International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347-6435(Online), Volume 1, Issue 3, December 2013)

Requirements Elicitation Techniques: Comparative Study

Omar Isam Al Mrayat[1], Norita Md Norwawi[2], Nurlida Basir[3]

[1,2,3] *Faculty of Science and Technology, Universiti Sains Islam Malaysia (USIM), Bandar Baru Nilai, 71800 Nilai, Negeri Sembilan*

***Abstract*— Over the years, software development failures is really a burning issue, might be ascribed to quite a number of attributes, of which, no-compliance of users requirements and using the non suitable technique to elicit user requirements are considered foremost. In order to address this issue and to facilitate system designers, this study had filtered and compared user requirements elicitation technique, based on principles of requirements engineering. This comparative study facilitates developers to build systems based on success stories, making use of a optimistic perspective for achieving a foreseeable future. This paper is aimed at enhancing processes of choosing a suitable technique to elicit user requirements; this is crucial to determine the requirements of the user, as it enables much better software development and does not waste resources unnecessarily. Basically, this study will complement the present approaches, by representing a optimistic and potential factor for every single method in requirements engineering, which results in much better user needs, and identifies novel and distinctive specifications.**

***Keywords*— Requirements Engineering, Requirements Elicitation Techniques, Conversational methods, Observational methods, Analytic methods, Synthetic methods.**

I. INTRODUCTION

Analysts and experts have reported considerable proofs in the research literature, related to software quality requirements that facilitate successful software delivery. [1] has reviewed 10 software project failures over the span of two decades. The study has specifically ascribed three of those downfalls to poor requirements analysis. It has been claimed that poor requirements are in eighth position in his list of the top 10 errors made by developers. Wiegers have argued "If you don't get the requirements right, it doesn't matter how well you execute the rest of the project" [2]. McGovern has declared that "the critical success factor will always be the accuracy and completeness with which the business requirements and goals are captured and traced to the associated details" [3].

According to [4], the intent of requirements analysis, is to elevate the probability of building right system, i.e., upon completion the system must satisfy the targeted customers and address their needs to a satisfactory degree.

Bergey, et al. have highlighted that if the quality requirements are not proficiently identified, the ensuing system cannot be properly assessed for success or failure, well before implementation. The [5] authors have stated that "It is widely acknowledged within the software industry that software engineering projects are critically vulnerable when these activities [elicitation, analysis, specification, and validation] are performed poorly".

II. BACKGROUND

Studies have exposed that problems associated with requirements engineering could cost 10-200 times more to rectify the program after its implementation, than if they were recognized during specifications [6, 7, 8]. Few other researchers have suggested that, the comprehensive amount of project budget due to requirements flaws is twenty-five to forty percent 25 to 40 % [9]. It is noteworthy that, requirements engineering is not only crucial, but will be disadvantageous, if the acceptable sources are not devoted early. Requirements Engineering (RE) "is an activity, which aim is to discover, document and maintain a set of requirements" [10, 11].

"The use of the term engineering implies that systematic and repeatable techniques should be used to ensure that system requirements are complete, consistent and relevant" [12]. Another definition according Software Test and Evaluation Panel (STEP) is "the disciplined application of scientific principles and techniques for developing, communicating, and managing requirements" [13]. Similarly, [14] have defined requirements engineering as "the systematic process of developing requirements through an iterative process of analyzing a problem, documenting the resulting observations, and checking the accuracy of the understanding gained".

Both these definitions indicate that requirements engineering is not just a single activity, rather it a process, which comprises a variety of phases and actions. [15] has pointed out that requirements engineering can be classified into elicitation, solution determination, specification, and maintenance.

International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347-6435(Online) Volume 1, Issue 3, December 2013)

Dorfman has segregated requirements engineering into five phases, such as, elicitation, analysis, specification, validation/verification, and management [16]. [17] have identified that requirements engineering has the following phases; eliciting requirements, modeling and analyzing requirements, communicating requirements, agreeing on requirements, and evolving requirements.

Even though we mostly agree with the classifications of the requirements engineering process mentioned above, however, we feel that the researchers have failed to include a phase to support requirements during the early stages of development. Therefore, we have considered requirements engineering as a process, which is composed of five distinct, but related phases. Figure I below shows these phases; however, we do not consider security requirements engineering to be a sequential process (as each phase can and should affect the others).

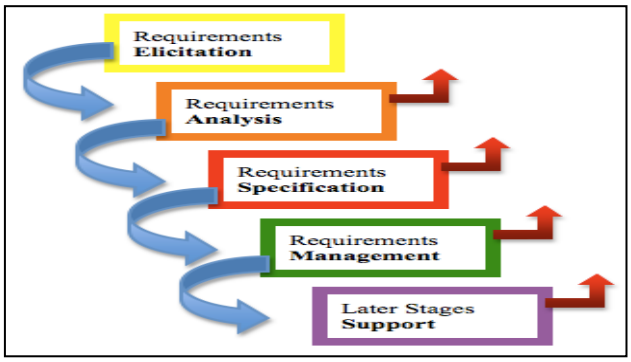

FIGURE I Requirements Engineering Phases Surveyed.

Source: Adapted from Sommerville. Software Engineering Book,(1998).

III. Requirements Elicitation

Quite a number of authors have stated that requirement elicitation is the first phase of the process, by which a project team determines the organizational needs that must be addressed by the project effort [18, 19, 20, 21]. Particularly, requirements elicitation concentrates on the preliminary pursuit of identified requirements and possibilities of the social actors (e.g., participants, users) most strongly associated with the system. A report from [22] claims that requirement elicitation is concerned with software requirements originate from and how the software engineer can gather them. It is the first stage in understanding the problem, which needs the software is required to solve.

In addition, [23] have defined requirement elicitation is the first step in the first step in gathering user requirements; it is the process of understanding and acquiring the needs of users and other stakeholders.

The term "elicitation" is used to capture the idea that, the individuals, who will be engaged within the system on a daily basis, are the appropriate source for the identification of needs and opportunities, and that designers must "draw" the requirements from such stakeholders. It is important to note that the requirements elicitation process focuses on "what*"* is expected to be achieved by the predicted system irrespective of "how*"* it is to be achieved. In the literature, a range of process models have been developed to identify the various features of the requirements phase of the software. One widely employed model suggests three fundamental stages, such as: elicitation, specification, and validation of requirements [19]. It is noteworthy that the requirements elicitation is concerned with the process of determining what issues must be addressed by a design effort. Requirements specification is constructed upon the elicitation activity, and comprises the specific documentation of the specifications for the system [24]. Generally, this constitutes both natural language explanations and official modeling techniques. Ultimately, requirements are carried out to assure the recognition of stakeholders for the requirements, which have been reported and which will be applied in pursuing design initiatives [25]. While this three-stage model indicates a linear strategy to develop requirements. The three phases are generally employed iteratively, often moving progressively in more detailed levels of requirements gathering.

Just as the requirements phase process is critical to the overall success of software design efforts, the requirements elicitation also plays a crucial initial role in the extensive requirements elicitation process. One of the crucial features of requirements elicitation is that, it is typically one of the most important components, by which the project team members acquire knowledge related to organizational domain. [19] have stated that "the importance of requirements elicitation cannot easily be overestimated; when you have to solve somebody else's problem the first thing you have to do is to find out more about it" (P.21).

The emphasis of the requirements process is generally presented as the process revealing the essential obstacles within the business context, generally known as "problem domain".

International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347-6435(Online) Volume 1, Issue 3, December 2013)

Consequently, traditional requirements elicitation strategies signify a shortfall based mode of inquiry, which is participating "in a study of what is unsuitable, not working, not up to standard, and in need of a 'fix'" [26]. In the following section, an overview of the most relevant techniques used in requirements elicitation are presented in the context of a standard software development process. Also, the classifications of requirements elicitation techniques are demonstrated and briefly highlight a number of the most widely employed methods of requirements elicitation and discuss the associated strengths and the challenges.

Overall, the goal is to force the analyst, user, and other stakeholders to understand the requirements they want to address. Defining requirements calls for effective interaction and open communication between the user and the developer to generate the necessary requirements they want to address and the information that can be used to develop the system that meets the needs of the user [27].

IV. Requirements Elicitation Techniques Categories

The requirements elicitation techniques facilitates the developers to have an understanding of the requirements of users, this phase allows the developers to recognize the requirements of stakeholders, other than the actual users of systems [28, 29]. Nevertheless, [30] have stated that elicitation is the process, which motivates the dynamic contribution of users in system development, which consequently increases the accomplishments and endurance of information systems. According to [31] specifications progression is an rigorous communication procedure among stakeholders and the programmers, therefore the four kinds of elicitation methods vary depending on the context of communication: observational, conversational, analytic, and synthetic [31]. Each and every kind provides a unique communication model among programmers and stakeholders, and demonstrates the characteristics of a strategy. Realizing the category of method facilitates engineers to comprehend various elicitation methods and guides them to choose a suitable technique, for requirements elicitation.

A. Conversational Methods

The conversational method provides a means of verbal communication between two or more people. As conversation is a normal means to convey requirements and concepts, and ask and answer questions, it is efficient to build and comprehend the issues and to elicit generic product requirements [32].

Methods in this group are also referred as verbal methods [32]. A standard conversational strategy is interview. It is generally used in requirements elicitation [20]. Other methods under this category include workshop, focus groups and brainstorming (Table I).

TABLE I
Conversational methods

Method	Literature support	Executor	Illustration
Interviews	[20, 33, 34]	An experienced analyst with generic knowledge about the application domain	Interviewer discusses the desired product with different groups of people and builds up an understanding of their requirements (Interviewer asks questions to the specialist or end-users, related to a particular topic)
Workshop, focus groups	[33, 34]	An experienced outside facilitator	Stakeholder representatives gather together for a short but intensely focused period to create or review high -level features of the desired products
Brainstorming	[34, 35, 36, 37, 38]	An experienced outside facilitator	Brainstorming is a kind of mini conference, held among six to ten experts. Members from different walks of life involve in the brainstorming, chaired by the organizer, who asserts the topic to be discussed

One of the most common types of social relationship is conversation. Generally people are pleased to express about the jobs they perform, and the challenges they encounter. The verbal needs and limitations are generally called non-tacit specifications. As oral communication is realistic and productive to gather non-tacit knowledge, the conversational methods constitute the principal approach to non-tacit requirements elicitation, by carrying out interviews, workshops or brainstorming sessions [39]. Generally, conversational strategies are extremely used in requirements development, however not by itself, they require combining other kinds of techniques, to accomplish the software development phase.

International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347-6435(Online) Volume 1, Issue 3, December 2013)

However, they require a lot of human efforts [18, 40]: conference set up and creating transcript and examining records of a live conversation, and consume a lot of time. On the other hand, it is concern to accomplish the elicitation process, particularly when employing workshop or brainstorming, e.g. organizing the meeting and making sure that the representatives are available for the meeting [20].

B. Observational Methods

The observational method provides a means to develop a rich understanding of the application domain by observing human activities [31]. In addition to non-tacit requirements, some requirements are obvious to stakeholders, but difficult to verbalize, which is called as tacit requirements [39]. Verbal communication is frequently weak, when gathering tacit requirements. Consequently, observing how people perform their routine work, facilitates gather information, which are challenging to explain in words. Methods under this group are illustrated in Table II.

TABLE II
Observational methods

Method	Literature Support	Executor	Illustration
Social analysis and Ethnographic study	[20, 39]	The observer must be accepted by the people being studied as a kindred spirit and must be sufficiently familiar that they carry on with their normal practices as if he was not there	An observer spends a period in a society or culture on making detailed observation of all their practices. User culture and work environment are observed
Observation	[33, 41, 17]		
Protocol analysis	[33, 39]		A subject is engaged in some task, and concurrently speaks out loud and explains his thought

Observational methods seem to be best suited, when individuals find it complicated to communicate their requirements and when analysts search for a improved comprehension of the perspective, where the preferred product has to be utilized [42]. Good examples of tacit information consist of the scheduled work, which individuals accomplish day-to-day, spontaneously and in the organizational or social contexts, which most likely affect the specifications. As people are acquainted with the perspective and scenario of their work, they do not deliberately contemplate about the schedules and the working environment. It is challenging for them to enunciate how work is performed, despite the fact that occasionally the routine work is simple to be revealed to others [20]. Therefore, to be engrossed in the actual work scenario, to acquire the observational facts, can assist engineers to thoroughly, understand the routine of work, the societal group, the organization, and the wider perspective, within which the product is used. As observation methods fall into the category of longitudinal studies, in general, it takes longer period than the other methods [43], which is considered as main disadvantage of such methods, especially when the project has tight schedule at the requirements stage. Besides this, observation requires compassion and receptiveness to the physical environment [31]. It is easy for observers to perceive a good image about the work context, but it is normally hard to specify and analyse their perception.

Observational methods are used for understanding complex societies, rather than making judgments about improving or supporting the ways of working [20, 31]. They are beneficial to discover fundamental elements of routine order, such as the standard design of work, and offer the most relevant information towards designing solutions [44]. Consequently, it is generally a good practice to begin with an observational method, to get a preliminary comprehension of the preferred product, when the development team falls short of experiences of product development in a given domain.

C. Analytic Methods

Analytic methods provide ways to explore the existing documentation or knowledge and acquire requirements from a series of deductions. These methods are illustrated in Table III.

International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347-6435(Online) Volume 1, Issue 3, December 2013)

TABLE III
Analytic methods

Method	Literature Support	Executor	Illustration
Requirement reuse	[20, 33,41, 45, 46, 47]	Documentation	In this technique the systems within the same product family is used to identify requirements of the desired system
Documentation studies /content	[20, 45, 46]	Documentation	A common method consisting of reading and studying available documentation for content that is relevant to and useful on the requirements elicitation tasks
Analysis Laddering	[48]	Expert 's knowledge	It involves the creation, reviewing and modification of hierarchical content of expert's knowledge, often in the form of ladders (i.e. tree diagrams)
Card sorting	[49]	Expert's knowledge	The expert is asked to sort into groups a set of cards each of which has the name of some domain entity written or depicted on it

A wide range of studies have focused on requirements of the preferred product, which consists of problem evaluation, organizational charts, specifications, user manuals of existing systems, research report of competitive systems in market. By understanding it, the engineers capture the information about the application domain, the work-flow, the characteristics of product, and map it to the requirements specification [48].

Furthermore, they recognize and reuse requirements from the specification of the heritage or identical products. It is always worth searching and filtering for reports and recorded information, relevant to the desired product. The mapping techniques are beneficial for knowledge acquisition, in analytic methods [50]. Multidimensional scaling [50, 51] allows users to obtain conceptual structure, to recognize aspects and identify cause-effect associations of a process, and variance analysis [50,52] to use existing system as a basis for determining new system requirements. Generally these techniques are considered as knowledge acquisition strategies; however they are also flexible in requirements elicitation. As explained in Table III, laddering [48] is utilized to elicit justification and explanation of technological terminologies or subjective terms, and to elicit, how specialists composite their knowledge about a field, and card sorting [49]. Generally, the analytic strategies are not essential to requirements elicitation, because requirements are grabbed ultimately from other sources, instead of end users and clients. Nevertheless, they form secondary variants, to enhance the performance and usefulness of requirements elicitation, particularly when the information from heritage or relevant products is re-usable.

D. Synthetic Methods

According to [39], the synthetic strategies incorporate various channels of communication, and offer models to illustrate the characteristics and relationship of system, they deliver good hints for requirement recognition, in the form of abundant semantic models. For instance, the prototypes offer an initial version of the system to the users, which can emphasize them about the functions which are usually in any other case ignored. Storyboard technique, which is categorized between scenarios and prototyping. It presents a procession of prospects beginning from sample components to live interactive reports [34]. Appreciative Inquiry (AI) is a combined method, which includes communication between clients and designers, examine the existing systems, tracking the behaviours of the users and visualize the future system or software, with all the necessary functions. It includes combined strategies, which enhance requirements elicitation process. Examples of synthetic methods are illustrated in Table IV.

International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347-6435(Online) Volume 1, Issue 3, December 2013)

TABLE IV
Synthetic methods

Method	literature support	Executor	Illustration
Scenarios, passive storyboards	[20, 34, 35, 53]	Analysts and stakeholder representatives communicate and coordinate to reach a common understanding of the requirements	This approach describes the precise details of the current and future processes, which comprises the actions and interactions between the users and the system
Prototyping, Interactive storyboards	[20, 34, 17]		It provides stakeholders with a concrete (although partial) model or system that they might expect to be delivered at the end of a project. It is often used to elicit and validate system requirements. Prototype generally represents and visualizes the actual parts of system
JAD/RAD	[35, 39, 54, 55]		It stands for Joint Application Development. Rapid Application Development and emphasizes user involvement through group sessions with unbiased facilitator. This approach is more or less similar to the brainstorming, however, it differs is one aspect, where the stakeholders and the users are also allowed to participate and discuss on the design of the proposed system
Contextual inquiry	[56]		It is a combination of open-ended interview, workplace observation, and prototyping. This method is primarily used for interactive systems design where user interface design in critical
Appreciative Inquiry (AI)	[57, 58, 59, 60, 61, 62]		Appreciative Inquiry (acronym "AI") is principally an organizational development method, which focuses on increasing what an organization does well rather than on eliminating what it does badly

The synthetic methods are generally integrated at other phases of the product development life cycle. As the purpose of synthetic methods is to enhance the communication among programmers and the clients, they are appropriate for various phases of the development process. They efficiently coordinate the requirements stage with the remaining development processes.

V. Comparison Between Categories Of Requirements Elicitation Techniques

The categories are due to the outcomes of dissimilarities between the requirements elicitation techniques, each and every group has some characteristics, which make all categories to have distinctive characteristics. The table below shows comparison among various groups of elicitation techniques and their benefits and drawbacks.

TABLE V
Comparison between categories of elicitation techniques

Category	Stakeholder Participation	Observable phenomena	Future system Knowledge
Conversational methods	Definitely stakeholders are the main participants	Not applicable	Some hints or guidelines are provided that might guide the future development of systems. Novel ideas might lead the analyst to envisage future system
Observational methods	Stake holders do not participate, as this method totally involves the observer to observe the activities of the end-user	The whole process is based on the phenomenon of observation (s), hence this criteria is very crucial	Observing the current methods are not easily applied to the development of future concepts
Analytic methods	Stake holders do not participate	A lot of factors, including the codes are observed	The analyzing of codes and other existing documents will pave way for getting knowledge about the future

International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347-6435(Online) Volume 1, Issue 3, December 2013)

			system
Synthetic methods	involves collaboration between stakeholders and systems analysts to identify needs or requirements	Some parts of the process is based on the phenomenon of observations, hence this criteria is very crucial also Behaviors of people are observed	The strong Synthetic methodspractices, make people to give more ideas about the future systems,. Hence gaining knowledge about future system is one of the biggest advantage of this method. Synthetic methods are a strategy for purposeful change that identifies the best of "what is" to pursue dreams and possibilities of "what could be"
Category	**Understanding the domain**	**Identifying sources of requirements**	**Predictive ability of unique attributes elicited**
Conversational methods	Diverse, knowledge sharing might make analyst to understand domains	Nil	Nil
Observational methods	Observational methods clearly makes the analyst to understand the domain	Nil	Depends on the observer
Analytic methods	Again, the code analyzing process will reveal the facts of domain	Definitely, this method needs makes the analysts to understand and identify the source of requirements	Nil
Synthetic methods	As the experts will be made to converse about the domain, it becomes very effective means to understand the domain. This type makes the analyst to unambiguously understand the domain	Synthetic methods generates volumes of data that provide great detail on the origins and consequences of local needs and resource constraints	It's one of the most important features is the predictive and find a unique attributes

Based on the explanation about the various categories of elicitation methods, it is evident that, each and every category has its own advantages and disadvantages. The above table has clearly illustrated the pros and cons of each method. The conversational methods have a lot of advantages, such as the actual fact, which is very helpful for collecting loaded information about the requirements. Apart from unearthing opinions, the interview technique indentifies feelings and goals of different individuals. With the help of interviews, it is easy to dig the details by asking follow-up questions. However the conversational methods have the following disadvantages: it is very difficult to master the skills of interviewing. The requirement elicitation depends a lot on the behaviour and attitude of interviewer. Moreover, the interviewer has to be always unbiased. Even though we can collect lot information with the interview method, we cannot assure for getting meaningful information.

The benefits of observational methods can be summarized as follows; the observational methods are better option for fetching basic aspects of routine order. Furthermore, they offer crucial information for designing solution. These methods are very handy, when the development team lacks experience about product domain. However, there are some problems in practicing the observational methods. The most critical drawback is that, observational methods need a lot of time, and these techniques are not good choice, when schedule is tight. Similar to conversational methods, the observational methods are also very difficult to understand thoroughly. Furthermore, observational methods need sensitivity and responsiveness towards physical environment. Many studies have discussed about the analytic methods and give the techniques that are related to this category as mentioned before, for example reusing the requirements of the existing system as common method of requirements elicitation. There are a lot of advantages of using the existing knowledge to develop the new product, which includes low cost and less time. Despite the disparities of type of users and stakeholders, this method is used very commonly particularly in developing user interfaces, database and security policies.

Many projects have failed due to the employment of inappropriate elicitation methods. One such example of big project failure is "Ariane 5 Flight 501 (European Ariane 5 expendable launch System)", where, the requirements specifications of Ariane 4 were reused.

International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347-6435(Online) Volume 1, Issue 3, December 2013)

However, the flight path of Ariane 5 was very much different, hence, the system developed using the requirements of Ariane 4 was unable to handle the Ariane 5 flight path, this is one of the most important disadvantages of analytic methods, because the failure will be after testing the new software or system, so this methods depends on the old code or software that might completely or partially different from the new software.

The synthetic methods, it is particularly valuable for stakeholders such as, business owners and end users who might not understand the technical aspects of requirements, however, will better relate to a visual representation of the end product. Prototyping as a example may be an interactive screen, a mock-up, a navigation flow or a storyboard. Simple, throwaway prototypes might be executed in the initial stages of discovery, and more detailed, interactive prototypes might be done once business requirements have been identified [31]. Another example under the same category is appreciative inquiry technique (AI), in order to clearly understand this technique, it is crucial to dissect the terms and comprehend the meaning in the context: appreciation means to recognize and value the contributions or attributes of things and people around us. Inquiry indicates the exploration and identification of possibilities of novel ideas. When combined, this term means that by appreciating what is good and valuable in the present situation, it is possible to discover and understand the means to institute positive change for the future [61].

The positive aspects of appreciative inquiry are: this approach is constructed upon the advantages of an organization or group; it understands things that are well implemented. Consequently, a very beneficial and optimistic influence on spirits, guarantee and value of individuals and groups, contributors can become empowered and encouraged. It is very easy to involve people, who do not generally get engaged in this kind of activity, due to the fact of the conversational style of questioning and specific focus on the participants [62, 63, 64]. However, the disadvantages of the appreciative inquiry are: it consumes time, needs periodic commitment, to motivate participants and occasionally additional works should be done to get people out of the SWOT (strengths weaknesses, opportunities threat) mindset.

VI. Conclusion

The beneficial factors of synthetic methods are: these methods is built upon the positive aspects of an organization or group, it recognizes factors, which are well executed; therefore, we can have a very effective and optimistic effect on state of mind, assurance, and value of individuals and groups. Contributors are motivated, because they feel very much appreciated. Due to the casual conversational style of questioning, and specific focus on the participants, it is easy to involve individuals in synthetic methods, who generally do not get involved in these kinds of activities.

REFERENCES

[1] Nelson, R. R. 2007. “IT project management: Infamous failures, classic mistakes, and best practices”. MIS Quarterly Executive. 6(2), 67-78.

[2] Wiegers, K. E. 2006. “More about software requirements: Thorny issues and practical advice”. Microsoft Press: Redmond Washington.

[3] McGovern, F. 2007. “Getting requirements right in the analysis phase”. Compuware White Paper.

[4] Zowghi, D., & Coulin, C. 2005. “2 Requirements Elicitation : A Survey of Techniques, Approaches , and Tools”.

[5] IEEE SWEBOK, R. 2004. "Software Engineering Body of Knowledge."

[6] Boehm, B. & Papaccio, P.1988. “Understanding and Controlling Software Costs”. IEEE Transactions on Software Engineering. SE-4, 10.

[7] McConnell, S. 2001. “From the Editor - An Ounce of Prevention”. IEEE Software. 18, 3.

[8] Romero & Mariona. 2010. “Sure: secure and usable requirements engineering”.Retrieved fromhttp://dl.acm.org/citation.cfm?id=1925644.

[9] Wiegers, K.2003: “Software Requirements”. Microsoft Press.

[10] Pamela Zave. 1997. “Classification of research efforts in requirements engineering”. ACM Comput. Surv. 29(4):315–321.

[11] Raimundas Matulevicius. 2005. “Process Support for Requirements Engineering A Requirements Engineering Tool Evaluation Approach”. PhD thesis, NTNU. Doctoral theses at NTNU. 142.

[12] Sommerville and Sawyer. 1997. “Requirements Engineering”. A Good Practice. John Wiley and Sons.

[13] Software Test & Evaluation Panel (STEP). 1991. “Requirements Definition Implementation Team: Operational Requirements for Automated Capabilities”. Draft Pamphlet (Draft PAM).

[14] Loucopoulos, P., and Champion. 1989. “R.E.M.: Knowledge-Based Support for Requirements Engineering”. Information and Software Technology.

[15] Davis, A. 1990. “Software Requirements: Analysis and Specification”. Prentice Hall.

[16] Dorfman, M. 1990. “Tutorial: System and Software Requirements Engineering”. IEEE Computer Society Press.

[17] Nuseibeh, B. and Easterbrook, S. 2000. “Requirements engineering”. a roadmap. in Proceedings of the Conference on The Future of Software Engineering, (Limerick, Ireland), ACM Press. 35 - 46.

[18] Goguen, J. & Linden, C. 1993. “Techniques for Requirements Elicitation”. 1st IEEE International Symposium on Requirements Engineering (RE'93), San Diego, USA, 4-6th January. pp. 152-164. http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=324822.

[19] Loucopoulos, P., and Karakostas. 1995. “V. System Requirements Engineering McGraw-Hill”. Inc., New York, NY.

[20] Kotonya, G., and Sommerville, I. 1998. “Requirements Engineering”. Processes and Techniques John Wiley & Sons. New York, NY.

[21] Escalona, M., & Koch, N. 2004. “Requirements engineering for web applications-a comparative study”. Journal of Web Engineering. 2(3), 193–212. Retrieved from http://citeseerx.ist.psu.edu /viewdoc/download?doi=10.1.1.153.5974&rep=rep1&type=pdf.

[22] IEEE SWEBOK, R. 2004. "Software Engineering Body of Knowledge."

[23] Browne, G. J., & Ramesh, V. 2002. “Improving information requirements determination: a cognitive perspective”. Information & Management. 39(8), 625–645. doi:10.1016/S0378-7206(02)00014-9.

[24] Vessey, I., and Conger, S. 1994. "Requirements Specification: Learning Object, Process, and Data Methodologies". Communications of the ACM. (37:5), pp 102-113.

[25] Wallace, D.R., and Ippolito, L.M. 1997. "Verifying and Validating Software Requirements Specifications". in: Software Requirements Engineering, R.H. Thayer and M. Dorfman (eds.), IEEE Computer Society Press. Los Alamitos. CA. pp. 389-404.

[26] Whitney, D. 1998. "Let’s change the subject and change our organization: an appreciative inquiry approach to organization change". Career Development International (3:7), pp 314-319.

[27] Guinan, P., Bostrom, R.P. 1986. “Development of computer based information system: A communications framework”. SIGMIS Database. 17(3), 3-16.

[28] Browne, G. J., & Ramesh, V. 2002. “Improving information requirements determination: a cognitive perspective”. Information & Management. 39(8), 625–645. doi:10.1016/S0378-7206(02)00014-9.

[29] Hickey AM, Davis AM. 2004. “A unified model of requirements elicitation”. J Manage Inf Syst 20(4):65.

[30] Farzan R, DiMicco JM, Millen DR, Dugan C, Geyer W, Brownholtz EA. 2008. “Results from deploying a participation incentive mechanism within the enterprise”.

[31] Zhang, Z. 2007. “Effective Requirements Development-A Comparison of Requirements Elicitation techniques”. Tampere, Finland, INSPIRE, 225–240. Retrieved from http://pdf.aminer.org/000/359/901/requirements_elicitation_with_indirect_knowledge_elicitation_techniques_comparison_of_three.pdf.

[32] Avison, D.E. and Fitzgerald, G. (eds.). 1995. “Information Systems Development: Methodologies, Techniques and Tools”. McGraw-Hill Book Company.

[33] Goguen, J. & Linden, C. 1993. “Techniques for Requirements Elicitation”. 1st IEEE International Symposium on Requirements Engineering (RE'93), San Diego, USA, 4-6th January. pp. 152-164. http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=324822.

[34] Leffingwell, D. and Widrig, D. 2003. “Managing Software Requirements - A User Case Approach”. 2nd Ed. Addison-Wesley.

[35] Zowghi, D., & Coulin, C. 2005. “2 Requirements Elicitation : A Survey of Techniques, Approaches , and Tools”.

[36] Tsumaki, T., & Tamai, T. 2005. “A framework for matching requirements engineering techniques to project characteristics and situation changes”. … of Situational Requirements Engineering …, 44–58. Retrieved from http://cui.unige.ch/db-research/SREP05/Papers/04.pdf.

[37] [37] Chauncey e. Wilson. 2006. “Brainstorming Pitfalls and Best Practices”. ACM. vol 13,issue 5.,pages 50-63. http://delivery.acm.org/10.1145/1160000/1151342/p50-wilson. pdf?key1=1151342&key2=0909760121&coll=ACM&dl=ACM&CFID=67531224&CFTOKEN=54899420.

[38] Linn Gustafson. 2008. “Requirements Engineering Verification validation”. University West, Course slides.

[39] Maiden, N. &Rugg, G. 1996. ACRE: “Selecting Methods for Requirement Acquisition”. IEEE, Software Engineering Journal. 11(3):183-19247.

[40] Christel, M.G. and Kang, K.C. 1992. “Issues in requirements elicitation Technical report”. CMU/SEI-92-TR-12 ESC-TR-92-012. Carnegie Mellon University. Pittsburgh, PA,80.

[41] Stephen Viler & Ian Sommerville. 1999. “Social analysis in the requirements engineering process: From ethnography to method”. IEEE. pages 6-13. http://ieeexplore.ieee.org/iel5/6303/16860/00777980. pdf?tp=&arnumber =777980&isnumber=16860.

[42] Viller, S. and Sommerville, I. 2000. “Social analysis in the requirements engineering process: from ethnography to method”. in Proceedings of the 4th International Symposium on Requirements Engineering. (RE'99), (Limerick, Ireland, 1999), IEEE CS press.

[43] Myers, M.D. 1999. “Investigating information systems with ethnographic research”. Communications of the AIS, 2.

[44] Hutchings, A.F. and Knox, S.T. 1995. “Creating products: Customer demand”. Comm. ACM. 38 (5). 72-80.

[45] Cybulski, J.L. and Reed, K. 2000. “Requirements Classification and Reuse: Crossing Domain Boundaries”. in Conference on Software Reuse. ICSR'2000, (Vienna and Austria), 190-210.

[46] Knethen, A.v., Paech, B., Kiedaisch, F. and Houdek, F. 2002. “Systematic requirements recycling through abstraction and traceability”. in IEEE Joint International Conference on Requirements Engineering.

[47] Woo, H.G. and Robinson, W.N. 2002. “Reuse of scenario specifications using an automated relational learner: a lightweight approach”. in IEEE Joint International Conference on Requirements Engineering. 173 - 180.

[48] Rugg, G., Eva, M., Mahmood, A., Rehman, N., Andrews, S. and Davies, S. 2002. “Eliciting information about organizational culture via laddering”. Information Systems Journal, 12 (3).215-229.

[49] Rugg, G. and McGeorge, P. 1999. “The concept sorting techniques”. The Encyclopedia of Library and Information Science. 65(28). 43 - 71.

[50] Byrd, T.A., Cossick, K.L. and Zmud, R.W. 1992. “A Synthesis of Research on Requirements Analysis and Knowledge Acquisition Techniques”. MIS Quarterly. 16 (1). 117 - 138.

[51] Wright, G. and Ayton, P. 1987. “Eliciting and modeling expert knowledge”. Decision Support Systems. 3 (4). 13-26.

[52] Hawgood, J., Land, F. and Mumford, E. 1978. "A participative approach to forward planning and system change". In Proceedings of the 2nd Conference of the European Cooperation in Informatics. (Venice, Italy, 1978), 39 -81.

[53] Stuart Anderson Massimo Felici. 2001. "Requirements engineering questionnaire". http://homepages.inf.ed.ac.uk/mfelici/doc/questionnaire.pdf.

[54] Coughlan, J., & Macredie, R. 2002. "Effective communication in requirements elicitation: a comparison of methodologies". Requirements Engineering. 7(2), 47–60. doi:10.1007/s007660200004.

[55] Charles F. Manski1 and Francesca Molinari. 2008. "Skip Sequencing:A Decision Problem In Questionnaire Design". North-western University and Cornell University. vol 2, pages 264-285. http://arxiv.org/PS_cache/arxiv/pdf/0803/0803.3875v1.pdf.

[56] Holtzblatt, K. and Beyer, H. 1993. "Making customer-centered design work for teams". Comm. ACM. 36 (10). 93 - 103.

[57] Cooperrider, D.L. & Srivastva, S. 1987. "Appreciative inquiry in organizational life". In Woodman, R. W. & Pasmore, W.A. (eds) Research In Organizational Change And Development. Vol. 1 (129-169). Stamford, CT: JAI Press.

[58] Cooperrider, D.L. & Whitney, D. 2001. "A positive revolution in change. In Cooperrider". D. L. Sorenson, P., Whitney, D. & Yeager, T. (eds.) Appreciative Inquiry: An Emerging Direction for Organization Development (9-29). Champaign, IL: Stipes.

[59] Barrett, F.J. & Fry, R.E. 2005. "Appreciative Inquiry: A Positive Approach to Building Cooperative Capacity". Chagrin Falls, OH: Taos Institute.

[60] Whitney, D. & Trosten-Bloom, A. 2003. "The Power of Appreciative Inquiry". San Francisco: Berrett-Koehler.

[61] Kelm, J.B. 2005. "Appreciative Living: The Principles of Appreciative Inquiry in Personal Life". Wake Forest, NC: Venet.

[62] Mrayat, O., Norwawi, N., & Basir, N. 2013. "Appreciative Inquiry in Eliciting Software Requirements". International Journal of Computer Science and Electronics Engineering (IJCSEE), 1(3).

[63] Gonzales, C. 2011a. "Eliciting user requirements using Appreciative inquiry". Empirical Software Engineering. 20(4), 65–84. Retrieved from. http://www.springerlink.com/index/X74905W1820K7187.pdf.

[64] Gonzales, C. K., & Leroy, G. 2011. "Eliciting user requirements using Appreciative inquiry". Empirical Software Engineering, 16(6), 733–772. doi:10.1007/s10664-011-9156-x.