SindhUniv. Res. Jour. (Sci. Ser.) Vol. 51 (03) 465-468 (2019)

http://doi.org/10.26692/sujo/2019.03.74

SINDH UNIVERSITY RESEARCH JOURNAL (SCIENCE SERIES)

Requirement Engineering Challenges in Distributed Software Development

F. A. ABBASI*, A. BURDI++, R. S. KHAN, S. H. F. NAQVI, M. S. ABBASI, A. R. NIZAMANI

Institute of Mathematics & Computer Science, University of Sindh, Jamshoro

Received 26th December 2018 and Revised 11th July 2019

Abstract: In process of software development, requirement engineering is one of the main pillars and important elements because of the role it plays. Requirements acts as baseline upon which the pacts between the team members, people using it and those purchasing it are affirmed. For an increasing number of globalization and technological reasons, it is imperative to note that software development has grown in recent years and has resulted in several changes that are crucial in the establishment of development projects. The change is supported and geared by the desire to work round the clock and capitalize on the extensive number of available resources, reduce on expenses and the urge to be near or closer to the users but the negative effect of such applications is the risks associated with gaps in communications. The distributed development of software is a complex phenomenon, facing the challenges like: geographical separation of project members, different time zones, different languages, different backgrounds, team structure, organizational structure, communication and technology to be used for communication and coordination.

Keywords: Requirement Engineering, Distributed Software Development, Global Software Development.

1. INTRODUCTION

The distributed development of software is a multifaceted phenomenon, facing the challenges like: geographical separation of project members, different time zones, different languages, different team structure, organizational structure, communication and technology to be used for communication and coordination. Requirements are used as a foundation upon which any agreements between the clients, users and the project team members are affirmed. The challenges associated with the requirements engineering are acknowledged as the major reasons that causes many failures in software development. The desired results when it comes to an SE project is made more specific during a RE stage phase. The RE Process revolves around the sub-processes related to analysis and requirement elicitation, specification, validation and requirements.

- RE is one of the crucial Phase in an SE project most significantly in DSD. Any failures or errors encountered during this stage may affect the entire life cycle of an SE project which in the end causes failures in trace ability. In case of misunderstood, imprecise or wrong SRS, there is a high chance that the project will witness high cases of faultiness in cost estimations and project schedules and costly expenses when it comes to rework attempts coupled with frustrations from the part of the stakeholder. High quality SRS are therefore essential for successful SE projects.

2. BACKGROUND

Requirements acts as a baseline upon which all the agreements put in place between the users and the project team members are affirmed. The RE process on the other hand is a mix of other processes such as requirements specification, analysis, requirement elicitation and validation. Elicitation and analysis of requirements revolves around the worm of stakeholders on a complete set of nonfunctional and functional attributes that should be included in the final complete software. As such, the final set of issues included should be an agreement between stakeholders and more significantly, their comprehension of their content must be clear and precise. In the wake of the specification of requirement stage, the collection of attributes is changed to a consistent and complete set of technically unique and specified requirements which are then accepted as software requirements specification. Due to globalization, an increasing number of software developers have been scattered geographically. In so doing, this is what gives DSD its attributes. When the distance ends up developing into worldwide, then it becomes Global Software Development instead of Distributed Software Development.

3. REQUIREMENT ENGINEERING

Requirement engineering is acknowledged as the procedure of pointing out the purpose by locating the stakeholders, their requirements and putting them in a manner that is agreeable to evaluation, scrutiny, communication and at implemented at a later stage. According to (2) RE is acknowledged as a structured process that involves a combination of events that are meant to validate, derive and maintains a systems requirement written material. Some of the combination of activities involved in this procedure include requirements negotiation and analysis, requirements elicitation and validation of requirements. For instance,

++Corresponding Authors Email: asad.buledi@usindh.edu.pk, faheem.abbasi@usindh.edu.pk, rida_khan5@live.co.uk sharif.abbasi@usindh.edu.pk, a.nizamani@usindh.edu.pk

*Institute of Information and Communication Technology, University of Sindh, Jamshoro

(Fig. 1) Below is a representation of a spiral model that is prevalent in RE procedures.

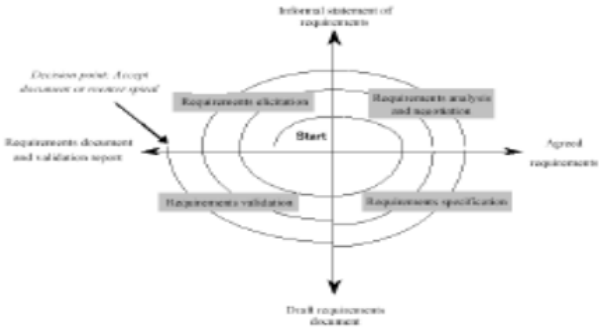

Fig. 1. A spiral model of the requirements engineering process

4. GLOBAL SOFTWARE DEVELOPMENTS

In the contemporary world, Global software Development is not considered a phenomenon. It grows on an annual basis and changes from a trend or new technology into something that is used get profit on a daily basis. Global software development nowadays is not a phenomenon. It expands with every year and turns from a trend into every day type of doing business. GSD enables the realization of reaching to an extensive number of resources and this tends to improve the efficiency of operation. GSD has also revolutionized the manner in Software Development is viewed in the modern era. For this reason, development of software in scattered or distributed surroundings is now incurring changes by encompassing related partners which are scattered based on time, culture and space.

Globalization has changed the style and manner in which information systems development is conducted today. Because of its influence, an increasing number of software developed by geographically co-located developers are now employed by other users in different locations. In so doing, this resulted in the concept of outsourcing and offshoring .offshoring is acknowledged as the main idea behind the GSD since it tries to showcase the nature of globalization which tends to curb geographic, temporal, social a distance across nations. In ancient Information systems development procedures, only two players are present, one who is providing that are responsible for development and selling of the systems and the clients that buy the software. GSD setting is different since it involves three players which include the providers, the clients and the users as shown in fig.1. For instance, a user based in America requested a company based in the America to develop a software for them. The firm can develop or manufacture the software on its own as in traditionally or request another company based in America to manufacture or develop it in what is known as outsource or offshore to other companies outside America, If the organization prefers to use the GSD, the manner of relationship changes leaving the organization as clients and it might request the providers to create a section of the system or the entire system.

Three Main players in GSD

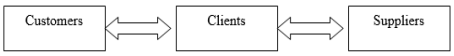

Fig. 2: GSD's Major players

Based on **(Fig. 2),** the clients are found as the connection between users and software provider and gather important data from clients and ensure that it reaches the providers. Moreover, acting as the upper level managers, the clients are responsible for managing the entire process and ensure that the final information systems are delivered to the clients appropriately.

5. DISTRIBUTED SOFTWARE DEVELOPMENT

Distributed development of software is a complex phenomenon, facing the geographical separation of project members, different languages and technology to be used for communication and coordination. The complexity even grows by considering the interplay between these issues: for example that cooperation technology is necessary to conduct distributed projects on one hand, and that such technology is a precondition which makes distributed development possible in the first place.

To deal with this complexity, the phenomenon of distributed software development can be structured by distinguishing between the following three interrelated topics: **(Fig. 3).**

1. Distribution itself: in what way can people and other entities be distributed?
2. Challenges: What are the challenges accompanying distribution?
3. Solutions: What are the solutions to deal with the challenges?

Fig. 3. Distribution itself can be distinguished by three dimensions

1. Physical distribution: team members or stakeholders are located at different physical places (different offices, buildings, cities, countries, continents);
2. Temporal distribution: team members or stakeholders are separated by time caused by different time zones or shift work as well as by different amount of hours spending on the specific project.
3. Organizational distribution: team members or stakeholders are located in different organizational structures; for example different organizations or different departments of one organization or different cross-organizational projects.

6. RESEARCH GOAL

The main objective of this research revolved around the development of Taxonomy for DS which encompasses the exploration of its common concepts. First, it commences by mentioning international sourcing or offshore sourcing as an important concept. International sourcing has been in existence for an extended number of years now. However, the important issue here is that service sectors are expanding and globalization is impacting opportunities that were seen as untouchables. For this reason, firms can help solve their challenges by looking for alternative options in other countries or in the same country.

Offshoring in some instances is acknowledged as outsourcing ort offshore outsourcing. However, what is confusing is that the concepts of offshore outsourcing. Offshoring and outsourcing are sometimes used interchangeably despite the fact that there are an extensive level of differences between them. Outsourcing when explained in terms of a corporate context implies a practice of an organization that involves transferring or the function of the organization to another individual. When this individual or party is situated in an international country, then offshore outsourcing becomes relevant. On the other hand, offshoring implies the transferring of the function or responsibility of an organization to another country irrespective of whether the responsibility remains in the company or not. On the other hand, outsourcing implies the assignment of non-core jobs or duties from internal production in a certain trade to an external entity that is well vast in that operation. One of the major misconceptions is the belief that all offshoring has an element of outsourcing. Some individuals even tend to argue that international offshoring is international outsourcing. Yet this is far from the truth.

As much as outsourced processes are assigned to third party individuals, offshore processes can either remain in the house or be assigned to another party. For this reason, the explanation of offshoring also encompasses organizations that establish dedicated captive centres or their remote locations meant to help them cut on costs. In sourcing can sometimes be explained as the opposite to outsourcing and is the assignment of duties to an internal party. It is decision made in trade to help in to manage and control of specific important competencies within an organization. In some instances, insourcing is also accepted as a situation where organizations establishes their captive process internationally by ensuring that they benefit from the cheaper environments while maintaining and control of their dealing processes and back-office work.

7. PROBLEM STATEMENT

The distributed development of software is a complex phenomenon, facing the challenges like: geographical separation of project members, different time zones, different languages, different backgrounds, team structure, organizational structure, communication and technology to be used for communication and coordination.

7.1. Proposed Solution

The pursuit of this research is to analyze the above stated issues of Distributed Software Development during requirement engineering, study the other approaches and finally proposes a process to address these challenges.

7.2 Process

The process that have been suggested purposes to curb the challenges incurred in distributing requirement process by ensuring that tasks of understanding and adaptation have been introduced to ensure that the SRS team is well taken care. The recommendations are made up of five action plans as shown in Figure 5 and explained below.

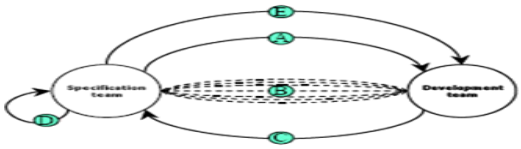

Fig. 4. Proposed process

7.2.1 A. The first version of SRS is finished and forwarded to the development team. While performing the responsibilities of specification, negotiation and elicitation, the specification team documents the first version of SRS. Immediately after completing the tasks, the written material is forwarded to the team of developers for adaptations. Moreover, in environments that are co-located, coding and modelling activities could commence at this moment. Yet, based on the research context, the challenges in distribution can lead to reduction in the understanding of SRS by the team of system developers. For instance, language and other differences can result in a miscomprehension of processes and lead to development of a product that does not meet the demands of the clients.

B. SRS Adaptation and analysis by the team of developers. Based on the engagement moment of the developers, this might be their first contact with the project. Developer's engagement happens at the beginning of every project and this is what makes them know the rationales and the needs. The level of demand or extent can be comprehended with the assistance of software such as Vision/Scope. Yet, it is sometimes challenging to know all the requirements evolution. Some new demands mighty be postulated and a number of interactions with the stakeholders might also come into question. The first action of the development team after being handed the project by the specification team is to try and comprehend the requirements and the primary context. As such, it purposes to curb any

potential sources of challenges that might occur. Impreciseness and ambiguity are common causes of language or other differences. If the team will incur many challenges in terms of this differences, the SRS might be re-documented again. The process of documentation offers more data relative to the information found in the already documented SRS. The process of rewriting or adapting the SRS allows the team to get more insights that are not written explicitly in the document. Many concerns are raised but are handles by the specifications team which curbs the impreciseness or the ambiguity as a result of language dissimilarities. An increase in communication is enhanced to help in the simplification of Step D and building agreements.

Clarifications in some instances can result in new users or clients especially if the developers are not aware of the data. As such, the quality of specification tends to increase. Moreover, re-documentation can also be employed in the standardization of inbound documents. Under normal circumstances, one team of developers can take the role of several specifications teams which will make the rewriting of SRS unique and different. The team of Developers can also make use of patterns of glossary and document, phrase structures and requirements formats to help in avoiding the creation of different formats for projectdocuments. In so doing, this necessity increases when an individual is taking into account metrics applications.

C. Adaptation of SRS is completed.it is then forwards to the Specification team for approval. As soon as it has been adapted completely, the new document set up in this process needs to be approved. AS soon as it has been approved, it is then sent back for verification by the specification team.

D. Approval and validation of SRS by specification team. This is done to ensure that after it has been adapted, it is still in line with the objectives and needs of the stakeholders.The main challenge here is determination of the effort requirements to validate the document and ensure that it still aligns with the needs of the client. Yet it is imperative to note that the communication set up in step C makes the specification tam aware of the process of adaptation and this is what curbs the efforts required to validate this procedure.

E. The Final Version of SRS is defined. Immediately after its approval, the definition of the final version of SRS is done and approved, the final version is employed as baseline of testing, coding and modelling software.

8. CONCLUSION

The prevalence of global distribution when it comes to software development has made the coordination and collaboration among software developers a difficult task because of the number of reasons. One of the reasons revolves around the distinction in time, physical barriers and language differences among others. A close to lack of informal communication among team members makes the coordination of a globally distributed is setting a difficult task.

To deal with these challenges, new processes and techniques are required. For this reason, this research purposes to present a strategy required to solving the differences associated with distribution. Based on the proposed strategy, communication issues such as language barriers are dealt with partially during the adaptation of SRS since it helps to reduce ambiguity and impreciseness during the re-documentation process. Challenges associated with time zone also arise but with an increase in development team understanding of the process needs, team communication frequency reduces. Moreover, the structure also offers the standards needed to fill the SRS.The phrase structures are also defined and are specific to a certain language which assists in curbing issues of ambiguity and improvising the clearness of the requirements. Because of the case study recommendations and proposed process, we are commencing empirical studies in two organizations and it will be based on two case studies made in two projects distributed around the globe.

REFERENCES:

Burg, J. F. M. (1997). Linguistic Instruments in RequirementsEngineering. Amsterdam: IOS Press.

Damian, D. (2007). Stakeholders in global requirements engineering: Lessons learned from practice. IEEE software, 24(2), 21-27.

Gumm, D. C. (2006) Distribution Dimensions in Software Development Projects: A Taxonomy. IEEE Software, September/October 2006, 45-51.

Kotonya, G. and I. Sommerville, (1998). Requirements Engineering: Processes and Techniques. Wiley Pub.

Lloyd, W., and J. Arthur, (2002) Effectiveness of Elicitation Techniques in distributed requirements engineering. Proceedings of the IEEE Joint International Conference on Requirements Engineering (RE'02).

Vibha S. and B. Sengupta, (2006) IBM India Research Lab, Satish Chandra, IBM T.J. Watson Research Center: Enabling Collaboration in Distributed Requirement Management, IEEE SOFTWARE Published by IEEE Computer Society.

Zave, P. (1997). Classification of Research Efforts in Requirements Engineering. *ACM Computing Surveys*, 29(4): 315-321.

Zowghi, D., (2002), Does Global Software Development Need a Different Requirements Engineering Process? Proceedings of International Workshop on Global Software Development – ICSE 2002, Orlando, Florida, USA, 53-55.