

IMPACT OF BAD REQUIREMENTS AND COMMUNICATION ISSUES ON SOFTWARE AND SELECTION OF OPTIMAL REQUIREMENT ELICITATION TECHNIQUES

MUHAMMAD ZUBAIR ASLAM

Bahria University Islamabad, Pakistan
Email: mzubairaslam13@gmail.com

ABSTRACT

Software has covered almost every aspect of human life, from airplanes to smart phones, everything is dependent on software now a days. As software covers lot of our work and make our work a lot easier but at the same time it is to be sure that lot of effort goes into making a software. Software failure has cost lot of money and human life as well. There are many factors which cause software to fail and most of the time it happens right at the end of the completion of the development of a software. In this paper, we will discuss several factors that why software fails and how it can cause companies to face major loss in terms of finance. This research paper mainly focuses on bad requirements, elicitation techniques and the issues faced due to communication.

Keywords: software requirements; requirements elicitation techniques; software quality; communication issues; software requirements engineering;



1. INTRODUCTION

Primary usage of Software Engineering (SE) is to have the high quality software under the budget and time restrictions [1]. This discipline of Requirements Engineering (RE) solely focuses at the multiple and different requirement activities and attempts to solve the issues related to it [2]. RE is a phase where we deal with all the issues related to the requirements and its solutions by using different means of methods. It is a branch of SE that deals with all aspects of requirements related issues and solutions [3]. Everyday many new software are developed, covering various aspects of life. From government organizations to private firms, from layman to a business man, everyone is in need and dependents on a software.

Now a days, Governments are the top consumers of the software. In 2003 United States had spent more than \$20.3 Billion in major IT projects and this figure went up to \$60 billion in 2004 [4]. It shows that how much nations are willing to spend on software and how software have forced people to spend so much for new inventions and making life easier for the people. The software around is like skyscrapers all around us but we don't know who built them, who designed them and who invested in them. We also don't know if they contain any defects or have some flaws that may result into massive disruption or financial chaos or catastrophic failure. According to a study, 70% of the organizations have at least one project failure within one year of project commencement. That shows how regular software failure is. To understand this phenomenon of software failure, let's take an example of a smartphone that contains hundreds of thousand lines of code and if one line has some error or shown vulnerability, can cause whole smartphone to stop working. In 2005 \$1 trillion was spent on hardware, software and other services worldwide.

Back in 1981 United States administration of federal aviation started looking to upgrade its ATC system, but efforts to make new system encountered many problems. In 1994 agency finally gave up the project and the price of project got three times than the estimated one (it went on to \$2.6 billion). Apart from it, delivery date went on to pass by several years. For 2016, it was estimated that around 21 million people are professional software developer around the world. And that shows that how serious this industry is and how difficult it is to work in there. Losses of Billions of dollars is not a unique thing as software project failures have caused people to lose their lives as well. Most of the time software failures occur like in airplane crashes. Just a pilot never wants to crash, investigation covered lot of aspect like wind speed, weather, pilot's training and maintenance record. Similarly, there are lot of factors that cause a software to fail. It may be the technical management, project management or development team incompetency that may lead to a failure. The biggest problem with software failure is that software failures for most part of time are avoidable and predictable. But unfortunately, most of the

enterprises do not see preventing failures as they think urgent. Even knowing this view can risks harming the enterprise or even may cause it destroying, but still it is happening. There are phases where software are tested and made sure that they do not have any failures or bugs in it before the deployment but still it is not enough for the software to predict 100% success ratio. Software testing phase of Software Development Life Cycle (SDLC) is dedicated to make sure that a software meets the requirements and does not possess any missing or faulty thing in it. But it is not possible to test the information technology system of any real size thoroughly. “Roger S. Pressman” in his classic book in the field of SE pointed that “exhaustive testing presents certain logistical problems. Even a very small program consist of 100 having nested path and a single loop which executes less than twenty times may require 10 to the power of 14 possible path to be executed” [5] if we need to execute the test for those 100 trillion paths, it would nearly take 3170 years even if we execute the evaluation in a millisecond. That shows we are never sure and to make sure if we have done enough testing or we know how much testing is enough to keep the software safe from failures.

2. LITERATURE REVIEW

A software project can be termed as successful, failed or challenged. A software is termed as successful if completed on given time and within the given budget, and have all the requirements completed and functionalities given by the stakeholders. A project is known as failed that was cancelled way before its compilation or it was completed but it was never actually implemented [6]. Here was a study by candid group in 2009 which revealed that only 34 percent of software projects succeeded, 22 percent were failed and 44 percent were challenged. That shows the mere ratio of those projects who were able to succeed against those which were failed or challenged. Most of the time it was requirements which resulted in the failure of the system/software. If we go into the details, inefficient RE process resulted into the 43% of the software failures, many time it was identified that 70% system failed because of the difficulty in identifying the requirements and 54% system failed because of requirements were not clear and organized [6, 7].

Back in 1994 only 16% of the systems were successful that shows how immature the processes were at that time [8]. Gause and Weinberg (1989) [9] pointed out some of the problems regarding requirements which were the cause of project failures, they said; requirements are difficult to translate into natural language, requirements have different level of details and types, if not in control they are difficult to manage and most of the requirements change during the development. Apart from it, many more reasons are there regarding requirements which resulted the software to fail: missing dependencies, poor planning, changes in requirements not handled and important requirements missed. The Standish group reported [10] that 53% of the project were challenged (over budget, late or missing features), 18% failed and only 29% of the projects passed (on budget, on time and containing all features).

These figures were not belonging to any particular size of the company, companies of all sizes showed the almost same numbers. In large companies, only 9% of the software were successful, 16.2% in medium and 28% in small companies. On the other hand, 60% of the projects were challenged in large companies comparing to 46% to medium and 50% to small companies. Same ratio was seen in failed number of projects as 30% projects were failed in large companies, 37% in medium and 21% in small companies (companies were organized in small, medium and large groups on the basis of their annual income). These numbers show that how the size of an organization and density of hierarchy have played role in the success of a software. Small company has less staff and simple hierarchy, resulted in a greater number of systems to be developed successfully. Apart from going over budget and over time, restarts were also one of the reasons why system failed.

2.1 Requirements engineering process

RE is considered as one of the main phases of SDLC where people from RE department and customer sit to finalize what the software should have or what the software should possess in terms of functionality. This is censorious stage as anything we decide here has long lasting impact on the rest of the system for the rest of the SDLC. In research different associated challenges have been reported that have an immense impact on the overall quality of the software [11, 12]. That is why good requirement refinement is very important at this stage [13]. RE phase is composed of several steps and it helps to learn and understand what the needs and demands of the customer are and what are the constraints on the system, understanding the real needs of the customers and handle the customer requirements [14]. The phases of RE are as follows [15].

2.1.1 Elicitation

The initial step of the RE process is known as elicitation. It is usually known as the phase where we discuss about what are the real needs of the both parties (customer side and development team side) [13]. This phase also involves the investigation where we define how program can achieve the objectives of system stakeholders and if no then what are the alternatives [16].

2.1.2 Analysis and negotiation

This phase is comprised of series of activities that are aimed to achieve and identify the problems in the system requirements and to achieve the agreement to improve the accommodation for system stakeholders. In the review phase, when an analyst finds any problem in the criteria, all these specifications that are collected are referred back to the previous phase (elicitation). We apply this method on unclear, incomplete and those requirements that are contradictory. In order to resolve the conflicts, the disputes found in the negotiation process should be resolved immediately and addressed individually with stakeholders.

2.1.3 Documentation

Once we have the specifications evaluated, documenting them is important so that we can formalize them with a proper process of specification. Teams organizes and finalizes the criteria in this process in such a way that we can achieve the accuracy and traceability. This phase is very important because often, the documents which were created during the specification will be used for the rest of the development phases.

2.1.4 Validation

This step of the series ensures that the documents and templates correctly reflect needs of all stakeholders as well as final draft of framework documented for contradictions, deviation from different standards and omission.

2.1.5 Management

Management is the final step of the lifecycle of software specifications, it is almost like a software system's maintenance. In this process, many of the most critical maintenance tasks involve updating the used or using criteria and the degree of developmental support.

3. TECHNIQUES FOR REQUIREMENTS ELICITATION

Elicitation of requirements is one of a dynamic process that involves multiple tasks with a range of methods and techniques that are available and the approaches to execute them. Many RE practices and experiments were only aimed to improve the elicitation processes through the production of different techniques. Below are some of the techniques used by different RE practitioners.

3.1 Traditional techniques

When projects become complex, there were needs to coordinate the work of development. The "waterfall" model is modified and the process has been developed for development work. When we are done with one phase the next phase starts along with a smooth timeline, the methods of initial stage of elicitation are now considered as standard techniques. This include common data collection methods like questionnaires, interviews, surveys and analysis of current documents like process models or guidelines and existing system user and other manuals.

3.1.1 Interviews

Interviews are considered as traditional method for communication. It is widely used technique by many practitioners for eliciting the requirements just because it is human-based social activity [17]. For gathering large amount of information interview is considered as an efficient approach. Interviewer's performance decides about how fast and useful information he can collect. Interview includes some general social skills not just asking the questions. We divide the interviews into two groups. Structured and unstructured. An interview is usually done by the person who have domain knowledge as well [18].

a) Structured

This type of interview consists of detailed information that is based on predefined fixed questions. It allows the requirement engineer to examine the level of understanding which a respondent has about the topic. The quality of information and usefulness is highly dependent on the quality of the questions [18]. Sometime people do use the templates as it provides the easier way to requirement elicitation. Structured interviews has one drawback that it limits the new ideas and exploration [17].

b) Unstructured

The unstructured one are solely made and set for the purpose of the discipline of anthropology and the sociology as a technique that usually elicit people's social activities and actualities. Unstructured interviews are only useful when focus is on understanding particular portent of a particular social environment [19]. These are mostly traditional in nature where we do not need any predefine questions. The information from the stakeholders is gathered through some open discussion.

3.1.2 Surveys/questionnaires

Questionnaire is used as one of the simple tools, it may be composed of open or closed questions at the initial phase of requirement elicitation. This is one of the methods used to gather requirements with less cost and time. It is an effective way to collect the data from the different stakeholders quickly along with avoiding the huge amount of irrelevant information and data. There is a drawback with this technique, some topic may get attention more than needed and some may get ignored. Questionnaire must be concise, clear and to the point [20].

3.1.3 Reading existing documents

Reports are very good source for knowing and understanding that what the customer wants in the form of software system and environment of customer's system. Reports also refers to manuals, input and output models and types. It also gives the information about the protocol guides, job description and implementation codes [13].

3.1.4 Introspection

The approach used for introspection is also a basis for the other methods used for achieving specifications. Introspection technique is only useful when the requirements engineer does not understand the program's environment and its goal, but possesses the experience of users and business processes [21].

3.1.5 Meetings

Informal or formal meetings may well evoke the requirements. This strategy is useful in the event of a conflict among the different stakeholders and parties. This technique is mostly used when we want to overcome some problems and to negotiate about something [13].

3.2 Collaborative techniques

In some different environments all techniques are efficient, good and useful. It is mostly a good approach to work with various techniques that allow to develop specifications. The integration of various techniques let the specialists to explore the basic concepts and enhance the general information and knowledge of the technology area. Rather than integrating the different techniques together, the integrated approach protocol builds the entire procedure by incorporating the conventional, cognitive and relational methods into common methodology. Expert and stakeholders shall collaborate and organize in multiple ways to understand the mutual point of desired product [17].

3.2.1 Brainstorming

This method is used for creating conceptual thought for applications that are to be built. It is a process where people from diverse departments and different stakeholders take part in the activity and debate in order to produce as many solutions for the problem as possible. Brainstorming method includes two stages. Generation process and aggregation. In the first part, we propose as many proposals as possible without even discussing the merits of the concepts. In the second phase we address, update and arrange the ideas. One of the benefits of this technique is that it promotes the new ideas and let the discovery of new and sophisticated methods for problems [21].

3.2.2 JAD/RAD sessions

Rapid Application Development (RAD) and Joint Application Development (JAD) highlight the customer interaction during meetings with a group in the presence of a facilitator. Major difference between Brainstorming and JAD is that JAD sessions are well established. They include the responsibilities and activities of each individual, moreover, partners and consumers are also allowed to address and participate as well [18]

3.2.3 Prototyping

Prototyping is one of the effective techniques of direct contact between experts and end users. It is also known as two-way strategy. It is used to get the feedback from the users. In this technique, we design the requirements and asked the end user to have a look at it and give the feedback [17]. This approach is considered as very useful when we have to design human computer interaction or when the consumers are not comfortable with the given solution [22].

3.3 Contextual techniques

It is a mixture of organizational perception and prototyping and unstructured listening. The goal is to collect the comprehensive state about the environments, patterns, stakeholders and their working procedures. This particular method is usually used for designing the systems which are interactive when the user interface designs are mostly difficult and critical [23]. Various techniques that are used under this group are given below.

3.3.1 Ethnographic technique

This technique is considered as a social science elicitation that supports conscious requirements. This technique was solely developed by and for the anthropologists to understand people who were in their cultural and social perception. The fundamental basis of this is human action. This technique is particularly used in usability and exploration of collaborative work environments and locations [24].

3.3.2 Discourse analysis

This technique is an extension of the ethnographic technique. It is composed of mainly 2 sub techniques called as conversational analysis and the speech as analysis. It presents a variety of methods and tools to examine systematically the basic parts of requirements. It uses the socio-linguistic methods to specify the linguistic characteristics of multiple types of discourse [17].

3.3.3 Socio-technical methods

It creates the informal version of the socio-technical structure. The people, the system and the organization are the main considerations in these techniques. This is the strategy to understand the problem and corporate sense [25].

3.4 Cognitive techniques

Cognitive techniques are the elicitation techniques that are mostly composed of multiple discipline paradigms, methods of knowledge elicitation are widely disseminated in the field of education, engineering of knowledge, philosophy, linguistic, anthropology and psychology [26].

3.4.1 Protocol analysis

Protocol analysis technique is mainly composed of the protocols that are either think-aloud or retrospectively based on the vocalizing behavior. The main advantage of this is the immediate verbalization of executive tasks, that is incorporated in work context. Thus, this approach has been shown to be very effective at exposing the issues of contact with the existing systems [24].

3.4.2 Knowledge acquisition technique

Many information learning ways and methods are present in literature and some of them are as follow.

- **Card sorting**

It is the basic elicitation method for the elicitation of the requirements. It has many optimistic phases that render it as an effective method for the elicitation of the requirements. Card sorting requires from the stakeholders to sort number of domain entities.

- **Repertory grids**

It is a basic method used by clinical psychologists. By defining small set of items like individuals, artifacts, user is asked to define certain structures such as slots, attributes. In addition it build the values that may be defined on a small scale between two end-point ranges [27].

- **Proximity scaling technique**

This method is domain specific and is a collection of measurements taken to define them. First step is peer-to-peer proximity assessment for the related domain elements that captures some of the tacit knowledge of the expert. Then an automated mathematical analysis is done to make multi-dimensional space to distinguish objects.

- **Laddering**

This is hierarchical technique of questioning which is devised from the grid technique that enables an ascending hierarchy of the definition to be developed. Laddering is suited for the use with the protocols where the performance of the document is used as specification reference [27].

- **Task analysis**

Task analysis technique uses the wide variety of observations with interview strategies to capture the final assessment of the information that specialists use to carry out the complex tasks. But with the complex tasks we mean those tasks where performance need to incorporate use of both automated (unconscious, procedural or strategic) and the controlled (conceptual, conscious) information to perform any task [27].

3.5 Innovative techniques

As many elicitation techniques are available, there is still a range of some unanswered issues that requires more study and research from research community. It is inevitable that some other strategies of elicitation may be established in near future by offering solution to given problems. Thus, as a consequence an advanced type of elicitation technique is created in which all the newly proposed strategies will be used [28]. Another method is throwaway paper model. It is a visual representation of what we want the system to be. Usually users are expected to generate feature through pen and share it with the requirement engineers. Graphic design tools are also used for this purpose. In most situations, the paper version is used as part of the usability check. Paper sample approach for the elicitation criteria is an appropriate method for small and some medium sized initiatives.

4. COMMUNICATION MODELS IN REQUIREMENTS ENGINEERING

This section provides some of the communication models that were proposed by researchers from multiple disciplines. The communication elements which we have mentioned are composed of different parts like source-receiver, feedback, encoder-decoder, contact and effect and noise [26]. Initial model explained is linear by Shannon-weaver. This model explains the process of communication having sources of information, a transmitter, a message, a receiver, a noise and destination. Transactional model is another model explained in [17]. This usually involves two or more than two participants who act with one another and react as well. A message which can be transmitted/exchanged between them only when both of the sender and the receiver take it in the common way. Another model explained it that the means of a message does not stay completely in message. Instead it is taken by the receiver based on their own background [29]. Because of differences in their background, this meaning can easily differ heavily from the needed meaning of sender. Communication is a usually a lot more than just an expression, it requires all the participating people to share the common quality attributes such as experience, language, cultural values and knowledge [26].

It is agreed that communication occurs only in a particular context. And it has four dimensions that are temporal, psychological, and physical. The physical dimension refers to the physical environment. Communication occurs in this environment. Social dimension possesses the relationship among people participating and norms and culture of normal society in which they communicate. The psychological part is composed of such things as friendly or unfriendly and formal or informal. The dimension of temporal contains the time in which the communication takes place. Berlo's model have the emphases on that communication is an interactive process [21]. He specifies the four main elements of communication which are "source, channel, message and testing". Perception and personality screen for each participant in the communication process has been highlighted by [27] and the argument which is given says that massive part of this will be encoded with many factors of the personality of the sender and perception of the environment. Barnlund's model defines that cues are those signals which a person processes from his environment [30]. Another model which is illustrated by Wenburg represented the communication process as an infinity symbol. This model has successfully demonstrated that communication is a process which never ends as shown in Figure 1.

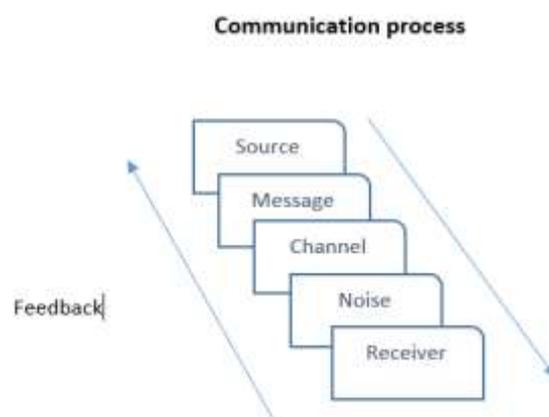


Figure. 1 Communication as a process

5. COMMUNICATION PROCESS AND CHALLENGES IN REQUIREMENTS ELICITATION

According to [28] [31] [32] activities of communications in eliciting the requirements can be composed of three phases. First one in acquisition of knowledge, negotiation, and its integration.

- **Knowledge acquisition**

In this phase the whole methods, facts do share the understanding e.g. vision, ideology, experience, knowledge and technology.

- **Knowledge negotiation**

As the whole process facts negotiation for the software requirement information.

- **Knowledge integration**

We can use model which we have described earlier in Figure 1 as a model for communication in the RE process as explained and defined in Figure 2. Source is the customer here, message is requirement, channel is technique, noise is communication challenges, the receiver is developer and the feedback is software requirement specification. In other words, one can claim that the connection between theory with requirement elicitation goes beyond to the simplest of communication fallacy.

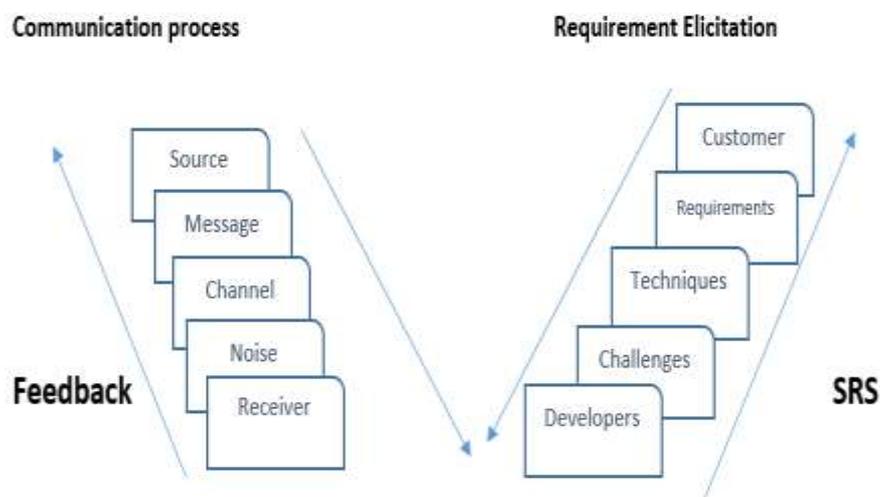


Figure. 2 Communication and requirement elicitation process

Acceptance of knowledge merely revolves on sides of the degree to the assurance, acceptance and satisfaction, users usually feel for the newly developed system. Feedback is usually referred to the communication which is gathered from the users using documents. These all challenges are in need to get reduced so that we can ensure the affective form of communication. The knowledge must be gathered so that the learning and understanding of requirements and commitments to changes should be made possible. The information and data needed for acceptance by all the parties.

Table. 1 Challenges in communication activities

Activities	Problems
Knowledge Acquisition	<ul style="list-style-type: none"> • Gap in understanding • Redundant aspects • Innovative Thinking
Knowledge Negotiation	<ul style="list-style-type: none"> • Commitment • Information Exchange • Shared perspective
Knowledge Acceptance	<ul style="list-style-type: none"> • Fear factor • Feedback • Change management

Apart from that, there are many mediums which used as communication mediums between developer and customer. The obtained result shows that the medium which was we often used are telephone, email, face to face and meetings. But yet there are still some problems in these mediums such as information was not consistent and mistakes in interpretation. Often the data and knowledge were delivered by using the different methods, thus the problem comes up since individual may have the different or multiple views and understanding of given topic.

These given situations demand us to improve it so that to ensure a good transmitting process of knowledge. The personality of all the customers and the development team also have influence on the acceptance and information delivery. The personality attributes consist of the quality and the commitment by team, support of the environment and personal skills of them. The result shows the lack of customer intent and cooperation, commitment and their capability contribute to the conflict in their personality. This situation causes the customer to have daily routine work excess, enough number of staff members and the turn-over.

Table. 2 Communication challenges

Criteria	Challenges
Type of Input	<ul style="list-style-type: none"> • Redundancy of information. • Frequent requirement changes • Changes of scope • Different information
Personalities Involved	<ul style="list-style-type: none"> • Lack in the cooperation • Lack in the commitment and their participation • Lack control of burden
Communication skills	<ul style="list-style-type: none"> • Lack in ability of sorting information • Lack in ability in proactive and instructive delivery information • Lankness about the organizing an idea • Lack of logic written
Medium of Communication	<ul style="list-style-type: none"> • Cannot access file. • Late responses. • Information which is informal • Information which is unrecorded.
Procedure	<ul style="list-style-type: none"> • Changes in the report types • Changes in the documents • Changes in criteria of acceptance

6. CONCLUSION AND FUTURE WORK

This paper displayed that how requirements have influenced the software and systems. How requirements were the cause of failures of different kinds of systems? We have also seen that how much money is spent on the software development. We have also seen the summary of how the RE prices by communicating all the significant activities which start at the requirements collection and end at their management. This paper also contains the short but explanatory summary of various methods and techniques used in RE all over the world. Though full endeavour has been made to observe all the major strategies of elicitation in paper. Study of these methods focused on some of the well-defined parameters which can be the work to be done in near future. Another effort can be made to identify current concerns and future prospects in relation to the requirements of the tendering process. This research may provide valuable advice to the requirement engineering practitioners on production of quality systems and software.

REFERENCES

1. Pandey, S. and K. Mustafa, *Risk Assessment Framework (RAF)*. International Journal of Advanced Research in Computer Science, 2010. **1**(3).
2. Pandey, S. and M. Batra, *Formal methods in requirements phase of SDLC*. International Journal of Computer Applications, 2013. **70**(13): p. 7-14.
3. Pandey, S. and K. Mustafa, *Recent advances in sre research*. Recent Advances in SRE Research, 2010. **2**(04): p. 1079-1085.
4. Hussain, A. and E.O. Mkpojiogu. *Requirements: Towards an understanding on why software projects fail*. in *AIP Conference Proceedings*. 2016: AIP Publishing LLC.
5. Pressman, R., *Introduction to Software Engineering*. Software Engineering: A Praccitioner's Approach, 2009: p. 27.
6. Kamuni, S.K., *Study of factors that induce software project overrun time*. 2015.
7. Asghar, S. and M. Umar, *Requirement engineering challenges in development of software applications and selection of customer-off-the-shelf (COTS) components*. International Journal of Software Engineering, 2010. **1**(1): p. 32-50.
8. Wiklund, D. and J. Pucciarelli, *Improving IT Projects Outcomes by Systematically Managing and Hedging Risk: An IDC Insight Research Document*. 2009.
9. Gause, D.C. and G.M. Weinberg, *Exploring requirements: quality before design*. Vol. 7. 1989: Dorset House New York.

10. Gulla, J. *Seven reasons why information technology projects fail*. in *SHARE Conference*. 2011.
11. Babar, M.I., M. Ramzan, and S.A. Ghayyur. *Challenges and future trends in software requirements prioritization*. in *International conference on computer networks and information technology*. 2011: IEEE.
12. Babar, M.I., et al., *Systematic reviews and mapping studies on software requirements: encyclopedia of objectives and issues for newbies*. *Journal of Software Engineering & Intelligent Systems*, 2017. **2**(2): p. 128-156.
13. Mead, N.R., *Requirements engineering for survivable systems*. 2003.
14. Babar, M.I., et al., *PHandler: an expert system for a scalable software requirements prioritization process*. *Knowledge-Based Systems*, 2015. **84**: p. 179-202.
15. Kotonya, G. and I. Sommerville, *Requirements engineering: processes and techniques*. 1998: Wiley Publishing.
16. Hofmann, H.F. and F. Lehner, *Requirements engineering as a success factor in software projects*. *IEEE software*, 2001(4): p. 58-66.
17. Zowghi, D. and C. Coulin, *Requirements elicitation: A survey of techniques, approaches, and tools*, in *Engineering and managing software requirements*. 2005, Springer. p. 19-46.
18. Sharma, S. and S. Pandey, *Revisiting requirements elicitation techniques*. *International Journal of Computer Applications*, 2013. **75**(12).
19. Zhang, Y. and B.M. Wildemuth, *Unstructured interviews*. *Applications of social research methods to questions in information and library science*, 2009: p. 222-231.
20. Adeel, K., S. Ahmad, and S. Akhtar. *Defect prevention techniques and its usage in requirements gathering-industry practices*. in *2005 Student Conference on Engineering Sciences and Technology*. 2005: IEEE.
21. Paetsch, F., A. Eberlein, and F. Maurer. *Requirements engineering and agile software development*. in *WET ICE 2003. Proceedings. Twelfth IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises, 2003*. 2003: IEEE.
22. Shams-Ul-Arif, Q.K. and S. Gahyyur, *Requirements engineering processes, tools/technologies, & methodologies*. *International Journal of Reviews in Computing*, 2009. **2**(6): p. 41-56.
23. Viinamäki, J. *Requirements elicitation: Contextual inquiry*. in *T-76.652: Seminar on User and Customer Requirements Engineering*. 2004.
24. Murugan, C.S. and S. Prakasam, *Comparison Of Different Techniques For Software Requirement Elicitation*.
25. Bryl, V., P. Giorgini, and J. Mylopoulos, *Designing socio-technical systems: from stakeholder goals to social networks*. *Requirements Engineering*, 2009. **14**(1): p. 47-70.
26. Cooke, N.J., *Varieties of knowledge elicitation techniques*. *International Journal of Human-Computer Studies*, 1994. **41**(6): p. 801-849.
27. Corbridge, C., et al., *Laddering: technique and tool use in knowledge acquisition*. *Knowledge Acquisition*, 1994. **6**(3): p. 315-341.
28. Vijayan, J. and G. Raju. *Requirements elicitation using paper prototype*. in *International Conference on Advanced Software Engineering and Its Applications*. 2010: Springer.
29. Urquhart, C., *Analysts and clients in conversation: cases in early requirements gathering*. *ICIS 1998 Proceedings*, 1998: p. 11.
30. Chirinos, L., F. Losavio, and A. Matteo, *Identifying quality-based requirements*. *Information systems management*, 2004. **21**(1): p. 15-26.
31. Viller, S. and I. Sommerville. *Social analysis in the requirements engineering process: from ethnography to method*. in *Proceedings IEEE International Symposium on Requirements Engineering (Cat. No. PR00188)*. 1999: IEEE.
32. De Vries, H.J., H. Verheul, and H. Willemse. *Stakeholder identification in IT standardization processes*. in *Proceedings of the Workshop on Standard Making: A Critical Research Frontier for Information Systems*. Seattle, WA. 2003.

AUTHORS PROFILE