J. Basic. Appl. Sci. Res., 4(4)114-119, 2014

© 2014, TextRoad Publication

ISSN 2090-4304
Journal of Basic and Applied
Scientific Research
www.textroad.com

A Study on RE Process Models for Offshore Software Development

Muhammad Usman Akram, Bilal Hassan, ShahidBhatti, Fahad Hassan, Jawad Bashir

[1]COMSATS Institute of Information Technology Lahore, Pakistan
[2]University of Management and Technology Lahore, Pakistan
[3]COMSATS Institute of Information Technology Lahore, Pakistan
[4]University of Lahore, Lahore, Pakistan
[5]COMSATS Institute of Information Technology Lahore, Pakistan

Received: January 6 2014
Accepted: March 11 2014

ABSTRACT

:Increasing trends of cheap and quality software development have raised a great interest in offshore software development. Sub continental software houses are much cheaper than the European or American market. Due to cultural social and linguistic differences the requirement gathering has become difficult for offshore software developers. In this study the requirement gathering issues for offshore software houses are investigated and then the standard requirement engineering models are compared with each other according to different metrics and a comprehensive survey among the sub continental software engineers is carried out to suggest a proper requirement engineering model according to the nature of the project.

KEYWORDS: Requirement Engineering, Offshore Software Development, Pakistani Companies, Project Management, Survey for requirement gathering

I - INTRODUCTION

Ever increasing demand of rapidly develop quality software in low-budget have developed a complete new set of software houses based in Sub-Continent (India, Pakistan, Bangladesh in some context) which are totally dependent on foreign projects, cheap labor have encouraged European and USA based vendors to outsource their development to such low-cost software houses this scenario is called Offshore-outsourced software development (O-OSD). Due to some cultural bounds and lack of communication there arise a number of factors which can affect the software development has identified number of risk factors involved in O-OSD [1].

O-OSD has become a highly favored topic for companies aiming at cost savings while achieving final product delivery within estimated time schedules. Still, this type of development has several challenges due to its inherent nature. For instance, decreased degrees of communication, lack of knowledge about customers' business domains, disputes on legal issues are big concerns [2, 3].

Low maturity of relationships increases the risk factors involved in development of offshore applications [4]. Italso further suggested that requirement gathering is the core phase due to its social nature to be affected. Due to the agile nature of the products especially when it comes to web application development the communication gap and lack of interaction with clients in offshore software houses the requirement gathering phase gets tricky.

It has been studied that the work dispersion has a negative effect on the adopted structured software process models [5]. Software houses have been known to develop their own process models which can either be direct usages of the process models mentioned in texts or they can also tailor the process models according to their needs. The actual process models being used in any software development environment might be somewhat different from those which are widely recommended. There is a need to study the working atmosphere of the software house especially of those working offshore. In offshore software development the requirements are either shipped via e-mails, Skype calls or direct phone conversations. Due to the social differences there is a fear that the actual requirements are interpreted differently as explained by the stake holders.

Due to the problems in communication and the dispersed nature of the working environment there is much need of studying the focus of the offshore software engineers on the requirement gathering phase of the software development.

In this Survey a number of offshore software developers are engaged to discover the requirement engineering practices of offshore software houses. Results are analyzed, discussed and the comparison of different requirement engineering process models is discussed according to different criteria in order to setup guidelines for the requirement engineers to choose the proper model according to the nature of the project.

***Corresponding Author:** Bilal Hassan, University of Management And Technology Lahore, Pakistan

Requirement Engineering Process Models

In recent days, requirement engineering (RE) a domain of software engineering has shaped itself as a complete process which requires thorough emphasize from Elicitation, Analysis, Specification, Validation to Management for the purpose of developing software products in an efficient and cost effective manner. Requirement gathering from the client is a fundamental stage in any process model to develop a successful software product for clients. Requirement Engineering in itself has become a complete process now with its own input and output [6].

Requirement Engineering Activities

The Requirement Engineering is composed of the following activities. In the comparison of RE models and the questioner these activities were analyzed in detail hence preventing the issues of merged activities in RE process which were identified by [7].

Project Creation: In this activity of the requirement engineering process model the team decide to whether or not to create the new project or the existing project can be modified to meet the requirements. Usually project creation takes much less time and this activity is neglected [8].

Elicitation: During the Elicitation phase the team gathers the requirements from different stakeholders of the system. In this phase the team tries to discover Boundaries and also identify the stakeholders involved and also how the goals and different tasks are being performed by each stakeholder [9].

Interpretation and Structuring: Right after the elicitation phase the discovered requirements from elicitation phase are understood, structured and analyzed a proper abstraction is applied to translate the requirements into programmable form. In some of the process model this phase is actually the part of elicitation phase. This is quite possible that interpretation and structuring of the requirements are actually done in elicitation phase or it can also be performed separately [10].

Negotiation: When the requirements are actually understood by the team then the team actually tries to negotiate with the clients to come up with a signed document in which all of the requirements are documented [10].

Verification and Validation: This activity aims to check that the requirements accurately represent the needs of the system (Kotonya, 1998) and that they are complete, correct and consistent [10, 11]. The technical experts or quality assurers also review the requirements.

Change Management: In this activity the change of the requirements are managed whenever a change comes in the requirements the team make sure that certain information is gathered for the change to in-cooperate into system [10]. Hence the change is actually evaluated in this phase increasing risk so the risk management and its impacts are also studied [9].

Requirements Tracing: Requirements tracing is used to track the origins of each requirement.

A requirement gathering phase in any software house is dominated by human, social and organizational factors which create a need to develop a complete model [12]. A lot of work has already been done analyzed and practiced and still there is a lot of room for improvements. There are five activities in any requirement engineering process model. In elicitation phase the requirements are discovered and gathered. Second the analysis phase comes where raw information is analyzed and documented. The requirement specification phase prepares a document containing all the specifications and roles of the product development team. The validation phase validates all the specification either these are implementable, feasible or not. Final phase is about management of requirement e.g. versioning, tracing and tracking etc. In order to carry out all these activities many researchers have proposed process models like Coarse-grain activity model (Linear RE Model), Linear Iterative Model, etc.

Coarse-grain activity model: First of all we will have a look towards Coarse-grain activity model which is one form of linear model. It provides an overall picture of the process and a foundation for other models as well [10]. It also describes the system requirement and design of different activities in the process but it doesn't tells that how to enact a process. But it has some problems. It is not designed to get user feedback; freezing of requirement is not available here. It does not deal with the risks, so no management is available for underlying risks.

***Linear Iterative Model*:** Linear Iterative model is a well-known model and is widely used in software industry [10]. It follows iterations for validation of requirements again and again until stakeholders are agreed and final system specification is achieved. It has no risk management capabilities. But it delicately deals with the issues like freezing of requirement and requirement validation.

***Spiral Model*:** The third famous model which is particularly designed keeping in view the idea of risk analysis and risk management is Spiral Model [10]. It is also based on iterative approach. The major contribution of spiral model is the discovery of requirements in different iterations. These requirements can be re-evaluated after each spiral. After each spiral the specification document gets more mature. This model is also widely used in software industry.

One drawback of this model is that it requires high cost and level of expertise due to risk analysis. It is commonly used for larger projects

Role Action Models: In Role Action Model, there are not only the actions/activities are explained but also the roles are been declared against each activity [13]. Every technical person associated to the project team is mentioned with their role and action. So it is very clear to all development team members, what their responsibilities are and which person they communicate for specific activities. Other models are also defining the roles, but they are in written form. But in role action model all roles are defined in its design. They are useful for process understanding and automation.

V Model: The one major contribution in the requirement engineering process models area is the introduction of V model developed and used efficiently by the German software industry [14]. The beauty of this model is that it validates your requirement at each level of project. The V model has the very clear initial state and goals by which it divides larger projects into smaller ones along with their complete functional and nonfunctional requirements. The V model also catches the risk factors while keeping in mind the quality as its important criterion. It also determines Risk Acceptance, safety and security levels. Analyzing Quality of Requirements is also the part of the V model.

II - Comparison of Requirement Engineering Process Models

The study of process models is of two types. In first attempt the widely used process models are selected and compared based on their activities. Here we find out that all process activities are not handled by all process models. In the second study based on the different criteria (level of project, risk management etc) was chosen for comparison of already selected process models. And we found that different process models have diverse properties and they perform differently. Both studies are discussed in later sections.

Process Activities Comparison among Process Models

All the selected models are reviewed against all the process activities and composed the result in the Table 1. The Table shows that some of the process models do not handle the whole five activities. The academician and the software industry personnel can easily find out their required model with the help of this table.

Model/Phase	Elicitation	Analysis	Specification	Validation	Management
Coarse-grain Activity Model	Y	Y	Y	Y	N
Spiral Model	Y	Y	Y	Y	N
Linear Iterative Model	Y	Y	Y	Y	N
Role Action Models	Y	Y	Y	Y	N
V model	Y	Y	Y	Y	Y

Table 1: Comparison of Models based on activities

Table 1 is compiled by thoroughly studying RE process models and their ability to carry out the RE process activities as mentioned in [9]. This comparison was done to facilitate the developers to check the feasibility of their selected process model which they would choose from Table 2 after carefully understanding the parameters of their project.

There are seven requirement gathering activities mentioned in section 0 for the sake of comparison the project creation activity is left out as it is independent of whichever process model is applied. From Table 1THIS is quite clear that every activity is carried out by almost every process model under consideration apart from the management phase. Only the V Model presents the opportunity to manage the requirement changing and tracing towards its origin. Coarse-grain, spiral, liner iterative and role/action carryout each activity but they do not present the requirement change management.

Criteria to compare Process models based on different parameters

Kotonya and Sommerville have mentioned that RE requirements change according to the nature of the problem under development [10, 15].In the second study different criteria is selected to compare the process models (user feedback, freezing of requirement, small project, large project, prepare prototyping (Requirements Tools, 2011), risk management) [16]. The study will help the software industry personnel and researchers to decide the

process models according to their requirement. The results are shown in Table 2. Different process models fulfill different criteria; this study can help the developers to choose the process model according to their criteria.
Coarse-grain requirement process model is pretty simple one in nature it is a linear process which carry out the activities in linear manner. This can only be used in very ideal cases in which the requirements are already pretty much clear so this process model can only be used for small projects in which the requirements are somewhat straight forward. This process model is not recommended for any added complexity. The spiral model is usually carried out in parallel to the spiral model of the software engineering process model. For small projects this is not recommended as the spiral is known to handle the intensive risk management and it usually increases the cost which is a big concern for small projects. Neither feedback nor the requirement freezing is handled in spiral rather it goes for iterations.

Parameter/ Model	Coarse-grain Activity Model	Spiral Model	Linear Iterative Model	Role Action Models	V model
Small projects	Y	N	Y	N	Y
Large project	N	Y	Y	Y	N
User feedback	N	N	Y	N	Y
Requirement Freezing	N	N	N	N	N
Risk management	N	Y	N	N	Y
Involve prototyping	N	Y	Y	Y	Y

Table 2: Comparison of Models for different user requirements

Linear iterative model does not carry the risk management thus reducing the overall cost and it also is iterative in nature in the cases in which the requirement freezing is required this doesn't provide ant mechanism to do that. Hence it can be used for small and also for large projects in which the risk management and the requirement freezing is not required. In its usages the role/action model is also similar to the iterative process model the only difference is that in role/action process model it is supposed that the team size is considerably large which is not the case in small projects so this process model is not suitable for small projects in every other regard this process model is suitable in every case in which the linear iterative model can be used.

In the V model the software validation and verification is major target for which the V model (Bucanac, 2004) was suggested [14]. This model is the modified coarse-grain model to conduct the validation and verification. This model does not provide the freezing of requirement either but it definitely provides the mechanism to get the user feedback in time.

III - The Survey of Offshore Software Developers/Companies

A Comprehensive Survey was being conducted from the offshore software houses and their team members who do outsourcing to gather the information of actual RE practices. Software houses and individuals working offshore practice different approaches towards requirement gathering of the upcoming projects. Questions were asked for three different kinds of projects listed in Table 3 for each activity of the requirement engineering. The questioned asked in the survey were designed according to the considerations presented by Meltzoff [17].

	Small Projects	Medium Projects	Large Projects
Team Members	1-10	10-20	20-50
Effort in Person Months	1-3	3-10	10-30

Table 3: Projects Evaluated

A careful consideration was made in choosing the population size and quality for the survey under discussion. Around 400 individuals were being contacted to fill out the questioner. The questions were focused on how much importance was given to each RE phase for small to medium projects. The population was selected according to the guidelines given in [18].The CEOs, Team Leads, Experienced Developers and the individual contractors were among the audience from which this survey was being conducted. A Series of Questions were asked from each one about each activity of RE which they practice in their development environments the form used can be accessed online [19].

	Small Projects	Medium Projects	Large Projects
Project Creation	50.0%	66.7%	77.1%
Elicitation	55.6%	80.6%	86.1%
Interpretation and Structuring	47.2%	75.0%	78.1%
Negotiation	36.1%	77.8%	81.8%
Verification and Validation	75.0%	77.8%	83.3%
Change Management	36.1%	72.2%	80.6%
Requirements Tracing	47.2%	61.1%	70.6%

Table 4: Survey Results

Table 4 presents the results of the survey rows are the projects. Three different kinds of projects were selected for the survey. The nature of the project was selected according to the project size the details are listed in Table 3.

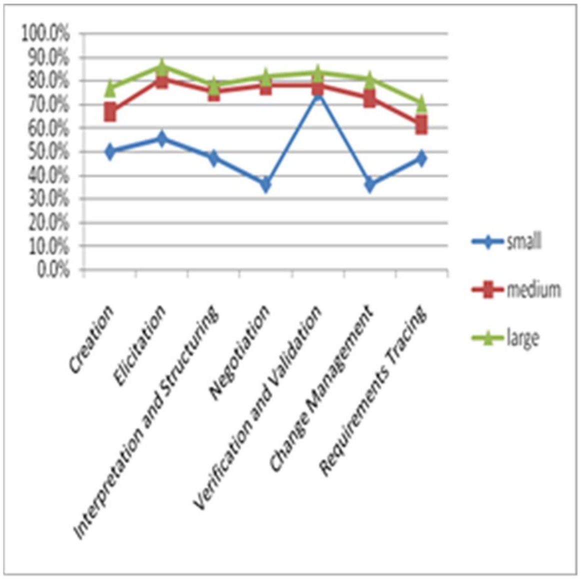

Figure 1: Plot of the results of the survey

The details of the survey are presented in Table, in order to get a better view of the results the same results are plotted in Figure. On the left is the percentage of the persons saying yes to the question asked for each kind of project that whether or not they consider the requirement engineering activity labeled horizontally in the graph in their own projects. The results for each activity for the small projects are on lower side except for the verification and validation phase. This is quite obvious that in reality the small projects always have low budget which result for the developers to spend less time in risk management. Verification phase is the phase for which everyone emphasized that they do consider it in every case whether the project is small or not. Elicitation phase is another considerable phase the graph of Figure 1 also shows that the developers do rate this step very high in development of medium to large projects. As this survey was conducted from the actual developers, team leads and CEOs of software houses located in subcontinent doing offshore software development the results are actually the picture of the requirement engineering processes being practiced.

IV - Conclusion and Future work

The problems of requirement engineering for offshore software development were identified and then a comprehensive survey of the suggested requirement engineering process models was done for different metrics. Two approaches were used on one the process models were compared for each activity whether or not that process model provides the base for each activity. Another approach was to compare the process models for different requirements from the development team perspective. At the end the comparison was mapped on the actual practices in offshore software houses. From the survey it is quite clear that developers rate the elicitation and verification phase as the

most important. Although the use of the RE process models are used in parallel to the SE process model being used in each project but from the study it’s suggested that V model for RE is better in practice.
The projects were evaluated according to the size of the projects. There is a need to carry out a study on the nature of the project one such study can be that are there any differences of the RE requirements for the developers for the web-based, desktop or mobile apps.

Acknowledgment:

The authors declare that they have no conflicts of interest in this research.

REFERENCES

1. Akram, M. U. (2013, 12). *Requirement Engineering Survey Form*. (University of Lahore) Retrieved 12 17, 2013, from https://docs.google.com/forms/d/1N9rRCMCTMKL036thImsgK635s6BgxYzTnVyH3BtFWl8/viewform

2. Astray, W. M. (2006). *Globalization and of shoring of Software: A Report of the ACM Job Migration Task Force.* NY: ACM.

3 Bucanac, C. (2004). *The V Model*. Ronneby: University of Karlskrona.

4. D. Pandey., U. S. (2010). An Effective Requirement Engineering Process Model for Software Development and Requirements Management. International Conference on Advances in Recent Technologies in Communication and Computing (ARTCom).

5. Houdek, F. a. (6-8 September 2000). Analyzing requirements engineering processes: a case study. *11th International Workshop on Database and Expert Systems Applications*, (pp. 983-987). Greenwich, UK.

6. Humphrey, W. S. (1995). *A discipline for software engineering.* Addison-Wesley.

7. Kotonya, G. a. (1998). *Requirements Engineering - Processes and Techniques.* UK: John Wiley & Sons.

8. Madhuchhanda Das Aundhe, S. K. (2009). Risks in offshore IT outsourcing: A service provider perspective. *European Management Journal* .

9. Meltzoff. (1998). Critical Thinking About Research: Psychology and Related Fields.

10. Narayan Ramasubbu, S. M. (June 2008). Work dispersion, process-based learning, and offshore software development performance. *32* (2).

11. Nuseibeh, B. a. (2000). Requirements Engineering: A Roadmap, The Future of Software Engineering. *ACM Press, Anthony Finkelstein (Ed.).*

12. Requirements Tools. (2011). (Atlantic Systems Guild Ltd) Retrieved 12 17, 2013, from http://www.volere.co.uk/tools.htm

13. S. U. arif, Q. k. (2009). requirements engineering processes, tools/technologies and thodologies. *International Journal of Reviews in Computing* .

14. Sacha Martin, A. A. (2002). Requirements Engineering Process Models in Practice .

15. SAKTHIVEL, S. (April 2007). Managing Risk in OffShore Systems Development. *Communications of the ACM*, *50* (4).

16. Sommerville, I. a. (1997). Requirements engineering: a good practice guide.

17. Steve Easterbrook, J. S.-A. (2007). Selecting Empirical Methods for Software Engineering .

18. Taylor, R. N. (2009). *Software architecture: foundations, theory, and practice.* Wiley Publishing.

19. Verner, J. M. (2012). Exploratory case study research: Outsourced project failure. *54* (8).