



SOFTWARE ENGINEERING REQUIREMENTS PROBLEMS AN INVESTIGATION STUDY IN JORDANIAN CONTEXT

MOHAMMAD TARAWNEH

Asstt Prof., Department of Computer Information System, AL-Balq'a Applied University, Jordan

E-mail: moh@yahoo.co.in,

ABSTRACT

This paper presents an overview of the field of requirements engineering (RE). It describes the main areas of RE practice. When requirements practices is good this may lead to accelerate the development of software. The process of defining business requirements birds the stakeholders with shared goals, vision and expectations. In order to increase the accuracy of requirements you have to involve substantial user in establishing and managing the changes to agree upon requirement so emphasizing that the functionality built which enable users make the important business tasks.

Keywords: *Requirements Engineering, Requirements analysis, Requirements Management.*

1. INTRODUCTION

Requirements engineering (RE) can be defined as problem context identifying, putting the customer's needs within that context and delivering a specification that meets customer requirements within that context. There are many requirements methods that purport to do this, like, soft systems methodology [2], scenario analysis [1], and UML [3]. Sometimes they work, sometimes they do not. The implication of such requirements methodologies, if we can label at least aspects of them as such, is that the application of 'x' method will produce the right requirements irrespective of the problem's characteristics. This is conventional wisdom and don't surprise, the creators and vendors of requirements methodologies claim, with only one exception [4] that their approach is a hammer and all problems are nails. But there are many factors other than just application of a requirements methodology that influence the success or failure of software projects in practice, in this paper we focus only on requirements engineering. As Davis and Hickey mention, that as a researchers we have to be aware of what is gonging in practice filed in order to put our research in that context [5], without (RE) methodology we will always practice our research in context free , bubble for practitioner view documentation regardless SW project successor failure. The main problem of this research is to investigate the main problems that meet to software project mangers on requirements elicitation for software projects in Jordanian firms.

2. LITERATURE REIEW

According to the Extreme chaos report [6], understanding requirements was ranked as number seven of importance for project success. The report recommends starting with base line requirements and then adding more features as needed. This can help to reduce requirements changes, help user and sponsor to see results faster and help project managers to prepare and link the need and criteria for the next phase of project. A study by Martin et; al. [7] found that the misunderstanding of basic requirements lead to longer project duration, weak or incomplete requirements and can lead to an unsuccessful project. Mursu et; al [8] mentioned that there is a possibility to develop unusable software due to misunderstood requirements. Management of Requirements establishing and maintaining an agreement between the client and supplier, that includes specific information about technical component, functionality and performance that will be included in the software release. This agreement forms the fundamental basis for planning, performing, estimating, and tracking the project's activities. As new requirements are added to the release or existing requirements are omitted or modified from the set, the release cost, schedule, and quality are changed. These changes to the requirements after the basic set has been agreed by both clients and maintainers are known as requirement's volatility. Requirement's volatility is common and will know in the software industry. Requirements volatility According to (Jones) [9] more than 70% of large



applications (i.e., over 1000 function points) experience.

Research by (Standish group) [10] and (Gibbs) [11] conclude that poor requirements at first and inadequate risk management lead to low quality and poor software delivery success rates. The Software Engineering Institute (SEI) believes that organizational processes are the main factor in the predictability and quality of software and requirements management is one of the key that contribute in their model of a mature organization [12]. A study by Boehm [13] identified continuous requirements changes as a significant risk. Mursu et al [8] suggested freezing requirements at a fixed point, an opinion is rejected by Keil et; [14], who argued that requirements should not be frozen at any point because this affects business environment, and leads to project with little or no flexibility in changing specifications. According to Addison and Vallabh [15], continuous requirements changes lead to delay in project schedule and affect the estimated budget. In addition Technical requirements should be defined early. In many cases a prototype is built and tested to develop a good understanding of the system's needs and requirements. A prototype is particularly having benefit in situations where the client is unsure about the requirements. A clearly defined requirements specification that agreed upon by both the client and the development team emphasize that the client's needs are understood before starting work design. The requirements document is, in effect, a contract between the client and the development team. It specifies what the product must do, but not how. It serves as a guide for design activities and as the baseline for controlling any technical changes that may be needed during the project.

3. METHODOLOGY

The study has adapted the descriptive, field and analytic methods. An office survey and reviewing of theoretical and field studies and researches were conducted in order to crystallize the bases of the research and to stand at the important previous studies which consists a vital support for the study through their epistemic axes. A comprehensive survey and analyzing data that collected from questionnaires by using statistical ways were conducted to implement the field analytic research. The study based on a developed questionnaire that organized depending on the previous questionnaires; this questionnaire has adjusted to fit the Jordanian environment. A personal interview

was conducted with some managers of engineering projects in Jordanian environment.

4. POPULATION OF THE STUDY

The population of the study consists of all Jordanian institutions (n=96) work in three key sectors as shown in the table (1) below. 100% of these institutions were selected for this study.

Table (1)
Population of the study.

Institution/sector	Number
Banks sectors	17
Software engineering	62
Insurance companies	17
Total	96

5. SAMPLE

The sample number was (276) which consists (100%) of Software engineering managers (males and females) at four sectors. (276) questionnaires were distributed to managers, the returned questionnaires were (242); (26) questionnaires were excluded because they weren't valid for statistical analysis, so the valid questionnaires were (216). Only 20 managers were interviewed because the others excused because of they were busy or in traveling.

Table (2)
Frequentative distribution of the sample

variable	category	number	Portion
Gender	Male	166	76.9%
	female	50	23.1%
experience	≤ 5 years	31	14.4%
	6-10 years	37	17.1%
	11-15 years	40	18.5%
	16-20 years	51	23.6%
	>21 years	57	26.4%
education	Baccalaurean	157	72.7%
	High certificate	59	27.3%

Table (2) shows that the most of the sample are males (n=166) which consist (76.9%) of the sample where the females portion consists (23.1%) of the sample. For education variable, bachelor degree took the high portion (72.7%) whereas high certificates portion was (27.3%). For experience

variable, the high portion went to (11-15 years) which consisted (18.5%); the lowest portion went to (5 years and less), (14.4%).

6. DATA ANALYSIS

Table (3)

Means and Standard deviations of subjects' perceptions towards standard requirement factor *significant at level ($\alpha \geq 0.05$)

Symptoms	Severity degree			T.Value	Sig
	Mean	Std	Severity		
There are management and following for project's requirements by teamwork	4.04	0.96	high	11.260*	0.00
Data inputs than are given to teamwork's are not clear, contrastive, unsuitable and surface.	4.26	1.01	high	13.526*	0.00
When reaching the final stage of the project, the need and desire for outputs could disappear.	3.83	0.99	high	8.443*	0.00
Teamwork jumps from task to another task before finishing the first task.	3.75	0.97	high	7.211*	0.00
Real problems are realized lately.	3.76	0.99	high	7.598*	0.00
Using the correct way in collecting system's requirements.	4.31	0.97	high	14.019*	0.00
There is a large amount of information bigger than individual ability to deal with	3.85	0.99	high	6.93*	0.00
Project and user's requirements change during work which leads to implement it later	3.81	1.03	high	6.36*	0.00
If we have two results that have same opportunity of occurrence, we get undesired result which means that matters don't go as we wish	3.79	1.03	high	6.28*	0.00
The project starts according to customer desire and current priorities of the plan with little affection by real	3.66	1.07	high	6.24*	0.00

Symptoms	Severity degree			T.Value	Sig
	Mean	Std	Severity		
objectives.					
The customer has raw ideas about the project which we intend to implement it	3.62	1.00	high	15.35*	0.00
The outer groups don't tell the teamwork all what they know about the project even if this information is very important for the project.	3.82	0.96	high	22.61*	0.00
The final user committed by working with teamwork in order to develop the system	3.69	1.04	high	16.84*	0.00
High level of interference from the final user with development team of software engineering projects	3.77	0.92	high	22.58*	0.00
Customers (people) think that work is easier /smaller than as it is.	3.58	1.05	high	9.79*	0.00
Customers change their thoughts towards project's plan or its features.	3.67	1.03	high	15.81*	0.00
There is a high level of trust between the final user and teamwork	3.74	1.04	high	19.20*	0.00
Teamwork affect by the large number of users and customers of the system.	3.70	1.01	high	17.87*	0.00
Teamwork don't know what suppose to do during the different stages of the project	3.54	1.02	high	9.78*	0.00
Ideality harms to plan and costs of the project as well as customer satisfaction.	3.61	0.99	high	11.11*	0.00
Total mean	3.67	0.58	high	17.52*	0.00



Table (3) shows that the general mean of items related to standard requirement variable in terms of its severity is (3.89), SD (0.54); the item there are management and following for project's requirements by teamwork, ranked the first rank with mean (4.13) and SD (0.92); whereas item the Teamwork don't know what suppose to do during the different stages of the project, ranked the last rank, M (3.3.), SD (1.02). Also the table (3) shows that the general mean of items related to standard requirement variable in terms of its frequency degree is (3.81) , SD (0.56). the item, there are management and following for project's requirements by teamwork, ranked the first rank, M (4.01), SD (0.98), whereas the item (96), , the project starts according to customer desire and current priorities of the plan with little affection by real objectives, rank ed the last rank, M= (3.59), SD = (99). The means for all items of this dimension were at high degree which indicates that the success of projects in terms of standard requirement, the outer groups don't tell the teamwork all what they know about the project even if these information are very important for the project., ranked the first rank with mean (3.82) and SD (0.96); whereas item (94), teamwork don't know what suppose to do during the different stages of the project, ranked the last rank, M (3.54) , SD (1.02).

Also the table terms of its frequency degree is (3.59), SD (0.60). the item the outer groups don't tell the teamwork all what they know about the project even if these information are very important for the project, , M (3.73), SD (0.98), whereas the item (94), which indicates that the success of projects in terms of standard requirement.

7. INTERVIEW ANALYSIS

Question) what are the main problems that meet to software project mangers on requirements elicitation for software projects in Jordanian firms?

Table (4)
Mean and percentages of requirements

Item	Mean of responses	Percentage	Rank
New ideas come from the last user after finishing the system	14.49	72.45%	4
User doesn't understand his requirements especially when	15.56	77.80	3

the system is new and there are no experts in field which needs to be computerized			
New administrative decisions may cause change in requirements	16.58	82.90	2
Person experience has an important role in identifying his requirements	17.83	89.15	1
Administrative movements force team work to deal with new persons	13.25	66.25	5
Documentation of requirements is needed in order to control change in requirements	10.74	53.70	6

The table (4) shows that person experience at his work has a big role in identifying his requirements clearly (89.15%). Some new administrative decisions may lead to change in requirements (82.90%). User's misunderstanding for his requirements, especially when the system is new, and if there are no experts in field which needs to be computerized (77.80%). New ideas that come from the user after finishing the system may delay delivering date, and the project may exceed the identified budget (72.45%). Administrative movements from one location to another force the team work to dealing with new persons (66.25%). Finally documentation models have to be used in order to control changes in requirements because that may affect other factors such as scope and reliable estimation (53.70%).

During the filed study that include 576 software project's in Jordanian firms the following classification was noticed that describe the different aspect listed below.



Typical Project Timeliness in this Project Management System (compared to originally approved project plan, answers must total 100%)						
Actual duration compared to original plan	<50% Very Early	51-94% Early	95-105% on-time + 5 %	106-149% Late	150-249% Very Late	>250% Maybe Never
% frequency of occurrence Row must = 100%	0%	23%	37%	19%	17%	4%

Typical Project Budget Performance in this Project Management System (compared to originally approved total cost budget, answers must total 100%)						
Actual cost compared to original plan	<50% Big savings	51-94% Savings	95-105% on-budget + 5 %	106-149% Excess cost	150-249% Large excess cost	>250% Broke the bank
%frequency of occurrence Row must = 100%	7%	18%	37%	27%	9%	2%

8. CONCLUSION

Upon the analysis of the study data through the program of statistical analysis (SPSS), the researcher approached the following results:

Table (3) shows that the general mean of items related to standard requirement variable in terms of its severity is (3.89), SD (0.54); the item there are management and following for project's requirements by teamwork, ranked the first rank with mean (4.13) and SD (0.92); whereas item the Teamwork don't know what suppose to do during the different stages of the project, ranked the last rank, M (3.3.54), SD (1.02).

From the researcher view, weakness of workers in collecting requirements and disability to use the right methods in collection process is from the reasons that lead to make these requirements unclear. Each project has its privacy in requirements collection process by the user and analyzing these requirements; also if these requirements don't identify accurately, this may cause failure to the project; documentation mechanism considered as an important requirement that is through revising available documents in Jordanian institutions there was no documentation mechanism for requirements in order to revise it with systems users when finishing these systems at the end. User's misunderstanding of needed requirements is one of the reasons that lead to imperfection of requirements and changing them during working on the system. Identifying project framework, clarifying and defining objective and users' involvement play an important role in defining the project requirements. User who will be

joined to the teamwork should be an expert in his work and use his experience in defining project requirements. Available culture play an important role in defining project requirements that is if this culture support computerization, the institution's vision will be clear which in turn helps in defining requirements; controlling sudden changes is possible through cooperation between the institution and system development team. Here, managers and systems analysts should manage these requirements efficiently in order to cope this important dimension in success of software engineering projects that is changing system requirements lead to: failure, exceeding the limited budget, delay in delivering time and user dissatisfaction.

For the importance of the big role that user involvement plays in success of software engineering projects, this dimension is very important to clarifying work objectives and making balance among teamwork members when they sharing their roles and enable them to know unclear things in the institution, so, involving the user within teamwork helps in resolving some of vagueness in specific items of the work which is to be achieved by software engineering team in the institution. User involvement process minimizing resistance change for new system that is the user who intended to be involved within teamwork should have a positive role to protect the project and tries to convince the others with the importance of the project for the institution in all; also this user should have a role in communication flexibility process between system development team and system users because he/she is the only one who can explain the teamwork and his /her coworkers views.



Finally, having a central repository for requirements is clearly correlated with project success. This is good news, because it is relatively easy to do. In fact, it is difficult to understand why a project would not have a central repository for requirements given the technology available today. As with any study, there are limitations to this research. While some of the limitations must be accepted, as little can be done to overcome them. The scope of the research has been limited to Jordanian firms. It is recognized that such firms may have cultural characteristics which distinguish them from other firms. Several limitations became apparent during the survey stage of the research. The poor response rate related to questionnaire, because most of software project managers don't understand the questionnaire words related to software projects symptoms. Future directions in research this research must be extended to investigate how the knowledge base systems can contribute to software project manager to deal with ambiguous requirements.

9. RECOMMENDATION

When requirements practices is good this may lead to accelerate the development of software. The process of defining business requirements binds the stakeholders with shared goals, vision and expectations. In order to increase the accuracy of requirements you have to involve substantial user in establishing and managing the changes to agree upon requirement so emphasizing that the functionality built which enable users make the important business tasks. Software requirements engineering encompasses the two major sub domains of requirements definition and requirements management:

Requirements Definition is the collaborative process of collecting, documenting and validating a set of requirements that encompasses an agreement among basic project stakeholders. Also requirements definition is subdivided into the main process areas of elicitation, analysis, specification and validation processes. From a pragmatic perspective, requirements definition serves for requirements that are good enough to enable the work team to deal with design, testing and construction at an acceptable level of risk. As discussed, the risk is defined as the fear of having to do expensive and unessential rework.

Requirements Management involves working with known set of product requirements throughout the product's development process and its operational life. It also involves managing changes to that set of

requirements throughout the project lifecycle. In practice, requirements management includes selecting changes to be incorporated within a particular release and emphasizing effective application of changes with no adverse apply on schedule, plane or quality. An effective and well requirements definition and management solution makes exact and complete system requirements, while it help organizations to improve communications in an effort to better bird IT with business needs and aims. It includes a set of industry best practices for each category, as well as tools to help and accelerate requirements activities. Strategies for better requirements A variety of practices can help software teams bridge communication gaps and do a better job of understanding,

Documenting and communicating customer needs. several best practices in the categories of requirements elicitation, analysis, specification, validation and management. We discovered that it is not the number of users involved that is important, but rather managing the size of the project in terms of functionality; it is not the requirements methodology per se, but rather use of an appropriate software development methodology into which the requirements methodology fits; it is not scope creep, but rather that scope is well defined when it creeps; it is not a project manager experienced in the application area, but rather a project manager who manages requirements effectively; it is not necessarily having complete requirements at the start of the project but rather completing the requirements at some stage during the project; and projects that had a central repository for requirements were more likely to succeed. The most important correlations for project success are to get good requirements and to manage those requirements effectively. Getting good requirements means a number of things. Some that are important are a high level of customer/user involvement, high-level sponsorship throughout, to scope the project effectively and it is critical to have a good project manager who can manage, rather than one who just happens to know the application domain.

Through the previous results, the researcher could prescribe a group of recommendations to how deal with requirements elicitation in Jordanian firms included in the following:

- **Requirements Elicitation**

Define the product vision and project scope.

The system vision is the long-term strategic concept below the most important purpose and form of the



new system. The vision could also describe the system's place and arrangement among its competition in its market or operating environment. The project scope is the part from of the product vision that the current project will address. The scope draws and plans the boundary between what is inside and what is outside for that project. Ambiguous project scope, the project will face open invitation to scope creep. Before the eliciting requirements, teams should have understood both the product vision and the project scope exactly to do their work in successes manner.

- **Identify stakeholders, customers and users.**

To software development groups, users are a subset of customers and customers are a subset of stockholders. Also user is subdivided into many classes according to their exactly need. This subdivision is help to the project outcome, its have benefit to get the key stakeholders commitments for their participate in requirement definition. Customer participation has a lot of help during the requirement management also their perspective is needed when the teams change the decision, assessing the application of these changes and adjusting the priority of requirement. Any software project have to identify exactly its main requirement decisions makers and the right way to make decisions to emphasize that right people who can make right decisions and on its time in right place.

- **Select product champions.**

Product champion can be defined as representative who will act as literal voice of customers in every user class, in perfect and ideal cases those product champion are real users who represent there peers in significant user classes in practical life. Products managers are often do this role in commercial software development organizations. The level of engagement of product champion should be determined from the first may be you need to share them in one or two workshops, and sustain engagement contact points between product champions frequently. Team work should choose the right representative to do this role product champions must understand exactly the business requirements which represented by the product vision and project scope. Good product champion should be collaborative and have good communication channels with other member's team to ask input resolve conflicts and solicit feedback.

- **Choose election techniques.**

The way analyst can use for requirement elicitation determined by the extent of stakeholder involvement and the access of analyst to stakeholders. When stakeholder is locally available workshop or work cases have a lot of benefit while when they are geographically separated questioners and survey right be necessary. Individual interview with experts have a lot of benefit to take the good information as are analysis models and building good and interactive prototype these elicitation techniques are not exclusive to have good with high quality and complete information elicitation technique analyst must use a lot of communication channels and the team have to excesses in variety of elicitation techniques.

- **Explore user scenarios.**

From the elicitation discussions that concentrate on users and how they use the system give greet result for requirement analysts. In general its more beneficial for the users to know the business tasks and usage goals them to describe and define all of the functionality they except to see in system. Team work should explore user scenario in order to emphasize that requirement they develop when it implement will allow users to gain their goals. But this is not sufficient to replace the need to define functional requirement in details.

Requirements Analysis:

- **Create analysis model.**

The general language that describe requirement specification found full of unclear vocabulary and a lot of gaps, its more preferable to represent requirement in many ways to give reader richer information, one of this good way is analysis model that visually represent information, graphical diagram allow reviewer to spot immediately missing requirements. Rather than examining this missing by full reading the textual specification. This model that give communication for team work with higher level of abstraction, which allow them to see big picture without getting mired in all of details.

- **Build and evaluate prototype.**

A prototype is a partial, preliminary or possible solution to the requirements. Which give chance for product champions to deal with a simulation or part of the final system they build on? Prototypes are more touchable than written requirements, invite prototype specifications, they are a way to bring use cases to real life imaginary mind when development team they make an experimental step into solution space, which is a valuable way to



assist and redefine requirements, but it cannot replace detail documentation of functional requirements. A prototype or group of screen designs does not give the logic that happens behind the scenes for businesses. A prototype has some disadvantages like it does not describe exactly the complete behaviour of the product when the users make some actions under certain conditions. Another disadvantage of a prototype is that it does not give exactly how all exceptions and error conditions are going to be dealt with, although this information is essential if teams want to build robust software. When used with right audiences, prototypes are a good way to help team analyze and develop existing requirements for a new system.

- **Prioritize requirements.**

Any software development organization has a limitation in resources and time, so any work team should determine which requirements are more important and urgent to put them first. This requirements prioritization gives the chance to team to apply the right step of user functionality in right sequence. Prioritization should be done in a collaborative manner that involves both customer and technical perspective in order to balance value of customer and cost and technical risk.

- **Look for ambiguities.**

Use of natural language in requirements writing is full with ambiguities, negative requirements, unclear subjective terms, complex logic, abstraction adverbs which may lead to different understanding by different readers. Team should agree on specific dictionary or terms before requirements are written to correct ambiguities that are cheaper for team in compilation to deal with disappointed customers.

- **Store requirements in a database.**

Storing requirements in commercial management tools (CMT) give the team a lot of advantage over the textual documents such as:

Easier to add additional information about different classes of requirement.

Give good mechanism for retaining requirements that have been proposed.

It is easier to track requirement status.

It also makes easier to deal with groups of requirements that are wanted for multiple releases in the future.

Facilitates communication and collaboration among distributed teams in organization.

Trace requirements into design, code and tests.

There is a lot of value to retain and link any software functional requirements to its origin may be by use

case or business rule, work team should include any small or trace information that connects functional requirements to design elements associated with it, codes, segments and tests to facilities and accelerate software maintenance. Requirement management tools help in managing traceability data.

Requirements Validation:

- **Review the requirements.**

Formal team review of requirements documents are the most and highlight effective quality practice that is available to software teams. This review gives an indicator on the understanding levels of requirements. The requirements analyst have to document requirements in order to emphasize they can communicate in clear, effective and efficient manner to various stakeholders. The analysts should make complete views about the requirements selected from textual requirements, scenarios, prototype, test...etc. that could be reviewed in order to get software that meets the business needs in good manner, all stakeholders of project should review the requirements.

Create test cases from requirements.

Work teams should start testing as soon as they get in hand some requirements, it has great value to find and check the problems in the use cases by driving test case from use cases and scenario.

Requirements Management:

- **Manage Requirements Versions.**

Requirements evolve, created and developed during all courses of projects so it's important to any project to track all versions of requirements specifications documents also individual requirements, this tracking of version helps teams to insure that all members of team use the current version of requirements baseline.

- **Adopt change control process.**

As the requirements have baseline, any modification on it must follow established change control process. Which give consistency in requirements change process, assessment, approval or rejection, before any requirements changed the team should agree on formal written change control processes in place.

Perform requirements change impact analysis.

In order to make appropriate decisions in change process developers have to evaluate the potential apply of these changes before committing to apply it.



- **Store requirements attribute.**

Requirements attribute help team to give full understanding of each requirement separately. Attribute to track must include priority, status, origin, assessment method, and risk and version number. So teams have to store attributes with requirements in order to emphasize that all essential data to communicate and prioritize requirements. Track the status requirements.

To track the project situation team can report the situation of each functional requirement from the base line, they can check it by number of possible requirements such as approved, proposed, implemented, rejected requirements. This tracking can easily validate the health status of project which avoids team UN essential status meeting.

- [12] Software Engineering Institute (SEI), Available http://www.sei.cmu/productlines/frame_report/productLS.htm [Accessed 12/1/2012]
- [13] Boehm B. W., "Software Risk Management Principles and Practice", IEEE Software Vol. 8 No 1, 1991, pp32-41.
- [14] Keil, M., P. Cule, K. Lyytinen, R. Schmidt. "A framework for identifying software project risks". Communications of the ACM, November 1998, pp. 76-83.
- [15] Addison. T and Vallabh. S. "Controlling Software Project Risks- an Empirical Study of Methods used by Experienced Project Mangers", Proceedings of SAICSIT 2002, pages 128-140.

REFERENCES:

- [1] Carroll, J. (ed.), Scenario-Based Design: Envisioning Work and Technology in System Development, Wiley Publications, 1995.
- [2] Checkland, P., Systems Thinking, Systems Practice, Wiley Publications, 1981.
- [3] Booch, G., Rumbaugh, J., Jacobson, The UML User Guide, Addison Wesley, 1999.
- [4] Jackson, M., Problem Frames, Addison Wesley, 2001.
- [5] Davis, A., and A. Hickey, "Requirements Researchers: Do We Practice What We Preach?", Requirements Engineering Journal, 2002, 7, pp107-111.
- [6] Standish Group, CHAOS, http://standishgroup.com/sample_research/PDFpages/extreme_chaos.pdf (Accessed 31/10/ 2006).
- [7] Martin, E. W., Dehayes, D.W., Hoffer, J. A. and Perking. W. C.(1994) Managing Information Technology : What Mangers Need to Know 2nd edition , prentice Hall, new jersey.
- [8] Mursu, A., Soriyan, H.A., and Olufokunbi, K.C. 1996. Toward Successful ISD in Developing Countries: First Results from a Nigerian Risk Study Using the Delphi Method
- [9] Jones C. 1994. "Assessment and Control of Software Risks"; Prentice-Hall International: Englewood Cliffs NJ; pp. 93-98.
- [10] <http://www.standishgroup.com>, Accessed from aabfs website at 31/10/2011
- [11] Gibbs, W. (1999) Software's chronic crisis. Scientific American, September, 468-73.