway: IEEE Press.