Importance of Requirement Management : A Requirement Engineering Concern

Dhirendra Pandey & Vandana Pandey

Deptt of IT, BBA University, Lucknow, Dr. C. V. R. University, Bilaspur, India
E-mail : Prof.dhiren@gmail.com, Vandanadubey7@gmail.com

Abstract **- Requirement engineering is first phase of software development processes and it is most important phase for every software development model. In requirement engineering phase we can gather the requirements from user and use this requirement to software development and produce software product that satisfy the user needs. In this research paper we describe the fundamental description of requirement engineering and present the basics dimensions of requirement engineering. Also, in this research paper we also give the basic idea of software requirement specification and present the concept of why requirement management is important for software development.**

Key words - Classification of Requirement, Dimensions, Requirement engineering, Requirement Management.

I. INTRODUCTION

Requirement engineering is relatively new term [1]. In system engineering. Requirements engineering is the science and discipline concerned with analyzing and documenting requirements [2]. It comprises needs analysis, requirements analysis, and requirements specifications [2]. In other words, requirements engineering (RE) means that requirement for a system are defined, managed and tested systematically. Even though requirements engineering has a fairly narrow goal – to determine a need and define the external behavior of a solution [3] it seems to be a challenge for organizations. Introducing requirements engineering is a change of behavior and culture and not just a change of process and technology [4]. It is also important that the cost of fixing a requirements defect later in the development stage is much higher than the cost of identifying and fixing it in the early stages of development. In order to do this the system requirements must be properly identified, analyzed and reviewed early in the development process. Requirements engineering is such a process that focuses on discovering, analyzing, documenting and managing system requirements. Several processes and techniques have been developed to assist requirements engineering activities [5].

A. Definitions of the Terms

The term requirement engineering is used to describe a systematic process of developing requirements through an iterative, co-operative process of analyzing the problem, documenting the resulting observations in a variety of representation formats, and checking the accuracy of the understanding gained [6]. RE is a transformation of business concerns into the information system requirements, "WHAT" the system needs in order to achieve the organizational goals. Requirements engineering process, is the other key tem used to describe the decomposition of RE into interacting non- linear activities. These proceed from informal, fuzzy individual statements of requirements to a formal specification that is understood and agreed by all stakeholders. The final term requirements engineering effectiveness is used as the measure of the accuracy and completeness of achievement of the RE process goals.

Requirement engineering process, is the other key term used to describe the decomposition of RE into interacting non – linear activities. These proceed from informal, fuzzy individual statements of requirements to a formal specification that is understood and agreed by all stakeholders. The final term requirements engineering effectiveness is used as the measure of the accuracy and completeness of achievement of the RE process goals. The effectiveness dimension is captured in such a way that it can be translated into meaningful quantitative statements concerning quality, cost and time schedule. Requirements management is the process of eliciting, documenting and communicating requirements. Requirements management play important role in success of software. It manages changes to requirements and maintains traceability in requirements documents. Requirements can be written using quality attributes known as software requirements specification. Success rate of product depends on process used by organization. Every company needs to assess their present approach in order to remain competent in dynamic market.

II. Dimension Of Requirement Engineering

Requirements are the description of how a system should behave. Requirements are also knowledge of application domain and constraints on operation of a system. Requirements management is the process of managing changes to the requirements. Requirements of a system change to reflect the changing needs of stake holders. They also change due to change in environment, business plans and laws. Requirements engineering is a process of discovering the needs of stake holders and documenting them for analysis, communication and implementation [7]. Many errors can be detected in the requirements phase. Davis [8] claims that fixing of errors detected in later stages of software development is more expensive than the initial stages. If errors are not detected in the requirements phase it leads to wrong product development.

Wrong requirements can also lead to wastage of valuable resources. Collecting requirements is not an easy task. Requirements engineering has critical problems which can be due to lack of stake holder's involvement in the requirement process. Lack of requirements management skills also leads to bad requirements engineering. Unclear responsibilities and communication among stake holders can also lead to bad requirements engineering. Functional requirements or behavioral requirements define functions of the product. Functional requirements include input that the software gets and output it generates. Non-Functional requirements or non-behavioral requirements are the properties of software such as portability, reliability, testability, efficiency and modifiability. Requirements are developed through requirements engineering. Requirements engineering is a process which include a set of activities such as requirements elicitation, requirements analysis and requirements negotiation and validation see figure 1. This process adopted to derive, validate and maintain a system requirements document. Requirements management is the agreement between software development organization and the customer. Both reach on agreement by stating. Communicating, reviewing and negotiating requirements. Ambiguous requirements, addition of requirements, less specification and insufficient user involvement are reasons for bad requirements generation.

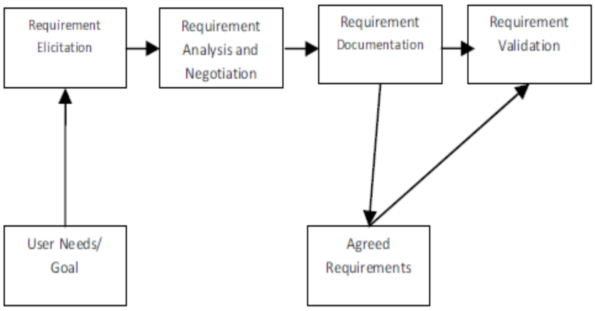

Fig.1: Requirements Engineering Process

A. *Requirements Elicitation*

Requirements elicitation is the process of discovering the requirements of the intended system. Elicitation is to interpret, analyze, model and validate information from stake holders. Researchers [9] mention that requirements elicitation has to deal with application domain knowledge. For example if the intended system is an automation of post distribution then the requirement engineers should have knowledge about the manual distribution of posts. Clear understanding of stake holders needs is an important issue in requirements elicitation. If needs are not clear than the right product will not be developed. A clear vision and scope is essential for the system to be built. Requirements should be documented and communicated among all stake holders. As there are many types of users such as novice users, occasional users, frequent users and expert users their application domain knowledge and experience should also be considered while taking requirements. Some researchers [7] discuss different elicitation techniques. Information about the organization work flow can be gathered by observing the employees. Interviews, questionnaires, surveys, process models are also good source of information. There is no standard technique to select elicitation technique it varies from project to project.

B. *Requirements Analysis*

Requirements analysis is to solve problems and reach an agreement on changes to requirements. As ever stakeholder has his/her own requirements therefore the requirements which cover main functions should be prioritized. Requirements analysis some time is done concurrently with requirements elicitation process. In this phase analysts read requirements. Problems are highlighted and reviewed. Conflicts among stake holders should be solved in this stage. Breaking down the functional requirements into suitable detail level will help the developers to understand and build the system. A checklist must be prepared for assessing each requirement. It helps in solving problems. Analysis model depict information at higher level of abstract than the textual description of the system requirements.

Different diagrams such as Entity Relationship Diagrams (ERD). Data Flow Diagram (STD), Activity Diagrams, and Use- Case Diagrams are used for depicting requirements at various levels of detail. Enterprise modeling is an organizational structure. It represents laws of the organization which affect its operation, goals and tasks. Data models are used to understand, manipulate and manage the data produced by information systems. Entity Relationship diagrams is suitable for modeling huge data. Modeling functional requirements deals with preparing models for the existing and future system. Functional requirements are also used for comparing the new features the intended system would have and facilities it will provides etc. Non – functional requirements are difficult to model the whole system as they cannot be measured individually. Once the requirements are specified they must be communicated among stake holders. This should be done in a manner which facilitates easy reading, analyzing and validating the requirements. There is no standard structure for this process. Making different stake holders to agree on requirements is a tedious task as every stakeholder wants to fulfill their own requirements. Setting priorities is a way to balance desired project scope against constraints of schedule, budget, staff and quality goals. This can be done by categorizing requirements in three levels

High level - these are requirements which are both are urgent and important. Medium level – these requirements are important but not urgent. Low level – these requirements are neither important nor urgent. This prioritization can be done on the basis of value, cost and risk. Ambiguous requirements can be identified by reviewing the requirements documents. Errors such as requirements conflicts and unrealistic requirements can be found out in reviewing the documents. As reviewing every document is tedious and time consuming checking critical document could save time. Written Requirements should be validated against actual requirements.

C. Requirements Documentation

In this section we discuss different requirements engineering notations. Entity Relationship Diagrams, Data Flow Diagrams, State Transition Diagrams, Activity Diagrams, and Use-Case Diagrams are used for depicting requirements at various levels of detail. Software Requirements Specifications and its attributes are also discussed.

Data Flow Diagram- DFD

DFD is used to understand the problem. It shows how the system works. DFD is a method of representing how data is processed by a system in terms of inputs and outputs. DFD represents processes, data flow, external entities and data storage. Some basic DFD symbols and notations are mentioned below. A process transforms incoming data flow into outgoing data flow.

Data stores are the repositories of data in the system. Data flows are pipelines through which information flows. External entities are objects placed outside of the system, with which the system communicates. These are sources and destinations of the system's inputs and outputs. A context diagram is a top level diagram. It is also known as Level 0 DFD. It contains one process node which generalizes the function of the entire system. From context diagram DFD is broken down into required low levels**.**

Entity Relationship Diagram – ERD

Entity Relationship Diagram is also used to understand the problem. It shows how the system works by illustrating the logical structure of data. An entity represents the things about which the information is required. Attributes contains the data relating to the entities. Relationship provides the structure needed to draw information from multiple entities an entity is an object about which you want to store information. A weak entity is dependent on another entity to exist. Attributes are the properties of characteristics of an entity. A key attribute is the unique, distinguishing characteristic of the entity. For example, person number is Sweden is a unique key attribute.

A multi valued attribute can have more than one value. For example, an employee entity can have multiple skill values. A derived attribute is based on another attribute.Relationship illustrates how two entities are related to each other. This is a type of relation which can be self – linked. For example, employees can supervise other employees.

Use Case Diagram

Basic objectives of Use approach are to describe functions of the system that the users need to perform. Use Cases describes an external view of the system. They are services provided by the system to its users. They capture only functional requirements and cannot capture non functional requirements. Some basic Use Case Diagram symbols and notations include. A rectangle represents system which has use cases. These are labeled with verbs that represent the system's function. Actors are the users of a system. A simple line illustrates relationships between an actor and a use case.

Formal Notation

Formal specification of requirements is based on mathematical notations. They are helpful in verifying incompleteness and correctness of the requirements. They also help to remove ambiguity of the requirements. Formal method basically focuses on data and its function. If automated tools developed for formal specification than this method would be more useful. Formal specification language has three components. Syntax specifies specific notation used for representing data. Semantics are used to represent system requirements. Relations are rules used to indicate objects proper function [9].

State Transition Diagram

State diagrams show behavior of a system. They present states of an object [10]. These diagrams are useful in describing the classes to understand the behavior of the object through the entire system. Rounded boxes are used to show objects which represent state of the objects. Arrows are used to show transitions in the next state. A super state of object is used to depict many transitions that lead to a certain state. It eliminates redundancy.

D. *Requirements Validation and Verification*

Once the problem is described, the different parties involved have to agree upon its nature. We have to ascertain that the correct requirements are stated (validation) and that these requirements are stated correctly (verification).

E. *Requirements Management*

Requirements management is the process of managing changes to requirements. Changes are inevitable because of system errors and better understanding development of customers real needs.

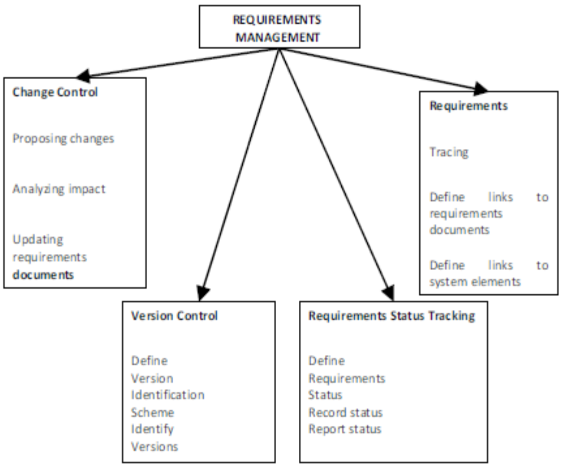
REQUIREMENTS MANAGEMENT
Change Control
Proposing changes
Analyzing impact
Updating requirements documents
Requirements
Tracing
Define links to requirements documents
Define links to system elements
Version Control
Define Version Identification Scheme
Identify Versions
Requirements Status Tracking
Define Requirements Status
Record status
Report status

Fig. 2: Requirements Management Activities

Requirements Management is carried out in parallel with other requirements activities in the software development. Activities of requirements management include keeping project plan undated with requirements, controlling requirements versions, tracking status of requirements and tracing the requirements.

III. Software Requirements Specification

The Requirements document or software Requirements Specification describes external behavior of software system. This document can be written by user or developer. A well written requirements document helps in meeting the desired goals of successful software within time limit. This requirement document can also have adverse effect on the software if not written with care. Errors that are found in the requirements document are easy to fix, cost less and consume less time. If these documents are handled properly errors can be reduced.

Fixing requirements errors during the design, coding or implementation phase will be a tedious task. There are two types of errors that can be found in requirements document. Knowledge errors, which are caused due to not knowing what the requirements are and Specification errors, caused due to lack of knowledge of experience of specifying requirements. A good requirements document has behavioral requirements and non-behavioral requirements. Behavioral requirements specify what the system does including its inputs and outputs. A non-behavioral requirement includes complete description of efficiency, reliability, security, portability and maintainability.

IV. Conclusion

This paper proposes software requirement engineering classification to improve the process of managing requirements. Software Requirements Specification help in documenting requirements in better way as collection of requirements is a crucial phase in software development. Making a perfect SRS is impossible therefore an SRS is considered good which contains as many quality attribute as possible. Dimensions of requirement engineering are analyzed and suggestions are made to select good requirement engineering process for specific area in software development.

References

[1] Ian Summervilley and P. Sawyer, “Requirement engineering- a good practice guide, John villey and sons, new york, USA, 2010.

[2] R. Thayer and M. Thayer, “Software Requirement Engineering Glossary”, 2nd Edition, IEEE Computer Society Press, pp 489-528, 2007.

[3] M. Davis and P. Hsia, “Giving Value to Requirement Engineering” IEEE Software, Vol. 11, No. 2, pp 12-15, 2009.

[4] S. Jocob, “Introducing Measurable Quality Requirements: A Case Study”, Proc. Of 4th IEEE International Symposium on Requirement Engineering, IEEE Computer Society Press, Limerick Ireland, pp. 172-179, 2009.

[5] Kullor, “Requirement engineering for Software Products”, The University of Carlag, Canada, pp. 1-12, 2011.

[6] Pohl, “The Three Dimension of Requirement Engineering, Proc. Of International Conference on advanced information System Engineering”, pp. 275-292.

[7] Bashar Nuseibeh, “Requirement Engineering: A Roadmap”, Japan, 2003.

[8] Ian M Davis, Requirement Engineering- Objects, Function and States, PHI, 2003.

[9] Kotonya and Summervilley, “Requirement Engineering”, John Villey and Sons. 2004.

[10] James Rambaugh, “The UML User Guide”, Addition Wesley, 2009.

[11] Karl, “Software Requirements”, Microsoft Press, 2006.

❖❖❖