INTRODUCING REFINED AGILE MODEL (RAM) IN THE CONTEXT OF BANGLADESH'S SOFTWARE DEVELOPMENT ENVIRONMENT CONCENTRATING ON THE IMPROVEMENT OF REQUIREMENT ENGINEERING PROCESS

Nirjhor Anjum[1] and Anwarul Kabir[2]

[1]Chief Analyst Officer, REVE Systems, Dhaka, Bangladesh.
[2]Associate Professor, Department of Computer Science, American International University Bangladesh.

ABSTRACT

The Software Companies of Bangladesh are using different types of agile models for software development. Although theoretically these models are worthy for small and medium projects, in practical case they are not so effective. In doing so, this paper tries to find out why do the agile models not suitable for Bangladesh's Software Companies and how do the problems that the Software Companies face for using the models can be solved. To reveal the answers, this study is based on survey and interview methods. Findings of this paper show that Bangladesh's Software Companies are facing different problems for implementing traditional agile models, such as, Communicational gap, lack of Documentation, unavailability of Prototype, Customer's lack of knowledge in the area of IT and many more. The study shows that if the Requirement Engineering Process is perfectly managed and some rules are modified in the traditional agile models, these problems can be solved. In doing so, a new model has been proposed by the study named Refined Agile Model (RAM) which is claimed to be better for Bangladesh rather than the traditional Agile Models. This model proposes a process flow which consists of Prototyping Cycle, Development Iteration Cycle and Additional Development Iteration Cycle. This new model also ensures a Requirement Engineer at Client End, sufficient documentation, preparation of prototype and presentation of frequent Demos. After ensuring these requirements in several real time projects, it was found that those projects were completed more effectively compared to all other old project experiences. Eventually, the paper concludes by mentioning that the Refined Agile Model (RAM) is the best model in the Bangladeshi software environment.

KEYWORDS

Agile methodology, Requirement engineering process, Software development life cycle.

1. INTRODUCTION

Requirement Engineering is the basic part of any kind of software development. Requirement Engineering involves requirement elicitation, requirement analysis, and requirement management. Inaccurate collection of customer expectation, wrong definition of requirement, improper analysis of requirement, and mismanagement of requirements etc. may cause implementation of an inefficient system. Sometimes it may cause the total failure of a software project.

DOI: 10.5121/ijsea.2019.10402

The Requirement Volatility is such a problem that may occur in different phases of a Software Development Life Cycle due to poor Requirement Engineering. In any pre-development phase if customer expectation is collected inaccurately by the Business Analyst, it must bring requirement volatility in the development phase. In the Project Analysis and Planning phase if the requirements are defined wrongly by the Technical Analyst, it will bring requirement volatility as well. If the System Architects do wrong analysis of the requirements it will also bring requirement volatility in the project. Again, if the Project Manager fails to manage the requirements in the development phase, it causes requirement volatility in both the development and post-development phases too.

One of the main causes of requirement volatility is less communication between software developer companies and client end's main stakeholders who are the decision makers for their projects. Though there are different well-known Software Development Methodologies like Spiral, Prototype, RAD and Agile etc, these cannot ensure flaws, especially in Bangladeshi software environment none of these models is found to be efficient due to constraints.

In this paperwe will be observing the present status of software development practices in Bangladesh, especially we are going to look into the communication gaps between developers and the client end users of the project which have severe impact on the entire project.

Section 1 of this paper focuses on the objectives for introducing Refined Agile Model (RAM) in the context of Bangladesh's software development environment. Besides this section explains elaborately about the background of the study. Furthermore, section 1 discuses methods that have been used to do this empirical study. In the same vein, section 2 of the paper sheds lights on the findings. On the other hand, Section 3 compares different methodologies that are used in Software Development Life Cycle practices and also tries to focus on the problems of SDLC practices in Bangladesh. Moreover, Section 4 of this paper introduces a new model named Refined Agile Model (RAM) with the help of requirement engineering process. This section also explains the efficacy of the new model to solve the problems of SDLC Practice in Bangladesh. Eventually, Section 5 of the paper highlights some research initiatives that can researchers will have been taken in this research arena in future.

1.1. OBJECTIVES OF THE RESEARCH

This paper aims at-

- To find the weaknesses of the models used in Bangladesh software development environment.
- To explore the software process model used by the Software Company in Bangladesh.
- To investigate the problems that the Software Companies are facing in Bangladesh, especially in requirement engineering while working for any domestic or offshore software project.
- Based on the problems, especially in connection to the Requirement Engineering process, this paper will try to introduce a process model that will suit in the context of Bangladesh's software development environment.

1.2. BACKGROUND STUDY

The Software Company is rather young compared to other industries in Bangladesh. Basically, in the early 1980s software development started in Bangladesh (Moretaza, 2016). Though majority of the Software Companies in Bangladesh were established in the late 1990s and onward, some of them have started their business in early 80s. In 1997 the Bangladesh Association of Software and

Information Services was established as the national trade body for software and IT service industry. The objective of BASIS was to promote the IT and Software Companies by providing with different types of supports. Currently there are more than 954 registered Software Companies with BASIS and around 2000 non-registered IT and Software Companies in Bangladesh who are providing different software solutions (BASIS, 2016). A large part of these Software Companies are working with foreign clients, therefore their main goal is to work with offshore projects. On the other hand, other companies are involved in providing software solution to the domestic clients.

In the domestic market most of the Software Companies generally provide customization and configuration service of the software systems that are already developed software products. However, other companies are involved with development of small to large website and software for their local clients from the scratch. However, it may be noted that the market for software system in Bangladesh is expanding. At present, maximum of these software projects are web based. It has been found from analysis that maximum of these projects use PHP, other use ASP.Net/C# platform, while rest few use JSP/J2EE platform.

It is for sure that, well defined requirement is the foundation of software project development because it provides a big support in the project estimation, analysis and planning phases. The success of a software project is also influenced by the quality of the requirements. Though the primary requirements are well documented in the beginning, still changes will appear during the software project development lifecycle. Requirement volatility is reported as one of the major reasons behind changes that occur in projects(Tiwana & Keli, 2014). The main problem of projects is due to increase in technical and business complexity now a days. Therefore, the requirement volatility management ability is the better way to bring success to software projects (Menachem, 1994).

Different studies on software projects have highlighted the growing concern posed by requirement volatility, and have stressed the need of proper management techniques in dealing with the situation. Various approaches have been discussed in the literature concerning management of requirements changes. Jones presents a list of methods or techniques which can be utilized to control the rate of requirements change which includes Joint Application Design, Prototyping and Configuration Management Change Control Board (Jones, 2009). Wiegers discusses about base-lining requirements with a purpose of controlling the requirements changes (Wiegers, 1999). Sudhakar presents suggestions on the contribution of process models towards management of requirement volatility (Sudhakar, 2005). His paper summarizes the advantages and disadvantages of various process models in handling volatile requirements. It is mentioned by Rajlich that adopting new practiceslike agile or iterative etc. can better manage the troubles regarding to requirement volatility (Rajlich, 2006).

The main goal of this research paper is to propound a new Agile Development Model, especially for Bangladesh, which can drive Requirement Engineering in a better way and can bring success to the software projects.

1.3. Research Methodology

The paper is based on both survey and interview method along with several case studies and analysis on real software projects to find out a solution for controlling Requirement Volatility.
At first, a structured set of questions was made where different questions were asked from each of the phases of a software development life cycle. To get an effective feedback on this questionnaire, 12 Business Analysts and Software Analysts were selected. Their valuable answers

on the questionnaire helped to identify Requirement Volatility as a major problem of Software Development Life Cycle. But, only Survey Method was not sufficient to find out all the problems. Then, to find the problems related to Requirement Volatility in details and the remedies that the Analysts took, later on, Interview Method was taken. Through interview with the Analysts the problem of Requirement Volatility was understood more clearly. Besides, the interviews gave some ideas on how the Analysts tried to handle Requirement Volatility in their projects. But, the found ideas were not sufficient to mitigate volatilities from the projects.

Next, as Case Study five best software projects were selected from a leading and renowned software company of Bangladesh. These five software projects were taken from different categories, different development model and different countries so that the paper may not have any limitation to the type of software development or location.
All these survey, interview and case study are made on the project managers and on such projects that faced requirement volatility even being renowned and large projects.

All these survey, interviews and case studies helped finally to find out the source, type, scope, and reasons of Requirement Volatility. Finally all these helped this paper to find the way of managing Requirement Engineering process effectively to bring success in software projects.
As the rate of Software Development in Bangladesh is not huge like foreign companies, this paperanalyzed on 12 most popular software projects of different categories; however, interview and questionnaire is taken from 25 people who are Senior Project Managers, Software Analysts, Business Analysts and Team Leaders. It is expected that, this data will be sufficient to do research on Bangladeshi Software Company.

2. Empirical Study and Problem Identification

Different methodologies were followed to examine and find the difficulties that Bangladeshi Software Companies face in the software development life cycle. At first Survey with questionnaire, next Interview from 12 Business Analysts and Software Analysts, and finally Case Study on five best selected software projects were taken in this regard. This chapter explains about the finding of problems that Bangladeshi Software Companies face while following traditional development models.

Survey Findings

First of all, we will be going through the Surveys and will find out the statistics to understand the findings from this survey:

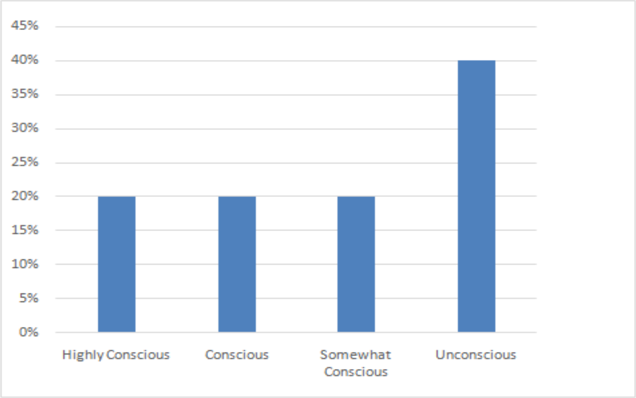


Figure 1: IT Consciousness of Customer

Figure 1 says that, maximum of the customers are unconscious or somewhat conscious about what they are expecting. Therefore, when customer is explaining their requirement, it is for sure that, any kind of wrong requirement can be received from customer. From this study we find that, a Business or Technical Analyst should be concerned about how IT concern his customer is actually, otherwise any mistake may occur while collecting requirements.

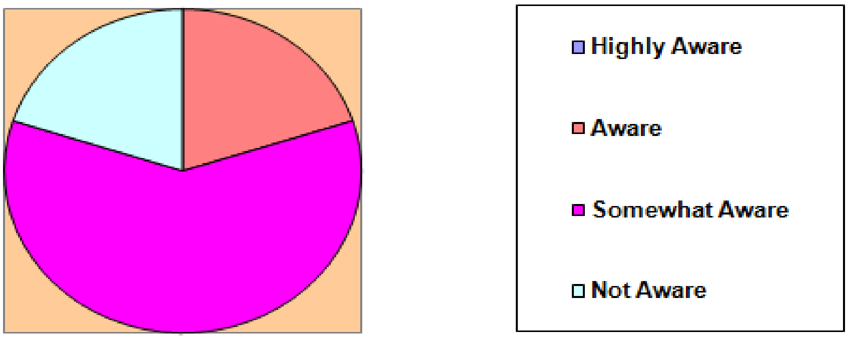


Figure 2: Customer's Understanding on own Requirement

From figure 2 we can understand that, 60% of the customers are somewhat aware and 20% of the customers are not aware of what they are expecting, while only rest 20% is aware of what they are actually expecting. From this question it is found that, to avoid requirement volatility we should know about what customer is expecting actually, because due to lack of technical knowledge customer sometimes can't understand what they are asking and whether it is feasible or not what they are asking.

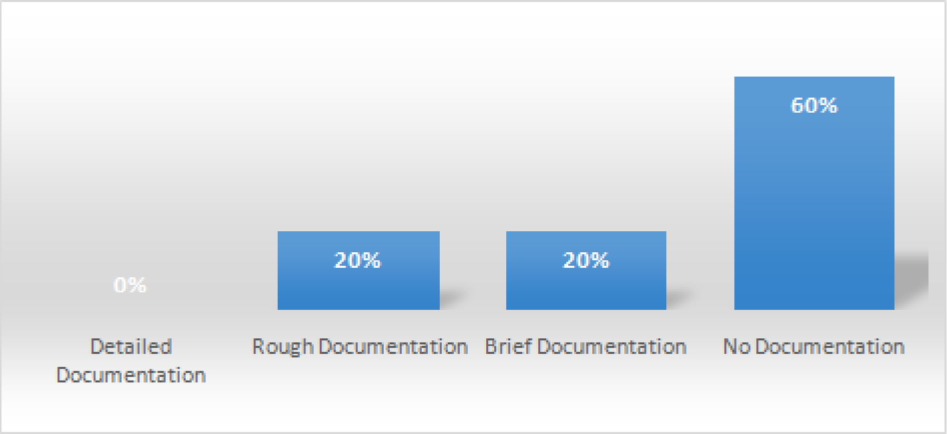


Figure 3: Documentation of Requirements from Customer

From figure 3 we can find that, 60% customers come with no documentation, while only 20% comes with brief and rest 20% comes with rough documentation. It rarely happens that customer comes with detailed documentation. This study makes us understand that, to protect a project from requirement volatility collecting customer requirement through documentation is very necessary. This might not cover all the requirements accurately but at least gives a concept on how customer is defining his requirements.

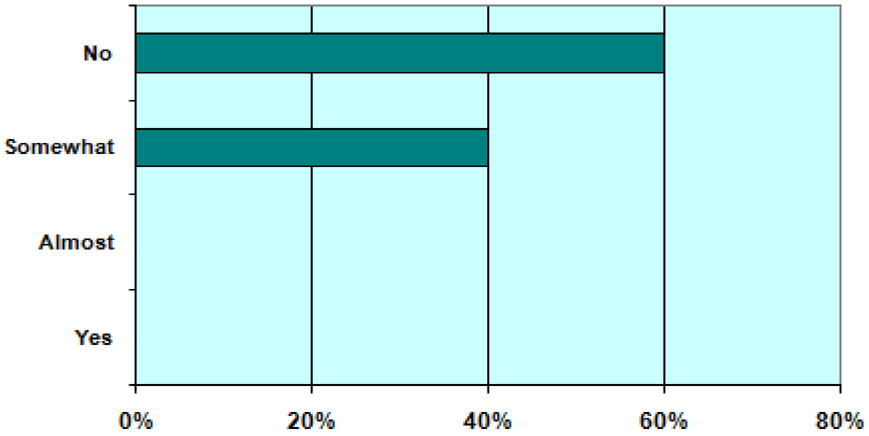


Figure 4: Making Customer clear about the Deliverables in the Requirement Analysis Phase

From figure 4 it is very clear that, in 60% of the cases it wasn't possible to make the customer visualize about what they are going to receive, and in 40% of the cases all the Developer Companies somewhat visualized the deliverables to the customers. Therefore, very clearly we can understand that, customer can't be sure about his entire requirement in the beginning, when gradually customer watches some parts of the development and understands technical things, then gradually he identifies and defines his further requirements more accurately. So, if the Analysts make a clear view to their customers about what they are going to deliver ultimately, and to do so if they make some prototypes or wireframes then easily requirement volatility will be avoidable.

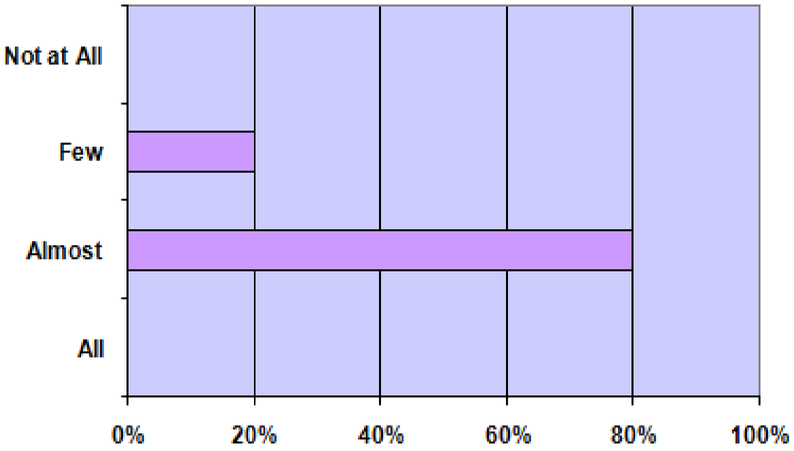


Figure 5: Feasibility of Customer Requirements

Figure 5 makes clear that, in 80% cases customer explained requirements are almost feasible, while in 20% cases few requirements are feasible while maximum are not. Therefore, it may happen in some project cases that many requirements provided by customer might not be feasible. If Analysts do mistake in evaluating feasibilities of customer requirements in the analysis phase, the requirement volatility may occur in later phases of the SDLC. This means, proper feasibility testing of customer requirement is highly important to protest requirement volatility.

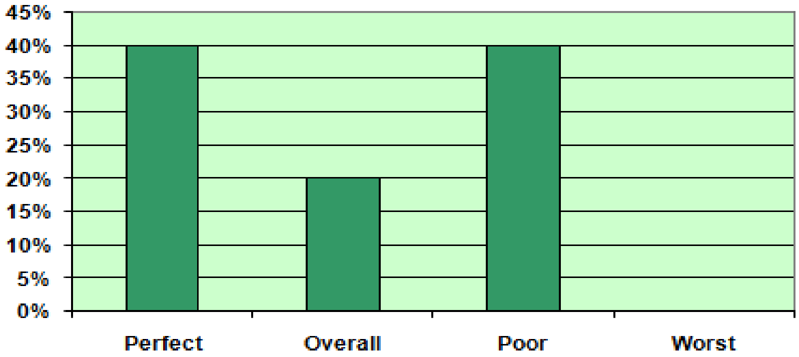


Figure 6: Understandability and Analyzability of Customer's Requirement by Analysts

According to figure 6, while 40% of the analysis and understanding of customer requirement by analysts are perfect and 20% is overall, at the other hand, 40% of the analysis is poor. If requirement is analyzed and understood poorly it must bring volatility in requirements. Therefore, wrong analysis of requirement by the Analysts may bring requirement volatility in the development phase, because when requirement is wrongly understood then requirement must change in later phases once the wrongness is understood.

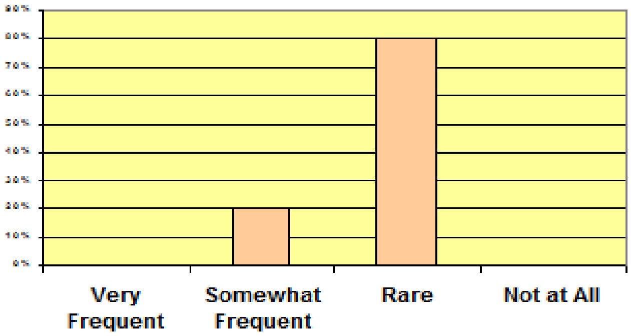


Figure 7: Frequency of communication in the Requirement Analysis Phase

The statistics of figure 7 shows that in 80% cases customer rarely communicates in the development phase, while 20% communication is somewhat frequent. This study makes a clear understanding that, due to rare communication in the requirement analysis phase it was not possible to collect all the requirements properly from customer. Therefore, proper communication is required between customer and analysts in the requirement analysis phase.

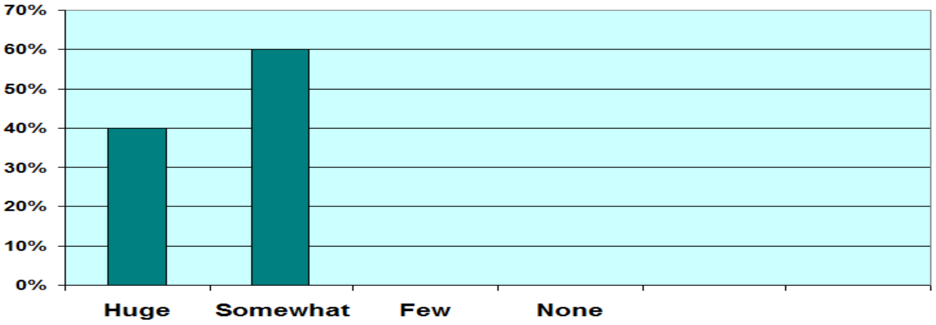


Figure 8: Arrival of New Requirements in the Development Phase

According to figure 8 it is found that, in 60% cases somewhat while in 40% cases huge requirement arrived in the development phase. From it we learn that, care should be taken to manage the requirements by the Project Managers. If a Project Manager fails in requirement management it may cause requirement volatility. Furthermore, incomplete collection of requirement in the analysis phase may cause volatility in the development phase.

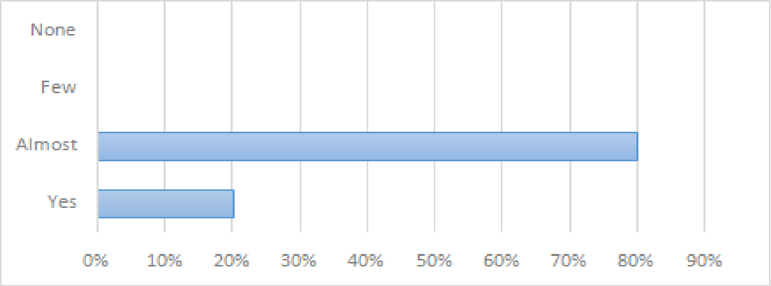


Figure 9: Number of Newly Accepted Requirements in the Development Phase

Figure 9 shows that, in 80% cases we accept almost all new requirements, while in 20% cases we accept all the requirements that customer provides while development phase is already running. From it we learn a lesson that, we should not accept any new requirement while a development phase is running if we really want to protest requirement volatility.

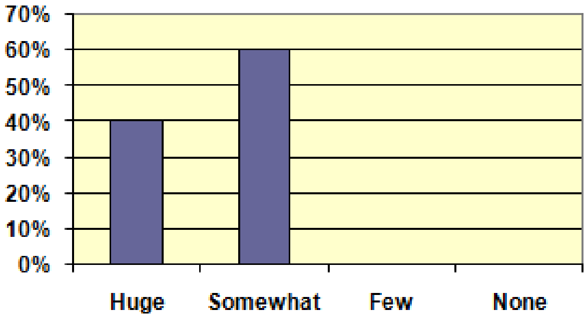


Figure 10: Number of Change Requirements that are accepted in Development Phase

Figure 10 makes us clear that, not only new requirements but we also accept huge change requests in 40% cases, and somewhat change requests in rest 60% cases. If the requirements are understood, analyzed, defined, documented, collected and managed properly, therefore if the requirement analysis phase is successful then change request never appears in the later phases of a SDLC.

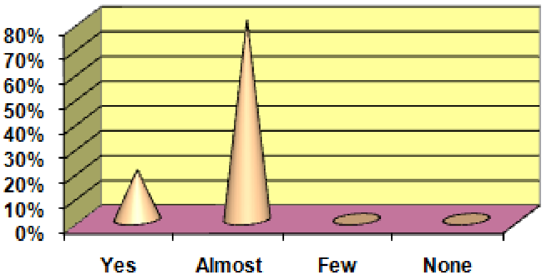


Figure 11: Delay in Delivery Timeline while New/Change Requirements are accepted

Figure 11 explains that, in 80% cases deadline is overall missed and in 20% cases it is highly missed due to accepting new/change requests from customer in the development phase. So, to avoid requirement volatility we always should avoid accepting new/change requests widely in the development phase. Therefore, we can accept any number of requirements in the requirement collection phase but not when the contracted development phase is running.

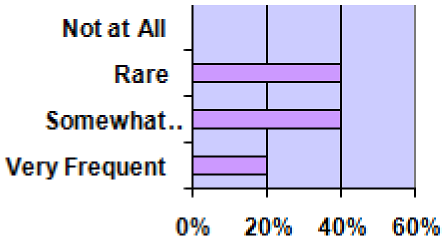


Figure 12: Frequency of communication in the Development Phase

It is found from figure 12 that, in 80% cases customer communicates the development team either rare and somewhat while the development phase is running. It is observed that where frequency of communication is higher, the requirement volatility appeared less therein. Therefore, to overcome requirement volatility problem, frequent communication between clients and development team is required.

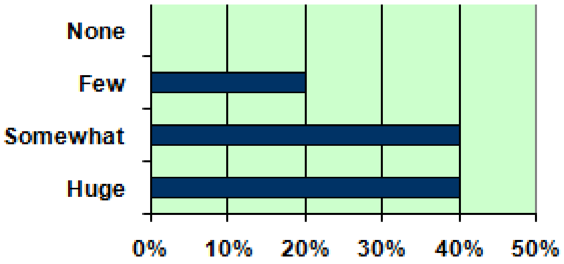


Figure 13: Number of Newly Accepted Requirements in the Testing Phase

From figure 13 it is found that, even in the testing phase customer gave huge requirement in 40% cases while somewhat requirements in rest 40% cases. This happens because when customer views a practical view of the development they understands more accurately about what they are going to get and what they actually need. Therefore, in testing phase requirement management is highly required by the Project Managers.

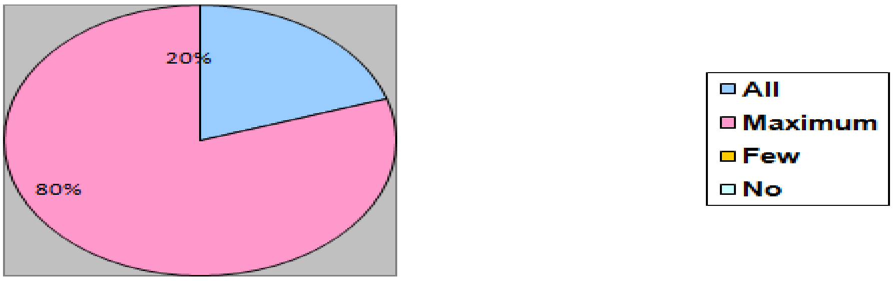


Figure 14: Number of Change Requirements that are accepted in Testing Phase

As according to figure 14, 80% cases the development companies accept new/change requirements from the customers in the test phase to complete the project successfully. But such acceptance of requirements lastly delays the timeline of the entire SDLC, and the project at last gets delayed. Therefore, if any new requirement appears in the testing phase, we can mark that another development phase but shouldn't include those within the running development phase. Otherwise the project will be victimized by requirement volatility.

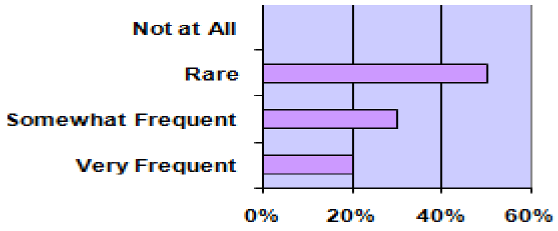


Figure 15: Frequency of communication in the Testing Phase

From figure 15 we find that, in 80% cases customer-developer communication is either rare or somewhat frequent. Though testers are overall aware of the requirements and testable checklists but still they cannot understand the actual customer requirements due to less communication opportunity with customers in the testing phase. If customer-developer/tester communication remains deep in the testing phase as well, this will help to mitigate requirement volatility from the end part of the SDLC.

Interview Findings

As questionnaire method is not sufficient to understand a software project, it's life cycle, issues and solutions, next interview method is added to learn more about how Requirement Volatility appears in real time software projects, and how the Project Managers (both Business and Technical) handle these volatilities practically. For the purpose of this paperinterviews were taken from 6 Business Analysts and 6 Software Analysts who were involved with some selected and most popular software projects.

From a set of interview it is found that, to avoid Requirement Volatility companies involve their Software Analysts in the Requirement Collection so that they can hear and understand the requirements from customer. Generally this phase is handled by Business Analysts, but to avoid collection of wrong/incomplete requirement companies involve their Software Analysts as well.
It is also found from the interviews that, if even the Agile Method is followed, to reduce Requirement Volatility proper and detailed documentation is maintained through the development life cycle by some companies.

Some interviewee have informed that, to avoid Requirement Volatility customer was kept tagged throughout the development process deeply and multiple demo is presented so that there might have no new requirement at the end of the project.

3. A Comparison Study of Different Methodologies in Bangladesh

There are different Software Development Life Cycles, among which the best known ones are:

- Waterfall Model
- Iterative Model

- Spiral Model
- V-Model
- Big Bang Model
- Agile Model
- Scrum
- XP
- RAD Model
- Software Prototype Model

Although the above mentioned models are used to find out the problems regarding the information and technology sector of the world, Scrum, Agile Model (Agile XP and RAD) are practiced in Bangladesh's industries which are related to information and technology (IT) sector. Due to this reason, this models are studied to identify the problems of the information and technology based industries of Bangladesh.

Problem 1

The findings reveal that as Bangladesh's software industries are growing rapidly from the last 10-15 years, customers are not much concerned about IT products and services. In doing so, in many cases customer expects things that are technically not feasible. As a result, the proper requirements collection process becomes difficult for the Software/Business Analysts.
On the other hand, Agile Model advantageous to that customer who is overall technology oriented. So, this is so much contradictory to the current scenario of Bangladesh.

Problem 2

From the paper findings it is very clearly found that the customers are very indifferent to study, understand, analyze and plan the requirements of their projects through proper documentations in Bangladesh. As a result, Software Companies compel to give less emphasis on the documentation. And, for this lack of proper documentations, if any team member of the project leave the project within development phase that makes an impediment to the new member who is replaced instead of member who leaves the project to understand the scope of the project properly. In this case, it is very clear that the Agile Model is not appropriate for Bangladesh because of its preference on the less documentation.

Problem 3

As Agile Model requires an overall plan of the project from customer, and it is not feasible in Bangladesh because here the maximum customers are not IT oriented and they cannot visualize what they are expecting for their business. Later when the development of their project starts, sitting with the vendor companies they begin understanding what they actually need. In doing so, customer cannot give all requirements in the beginning, and his requirements increases throughout the development phase. As a result, Software Companies cannot visualize the final outcome of the project and unable to stay stick to a perfect project plan.

Problem 4

Agile and RAD methodologies demand an uninterrupted communication between customers and Software Companies during the development phase of the project. But in the case of Bangladesh's Software Companies, after giving the work order, customers are indifferent to communicate with the companies. When the deadline comes, they start to knock the developers for delivering the project.

Problem 5

Agile Model requires complete dedication of the software developers to the project. But findings shows that in Bangladesh, the practical scenario is different. Here, it is impossible for any developer to work dedicatedly for one or at two project. It frequently happens that each developer is tagged with multiple software development and maintenance projects.

Table of Problem Findings on SDLC Practice in Bangladesh:

From the comparative studies of the different methodologies of Bangladesh, it is clear that maximum agile and RAD methodologies attributes have some problems for which they are not suitable for Bangladesh Software Companies. In doing so, the following table explains the problems and their results and effectiveness of these Agile and RAD models in Bangladeshi Industry:

ISSUES	**RESULT**	**Can the SDLC give Solution?**		
		Scrum	**XP**	**RAD**
Customers are not much IT concerned and can't express their exact requirement accurately	• Many of customer requirements become unfeasible • Wrong requirement is collected from customer	No	No	No
Customer don't provide documentation for their requirement	• Due to less documentation, detailed requirements are not collected from customers	No	No	No
Development Companies rarely do detailed documentation of the projects	• If Team Member leaves it becomes very difficult to let new Members understand about currently running project	No	No	No
Customer can't provide their entire requirement in Requirement Analysis phase	• More requirement arrives in the Development and Testing phase • The final outcome remains unclear and uncertain	No	No	No
Customers don't keep much communication in Development and Testing Phase	• Customer can't understand about the final outcome of the development unless the final phase reaches	Partially	Partially	No
New Requirement arrives in the Development and even in Testing Phase from customer	• Timeline increases	No	No	No

To keep customer satisfied, the Software Companies accept new requirements even when a Development and Testing Phase is running	• Project delay occurs	Yes	Yes	Yes
Developers are not fully dedicated for one single project, and is involved with others	• Project delay occurs • Project fails to reach quality	Partially	Partially	Partially

In this way, it becomes clear that, theoretical software development methodologies are failed to suit with Bangladesh's Software Environment. This outcome of the problem inspires to find a better model which can be best for the Bangladesh's Software Environment.

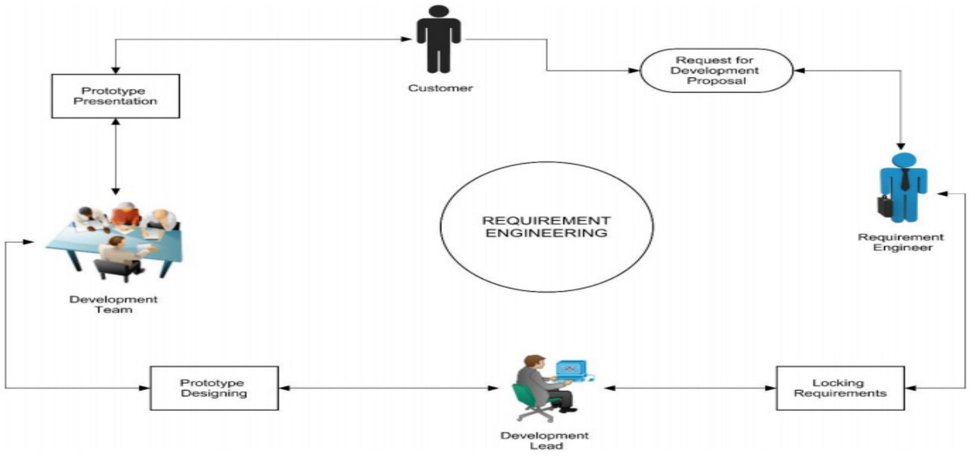
4. Introducing Refined Agile Model (RAM) in the Context of Bangladesh Environment Concentrating on the Improvement of Requirement Engineering Process

From the study on Bangladesh's Software Companies it is found that, no traditional and theoretical model can be suitable for the companies.

It is expected that the newly proposed Refined Agile Model (RAM) can be a fruitful model for developing software in the Bangladesh's software industry.

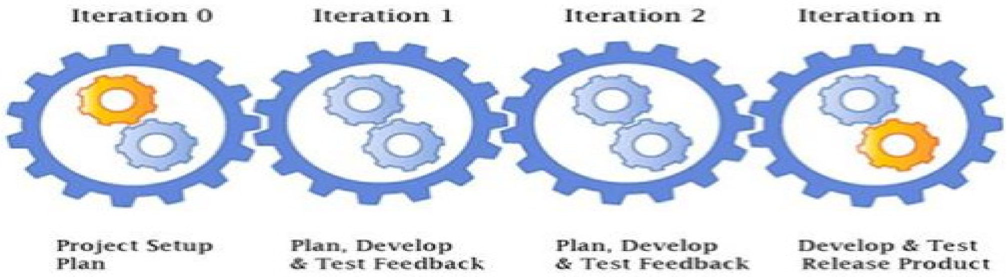
Requirement Engineering Process:

Refined Agile Model (RAM) proposes that Customer will not be responsible for giving plan, sharing idea, describing requirements or keeping communication. The Software Company will appoint a Requirement Manager and a Requirement Engineer who will be visiting Customer premises, will know their business need, will understand what exactly they need and will act like Customer with the Development Team. These Requirement Manager/Engineer are from IT background with knowledge in Business Area. So, they will better understand the need of Customers. In this cycle the Requirement Engineer/Manager will not only lock requirements but also bring an approximately perfect Prototype of what customer is expecting. This will help he Customer visualize that what he is going to receive finally.

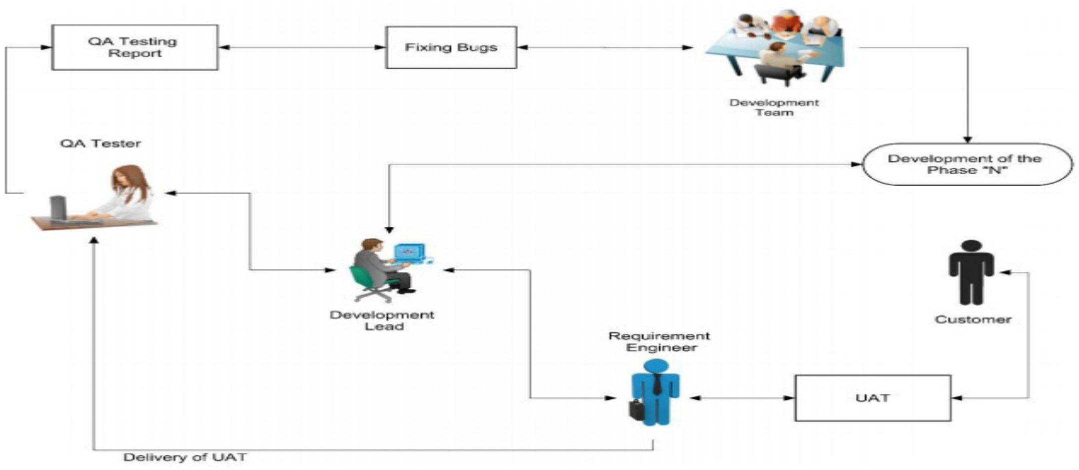


Iteration Cycle Development Process:

Once the final Plan and Prototype is ready, next the entire Project Plan is parted into several Iteration Cycles. In each iteration a phase of task will be completed.


Iteration 0
Iteration 1
Iteration 2
Iteration n
Project Setup Plan
Plan, Develop & Test Feedback
Plan, Develop & Test Feedback
Develop & Test Release Product

Here as per Refined Agile Model (RAM) no new requirement will be accepted within the planned iteration cycles. For additional requirements new iteration cycles will be defined after completion of the actually planned iteration cycles. This will avoid increase of time and confusion within the development.



Overall Process Flow:

In the first cycle “Prototype Locking” will be done, which will make the customer and developer company sure about the final outcome.

And, next there will be ‘n’ number of Iteration Cycles to complete the project development which will be defined by the Project’s Technical Lead discussing with Requirement Engineer/Manager. No new requirement will be accepted within these cycles.

Finally, to meet customer desire, with the new requirements (that customer asked even after development of the final Plan and Prototype) another `n` number of Iteration Cycles will be cleated and continued.

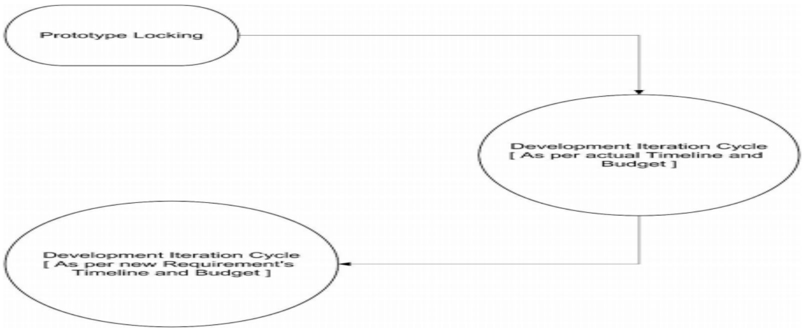

Prototype Locking
Development Iteration Cycle [As per actual Timeline and Budget]
Development Iteration Cycle [As per new Requirement's Timeline and Budget]

Figure: Overall Process Flow

Therefore, the followings are the salient features of the proposed model:

1) In the Requirement Analysis Phase a Requirement Engineer from Software Company will be sent to Customer-end (under the supervision of a Requirement Manager) who will be giving sufficient time at the Customer Premises to understand customer's need. Point to be noted here that, this Requirement Engineer and Requirement Manager will act on behalf of the Client always throughout the project development and will deal requirements on behalf of Client. Their role will be understanding Client's need and acting like Client with the Software Developing Company.
2) After collection of the requirements by Requirement Engineer, these requirements will be listed down and locked by the Requirement Manager who will be vetting and confirming these requirements from Customer.
3) Then, the Requirement Manager will make a Prototype with the help of an UI Engineer or Documentation Expert. This will help the customer to visualize about what he/she is ultimately going to get in the delivery. Thus, the Prototyped Model is practiced in this phase.
4) Next, the Project Manager will make an almost complete plan of the entire development. As per the proposed rule, the maximum possible plans should be done in the beginning, and the Unseen and Undefined Requirements will be added later on in different Phases.
5) Project Manager will make different Managed Iteration Cycle of Development to break-down the entire development into different Phases. The Requirement Manager will help to set a priority sequence of Phases and Tasks.
6) In each Managed Iteration the Cycle-Plan, Development, Testing, Feedback (UAT), Fixing, and Deployment of each Phase will be done. In each Phase's UAT phase the Requirement Manager will be locking UAT feedbacks sitting with customer, if required in the Customer's premises.
7) If one Managed Iteration Cycle has any missing or additional requirement, those are developed and managed after the Contracted Development completes. Therefore, the Cycles will be added totally separately. Therefore there will be no addition of requirement within an ongoing cycle. Thus the entire process is Incremental as well.

Some other recommendations for the proposed model:

1) Understanding the IT knowledge depth of customer is very important, because this will help to evaluate, deal and collect requirement from customer accordingly.
2) Practically it is very tough to gather the entire requirement in the analysis phase, and some requirements will remain uncollected, therefore new requirements will appear while development is ongoing or even later for sure. But still, Analysts will try to collect the maximum possible requirements in the Requirement Analysis phase.

3) Technical Analysts should also be tagged while collecting requirement from customer if the Requirement Engineer is a non IT person. This will help to get vetting on the feasibilities of what customer is asking for.
4) Making customer clear about the feasibility of his expectations is necessary, because this helps to avoid future impractical expectation of customer in the project. Therefore, it is very important for the Analysts to identify the feasibilities of client requirements accurately.
5) Proper documentation is required to collect requirement, prepare final deliverable list, generate Software Requirement Specification (SRS), make QA and UAT checklist etc. so that in each phase can be specified and locked through proper documentation.
6) Presenting prototype, wireframe and demo of the final product is necessary to let customer have a clear view on what they are going to get once the development is completed. This will help the Analysts help to avoid wrong analysis of customer requirements as well.
7) Frequent communication from the beginning phase to deployment phase is highly required to avoid any type of Requirement Volatility. This frequent communication will be done by the Requirement Manager. If number of demo presentation is increased through the entire development phase, then the risk of wrong development decreases; because each demo presentation phase gets vetting from customer.
8) The Project Managers have to be bold and logical to reject any new/change requirement within a running development phase if it is not feasible accept those within the running one.
9) If any new requirement appears in the development or testing phase, it should be tagged with the next development phase, but not with the running one.
10) QA Testers should be tagged with the project in the Requirement Analysis phase to prepare accurate checklist for testing. QA Tester should also be tagged with customer while they are providing feedback in the testing phase so that UAT of customer and feedback of QA Tester matches perfectly.

4.1 Validation of the Methodology

To validate that “Refined Agile Model” works better than theoretical agile models and other models, it was implemented on two projects named:

- Training Management System, Implemented for Sudokkho NGO
- MicroERP System for E-Commerce Business, Implemented for Proshiddho Limited

RAM worked wonderfully and a table is presented below that explains how RAM solved the problems that theoretical agile couldn’t solve:

PROBLEM	SOLUTION BY RAM
Customer can’t provide their entire requirement in Requirement Analysis phase	A Requirement Engineer (who is from Engineering background with concepts of Business) visited client’s premises 5 times and understood their business need. Finally he created the Requirement List on behalf of the customer and explained it to the Development Lead
Customer don’t provide documentation for their requirement	The Requirement Engineer and his senior, the Requirement Manager prepared the Requirement List and made proper Project Documentation on behalf of the customer.

Customer are not much IT concern and can't express their exact requirement accurately throughout the Development phases	Throughout the entire Development Phase the Requirement Engineer was dealing with the Development Lead as Customer on behalf of the Customer. The Requirement Engineer checked the development frequently and gave his feedback when he found mismatch with Customer's requirement.
New Requirement arrives in the Development Phase	As a Prototype of the entire project was prepared in the beginning and was acceptance and signed by the Customer, therefore there were be no scope of requirement volatility. Customer's new/change requirements were moved to make new iteration cycle after completion of the planned iteration cycles.
Customers don't keep much communication in Development and Testing Phase	As there were Requirement Manager and Engineer who acted as Customer to the Development Lead/Team there were no need of Customer to sit with the Development Company. However, this happened because Requirement Manager/Engineer kept frequent communication with Customer to synchronize them about the Development.
To keep customer satisfied, the Software Companies accept new requirements even when a Development and Testing Phase is running	At first all possible requirements were collected, then these were divided into some Development Phases (Iteration Cycle). The overall plan was done in the beginning and prototype was made as per that plan. As the Prototype was signed, and Requirements were locked properly, customer didn't pressurize to add much new/change requirements.
Developers are not fully dedicated for one single project, and gets involved suddenly with other projects, therefore there remains scope of change in project team.	As Requirement Engineer and Manager documented the projects properly and listed all Requirements perfectly, it helps new team member to understand about requirement and from where what to do.

So, finally, the above results were evaluated and the theoretical agile models and RAM was compared, and it is found that RAM performs better. Here goes the comparison chart among theoretical Agile Models and Refined Agile Model (RAM):

PROBLEM	**Can the Model give solution?**			
	Scrum	**XP**	**RAD**	**RAM**
Customer can't provide their entire requirement in Requirement Analysis phase	No	No	No	Yes
Customer don't provide documentation for their requirement	No	No	No	Yes
Customer are not much IT concern and can't express their exact requirement accurately throughout the Development phases	No	No	No	Yes

New Requirement arrives in the Development Phase	No	No	No	Yes
Customers don't keep much communication in Development and Testing Phase	Partially	Partially	No	Yes
To keep customer satisfied, the Software Companies accept new requirements even when a Development and Testing Phase is running	Yes	Yes	Yes	Yes
Developers are not fully dedicated for one single project, and gets involved suddenly with other projects, therefore there remains scope of change in project team.	Partially	Partially	Partially	Yes

So, this way, implementation of Refined Agile Model (RAM) in the two selected projects it is found that this newly proposed model truly works better in the Bangladesh's Software Development Environment.

5. FUTURE WORK

In this paper different type, price and volume of software projects were analyzed to understand different SDLC and their implementations and obstacles in the Bangladesh's Software Company. All these experiments have found that, the requirement engineering process is not being proper and requirement volatility in huge and uncontrollable in many cases.

However, all the observations of this paper were made in the Bangladeshi Software Company regarding to their SDLC practices. Therefore, this paper is limited within the Bangladeshi Software Company. This same study can be made on the foreign companies and the projects as well. In this way this study will have a global and wider solution for the problems occurring in the requirement engineering process.

Besides, the proposed Methodology will be best suitable for Medium to Large scale Software Development, but is not good for small application development. So, further study can be done to suit the Development Methodology for all type of developments.
As a further scope of study, this same research can be done for the developments that were done on the open source CMS and other regular type of web based projects.

6. CONCLUSION

This paper tries to find out that why does the Agile Model not suitable for Bangladesh's Software Companies and how do the problems that the Software Companies face for using the methodology can be solved. Methodologically, this paper collected data through survey and interview. The findings of this paper reveal that agile development model is good for small and medium projects and in favor of the incremental model of delivery, many Software Companies in Bangladesh think it is suitable for local software projects. But the working environment of software development of Bangladesh does not very compatible for using Agile Model.

In the case of Agile Model, it is required to keep frequent communication with client from the beginning of the project up to its end. But, in practical it becomes difficult for Software

Companies to sit and communicate very frequently with their customers in Bangladesh. This ultimately leads to a communication gap between clients and Software Company. For such communication gap neither Software Companies get scope to understand client’s need properly nor clients understand that whether the Software Companies are understood his expectation accurately or not.

Another big problem with traditional Agile Model in Bangladesh is proper documentation. In Agile Model detailed documentation is not required. But in the case of Bangladesh, maximum clients change their requirements randomly. To manage a project understanding and managing the change-requests properly is highly required. And, due to lack of documentation in Agile Model project managers sometimes fail to understand and manage the changing behavior of the projects. Besides, as in maximum of the Bangladeshi Software Companies resources are limited, therefore in case of large projects the Companies shifts their developers suddenly from one project to another based on their need. But due to lack of proper documentation, newly joined developers cannot understand the project suddenly which also hampers the development of the project.

Moreover, in Agile Model there is no distinction between small or big changes. As every change has a cost and in Bangladesh as random change requests appear from client-end within a development life cycle, therefore the traditional Agile Model can’t fully identify the effort against the large changes. This ultimately increases the cost of project which may bring loss for the project.

All these studies mainly highlighted the problems of various models, especially Agile, that increase Requirement Volatility. But, there are some other minor SDLC practices which are not studied and analyzed. Moreover, these studies are not done on the Bangladeshi software projects. For this reason, the paperhas given priority in finding the remedy of Requirement Volatility and thus finding a way to manage the Requirement Engineering process effectively in the context of Bangladeshi Software projects.

In this paper, Requirement Volatility is found as a major problem that appears in the traditional software development life cycles (SDLCs) that are practiced in Bangladesh.

To ensure the solutions of the problems a better solution of new Software Development Methodology is proposed which is named as prototyped Agile Model.

Refiled Agile Model prepares a Prototype that defines an almost complete Project Plan in the beginning. Next, the entire project plan is placed in different iterations and are delivered time to time as per customer requirement. In each iteration the requirement collection, development, testing and deployment will be incremental but will be within the scope of the designed Prototype. Thus the entire model becomes a combination of Incremental Iterative (Agile) and Prototype Model.

In conclusion, by using the new model named “Refined Agile Model” practically, positive outcome was found. So, it is expected that overcoming all the bindings of traditional and theoretical models this model will be able to server the purposes of managing Software Development Life Cycles in Bangladeshi Software Company.

REFERENCES

[1] Moretaza, T. (2016) Ensuring priority for local Software Companies first challenge. The Independent. [Online] 27th June. p.14. Available from: http://www.theindependentbd.com/post/49371. [Accessed: 03rd July 2016].

[2] BASIS (2016) BASIS Members Directory 2016. [Online] Available from: http://www.basis.org.bd/index.php/members_area. [Accessed: 28th July 2016].

[3] A. Tiwana, and M. Keil, "The one minute risk assessment tool," Communications of the ACM, 2004.

[4] M. Ben-Menachem, Software Configuration Management Guidebook, McGraw-Hill International (UK) Limited, 1994.

[5] C. Jones, Software Engineering Best Practices: Lessons from Successful Projects in the Top Companies, McGraw-Hill Osborne Media, 1st ed., 2009.

[6] K. Wiegers, Software Requirements, Microsoft Press. 1999.

[7] M. Sudhakar. Managing the Impact of Requirements Volatility. Master Thesis. Department of Computing Science, Umeå University, Umeå, Sweden. 2005.

[8] V. Rajlich, "Changing the paradigm of software engineering," Communications of the ACM, vol. 49, no. 8, August 2006.

[9] Costello, R. and Liu, D. (1995), "Metrics for Requirements Engineering": Journal of Systems and Software, Vol 29 (No. 1), pp. 39-63 MIL-STD-498. 1994. Software [10]Development and Documentation. U.S. DoD.

[11] T. JavedManzil, M. Quiser, and S. Durrani, "A study to investigate the Impact of requirements Instability on Software Defects", ACM SIGSOFT Software Engineering Notes, 29 (3), May 2004, pp:1-7.

[12] Zowghi, N. Nurmuliani, A study of the Impact of requirements volatility on Software Project Performance, Proceedings of the Ninth Asia-Pacific Software Engineering [13]Conference , APSEC 2002, Gold Cost, Queensland, Australia,04-06 Dec 2002, pp:3-11.

[14] Lamsweerde, A. Requirements Engineering in the Year 00: A research perspective. In proceeding of the 22nd International conference on Software Engineering (ICSE'2000), Limerick, Ireland, 5-19, ACM Press.