An Empirical Study on the Current Practice in Software Project Management – the Requirement Specifications

Yusmadi Yah Jusoh, Nor Hidayah Zainal Abidin, Che Suhana Che Wil,
Fatimah Abdullah Moalim, Nurul Huda Wahab and Noraini Che Pa

Abstract—Requirements specification is one of the most crucial processes in software development projects. Without well-specified requirements, the project manager could not planning and design a good project and team members could not understand what and how to do and things get worse when the user may not know what can be expected from the project. The objective of the study is to identify factors that influence the specification requirements associated with the successful of project development from the literature review and to verify the best factors identified through the real-world practice. Identifying such factors can help companies around the world in an effort to improve the quality of software development, especially in the specification requirements that lead to the success of project management. To deal with research question, we use the literature review and empirical survey approach. The result shows the factors that most significantly affect the requirements specifications is effective requirements communication chain and the lowest factor is the defined project/application domain. This study expected to address some of the factors that affect the specification requirements that is identified can help practitioners to develop a good strategy in the planning for the project success.

Keyword—Requirement Specification, Requirement Engineering, Project success Factor

The authors are with the Department of Software Engineeringand Indormation Systems, Faculty of Computer Science and Information Technology, Universiti Putra Malaysia, 43400 Serdang, Selangor, Malaysia, yusmadi@upm.edu.my, nhidayah.zainal@yahoo.com, csuhana@gmail.com, fatima.m369@gmail.com, hudawahab86@gmail.com, norainip@upm.edu.my

I INTRODUCTION

Requirements engineering (RE) and its importance have been recognized in the software industry (Berry et al., 2005). From the past study mentioned that the quality of the requirement could affect the successful of the software development (Tamai and Kamata, 2009). However, it still has some practitioners or decision-makers in businesses or institutions still do not realize the importance of the requirements specification in IT development practices (O. Skroch, 2010).

Requirements engineering (RE) applies starting from the initial software development although it is still considered unofficial to help identify and propose the selection of the right solution and the technology needed for the development of the software. To make it more useful, it needs to be documented and organized neatly in the document. Documenting the requirement is not only the process of writing the needs as seen by the user. Requirements specification is a vital link in the overall design of the software and tries to give meaning to the overall goal of software development and provides a comprehensive description of the implementation purpose of the software and environment for software under construction. The requirement specifications document fully describes what will be done and how it is expected to be implemented and also always expressed in precise and explicit language functions and capabilities of the system software must be provided, as well as any constraints which the system must remain in it. According to (Niazi, Mahmood, et al., 2016) believe that requirement specification is important because it is an official statement of the system requirements for all stakeholders. It serves as an action plan and also a bilateral basis to make sure that both the client and the organization understand the needs of software from a particular perspective on the point and specific times. The system requirements evolve over time, at any stage of software development. This information will be documented and updated in the specification requirement document.

The software development process basically starts with defining the requirements specified by the basic principles as a decisive foundation and driver of development activities.

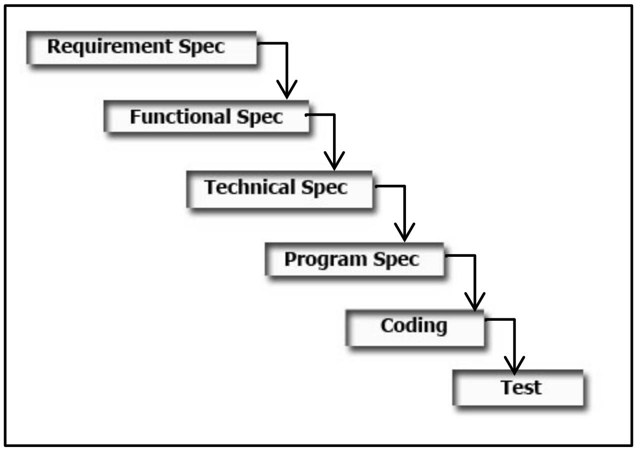

Figure1: Sequences of Development Activities (Graham, et al., 2008).

Requirements specification becomes an input to the functional specification activities and subsequent activities such as, technical specifications, program specifications, coding and testing. When the requirements specification does not meet the needs, specifications or certain standards, it will affect the subsequences process in the development phase and as a whole will give more impact in the software development.

A studies conducted recently identified has shown several factors that have been proven to give an impact to the success of requirements specification and will be elaborated in this study. In addition, the findings of the study conducted also will be elaborated further. We also identify the success factors through a systematic literature review (SLR) and verify through a questionnaire in the real-world practices. Identifying these factors through the study of literature review in advance helps us to achieve the research objective in real-world practice today. Our ultimate focus in this study is to identify the best factors affecting specification requirements for determining a successful software development. To illustrate the success factors for creating an effective requirement specification, we made some research on the sample drawn from different sectors including construction, education, government and public administration, finance and insurance, health care and social assistance, hotel and food services, transportation and warehousing, manufacturing and other sectors that related to software construction.

II. RELATED WORKS

Requirements Engineering in software projects is considered as the vital process in software development (Verner et al., 2005; Hull et al., 2010; Zakaria et al., 2011; Robinson and Vlas, 2015; Ibrahim and Darwish, 2015; Bormane et al., 2016; Abbas, 2016). Requirements are the driving force behind the development of a software project (Iqbal et al., 2012). According to Iqbal et al. (2012) and ur Rehman et al. (2013), each phase in software development like analysis, design and testing etc., directly or indirectly, depends on the requirements.

A problem in the software requirements specifications can directly impact project success, since it forms a basis for subsequent activities, affects the design of the system architecture and contributes to the quality of the software (Wiktorin, 2003; Hull et al., 2010). Most of the causes of project failures, cost and schedule overrun are often traced back to requirement engineering issues such as requirement creep, poorly documented requirements, requirements that are impossible to comply with (inverse requirements) and requirements that remained futile to meet the user needs (Pohl, 2010; Iqbal et al., 2012; Kalinowsk et al., 2016; Fernandez et al., 2016). In 2009, Tamai and Kamata conducted a study to analyze the relations between requirements specification quality and project success. They investigated 32 projects started and completed during the period of 2003 - 2005 in a large business application software development division of a company in Tokyo. Various statistical analysis techniques over the software requirements specifications quality data and project outcomes (Tamai and Kamata, 2009). Some interesting relations between requirements quality and project success or failure were found. (Tamai and Kamata, 2009).

Nasir et al. (2015) conducted a four-round Delphi to determine the degree to which the PMBOK can address the identified critical success factors for software projects. Nasir et al.'s (2015) study shows that the PMBOK provides a very good framework for addressing the critical factor clear requirements and specifications and has a significant impact on the software project's success, but it may not be the most effective way of doing things. The PMBOK provides a good framework for addressing the critical factors: clear objectives and goals and frozen requirement, but there are minor missing activities that may impact the software project's success (Nasir et al., 2015).

The requirements engineering practices have been defined as a key issue that affects the success rate of projects in software industry (Basharat et al., 2013). According to Sethia and Pillai (2013) and Abbasi et al. (2015), requirements elicitation is an important activity in the requirement engineering. Requirements elicitation begins with identifying stakeholders of the system and collecting raw requirements from various viewpoints (Pandey et al., 2010; Abaasi et al., 2015; Bormane et al., 2016;). Stanley and Uden (2013) argued that central to the problems in project failure is that there are many different stakeholders involved in the project and each often has conflicting interests. Stakeholder conflict is the biggest challenge in requirement process (Asghar and Umar, 2010; Zakaria et al., 2011, Bormane et al., 2016). Mauger et al. (2010) and Bormane et al. (2016) states that the requirements elicitation process includes these five principal types of activities: understanding of the application domain, identifying the sources of requirements, analysis of stakeholders, selecting techniques, approaches and tools, and eliciting the requirements of stakeholders and other sources.

Quispe et al. (2010) performed a diagnostic study to identify requirements engineering practices in very small software enterprises in Chile. The study consists of a survey and a focus group that were periodically conducted with experienced project managers of VSSEs. The project's scope expands as clients require additional changes, often with inadequate changes (Quispe et al., 2010).

The software project should have a proper identified scope with clear objectives and goals to be successful (Nasir and Sahibuddin, 2011; Atkins, 2012). Atkins (2012) conducted a survey of project managers and requirements engineers to determine what skills, qualifications, or experiences correlate with project success. The survey for project managers involved 10 questions about the project manager's experiences and qualifications and 23 questions about each project, while the survey for requirements engineers involved 13 questions about their experiences and qualifications and 19 questions about each project (Atkins, 2012). Atkins (2012) argues that the scope of the project cannot be defined without good requirements and a project cannot succeed without good requirements.

Software development is considered to be a dynamic process where demands for changes seem to be inevitable (Sommerville, 2010). The constant changes in requirements, known as requirements volatility, during the software development causes many projects failure and some to be completed partially (Mundlamuri, 2005).

In 2012, Singh and Vyas discussed the requirements, volatility in requirements, causes of requirement volatility and then the impact of requirements volatility on project schedule, project cost, project performance, software quality and software maintenance. Singh and Vyas (2012) explored the positive implications of the requirement changes. Due to the increasing size and complexity of software systems, there is a growing demand for intelligent approaches that can help to improve the quality of requirements engineering processes (Maalej et al., 2009; Felfernig et al., 2010; Mobasher & Cleland-Huang, 2011; Renzel et al., 2013; Ninaus et al., 2014). Ninaus et al. (2014) argued that low-quality requirements are a major reason for the failure of a project. The quality of the requirements elicited during the elicitation phase of requirements engineering depends mainly on how well the requirement elicitation technique, approaches and tools are selected (Abbasi, 2013). Abbasi (2013) conducted a study to identify the attributes of elicitation techniques, projects and the stakeholders which influence the elicitation technique selection process and proposed a Fuzzy Logic based intelligent requirement elicitation technique selection model which reduces the human biasness while elicitation technique selection.

Ninaus et al. (2014) proposed INTELLIREQ, intelligent techniques for software requirements engineering. The INTELLIREQ environment is based on different recommendation approaches that support stakeholders in requirements-related activities such as definition, quality assurance, reuse, and release planning (Ninaus et al., 2014). Ninaus et al. (2014) provided an overview of empirical studies related to the INTELLIREQ environment and the business benefits. Among the major advantages that can be expected from the INTELLIREQ environment are an increased reuse of requirements, active guidance of stakeholders, increased consistency in requirements models, and reduced time efforts needed for the construction of requirement models (Ninaus et al., 2014). INTELLIREQ environment supports early requirements engineering where the major focus is to figure out and prioritize high-level requirements in software projects (Ninaus et al., 2014). There is an important number of RE tools currently available on the market but, unfortunately, existing requirements engineering tool lists do not usually provide detailed and precise information about the tools they catalogue (Carrillo et al. (2015). Carrillo et al. (2015) conducted a quantitative study to demonstrate the commonalities and differences between current requirements engineering tools. Carrillo et al.'s (2015) study can help practitioners to decide which tool is the most suitable among several alternatives, according to their particular needs.

III. RESEARCH METHODOLOGY

The study aims to analyze the critical success factors in software projects. A survey method is selected as the research strategy to identify the impact of the top 5 identified factors by Niazi et al. (2016) have on software projects success. This paper focuses on requirements specifications and its sub factors, the fourth most success factor mentioned in Niazi et al.'s 2016) study with a frequency of 41%, i.e., 48 papers.

A. *Survey Design*

The questionnaire instrument consists of four part: i) Organization Background; ii) Respondent Background; iii) Projects Background; and iv) Projects Success Factors. The questionnaire contained primarily multiple choice questions or could be answered with few words but few open questions were also included to provide the respondents with the possibility of expressing their own viewpoints. Background information about the organizations, respondents, and projects played an important role in the survey. Projects success factors part included 14 questions related to requirements specification and its sub factors. Questions about requirements specifications sub factors used likert-type scale on an ordinal scale of 5 and defined for each a maximum value (e.g., "totally agree"), a minimum value (e.g., "totally disagree"), and the middle ("neutral"). Google Forms used to create an online survey and responses collected in an online spreadsheet.

B. *Data Collection*

The online survey link was sent to IT practitioners from different organizations varied in areas, sizes, ages, etc, i.e., finance and insurance industry, education industry, and health care and social assistance industry. This resulted in 155 responses. Answering all the questions in the questionnaire was compulsory; therefore, there were not incomplete answers.

C. Data Analysis

The data collected was analyzed using frequency counting for multiple selection questions and descriptive statistics. Hence the analyzed data were gathered from an anonymously conducted survey, we are unable to validate our results with the participants which forms a threat to the validity. The survey data analysis objective is to investigate the influence requirements specifications and its sub-factors have on software project success.

IV RESULTS

We had distributed about 300 questionnaires to IT practitioners from different organizations and we got 155 responses. In this section, we presents the results from our questionnaire survey which were categorized into 4 parts.

IV.a Organization background

Among the 155 respondents, 66 (42.6%) were from finance and insurance industry followed by 34 (21.9%) from education industry, 32 (20.6%) from health care and social assistance industry, 19 (12.3%) from manufacturing industry, and others from telecommunication and construction industry.

Other than that, 54.7% respondents are from private sector. While 35.5% are from public sector and others are from semi-government and non-profit organization. The details are shown in Figure 2.

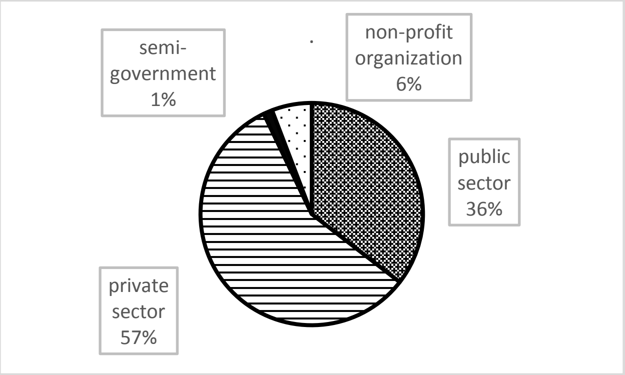

Figure2: The Category of Respondent's Organization

Most of their company are sited in various location, (65.8%). There are about 51.6% respondents are working in large scale company which is have more than 100 employees followed by 29% from small company (1-50 employees) and 19.4% are from medium company (50-100 employees).

IV.b Respondent background

A total of 32.3% respondents have 1 to 2 years' working experience, while 28.4% have 3 to 5 years' experience, followed by 21.3% less than 1 year experience, 14.8% have 5 to 10 years' experience and only 3.2% have more than 10 years' experience. About 21.3% from them are specialize in project management field. Figure 3 shows the details about respondent's specialization.

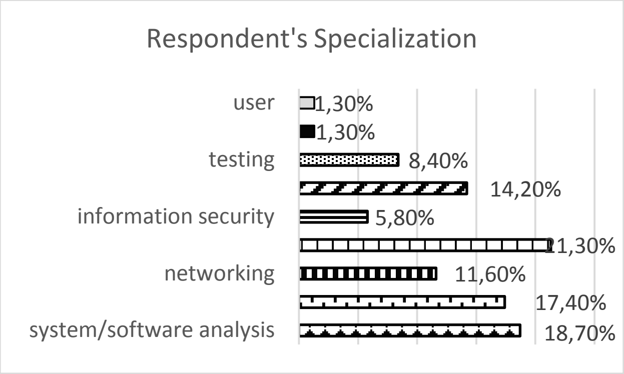

Figure 3: The percentage of respondent specialization

IV.c Project background

There are 38.1% respondents who works for new system development while others are working for redesign and enhancement of the existing system. Most of them (49%) spent 7-12 months for project development, 28.4% spent 1-6 months and 22.6% spent more than 12 months. Apart from that, 51.6% of respondent had in-house client for their project and only 48.4% focus on external client. There are 42.8% respondents have 1 to 100 clients, 20.6% have 100 to 1000 clients and 36.8% have more than 1000 clients. The result shows that they are working in large scale project.

IV.d Requirement specification

In the software engineering literature, project management and requirements management activities are crucial in order to achieve the high quality software system [44]. There are 127 (81.9%) respondents who had encounter requirement change during software development process as shown in Figure 4. The changing requirement specification can cause cost and time overrun and automatically can contribute to project delay. Project delay is unexpected adjournment of a project because of some event. Sometime, it is uncontrollable and can cause of project cost overrun. According to our survey, there are 117 from 155 respondents had encountered project delay and cost overrun.

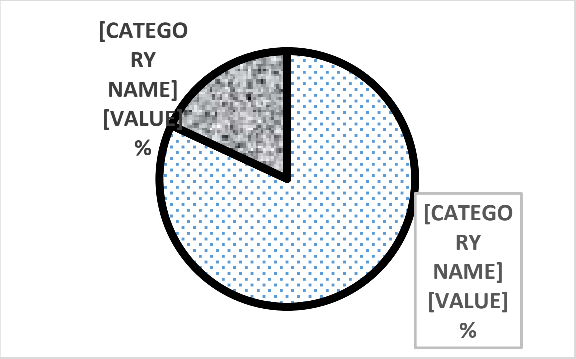

Figure 4: Number of respondents who had encounter requirements change.

In this survey, we also discovered that well-defined requirement in early stage plays a vital role in project success. There are about 65.8% of our respondents are agreed that their project requirements are well-defined in early development phase. While, other 34.2% are disagreed. A correct and complete requirement is necessary for various software projects [45]. The well-defined requirement can help project manager to decrease the percentage of requirement change.

We had identified that 34.2% of our respondents have high level authority in their project in order to achieve their project goal. This level authority gave them an ability to act on behalf of the organization or project stakeholders. The details are shown in Figure 5 below.

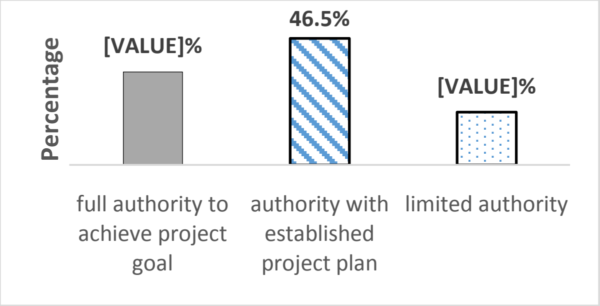

Figure 5: The authority level of respondents in project

We had categorized the requirement specification factor into 10 sub-factors. All these sub-factors are essential to project success.

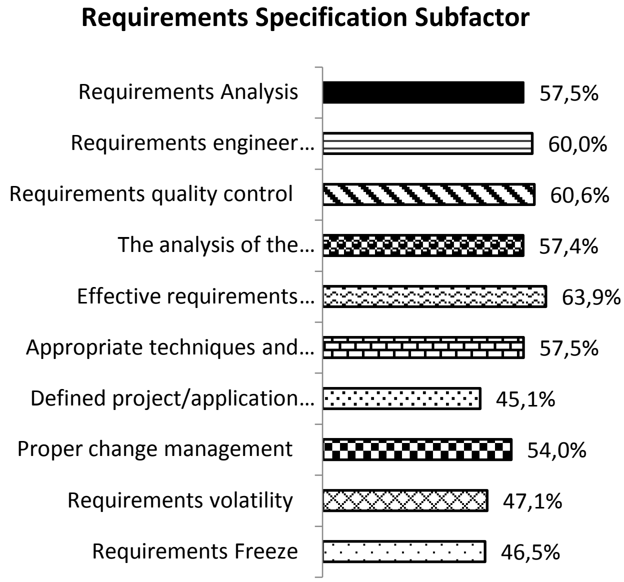

Figure 6: Requirement Specification sub-factors

From the questionnaire, there are 10 questions for respondents to rate on the Likert scale. The percentage of the requirements specifications sub-factor were derived from the scale that shows the combination of *Agree scale* and *Strongly Agree scale.* The details were shown in Figure 6. The effective requirements communication chain obtained the highest percentage which is 63.9%, where the respondents agreed that it is the main caused of project success. This is followed by requirements quality control, 60.6% respondents believes that verification and validation of the developed functionalities is critical to check whether the software system meets the required specifications.

Other than that, 60% agreed that Requirements Engineer should have sufficient skills and competencies of related knowledge. Such abilities affect and provide the sufficient support to the project success. 57.5% highlighted that appropriate technique and methodology should be applied accordingly to extract the good quality of the requirements as it will affect the project success. The same percentage shared with the facts that the identification and analysis of the stakeholders' requirements have high impact in the whole project requirements and processes. However, with only 1% differences from the previous points, performing the analysis of the stakeholders gets 57.4% agreed percentage from the respondents that it helps to determine the correct people with the correct skills to elicit the requirements.

Apart from that, 54% believes that proper change requirements management helps to plan efficient communication strategies and minimizes resistance of the changes to the requirement specification. Meanwhile, 47.1% agreed that requirements volatility is the main cause of the failure project. For requirements freeze, 46.5% agreed that it helps to maintain the stability of the software development process from the changes that may occur during the development process. Lastly, 45.1% point out that every project team members should have clear defined project/application domain as it is the key concept in meeting the customer need.

V. DISCUSSION

From the result of this study, it shows that most of the respondent agreed to the statement we have provided and it show that this happens in real situation and it should be improved.

1. Effective Requirements communication during requirements gathering is very important in order to achieve successful project status. It defined that establishing effective requirements communication chain helps the requirements engineer to control the users, prevent scope creep, run meaningful and effective meeting, and helps stakeholders to elicit the correct requirements and avoid them to deviate the objective of the requirements gathering session. Good interaction and communication with a project's clients—those who provide a project's requirements and determine its success—is essential for obtaining high quality requirements (Atkins, C., 2013).
2. An organization that practice Agile software development methodology usually will perform requirements quality control by validating and verifying the developed features are develop as per agreed requirements. As Agile team constructs the products on series of iterations, and adjust the future development direction by using the feedback from the user (Wiegers, K., & Beatty, J.,2013), the validation will be done repeatedly until all the requirements has been developed and issues has been resolved. This is the critical stage where there will be complaints, arguments and management involvements. If the situation is not being handled properly, it might be inviting the risk department interference, audit issue or even legal department involvements.
3. The requirements engineer should play the role in setting the right direction, filtering, analyzing and finalizing which requirements are “Must Have”, “Nice to Have” and “Not Recommended” or “Out of Scope”. This relate back to the knowledge, interpersonally and sufficient skills that every requirements engineer should have. According to Cynthia 2013 publication, requirements engineers with the professional certification and higher number of experience in doing IT projects are more likely to have a successful project (Atkins, C., 2013). Certified CBAP (Certified Business Analyst Professional), PMP (Project Management Professional) or others related certifications and experience helps to apply the

appropriate techniques and methodologies to extract the good requirements.

4. Performing the analysis of the stakeholders and their requirements are among the crucial part in finalizing to the agreed requirements. Cynthia mention that, lack of user/stakeholders' involvement can prevent any overlooked users, insufficient user involvement, lack of user input and incomplete requirements and and the stakeholders with no IT Project background or experience tend to elicit the requirements that follow their own personal style (Atkins, C., 2013). But the main purpose of the digital transformation is to improve the business processes and workflow while eliminating the legacy working culture. The client team need to ensure the correct person are being appointed to elicit the requirement. Else, Requirements Engineer need to put an extra effort to ensure that the requirements gathered are meeting the roadmap.
5. Proper change requirements managements have to be followed according to the industry's best practice to ensure effective communication strategies are in place while minimizing the change requirements. This requirements volatility commonly happens during the requirements specifications process as the user don't know what they want until they see the product. Requirements Volatility also can be referred to the changes of the requirements that took place during the software development that often results in growth of requirements size from the time of the initial requirements specifications to final requirements of the system development (Singh, M. P., & Vyas, R., 2012). The keep-changing requirements is dangerous and can affect many other factors such as dragging the project timeline, deviate project objectives, incur unnecessary or hidden cost and wasting time. If this is not being controlled, it will eventually cause the project failure.
6. Requirements freeze are recommended to be implemented to prevent from changes that might occur and maintain the stability of the software development process. Software Development Methodology such as Waterfall and Agile Scrum offer requirements freeze where once the requirements has been signed off, the user can't change or add any other requirements (Deemer, P., Benefield, G., Larman, C., & Vodde, B., 2010). New requirements can only be added at new sprint or maintenance stage with different project timeline, cost and man days. Therefore, this will help the project team members to have better and clear defined application domain as it is the key to meet the real customers' need.

In the Table 1, we presents the result for identified requirement specification sub-factors which were arranged in decreasing order of percentage.

Requirements Specification Sub-factors	**Percentage**
Effective requirements communication chain	63.9%
Requirements quality control	60.6%
Requirements engineer sufficient skills and knowledge	60.0%
Appropriate techniques and methodologies	57.5%
Requirements Analysis	57.5%
The analysis of the stakeholders	57.4%
Proper change management	54.0%
Requirements volatility	47.1%
Requirement Freeze	46.5%
Defined project/application domain	45.1%

Table 1: Requirement Specification sub-factors

Generally, from this study, we can understand that using the proper communications, techniques, methodologies with sufficient knowledge and skills will help in gathering the good requirements specification (Atkins, C., 2013). It also helps to prepare and overcome any unexpected outcome and issues that might occur during the process. Thus, by having good requirements, it will give high impact and being the main factors to determine the project success.

VI. CONCLUSION AND FUTURE WORK

This paper reports a research effort aimed at exploring the effects of factors that influence and affect the requirement specifications used for the software development that lead to the project success. There are 10 factors identified in the literature review that impact the requirement specification as outline in this paper.

The study was conducted on 155 respondents from various sectors that the majority from the private sector, followed by the public sector and the rest from semi-government and non-profit organizations. Most of those companies are located in various locations and can be categorized as a large-scale company. All respondents, selected from different industries such as financial and education industries, insurance industry, health care and social assistance industries, manufacturing industries, and the others from telecommunications industry and construction. The level of experience is also one of the measurement input in this study.

Based on the study, we found that the 3 out of 10 factors identified significantly affect the requirements specifications begins with the effective requirements communication chain in which the majority respondents agreed that it is the major factor of project success, followed by the requirements quality control laid for the 2nd place, and the 3rd place is requirements engineer sufficient skills and knowledge. The lowest factor selected by majority respondent is for the defined project/application domain.

This study has expected to provide useful information for further studies in requirement engineering (RE) as a project success factors in a software construction. This study recommends future work to investigate the attributes that affect the success factors that need to be identified such as level of education, level of experience, related skill and so on in the engineering requirements and find the relationships, dependency and its correlations between the attributes identified as a determinant of success or influence the requirements engineering.

REFERENCES

[1] Berry, D., Damian, D., Finkelstein, A., Gause, D., Hall, R., Simmons, E., Wassyng, A.: To do or not to do: If the requirements engineering payoff is so good, why aren’t more companies doing it? In: Proc. 13th International Requirements Engineering Conference (RE 2005), p. 447. IEEE, Los Alamitos (2005)

[2] Tamai T., Kamata M.I. (2009) Impact of Requirements Quality on Project Success or Failure. In: Lyytinen K., Loucopoulos P., Mylopoulos J., Robinson B. (eds) Design Requirements Engineering: A Ten-Year Perspective. Lecture Notes in Business Information Processing, vol 14. Springer, Berlin, Heidelberg

[3] O. Skroch, Developing Business Application Systems, DOI 10.1007/978-3-8349-8858-4_4, © Gabler Verlag | Springer Fachmedien Wiesbaden GmbH 2010

[4] Niazi, M., Mahmood, S., Alshayeb, M., Qureshi, A. M., Faisal, K., & Cerpa, N. (2016). Toward successful project management in global software development. *International Journal of Project Management*, *34*(8), 1553-1567.

[5] Bormane, L., Gržibovska, J., Berziša, S., & Grabis, J. (2016). Impact of Requirements Elicitation Processes on Success of Information System Development Projects. Information Technology and Management Science, 19(1), 57-64.

[6] Sommerville, I. (2010). Software engineering. Pearson.

[7] Mundlamuri Sudhakar, – Managing the Impact of Requirements Volatility‖, Master Thesis, 2005, Department of Computing Science, Umeå University, SE-90187 Umeå, Sweden.

[8] Mauger, C., Schwartz, T., Dantan, J. Y., & Harbouche, L. (2010, December). Improving users satisfaction by using requirements engineering approaches in the conceptual phase of construction projects: The elicitation process. In Industrial Engineering and Engineering

Management (IEEM), 2010 IEEE International Conference on (pp. 310-314). IEEE.

[9] Stanley R., Uden L. (2013) Why Projects Fail, from the Perspective of Service Science. In: Uden L., Herrera F., Bajo Pérez J., Corchado Rodríguez J. (eds) 7th International Conference on Knowledge Management in Organizations: Service and Cloud Computing. Advances in Intelligent Systems and Computing, vol 172. Springer, Berlin, Heidelberg.

[10] Verner, J., Cox, K., Bleistein, S., & Cerpa, N. (2005). Requirements Engineering and Software Project Success: an industrial survey in Australia and the U.S. Australasian Journal of Information Systems, 13(1). doi:http://dx.doi.org/10.3127/ajis.v13i1.73

[11] Hull, E., Jackson, K., & Dick, J. (2010). Requirements engineering. Springer Science & Business Media.

[12] Robinson, W., & Vlas, R. (2015). Requirements evolution and project success: an analysis of SourceForge projects.

[13] Zakaria, N. H., Haron, A., Sahibuddin, S., & Harun, M. (2011). Requirement Engineering Critical Issues in Public SectorSoftware Project Success Factor. International Journal of Information and Electronics Engineering, 1(3), 200.

[14] Dorsey, P. (2000). Top 10 reasons why systems projects fail. Retrieved February, 10, 2005.

[15] Zhu, Y. (2009). Requirements engineering in an agile environment.

[16] Abbas, J. (2016). Quintessence of Traditional and Agile Requirement Engineering. Journal of Software Engineering and Applications, 9(03), 63.

[17] Ibrahim, M. H., & Darwish, N. R. (2015). Investigation of Adherence Degree of Agile Requirements Engineering Practices in Non-Agile Software Development Organizations. IJACSA) International Journal of Advanced Computer Science and Applications, 6(1).

[18] Kumar, M., Shukla, M., & Agarwal, S. (2013, December). A Hybrid Approach of Requirement Engineering in Agile Software Development. In Machine Intelligence and Research Advancement (ICMIRA), 2013 International Conference on (pp. 515-519). IEEE.

[19] Iqbal, S., Naeem, M., & Khan, A. (2012). Yet another Set of Requirement Metrics for Software Projects.

[20] Quispe, A., Marques, M., Silvestre, L., Ochoa, S. F., & Robbes, R. (2010, November). Requirements engineering practices in very small software enterprises: A diagnostic study. In Chilean Computer Science Society (SCCC), 2010 XXIX International Conference of the (pp. 81-87). IEEE.

[21] Atkins, C. (2012). An investigation of the impact of requirements engineering skills on project success (Doctoral dissertation, East Tennessee State University).

[22] Nasir, M. H. N., & Sahibuddin, S. (2011). Critical success factors for software projects: A comparative study. Scientific research and essays, 6(10), 2174-2186.

[23] Nasir, M. H. N. M., Sahibuddin, S., Ahmad, R. B., & Fauzi, S. S. B. M. (2015). How the PMBOK Addresses Critical Success Factors for Software Projects: A Multi-round Delphi Study. JSW, 10(11), 1283-1300.

[24] Singh, M. P., & Vyas, R. (2012). Requirements volatility in software development process. International Journal of Soft Computing, 2, 2012.

[25] Felfernig, A., Schubert, M., Mandl, M., Ricci, F., & Maalej, W. (2010, May). Recommendation and decision technologies for requirements engineering. In Proceedings of the 2nd International Workshop on Recommendation Systems for Software Engineering (pp. 11-15). ACM.

[26] Maalej, W., & Thurimella, A. K. (2009, September). Towards a research agenda for recommendation systems in requirements engineering. In Managing Requirements Knowledge (MARK), 2009 Second International Workshop on (pp. 32-39). IEEE.

[27] Mobasher, B., & Cleland-Huang, J. (2011). Recommender systems in requirements engineering. AI Magazine, 32(3), 81-89.

[28] Renzel, D., Behrendt, M., Klamma, R., & Jarke, M. (2013, July). Requirements bazaar: Social requirements engineering for community-driven innovation. In Requirements Engineering Conference

(RE), 2013 21st IEEE International (pp. 326-327). IEEE.

[29] Ninaus, G., Felfernig, A., Stettinger, M., Reiterer, S., Leitner, G., Weninger, L., & Schanil, W. (2014, August). IntelliReq: Intelligent techniques for software requirements engineering. In Proceedings of theTwenty-first European Conference on Artificial Intelligence (pp. 1161-1166). IOS Press.

[30] Asghar, S., & Umar, M. (2010). Requirement engineering challenges in development of software applications and selection of customer-off-the-shelf (COTS) components. International Journal of Software Engineering, 1(1), 32-50.

[31] Kalinowski, M., Felderer, M., Conte, T., Spínola, R., Prikladnicki, R., Winkler, D., ... & Wagner, S. (2016, January). Preventing incomplete/hidden requirements: reflections on survey data from Austria and Brazil. In International Conference on Software Quality (pp. 63-78). Springer International Publishing.

[32] Fernandez, D. M., Wagner, S., Kalinowski, M., Felderer, M., Mafra, P., Vetrò, A., ... & Männistö, T. (2016). Naming the pain in requirements engineering. Empirical Software Engineering, 1-41.

[33] ur Rehman, T., Khan, M. N. A., & Riaz, N. (2013). Analysis of requirement engineering processes, tools/techniques and methodologies. International Journal of Information Technology and Computer Science (IJITCS), 5(3), 40.

[34] Pohl, K. (2010). Requirements engineering: fundamentals, principles, and techniques. Springer Publishing Company, Incorporated.

[35] Pandey, D., Suman, U., & Ramani, A. K. (2010, October). An effective requirement engineering process model for software development and requirements management. In Advances in Recent Technologies in Communication and Computing (ARTCom), 2010 International Conference on (pp. 287-291). IEEE.

[36] Basharat, I., Fatima, M., Nisa, R., Hashim, R., & Khanum, A. (2013, October). Requirements engineering practices in small and medium software companies: An empirical study. In Science and Information Conference (SAI), 2013 (pp. 218-222). IEEE.

[37] Abbasi, W. A. (2013). Fuzzy Logic Based Intelligent Software Requirement Elicitation Technique Selection Model (Doctoral dissertation, Iqra University Islamabad).

[38] Abbasi, M. A., Jabeen, J., Hafeez, Y., Batool, D. E. B., & Fareen, N. (2015). Assessment of Requirement Elicitation Tools and Techniques by Various Parameters. Software Engineering, 3(2), 7-11.

[39] Sethia, N., & Pillai, A. S. (2013). A study on the software requirements elicitation issues–its causes and effects. In World Congress on Information and Communication Technologies.

[40] Carrillo, D. G. J. M., Nicolás, J., Fernández, A. J. L., Toval, A., Ebert, C., & Vizcaíno, A. (2015). Commonalities and differences between requirements engineering tools: A quantitative approach. Computer Science and Information Systems, 12(1), 257-288.

[41] Niazi, M., Mahmood, S., Alshayeb, M., Qureshi, A. M., Faisal, K., & Cerpa, N. (2016). Toward successful project management in global software development. International Journal of Project Management, 34(8), 1553-1567.

[42] Wiegers, K., & Beatty, J. (2013). *Software requirements*. Pearson Education.

[43] Graham, D., Van Veenendaal, E., & Evans, I. (2008). *Foundations of software testing: ISTQB certification*. Cengage Learning EMEA.

[44] Kumar, S. A., & Kumar, T. A. (2011). Study the impact of requirements management characteristics in global software development projects: an ontology based approach. *International Journal of Software Engineering & Applications*, *2*(4), 107.

[45] Boota, M. W., Ahmad, N., & Masoom, A. H. Requirement Engineering Issues and Their Solutions.