EasyChair Preprint

№ 8566

How to Requirements Gathering/Effect on Software Development

Haseeb Munir, Babar Ali and Hammad Saeed

August 3, 2022

How to requirement Engineering/Gathering effect on Software development

Haseeb Munir
(Faculty of computing and IT)
19101001-079@uskt.edu.pk

Babar Ali
(Faculty of computing and IT)
19101001-087@uskt.edu.pk

Hammad Saeed
(Faculty of computing and IT)
19101001-169@uskt.edu.pk

ABSTRACT:

Requirements engineering / Gathering is one of the main parts of software development because it defines the user requirements and specification what user want and also define what has to be developed. In requirements Gathering we analysis the user requirements and verifying the requirements. If you not get requirements properly your software cannot work perfectly and the user cannot satisfy with your products. In this paper show why requirements Engineering / Gathering important for software development and focused on requirements Gathering Techniques and find out which techniques is best for requirements engineering and gathering process. We aimed at investigating which evidence helps to strengthen the process of requirements gathering management, planning including identification and measurement. We identified some causes related to requirements gathering in software development process, existing strategies to helping the identification and measurement, and metrics to support the development process.

INTRODUCTION:

Requirements gathering is one of the most essential parts of any project and adds value to a project on multiple levels. When it comes to smaller budgets, tighter timelines and limited scopes, exact documentation of all the project requirements become crucial. Requirements gathering is easier said than done, it is generally an area that is given far less attention than it needs. Many projects start with basic lists of requirements only to find out down the line that many of the customers' needs may not have been fully understood and implemented. Requirements gathering is an exploratory process that involves researching and documenting the project's exact requirements from start to finish. Effective requirements gathering and requirements management start at the beginning of the project. And after Requirements gathering, we have requirements engineering process that's is exactly use for requirements gathering faces and the Requirements engineering (RE) is one of the most difficult areas within the software development process because it decides and defines what has to be developed. Thus, RE is one of the branches of software engineering that arose from the need to solve the difficult tasks of collecting, analyzing, and verifying the software requirements. According to Christel and Kang Katona and Sommerville, and Birnbach et al. the RE process is frequently described by the following activities: elicitation, analysis, specification, validation and verification, and management. This study is focused in the first stage of RE, requirements elicitation, where almost all the RE's time is occupied by diverse problems when stakeholders interact to obtain quality requirements. Moreover, within the requirements elicitation process, there is a set of necessary activities: the stakeholder identification and the negotiation and selection of techniques to elicit the wishes and needs of the various stakeholders. In this context, there are diverse techniques for discovering and obtaining the software requirements, but it is necessary to have the knowledge to identify which ones are the most suitable for a specific project. Thus, considering that in order to improve software quality, it is necessary to improve the quality of the obtained requirements, it is crucial to improve the selection of the techniques used by the requirements engineer to discover the stakeholders' needs. From this perspective, it is important to consider that requirements elicitation does not just happen by itself. This process is strongly related to the context in which it is carried out, the specific characteristics of

the project, the organization, the environment, the experience, and knowledge of the analyst, as well as the characteristics of the elicitation technique employed. Software engineering, especially during the early phases, is a human-centric activity. Software engineers must gather customer needs, translate those needs into requirements, and validate the correctness, completeness, and feasibility of those requirements. Because of the involvement of various people in this process, there is the potential for human errors to occur. Cognitive Psychology researchers have studied how people make errors when performing different types of tasks. This line of research is called human error research. In this paper, we apply the findings from human error research to analyze and classify the types of human errors people make during the requirements engineering process. The remainder of this paper is organized as follows. Literature Review is discussed in section 2. In section 3 we analysis and discuss the results of different requirements gathering Process on the basis of different parameters.in section 4 We discuss conclusion.

LITERATURE REVIEW:

The main objective of requirements engineering (RE) is to identify the demands of stakeholders, which are people or organizations that will be affected by the system and possess influence, direct or indirect, over the system requirements. Consequently, it is essential to understand the issue and its context, elicit the requirements for the system, analyze, document, and validate them. The research question in this study is based on the following gap: "What are the main techniques, characteristics, and challenges of employing agile methods in the collection and specification of requirements?". We described the employment of the collection and specification of agile requirements adopted by software development companies, focusing on techniques, characteristics, and challenges of the technique practitioners. A survey was carried out, where forty-six (46) participants and practitioners from thirty-one (31) Brazilian companies (public and private) answered about the numerous collection and specification techniques of agile requirements. The results exhibited an overview of the employment of these techniques in private and public organizations.

TD refers to problems caused when software development tasks are pending or inefficiently executed (Kruchtenet al., 2012). While these actions can provide benefits in the short term, such as increased productivity, there are risks to the project, hindering its evolution (Guo et al., 2016; Rio set al., 2018b). With this, TD includes items usually con-trolled in a software project, such as unimplemented features. Also, it covers less visible aspects such as code smells and outdated documentation (Brown et al., 2010). Initially, TD had the focus on coding activities (Cunningham, 1992), but in the advancement of investigations, the concept was expanded to cover the other phases of soft-ware development, for example, in requirements engineering (Li et al., 2015a). According to Ernst (2012), an inappropriate-ate elicitation or analysis of requirements causes errors that increase the incidence of TD in software projects. In this context, the technical debt of requirements may occur intentionally, for example, in the case of consciously choosing not to execute the elicitation process strictly; or unintentionally, in cases where the requirements engineers are inexperience and may not have the skills needed to perform technical and long-term procedures (Rios et al., 2019).

One of the most important issues that software businesses have to address is the dynamic change in process requirements. Therefore, Requirement Change Management (RCM) in software development is essential for success of a software project. RCM is a collaboration driven process and successful RCM requires communication and co-ordination between stakeholders. Unsuccessful RCM can lead to high software cost, delayed schedules, volatile requirements, and end-less testing, ultimately cause project failure and harm the business. Global software development (GSD) is a software development paradigm in which development activities are carried out by experts who are based in different

locations around the world in order to develop successful products for a corporation. To achieve financial benefits, there is growing interest among international software industries in applying GSD. Outsourcing development to supplier firms in low-cost countries has become increasingly important owing to the significantly lower development costs. However, GSD causes several issues for practitioners that do not exist in software projects that are developed at the same physical location (collocated projects). Because development teams are based in several different physical locations, differences in ethnicities and time zones have an adverse impact on communication and coordination thereby resulting in insufficient communication, skills, abilities, and trust among development teams.
Most of the time root cause of failures of software development projects are the poor qualities. As per the commonly used method of the software quality measurement is the empirical method, and few researchers adopted an AHP and Fuzzy technique in determining the software quality.
There are many challenges associated, that it must be clear, correct, consistent, unambiguous, modifiable, verifiable and traceable, with a good quality SRS. Ambiguity is one of the major problems in the requirements specification. Ambiguity rarely surfaces during the development of a requirements model. Software Projects have low achievement ratios yet nowadays because of lack of user interests, delay in delivery, complex requirements or due to unnecessary requirements. Requirement prioritization increased user interest as it allows them to choose their preferred requirements which benefit them. Requirement prioritization helps us to get rid of dissimilarity among the distinct stakeholders. [Karl & Ryan] says that requirement prioritization allows shareholders to allot their assets on their priority requirements. Whereas [Hatton] view Software Requirement prioritization as compulsory now a day for the successful development of software projects.
However, software quality is elusive it is not easy to define or measure. In this context, quality requirements (QRs)play a crucial role in dealing with quality. QRs are the desired qualities of a system to be developed, such as maintainability, reliability, availability, usability, and integrity. Although QRs have some similarities to their functional counterpart—namely, functional requirements—they are unique in other respects, including their meaning, how they are expressed, and how they are measured. While these challenges exist regardless of the approach used to develop software, they are particularly prominent in agile software development (ASD) which entail incremental and iterative software development methods guided by agile manifesto and rapid software development (RSD)1. AS methods (e.g., Scrum, and XP), and RSD approaches (e.g., continuous deployment, continuous delivery, and Devashish extend ASD' capability by shortening the time to delivery of software), are widely adopted throughout the industry because they place focus on continuous delivery of valuable software and customer satisfaction. In such approaches, functional requirements tend to be favored over QRs leading QRs to be improperly documented.

RESULTS AND DISCUSSION:

We analyze and discuss different requirements gathering process by choosing different parameters through which we analyze which Processes is give better performance for requirements management in software development process.

Identification:

the process of visualizing the technical debt, identifying its causes and other attributes present in software development that led to its existence. This activity is crucial for the proper management of TD.

Measurement:

Analyzes and quantifies the costs and efforts required to assist in decision making regarding technical debt reimbursement.

Prioritization:

Organize the payment of technical debts in relation to importance, analyzing factors such as technical issues and financial impacts.

stakeholder:

analyzes a company stakeholder and can either affect or be affected by the business. The primary stakeholders in a typical corporation are its investors, employees, customers, and suppliers.
Low client availability:

Client Availability:

shows the days and times when a client is available to be scheduled for an appointment low client availability effect the software development process.

Sr no	Author	parameters	Percentage% and responds
1	Roberta Fagundes et.al	Identification	86%
2	Woubshet Behutiye et.al	Measurement	71%
3	Muhammad Azeem Akbar et.al	Prioritization	60%
4	Diego Lisboa et.al	Interview with stakeholders	56.6%
5	Juan Carlos Barata et.al	Low client availability:	39.5%

CONCLUSION:

In this review paper introduced how to different types of requirements effect on software development. The requirements make it easier to develop the question list as well as to identify gaps in knowledge. The objective is to ensure that the product to be developed is fully understood from all angles. In software engineering terms, the focus includes both functional and nonfunctional requirement. he results of this systematic review show different contexts that can lead to the emergence of the of requirements, involving causes for intentional or unintentional requirements, a collaboration of clients and stakeholders, elicitation and documentation of requirements and pressure of schedule, for example. Along with this evidence, other strategies that help identify and measure the requirements of were mapped, emphasizing manual management, with different application proposals. However, the results highlight the lack of automated resources focused on this type of specific requirements.

As future proposals, research will be invested based on the gaps identified in this work, adjusting with the evidence already placed and new empirical studies. Also, the development of a guide to support the requirements' identification and measurement is currently at early stage. The guide will consist of evidence from the literature and information gathered through survey in the software industry. Its objective is to help professionals identify and measure the existing requirements in their projects, knowing instructions and metrics to measure the data necessary for its resolution.

REFRENCES:

1. https://www.sciencedirect.com/science/article/abs/pii/S1045926X18301411
2. https://www.researchgate.net/publication/272863222_A_Review_of_Requirement_Engineering_Issues_and_Challenges_in_Various_Software_Development_Methods
3. https://ieeexplore.ieee.org/document/9325916
4. https://www.academia.edu/43750611/A_framework_for_software_requirement_ambiguity_avoidance
5. https://www.academia.edu/43750611/A_framework_for_software_requirement_ambiguity_avoidance
6. https://www.researchgate.net/publication/352016931_Identification_and_Measurement_of_Technical_Debt_Requirements_in_Software_Development_a_Systematic_Literature_Review
7. https://onlinelibrary.wiley.com/doi/abs/10.1002/sys.21572