

Appreciative Inquiry in Eliciting Software Requirements

Omar Isam Al Mrayat, Norita Md Norwawi, and Nurlida Basir

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 exceeding the budget are considered foremost. In order to address this issue and to facilitate system designers, this study has developed a modified user requirements elicitation technique, based on principles of Appreciative Inquiry (AI), which was proposed by Carol K. Gonzales and Gonyea Leroy. Appreciative Inquiry 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 comprehending the appreciative inquiry strategy; this is crucial to determine the requirements of the user, as it enables much better software development and does not waste resources unnecessarily. It is essential for all the end users to persistently participate in all the stages of appreciative inquiry, along with developers. Basically, AI will complement the present approaches, by representing a optimistic and potential factor for a proposed system, which results in much better user needs, and identifies novel and distinctive specifications, which has been presented as a case study in this paper.

Keywords—Appreciative Inquiry, Eliciting User Requirements, development, Technique.

I. INTRODUCTION

ACQUIRING the user requirements is one of the most crucial aspects in developing information systems, as requirements choices are impacted by rudimentary and unclear information; nevertheless the quality of the requirements is very crucial for the success of any project [1], [2].

According to [3] requirements elicitation needs a lot of efforts and it is not an easy task. [4] have stated that, perspectives, cognitive model and anticipating the variances between the users and analysts make requirements elicitation a challenging and contradicting task. In most of the instances, customers tend to be confused about their actual requirements. in few other instances, the existing work methods do not match with the expectations of management. Elicitation is a strategy, which will be used by the developers for understanding the needs. Therefore, elicitation is the initial phase in collecting user requirements; it is the method used for

understanding and acquiring the requirements of users and other stakeholders [5], [6].

Requirements elicitation is a one of the much focused topic among researchers and technologists [7]. Vague and misinterpreted user requirements are considered as one of the root causes of software development failures [7]. Despite the existence of a lot of strategies for collecting users' requirements, still, quite a number of software development projects fail, due to inadequate user requirement gathering, furthermore these failures are persistent and costly problem.

It is essential to note here that, generally, the confusion of choosing, or picking the wrong technique to collect user requirements, make the developers fatally fail in system development. For identifying the needs of users, it is essential to create a process, so that the delivery of information can be more real. Collecting requirements of stakeholder is a preliminary, but ongoing and extremely vital phase in software development. The requirement phase is subject to a huge amount of errors, which are swayed by crucial aspects included in requirements elicitation. Consequently, this study presents the following research question: despite the existence of a lot of strategies, for collecting users' requirements, still, why do a number of software development projects still fail?

II. BACKGROUND

A. Techniques Overview

As discussed earlier, collecting and assessing the requirements of users are very crucial for the accomplishment of systems development as expected; on the other hand, due to the discrepancy of the conventional methods used for gathering user requirements, they will not always produce the expected outcomes. There are many cases, where the users refrain to reveal their actual job responsibilities, as they do not want to show what they are really performing; therefore they could possibly end up in providing incorrect information. Subsequently, this may result in creating a distended system, with unnecessary functions.

As mentioned earlier, it is crucial to collect accurate user requirements for the success of system development, where, inaccurate information collected will confine the success. The traditional approaches are erroneous due to a lot of factors; observation method is one of the more widely used, traditional approaches [4], as the name implies, the developers observe the specific execution of current processes by the users,

Omar. Isam Al Mrayat¹, Norita Md Norwawi², Nurlida Basir³ are from Faculty of Science and Technology, Universiti Sains Islam Malaysia (USIM), Bandar Baru Nilai, 71800 Nilai, Negeri Sembilan (¹ Phone: +60182895078. Email: omar_sql@yahoo.com²Email:norita@usim.edu.my³ Email: nurlida@usim.edu.my)

without directly interfering. Generally, this strategy is employed in association with other methods, such as, interviews and task analysis [8], [9]. This clearly indicates that, observation method does not stand on its own, rather it needs the support of few other techniques, and consumes a lot of time; user comments and observation of analysts might disrupt tasks. Furthermore, observation method is very expensive, and demands considerable expertise and efforts from the analyst, to fully grasp and describe the activities being carried out. The efficiency of observation and other methods can differ, as users tend to modify their activities, when they are observed [10]. According to [11] the observation method consumes a lot of time, it is not suitable for projects with tight deadlines, and moreover observation needs level of sensitivity and responsiveness to the physical environment. On the other hand, interviews [12] are most likely the most conventional and generally utilized strategy for collecting requirements, as they are basically human-based cultural activities. [8] have stated that, interviews need to get access and incorporate a wide variety of perhaps varying viewpoints from several users. According to [13], information collected through interviews might considerably differ, based on the expertise of the interviewer, as a result, there is a lack of standards in implementing this technique, to acquire the needs; for instance, if the analysts separately employ this technique with the same users, probably the outcomes of the two interview sessions will differ. [14] has stated that, there is a risk of completely neglecting some topics, which is yet another issue of unstructured interviews, which focuses too much only on some areas. Moreover, interview method needs vast experience of the interviewer, who must be capable of choosing the most suitable interviewees [15]. On the other hand, it is challenging to determine the limitations of the proposed software and the organization processes, by employing this approach [9].

Prototype is another type of elicitation technique; they do not facilitate assessment of very good details [8]. Nevertheless, this method also consumes a lot of time to be constructed, and if the users feel that the prototype is inappropriate; all the efforts end in vain. [4] have stated that, prototypes are used in association with other elicitation techniques, such as interviews and joint Application development (JAD); in a lot of instances, prototypes are costly to develop, especially in terms of time and effort; which means that, this technique not ideal strategy. According to [16] with prototyping the requirements are evidently conveyed between the respondents of the sessions, but have to be open and require enhancements, for achieving the same level of clarity from other stakeholders. On the other hand, prototyping is limited with interaction with specific users, rather than with many users. [17] states that, it seems that, prototyping can also be effortlessly merged with other requirements elicitation techniques. Nevertheless, prototyping is more expensive, than all the other techniques, and can lead to incomplete software setup.

B. Appreciative Inquiry

[18] have proposed principles of appreciative inquiry. The aim of appreciative inquiry is to help organizations to develop, procedures and system, by using the current productive cases in software development. It is essential for the developer to concentrate on the preliminary feasibility studies, apart from finding the actual user requirements, for the software development. It is crucial to conduct elicitation process, using the appreciative inquiry, to accomplish these goals of the system, whereby avoiding wastage of time and money.

The method of appreciative inquiry is used in evaluation of user requirements, due to the fact that, developers lack or get unwanted information. As discussed earlier, for the purpose of identifying the needs of users, a process has to be created, whereby the information delivery can be more actual; which helps to develop better processes and new software, which can meet the current and future requirements. [18], states that, "Appreciative Inquiry was evaluated for its effectiveness with eliciting user requirements in different circumstances".

The appreciative inquiry will be carried out with the main stakeholders of the system (organization), including, users of the future software, owners, technical staff and the management teams. [19] states that, "When users participate in system design, they are able to communicate their needs and problems that they hope to solve. Otherwise, if their needs are not met, the system will not be used".

C. Eliciting User Requirements Using Appreciative Inquiry

1. Core process

As illustrated in figure 1, this process employs iterative 4-D cycle such as, Discovery, Dream, Design and destiny [20].

a) Discovery phase

In this stage, the user focuses at 'what is', where the developer identifies, the existing situation of organizations. In this phase, the stakeholders will present their knowledge, expertise, and how they have been managing organizations so far. They will reveal their testimonies of success in the organization and how they have achieved it. Three fundamental questions were formulated by Cooperrider, for performing these evaluations:

- "Describe the greatest experience of your lifetime?"
- "What is that you value most in you, your work and organization?"
- "What are the core factors that give life to your organization?"

b) Dream phase

Focuses on 'what might be'. This phase investigates the perception of users in terms of future of the system especially, and organization as a whole. Furthermore, in this phase the creativity of the stakeholder will be motivated. The stakeholder will be capable of proficiently contributing, due to the fact that, they have already accomplished the discovery phase. Hence, there must be no limitations to understand the dreams of stakeholders.

c) *Design phase*

This phase focuses on ‘how can it be’, where the customer and the developer examine the feasible size of the system. Based on the recognized strengths highlighted by the stakeholder, consequently the dimension and the strength of the system can now be derived. The role of organizations, their associations and procedures are viewed in the external point of view.

d) *Destiny phase*

This phase focuses on the ‘what will be’ factor, this phase is more definite than the dream phase, where the developer and user focus on the expected software, which will be developed and employed. This phase is aimed at defining the activities that have to be considered. Later, this phase supports the organization to gather information, as the ideas about the system, which need to be designed become much more apparent. Here the contribution of stakeholders will be obvious and a typical agreement ought to be offered, as how the system will be designed.

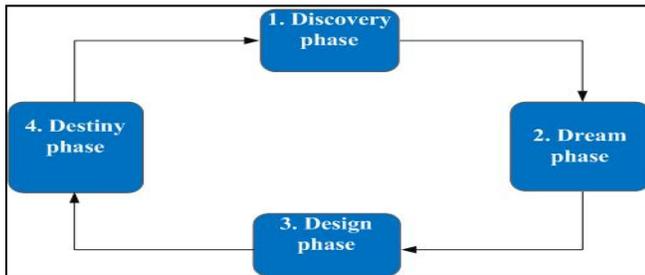


Fig. 1 Iterative 4-D AI

2. *Methods*

Methods employed in this process, and consequently how they are utilized in the development of software: effort will be made to examine the present cases for effectively addressing an successful and precise method for software development. Various cases will be used to indicate how several areas can get benefitted from this process as shown in the table below:

3. *Focusing and gauging*

Focusing and gauging the results of the user requirements, such as, kind of organization, significance of the proposed sub-programs, which the user needs. Few of the areas, which need to be considered, are:

- 1) Availability of feedback in the existing system must be considered.
- 2) Feedback for new process from the appreciative inquiry must be ensured.
- 3) Functionality of the system must be enhanced.
- 4) Effectiveness of the user and development teams must be assured.
- 5) Understand the significance of participating in the requirement of appreciative inquiry.

TABLE I
SOME CASES

Case Study	Method	Measurement	Stake holders
End user	The traditional method of gathering information is compared against appreciative inquiry process.	Gauging the requirements and inputs from the users.	End users and developers of the system. Past system developers may also be included in this case.
Customers & developers of system	Analyzing the requirements of the customers, especially external to the organization and developing documentation.	Investigating other retail or software.	Customers of the organization and developer team members.
Project Members	Study the multiple cases based on the results of appreciative inquiry.	Measure the requirements of the stakeholders and developing teams.	Users, testers of the system, development team.

4. *Requirements of the study*

- The project proposed by a software development team, has to fundamentally meet the target of the following perspectives: functionality, quality and future development.
- Functionality is the aspect related to the features of software, which must be working perfectly that makes it usable. Functions represent ‘what’ the software must do, such as, keeping track of the stock and inventory, computing taxes etc.
- Quality is the aspect associated with the evaluation among different software within the same organization, and perhaps towards software in other organizations of similar business model. Quality may also additionally engage in offering an edge against competitors of an organization. Discrepancies of requirements are assessed by the level, which a requirement alters during the development process [8]. Therefore, it is evident that, requirements of software might even change, while it is being developed. Future needs focus on the new avenues that the business will take in the near future. Existing trends in business might change, which results in making the system less useful or redundant. For example, a website currently selling mobile phones, might want to enter the market for other electronic gadgets such as, tablets; in this context, the current website may be able to cope with different models and prices of mobile phones, but it may possibly required to include models and prices of tablets as well.

III. CASE STUDY

In this study we have selected accounting field to make our case study, because this field is informative and full of tasks, which must be implemented and have too much of requirements to be achieved in the future software. We will use the same questions recommended by [20].

A. Discovery Phase

- What were your hopes and dreams, when you chose this project?
A: To deliver an effective system to the client
- Based on your past experiences, what was the greatest experience you have had with a project, when were you most successful and satisfied?
A: The trust by the customer, and working in front of them, with their participation are the greatest achievements. It gives us immense satisfaction, when we would satisfy the customers with minimal effort and time, within the International Financial Accounting (IFA).
- Did you get any help from your friends/colleagues? Were you able to help them in return?
A: Yes we did receive assistance from our friends and colleagues, and of course we reciprocated the help, based on our experience, we were able to solve the problems of our colleagues.
- Did you experience any unexpected incidents or face a difficult challenge? What did you learn from those?
A: Yes, every difficult challenge we face is our learning curve, above all it boosts our courage of facing any other bigger challenge. For example (Real case), converting manual accounting entries to the code example: the sales code is 01, this leads to solve a big challenge for our organization. Because the traditional (manual) process needs a lot of time, effort and margin of error is high, also checking in this case is difficult.
- What conditions are contributed to that extraordinary level of success and satisfaction?
A: The most important aspect is manually migrating data, this process demands a great deal of time and hard work, this process must be carried out with high precision, which is very challenging in our field. But we are now very much relieved now, as the new software can migrate the data after each movement and update records and data, the budget with high accuracy, and at low cost.
- What do you value most about yourself and your capabilities, as a member of the team, or as a contributor of the project?
A: We can get involved and play a significant role in building the foundation of this software, by using our expertise and skills, to transform the manual processes towards the computerized processes. This will help us to get over the complications faced by us, especially subtracting the process of exports, and store the outcomes again in table; the new software will also help us to, make a query, related to inventory any time we wish; to

know the financial situation in the organization, and business movements at any point of time.

- What do you value most about yourself as a member of the organization and/or member of the team?
A: I value both, as a member in the organization I perform my duties to solve problems, and provide best services to our customers; and at the level of team, we focused on the software development, which is essential in putting things required of the system.
- What are the most important attributes that support your highest levels of success and satisfaction?
A: Apply all kinds of accounting transactions in future system, inform the customers about all the operations in page of its own, to get their opinions about the tables and results.

B. Dream Phase

- What results do you expect from a team or project?
A: I expect all the following: Income list, List cash expenses, Budget page and customers Feedback.
- What do you envision as an ideal project in the future after many years (when, you have grand children)?
A: To be the most powerful and largest accounting firm in the Market
The next phases will use answer these questions to know "how can be and what will be".

C. Design Phase

- How can it be?
A: Software Size: all types of accounting transactions, notify the customers about all the operations.
Strength Identification: better services, customers trust, working in front of customers with their participation, work under International Financial Accounting (IFA), automatically migrate the data, high accuracy, make a query regarding inventory at any moment, account all the following: Income list, List cash expenses, Budget page and customers Feedback.
Organization Roles: customer participation, get account customer feedback, outsider's viewpoint.

D. Destiny Phase

- What will be?
 - a) Requirements: (convert manual processes to computerized processes).
 - 1) Converting manual accounting entries to the code.
 - 2) Migrating data after each movement, and updating records and data.
 - 3) Migrating data up to the budget.
 - 4) Storing the results of the operations in table.
 - 5) Making a query about inventory sat any time
 - 6) Showing business movements at any moment.
 - 7) Applying all kinds of accounting transactions.
 - 8) Informing the customers about all the operations and results in isolated page.

- b) Timeline: This system will be built within six months.
- c) Resources: Old data, users, stakeholders, developers and customers.

After completing this case study, using appreciative inquiry method, we had used another technique to study the same case, to prove the efficiency of appreciative inquiry in finding unique requirements, as against the other technique:

It is practically impossible for the researchers to gather users' requirements, just with observation technique alone, so we have used interview technique in association with observation, to obtain the user requirements as much as possible, by consuming the same amount of time used in achieve appreciative inquiry processes. We got only four, user requirements. The table below is illustrates the results of the case study.

TABLE II
TECHNIQUES COMPARISON

Attribute Technique	Appreciative Inquiry	Observation	Interview	Observation & Interview
User participation	Yes, Too Much	No	Yes	Yes
Unique requirements	Yes	Some Times	Some Times	Some Times
Requirement number	8	0	3	4
Understand system	Yes, Too Much	No	Yes	Yes

IV. RELATED WORK

Appreciative Inquiry has been proposed by [20], they conducted some studies using a process and common set of measures, by using comparisons as a means of identifying recurring practices, and also examines improvements; the first study was a comparative case study with end-users, to identify if appreciative inquiry would enhance requirements collection and user attitudes, as against brainstorming. The second study was a controlled user study, where they have compared appreciative inquiry, with brainstorming, to identify how effectively the previous results from the case study, could be replicated in a controlled experiment. The third study highlighted the works of previous studies, however assessed the use of appreciative inquiry with a number of genuine project teams of students, in an undergraduate Computer Information Systems (CIS) course, to evaluate the outcomes of needs and behaviors in different project phases. In order to consistently improve generalization of the results, the fourth study has been carried out, as multiple case studies, by employing a comprehensive duplicated processes and measures, for cross-case evaluation with the actual project teams of CIS course students. The fourth study has been aimed at further evaluating the effectiveness of appreciative inquiry (requirements and attitudes), in numerous case studies, which includes achievable relevant aspects, such as, efficiency of team, identified stress, and emotional intelligence, which

assisted to describe results. They have achieved positive results, in all the four case studies, and they were able to get distinctive and new requirements. However, our study is different from the above, as we have compared the results of appreciative inquiry in the case study, with two traditional techniques such as, observation and interview.

V. CONCLUSION

The beneficial factors of appreciative inquiry are: this approach 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 appreciative inquiry, who generally do not get involved in these kinds of activities. As explained earlier in case study, the appreciative inquiry is a very effective method in eliciting software requirements. By using this technique, unique and new requirements will be obtained, which consequently positively impact the proposed software. It is noteworthy that, success of any software development depends on the process of collecting users' requirements, and the technique that has been used in the software industry process.

REFERENCES

- [1] Christ of Ebert. (2004). Practical Requirements engineering solutions, University of twente, IEEE publications, volume 21, issue 2, pages: 16-18 <http://csdl.computer.org/comp/mags/so/2004/02/s2016.pdf>.
- [2] Suzanne Robertson. James Robertson. (1998). Mastering the requirements process. Addison Wesley.
- [3] Merlin Dorfman. (1999) Requirements engineering. Carnegie Mellon University, interactive publishers, <http://www.sei.cmu.edu/news-at-sei/features/1999/mar/Background.mar99.pdf>.
- [4] Zowghi, D., & Coulin, C. (2005). 2 Requirements Elicitation : A Survey of Techniques , Approaches , and Tools.
- [5] Browne GJ, Ramesh V (2002), Improving information requirements determination: a cognitive perspective. *Inf Manage* 39(8):625.
- [6] Hickey AM, Davis AM (2004) A unified model of requirements elicitation. *J Manage Inf Syst* 20(4):65.
- [7] Herrmann, A., Paech, B. (2009). Practical challenges and requirements prioritization based on risk estimation, *Empirical Software Engineering*, 14(6), 644-684. Hickey AM, Davis AM (2004) A unified model of requirements elicitation. *J Manage Inf Syst* 20(4):65.
- [8] Barney, S., Aurum, A., & Wohlin, C. (2008). A product management challenge: Creating software product value through requirements selection. *Journal of Systems Architecture*, 54(6), 576-593. doi:10.1016/j.sysarc.2007.12.004.
- [9] Gunda, S. G. (2008). Requirements Engineering: Elicitation Techniques.
- [10] Myers, M.D. (1999). Investigating information systems with ethnographic research. *Communications of the AIS*, 2.
- [11] Zhang, Z. (2007). Effective Requirements Development—A Comparison of Requirements Elicitation Techniques. *Software Quality Management XV: Software Quality in the Knowledge Society*, E. Berki, J. Nummenmaa, I. Sunley, M. Ross and G. Staples (Ed.) British Computer Society, 225–240.
- [12] Holtzblatt, K. (1995). Requirements gathering: the human factor. *Communications of the ACM*. Retrieved from <http://dl.acm.org/citation.cfm?id=203361>.
- [13] Goguen, J. (1993). Techniques for requirements elicitation. *Requirements Engineering*, 1993.. Retrieved from http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=324822.

- [14] McGraw, K. (1989). Knowledge acquisition: principles and guidelines. Retrieved from <http://orton.catie.ac.cr/cgi-bin/wxis.exe/?IsisScript=LIBROSCO.xis&method=post&formato=2&cantidad=1&expresion=mfn=011358>.
- [15] Pan, D., Zhu, D., Johnson, K. (2001). Requirements Engineering Techniques.
- [16] Raghavan, S. (1994). Lecture notes on requirements elicitation. Retrieved from <http://oai.dtic.mil/oai/oai?verb=getRecord&metadataPrefix=html&identifier=ADA278536>.
- [17] Tuunanen, T. (2003). A new perspective on requirements elicitation methods. JITTA: Journal of Information Technology Theory & Application, 5(3), 45–62. Retrieved from <http://moodle.stoa.usp.br/file.php/1359/WAEUs-Tuunanen2003.pdf>.
- [18] Carol K. Gonzales & Gony Leroy ,(2011), Eliciting user requirements using Appreciative inquiry, Springer Science+Business Media, LLC.
- [19] Avital M, Boland RJ, Cooperrider DL (eds) (2008) Designing information and organizations with a positive lens (vol. 2). Elsevier Science, Oxford.
- [20] Cooperrider, W., Stavros (2008), Appreciative Inquiry Habdbook for Leaders of Change (2nd ed.), Brunswick, OH, Crown Custom Publishing, Inc.