RESEARCH ARTICLE OPEN ACCESS

Software Engineering: Requirements Elicitation Techniques, Issues and Challenges in Industry

S. Vinay Kumar*, Dr.N.Kasiviswanath**, M. Haritha***
**(Assistant Professor, Department of Computer Science & Engineering, G. Pulla Reddy Engineering College (AUTONOMOUS): KURNOOL, Kurnool, Andhra Pradesh mail id:vinay.gprec@gmail.com*
*** (Professor & HOD, Department of Computer Science and Engineering, G. Pulla Reddy Engineering College (AUTONOMOUS): KURNOOL, Kurnool, Andhra Pradesh) mail id:gprechodcse@gmail.com*
*** *(Assistant Professor, Department of Computer Science & Engineering, G. Pulla Reddy Engineering College (AUTONOMOUS): KURNOOL, Kurnool, Andhra Pradesh mail id:mharitha22@gmail.com*
Corresponding Author ; S. Vinay Kumar

ABSTRACT : This paper highlights various issues and challenges of requirements discovering techniques in the industry, and is based on the analysis of the previous data. Drawing out the requirements from the stakeholders it is important, but it is difficult to identify the requirements present in the user mind. Hence many different kinds of techniques have been defined for the process of requirements elicitation. This paper gives an overview of these techniques and also highlights the key problems faced by the industry in spite of this set of techniques. The paper also presents the need for innovative techniques that can upgrade the existing ones. Methods/Statistical Analysis: A literature review method was adapted. The literature search was conducted based on several keywords, namely, requirements elicitation, requirements challenges and validation techniques. Subsequently, critical analysis of the available literatures was conducted using the thematic analysis approach.
Keywords - Requirements Elicitation Techniques, Requirements specification, Stakeholders, Validation.

Date Of Submission: 24-01-2019 Date Of Acceptance:08-02-2019

I. INTRODUCTION

In the Software Development Life cycle, communication phase is the most important stage where requirements are gathered. Project initiation and requirements gathering are the starting stages which involve the requirements elicitation phase. This needs the identification of the different stakeholders who will be the key participants of the initial project requirements discovering process. Stakeholder identification includes identifying the people who are in the entry level, middle level, and exit level. In Industry different kinds of techniques are defined in a number of ways based on the different viewpoints of different customers to grab the requirements from them: by simply asking questions to customers, by predefined agenda, or by some open interviews, or by Quality Function Deployment (QFD), or by Brain storming sessions etc...

II. REQUIREMENTS ENGINEERING

Requirements Elicitation is the process that involves all the conventional and advanced methods that elicit the requirements from the customers. But even with this powerful set of techniques the best and exact requirements can't be drawn out from the customers. So for the challenges that are faced by the industry people, there is a well known technique called the Requirements Engineering process {RE} **RE** is the process of establishing the services that the customer requires from a system and the constraints under which the system is developed and operates. The **requirements** themselves are the descriptions of the system services and constraints that are generated during the requirements engineering process. Requirements may range from a high-level abstract statement of a service or of a system constraint to a detailed mathematical functional specification. As much as possible, requirements should describe **what** the system should do, but **not how** it should do it.

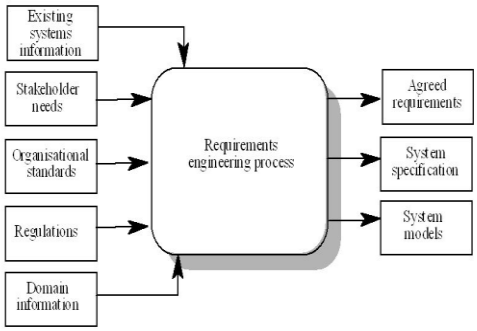


Fig.1 Requirements Engineering Process

III. REQUIREMENTS ELICITATION PROCESS

To gather the requirements, a feasibility study is to be done at first. Based upon the feasibility reports that are generated we need to take the decision whether the process is to be stopped or to be continued with the project. The next stage is requirements elicitation process from which some system models are generated.

2.1 Techniques for Requirements Discovery process

The various requirement discovery techniques are:

2.1.1. Viewpoints
2.1.2. Scenarios
2.1.3. Brainstorming Sessions
2.1.4. Facilitated Application Specification Technique (FAST)
2.1.5. Quality Function Deployment (QFD)
2.1.6. Use case Approach

2.1.1 Viewpoints: The concept of views and viewpoints is widely used across the architectural community, having originated back in the 1970's where Ross's Structure Analysis and Design Technique used them. The concept of Views became widely accepted following the development of Kruchten's 4 + 1 architecture view model, they have also been formalized in the ISO/IEC/IEEE 42010:2011, *Systems and software engineering — Architecture description*.

The principles of views and viewpoints are defined in slightly different ways in different places; the definitions adopted by IASA are:

- An architectural view is a representation of one or more aspects of an architecture that illustrates how the architecture addresses the concerns held by one or more of its stakeholders.
- A viewpoint is a collection of patterns, templates, and conventions for constructing one type of view. It defines the stakeholders whose concerns are reflected in the viewpoint and the guidelines, principles, and template models for constructing its views.

An example would be to use an operational viewpoint to create a view targeted at the Help Desk manager. The viewpoint is the template that contains information relevant to the operations of the system, and a view is the end product delivered to someone interested in maintaining the operational capability.

IT architects must have the ability to compare/contrast concept of views, viewpoints, and perspectives, understand the differences between them and how they work together to describe an architecture. They must be able to discern various stakeholder groups typical of IT development projects, describing the typical viewpoint of each group, and determine the set of views needed to satisfy project requirements.

OVERVIEW:

One of the main activities and responsibilities of the architect is to present the vision of the system to the stakeholders interested in it. In anything other than the most trivial of systems it is not possible to show this in a single diagram, although sometime this is attempted, so the concept of views and viewpoints has been developed to provide the appropriate information in an appropriate way for each set of stakeholders.

The architect has to understand the organization and the problem space to identify the appropriate framework to use for a particular architectural problem.

The ISO 42010 defines the relationship between the elements of architecture, its stakeholders and the description, as shown below:

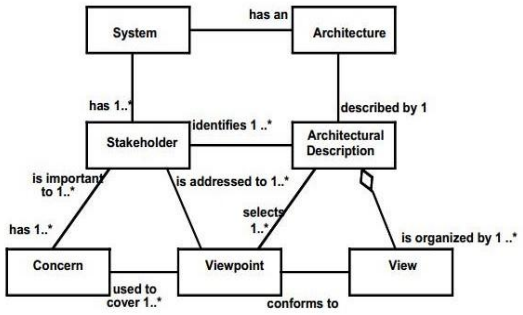


Fig.2 Different views of the complete software

Most architecture documents include a set of views and viewpoints which comprise the main content of the document. In a general sense the key viewpoints common to many architectural frameworks will include a description of the functions required, the data structures to be held, a description of the processing to be carried, how the software is developed and managed and how the software is deployed on the infrastructure.

The views of the model are concerned with:

- **Logical view**: The logical view is concerned with the functionality that the system provides to end-users. UML Diagrams used to represent the logical view include Class diagram, Communication diagram, and Sequence diagram.
- **Development view**: The development view illustrates a system from a programmer's perspective and is concerned with software management. This view is also known as the implementation view. It uses the UML Component diagram to describe system components. UML Diagrams used to represent

the development view include the Package diagram.

- **Process view**: The process view deals with the dynamic aspects of the system, explains the system processes and how they communicate, and focuses on the runtime behavior of the system. The process view addresses concurrency, distribution, integrators, performance, and scalability, etc. UML Diagrams to represent process view include the Activity diagram.
- **Physical view**: The physical view depicts the system from a system engineer's point of view. It is concerned with the topology of software components on the physical layer, as well as the physical connections between these components. This view is also known as the deployment view. UML Diagrams used to represent physical view include the Deployment diagram.

2.1.2. Scenarios:

The description of the architecture is illustrated using a small set of use cases, or scenarios which become a fifth view. The scenarios describe sequences of interactions between objects, and between processes. They are used to identify architectural elements and to illustrate and validate the architecture design. They also serve as a starting point for tests of an architecture prototype. This view is also known as use case view**.**

2.1.3. Brainstorming Sessions:

Brainstorming is used in a wide variety of ways, whether it's one person on their own or many people working together it can always be of some help. It's a cheap and easy way of getting ideas on how to solve problems. It is the most widely used and creative tool around at the moment as it helps to get your mind on the right track by sharing all the information you know either for yourself, or to the rest of the group. It is the newest problem solving concept. It was made popular in 1953 by Alex Osborn who described brainstorming as "a conference technique by which a group attempts to find a solution for a specific problem by amassing all the ideas spontaneously by its members". It is a special technique that makes you think about different answers and questions you hadn't thought about before. There are a few rules for brainstorming techniques, otherwise there might not be a solution to the problem is people get too out of control. Disadvantages of Brainstorming are as follows:

• Can take a long time to work out the kinks if the group is not organized properly.

• Can be hectic, leading to people being afraid to speak their opinion.

• Can go into too much detail that some things may not be used.

• Some people who are more outgoing then others may end up taking over the session with all their ideas. This form of leadership may lead to people being afraid to talk themselves.

• Not recommended for larger groups of people, as there could be too many trying to have their say at the same time.

• Can have repeats of opinions if people aren't paying close enough attention.

• May not end up with usable solutions.

2.1.4. Facilitated Application Specification Technique (FAST):

Its objective is to bridge the expectation gap – difference between what the developers think they are supposed to build and what customers think they are going to get.

A team oriented approach is developed for requirements gathering.

Each attendee is asked to make a list of objects that are-

1. Part of the environment that surrounds the system
2. Produced by the system
3. Used by the system

Each participant prepares his/her list, different lists are then combined, redundant entries are eliminated, team is divided into smaller sub-teams to develop mini-specifications and finally a draft of specifications is written down using all the inputs from the meeting.

2.1.5. Quality Function Deployment (QFD):

Mainly we are defining these are the quality requirements focused on customers satisfaction. The major steps involved in this procedure are –Identify all the stakeholders, e.g. Users, developers, customers etc. list out all requirements from customer. A value indicating degree of importance is assigned to each requirement. In the end the final list of requirements is categorized as It is possible to achieve It should be deferred and the reason for it. It is impossible to achieve and should be dropped off.

2.1.6. Use case Approach:

This technique combines text and pictures to provide a better understanding of the requirements. The use cases describe the 'what', of a system and not 'how'. Hence they only give a functional view of the system.

IV. BACKGROUND AND RELATED STUDIES

3.1. There are different issues related to requirements elicitation techniques that are identified and proven them as ineffective

requirements elicitation techniques: Lack of stockholders involvement, Identification of Stakeholders, Requirements' Volatility, Stakeholders' Confusions, the Problem of Scope. Based on the previous analysis identified the percentage of Issues is shown as:

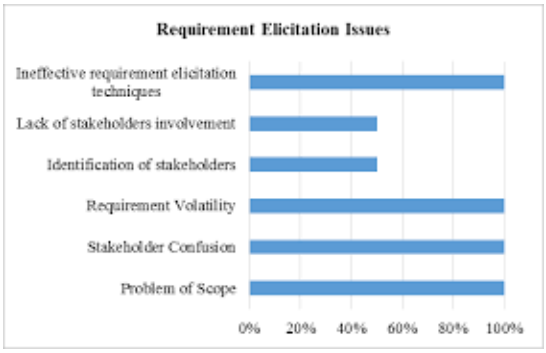
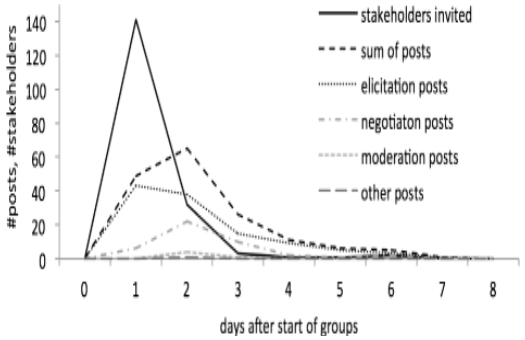


Fig.3 Represents the Requirements Elcitation Issues.

3.1.1. There are a number of issues and challenges encountered during this process. Some of them are as follows:
Understanding large and complex system requirements is difficult:
The word 'large' represents 2 aspects:
(i) Large constraints in terms of security, etc. due to a large number of users.
(ii) Large number of functions to be implemented.
The complex system requirements include those requirements which are unclear and difficult to implement.

- **Undefined system boundaries:** There might be no defined set of implementation requirements. The customer may go on to include several unrelated and unnecessary functions besides the important ones, resulting in an extremely large implementation cost which may exceed the decided budget.
- **Customers/Stakeholders are not clear about their needs:** Sometimes, the customers themselves maybe unsure about the exhaustive list of functionalities they wish to see in the software. This might happen when they have a very basic idea about their needs but haven't planned much about the implementation part.
- **Conflicting requirements are there:** There is a possibility that two different stakeholders of the project express demands which contradict each other's implementation. Also, a single stakeholder might also sometimes express two incompatible requirements.
- **Changing requirements is another issue:** In case of successive interviews or reviews from the customer, there is a possibility that the customer expresses a change in the initial set of specified requirements. While it is easy to accommodate some of the requirements, it is often difficult to deal with such changing requirements.
- **Partitioning the system suitably to reduce complexity**: The projects can sometimes be broken down into small modules or functionalities which are then handled by separate teams. Often, more complex and large projects require more partitioning. It needs to be ensured that the partitions are non-overlapping and independent of each other.
- **Validating and Tracing requirements:** Cross-checking the listed requirements before starting the implementation part is very important. Also, there should be forward as well as backward traceability. For eg, all the entity names should be the same everywhere, i.e., there shouldn't be a case where 'STUDENT' and 'STUDENTS' are used at separate places to refer to the same entity.
- **Identifying critical requirements:** Identifying the set of requirements which have to be implemented at any cost is very important. The requirements should be prioritized so that crucial ones can be implemented first with the highest priority.
- **Resolving the "to be determined" part of the requirements:** The TBD set of requirements include those requirements which are yet to be resolved in the future. The number of such requirements should be kept as low as possible.
- **Proper documentation, proper meeting time and budget constraints:** Ensuring a proper documentation is an inherent challenge, especially in case of changing requirements. The time and budget constraints too need to be handled carefully and systematically.

Consider a case study regarding the number of posts and number of stakeholders involved in the requirements engineering process



3.1.2. Why use this technique?

When a business analyst starts his assignment, he has to get more insight into organization, stakeholders, project and the existing system (if applicable) in a short time frame. He can attain this knowledge by studying the

available documentation. This technique is alternatively called the *background reading*. It is a widely applicable technique, and is not just specific to only Business Analysis, but it can also be used during the requirements elicitation process for effective requirements gathering.

3.1.3. How to use it?

These three steps described below will help you to use the background research technique for stakeholder's identification:

Step 1: Search for documents related to the project for which you work. This could include corporate reports, organization charts, descriptions of existing systems, job descriptions, policy manuals, feasibility study reports.

Step 2: Look for similar activities / projects within the company.

Step 3: Read these papers and put all the identified officials, managers or groups to your list of stakeholders.

Tips: This technique is useful for analysts who are new to the organization, thus external consultants in particular. Background research together with organization charts will help business analyst to get more insight information about the organization.

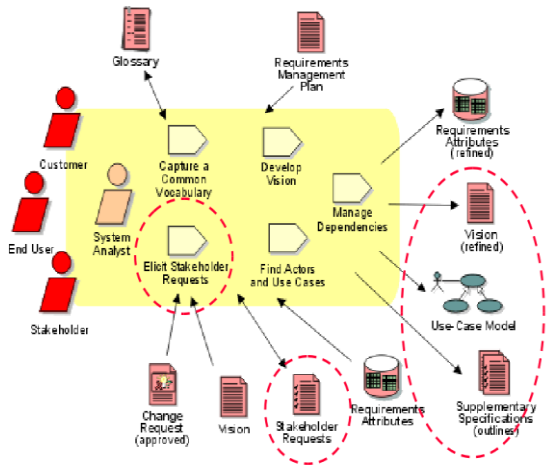


Fig.4 Requirements from various Sources

V. INNOVATIVE TECHNIQUES

In spite of numerous elicitation techniques available in RE, there are still several unresolved issues, which require further attention by the research community. It is expected that some other elicitation techniques may be evolved in future offering solution/s to these issues. Accordingly, an innovative category of elicitation techniques has already been proposed under which all the newly proposed techniques will be put forth. One of such innovative techniques is the Throwaway Paper Prototype. A paper prototype is a visual representation of 'what the System will look like'. Normally, users are expected to draw the features through a pen/pencil on paper and share with Requirement Engineers. Alternatively, a graphics program can also be used for the purpose. Although, in most of the cases, paper prototype is used as a part of the usability testing, where the user gets a feel of the User Interface, by applying the same technique in requirements elicitation, satisfactory results have been obtained. These tryout results indicate that the paper prototype method for requirements elicitation is a suitable method for small and medium sized projects. Other such techniques may also be placed in this category in future.

IV. CONCLUSION

In this paper different issues in using requirements elicitation techniques are discussed. As requirements elicitation is the most important and the initial-most step towards making new software and requirements elicitation techniques are the key to success in developing any new software, selecting the right technique is important. But there is no technique which can accomplish all the demands of requirements elicitation because there are so many issues like confusion of project scope, inadequate stakeholders' involvement, lack of communication and negotiation skills, ineffective techniques, strict time constraints, improper documentation, lack of requirements management and inconsistent requirements. Also there are some issues that are faced when implementing the elicitation techniques which must be resolved so that software can be developed as it is required. The results gathered are also analyzed using various parameters. For the future work, a systematic strategy for selecting the best and appropriate techniques for software requirements gathering may be developed.

REFERENCES

[1]. https://iasaglobal.org/itabok/capabilitydescriptions/views-and-viewpoints/.

[2]. https://www.researchgate.net/publication/269302285_Requirements_elicitation_Issues_and_challenges

[3].]https://pdfs.semanticscholar.org/692b/30d03ab7f1fd797327d26d2392008b8e3a93.pdf

[4]. Pandey, S. K. and Mustafa, K. 2010. Risk Assessment Framework (RAF). International Journal of Advanced Research in Computer Science. Vol. 01. pp. 423-432. International Journal of Computer Applications (0975 – 8887)Volume 75– No.12, August 2013 39

[5]. Pandey, S. K. and Batra, Mona 2013. Formal Methods in Requirements Phase of SDLC. International Journal of Computer Applications. Published by Foundation of Computer Science. New York. USA.pp.7-14

[6]. Pandey, S. K. and Mustafa, K. 2010. Recent Advances in SRE Research. International Journal of Computer Science and Engineering. Vol. 02. pp. 1079-1085.

[7]. http://www.sociology.org.uk/methsi.pdf.Last Retrieved on July 10, 2013
[8]. .https://www.hilcoe.net/docs/papers/Volume1N2/V1N2Paper7.pdf
[9]. Hofmann, H.F. and Lehner F. 2001. Requirements Engineering As a Success Factor In Software Projects.IEEE Software. IEEE. 18(4). pp. 58-66.
[10]. Chen, Ying 2012. Requirement Elicitation Techniques. http://www.umsl.edu/~ycnx6/.
[11]. Joaquín Nicolas., Ambrosio Toval., "On the generation of requirements specifications from software engineering models: A systematic literature review," Information and Software Technology., Elsevier journal., 2009, PP. 1291–1307
[12]. https://pdfs.semanticscholar.org/5c73/bc0016dccf95972b10facb3d6219be28397f.pdf
[13]. Issues in Requirements Elicitation Michael G. Christel Kyo C. Kang September 1992

S. Vinay Kumar" Software Engineering: Requirements Elicitation Techniques, Issues and Challenges in Industry" International Journal of Engineering Research and Applications (IJERA), vol. 9, no.2, 2019, pp 71-76