

The Significance of Requirement Engineering in Software Development

¹Rani Singh, ²Manish Kumar

Assistant Professor
Information Technology
Ganpat University, Mehsana, Gujarat

Abstract- At present, the process of software development comprises a series of steps, commencing with the feasibility study phase, followed by requirement gathering and analysis, design, coding, testing, and maintenance phases. Each of these phases is integral to the success of the software under development. Among these stages, requirement gathering and analysis, also known as requirement engineering, plays a pivotal role. This phase is critical because it ensures that the software aligns precisely with the customer's needs and requirements. It is during the requirement engineering process that customer needs are collected and thoroughly analysed. This phase holds great significance as the end user of the software is the customer. Therefore, it is imperative that the software caters to all of the customer's needs. This paper provides an overview of various requirement engineering approaches adopted by software development companies and organizations for gathering, comprehending, and addressing all sort of anomalies or inconsistencies in the collected requirements.

Index Terms- Requirement Engineering, Ontology, Requirements collection, Requirement change management.

I. INTRODUCTION

Different Software Development Organization follows many different ways for Software development. Initially when the software were developed at early times without any strategies software always leads to software crisis, It was believed that Software Development Life Cycle (SDLC) model can be the solution for the software Crisis. Software development life cycle models follows series of activities initially from feasibility study i.e. studying the economical, technical, operational and legal feasibilities of the software that will be developed. After the phase exit criteria of feasibility study, the software development goes to requirement gathering phase where the software requirement is gathered from the ultimate user and then the gathered requirement is analysed and on the basis of it SRS documentation is done that has function requirement, non-functional requirement and goals of implementation, then design phase where design document is maintained which includes low-level design and high level design, it also includes function oriented design and object oriented design, the next phase is the coding phase where the programmer adhere some common guide lines for writing the codes, the coding guidelines adhered is in such a way that if any teams members change, the code should be clearly understandable. After coding is done and all syntax error is removed the code goes into code review technique which includes code walk through and code inspection. The next phase is testing phase which includes unit testing (testing individual models using drivers and stubs) then integration testing (integrating modules one by one and testing it individually) then is the system testing (testing software overall), the entire testing scenario includes two types of testing, black box testing where the internal structure of the module is not known testing is done to check in outer structure using equivalence class partitioning, boundary value analysis and clean room testing. The next type is the white box testing technique where the internal structure of the module is tested based on statement coverage, path coverage, branch coverage, condition coverage and many more. Finally when all validation and verification testing is done, then comes the maintenance phase which includes all the changes which the software company or organisation has to adhere if the organisation wants that software to be used for longer period of time, or software being developed doesn't become the legacy product. So maintenance phase includes corrective, perfective and adaptive maintenance. Fig 1. Diagrammatically explain all the SDLC phases.

This paper mainly Focuses on Requirement gathering and analysis phase of the software development.

Software Requirement engineering played a major role in both academic and industry for both global and local market(domain), but the process of requirement engineering is not that simple and straight forward, it requires proper planning and strategies as for now the software is with user centric functionality[1]. Requirement elicitation is the first and important step to understand the user's requirements by different approaches and techniques. This step starts from requirement elicitation then requirement analysis, requirement documentation, requirement validation and requirement management [4].

The main focus of the researchers on requirement engineering is to study about the challenges which generally the software development team faces on requirement elicitation and they are as follows:

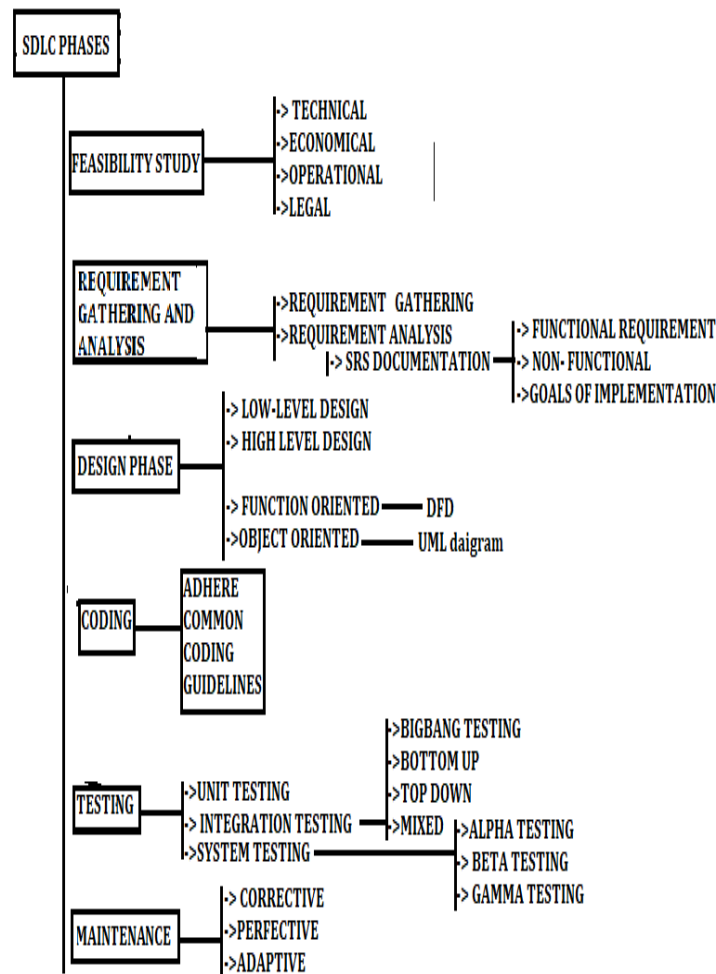


Fig1: Phases of SDLC

- The demand of rich communication i.e. difference of geographical location like socio- culture difference [4].
- The development of semantic technologies which leads to the concepts of ontology [2].
- The dynamic behaviour of the software for managing the change in requirement by the customer to support iterative development of software [3].
- Information gathered during requirement elicitation generally lacks in quantification, scalability, prioritization and requirement interdependency [6].

II. SURVEY

Semantic technology in the area of requirement engineering has provided immense opportunity for applying set of concepts and categories (ontology) for requirement classification. Currently the implementation of requirement classification techniques has not manoeuvre the semantics of requirements [3]. If the requirement gathered from the user or customer has classification or different categories, it would result in better quality and satisfied software product. This paper proposed ontology using semantic technique which is based on Methontology and its implementation is done using protégé. The data collected in requirement engineering is tested using validation and verification testing approach.

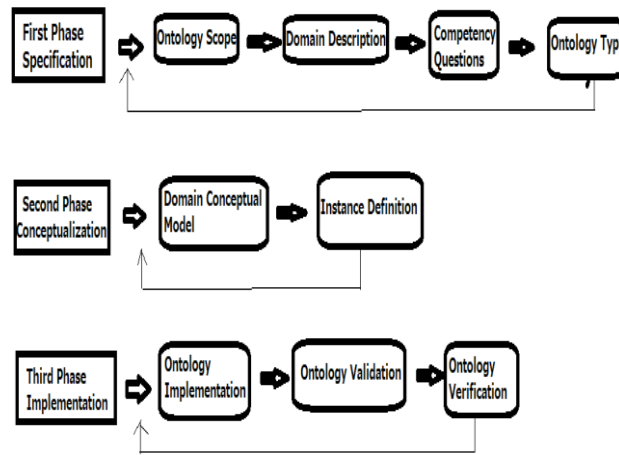


Fig 2: Development process of Ontology[3]

RCO (Requirement classification ontology) is used to classify the semantic in the requirement gathered, for the same RCO uses Methontology.

This paper has clearly explained the different ontology development process using RCO. Ontology development process includes 3 phases, the first phase is: Specification phase, the second phase is Conceptualization and the third phase is Implementation [3].

The First phase includes Ontology Scope which describes the boundaries of ontology for finding what should be considered. After Ontology scope is the Domain Description, here the most required knowledge to build RCO is investigated and explained. It includes data and procedural requirements, performance requirements, usability requirements, security requirements and cost variability, deadline and maintainability requirements. After Domain description, the next step is Competency Questions which is the most important step, especially in validation phase of the build ontology. In Competency Questions, few questions are generated related to the understanding of the requirement. The last step of the 1st phase is Ontology type, where different categories of Ontology are decided like upper ontology, mid-level ontology, domain level or application level ontology.

The 2nd phase is Conceptualization, Which includes domain conceptual model. Here data dictionary is formulated that describes some concepts of the requirement domain; UML is used for better understanding of the core concepts of the domain. Based on the understanding, different RCO instances are created like which requirement will be of which instance.

The 3rd phase is implementation phase where the researchers have transformed the RCO in reality based on some development tool like protégé.

The content of RCO evaluation is done on the basis of Ontology Validation and Ontology Verification based on Taxonomy evaluation, Answers of Competency Questions in respect to function and non-functional requirement. Even FOCA and GQM (goal/ question/ metric) verification process is considered. Fig 3 describes the FOAC approaches used in RCO evaluation.

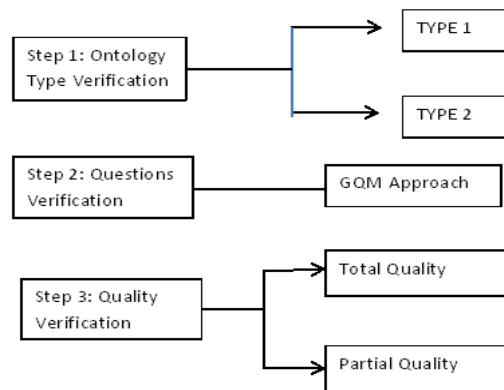


Fig 3: FOCA approach [3]

In Quality Verification partial and total Quality verification method is considered.

In conclusion, this study has provided a comprehensive overview of the development and evaluation of an ontology, referred to as the RCO, designed for the purpose of classifying requirements within the context of Requirements Engineering (RE). The motivation for this research stemmed from the scarcity of existing studies focusing on ontologies capable of classifying requirements within the realm of RE.

After requirement gathering, requirement change is generally very common because of missing necessities, boosting prerequisites, omitting repetitions, these reasons are common but need to be taken into consideration. Requirement changes must be highlighted during operational mode of software development or when the requirement framework is installed [1, 4].

Requirement change management in the single place or site is not so difficult but in distributed environment managing changes for good software development requires pre-requisite knowledge and understanding [4]. Project management teams usually need to have plans, schedules, and data collection strategies from all internal and external sources [4, 10] on prior basis before the requirement elicitation starts.

In this paper researcher has considered the recent advancement of global software development (GSD)[4] whose market is growing at the faster rate. Global software development addresses the changes in the cultural and geographical differences. Requirement change management and requirement engineering both are considered as a very difficult task for GSD. To resolve the problem caused due to GSD the strategies used in this paper is blind interviews and survey of the researchers for the understanding of the proposed framework and statistical tools are also applied to analyse the collected data.

For any organizations which are global based, collection of customer requirement from different socio culture thinking is a very difficult task. Here RE demands quantity and pure requirements analysis and collection. For any globally distributed framework data collection and analysis is difficult to manage because the data and requirement changes continuously or rigorously.

Project management plays a major role for this ever changing situation. Previously many project management tools have been used but those tools used have limited scope. For requirement collection some key points are their which are considered by the manager, those key points are connected with the RCM and RE methods [1, 4]. In this paper RE framework proposed which broadly categorized in three phases, 1st phase is communication and documentation phases where communication, analysis and negotiation and documentation is done. The 2nd phase is the analysis and management phase which includes risk management and analysis, requirement management, validation and verification and tools and techniques. The 3rd phase includes Finalization of SRS which includes tools confirmation, sites declaration and adoption procedures.

The RCM framework in GSD is categorised in 3 phases communication phase, analysis phase and development and documentation phase, all these phases includes time boxing [4].

The approvals of both RCM and RE framework is done using interview and survey, both the strategies includes questionnaires. For the survey associated with RE and RCM exercise in GSD the target persons are key leaders, analyst, program director, project manager and requirement engineers. Snowball data collection technique is used for the survey which is basically a web based survey. Fig 4 deals with the different RE framework [9].

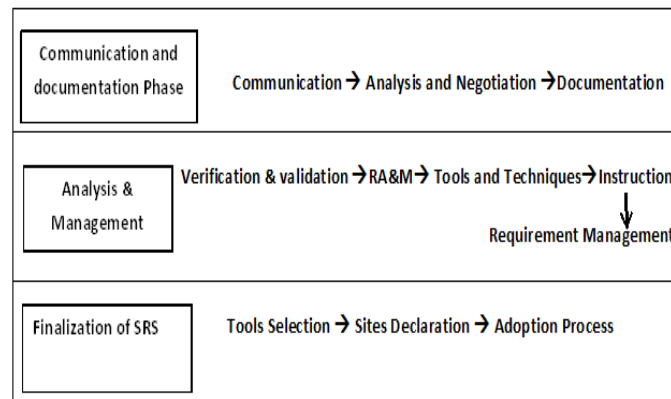


Fig 4: RE framework in GSD

Manual survey was also adopted to find the actual view after the implementation of the framework proposed. After survey statically analysis is done, to find the results of both RE and RCM framework, comparisons are done by manual survey, web based survey and interviews. After this actual role of project manager comes where all the complexities, internal unknown facts are identified and resolved. So this paper, concluded with the fact that for requirement elicitation, Requirement understanding, and requirement analysis communication plays an important role. For any type’s communication, survey and interviews were conducted and the data are analysed using statistical tools [5, 6].

As already discussed, that requirement gathering and analysis plays a major role in customer satisfaction about the software product development. Above papers which were reviewed has already mentioned many ways from where the actual requirements of the customer should be gathered and understood, but the problem faced due to requirement prioritization [6] was not covered on those papers. Requirement gathered from local domain or global domain, either from same culture people or different culture people, surveys or interviews, has too much of data, for the project manager doing analysis of those collected data i.e. from where to start or which module should be studied or understood first is the challenging task, so here comes the requirement of requirement prioritization (RP).

. This paper was a survey paper where they surveyed many paper, 122 relevant studies were selected to address the defined research questions. 108 RP techniques were identified and analyzed with respect to their benefits, prioritization criteria, size of requirements, types in terms of automation level, and their limitations; 84 prioritization criteria were disclosed with their frequency usages in prioritizing the requirements[6] and according to these papers all the present prioritization technique lacks quantization, scalability, are time consuming, have too much requirement dependencies and requires highly professional interventions. Identifying RP domain is significant is software development and even for the stake holders involved in the RP process.

Any software being developed follows SDLC phases, where among all phases requirement engineering leads to the challenging one. There are many strategies being developed for understanding the customer requirements in proper way for software development process. But based on the recent trends, the challenge is to follow SDLC phases in the case of mobile application development as the technology, software, requirements changes very rapidly [2]. So the strategies followed in traditional software development cannot be followed in mobile development application. So this paper survey the most common issues that arise in mobile application development and in its requirement phase like development cycle time device limitations, cost factor, resource effectiveness, quality assessment and domain understanding. The researchers did the systematic literature review (SLR) for understanding the requirement elicitation and its challenges. In this paper a comprehensive set of 22 elicitation techniques were measured and among that eight different categories are made based on their applicability [2][11].

Requirement gathering procedures are divided into two main categories in case of mobile application development: User requirement and System Requirement. Requirement gathering includes Interviews, Questionnaire, Social analysis, prototyping, requirement reuse [7], brainstorming, methods of scenarios. The challenges discussed in literature are stakeholder related, requirement related, communication related, human factor and organization related. Research methodology here covers all the aspects of SLR, majorly divided in 5 steps: Question formulation for understanding the requirement, data selection, past study selection, evaluation and analysis and results [2].

Data collected is based on inclusion and exclusion criteria. Several procedures of quality Assurance are also followed. Fig 5 elaborates different data search techniques [8].

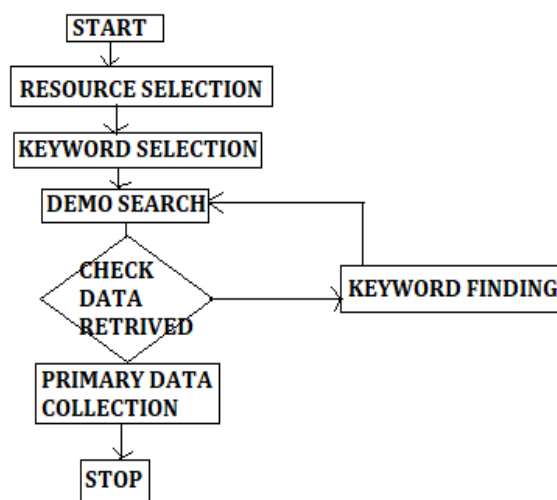


Fig 5: Data Search technique

III. CONCLUSIONS

Software development, whether it's for a small or large organization or for dynamic software like mobile applications, adheres to all the stages of the Software Development Life Cycle (SDLC) from assessing feasibility to the maintenance phase. This approach helps mitigate the problems associated with legacy software and software crises that were prevalent before the introduction of traditional SDLC models. However, for any organization, customer satisfaction takes precedence. Customer satisfaction is closely linked to the second phase of SDLC, where requirements are gathered, understood, and analyzed. The project manager collects customer requirements to comprehend the user's software prototype, a process known as requirement engineering. This survey paper examines various requirement elicitation techniques in recent practices.

According to the survey, recent trends favor questionnaires and interviews as effective methods, but for global software development, the snowball technique, a web-based survey, is beneficial for a deeper understanding of customer needs. The collected data is analyzed by the requirement engineering team using statistical tools or algorithmic studies. Particular attention should be directed towards mobile application software since it necessitates frequent updates and relies on automation techniques.

REFERENCES:

1. A. A. Alsanad, A. Chikh and A. Mirza, "A Domain Ontology for Software Requirements Change Management in Global Software Development Environment," in *IEEE Access*, vol. 7, pp. 49352-49361, doi: 10.1109/ACCESS.2019.2909839, 2019.
2. H. Dar, M. I. Lali, H. Ashraf, M. Ramzan, T. Amjad and B. Shahzad, "A Systematic Study on Software Requirements Elicitation Techniques and its Challenges in Mobile Application Development," in *IEEE Access*, vol. 6, pp. 63859-63867, 2018, doi: 10.1109/ACCESS.2018.2874981, 2018.
3. H. Alrumaih, A. Mirza and H. Alsalamah, "Domain Ontology for Requirements Classification in Requirements Engineering Context," in *IEEE Access*, vol. 8, pp. 89899-89908, 2020, doi: 10.1109/ACCESS.2020.2993838, 2020.

4. M. Shafiq et al., "Effect of Project Management in Requirements Engineering and Requirements Change Management Processes for Global Software Development," in *IEEE Access*, vol. 6, pp. 25747-25763, 2018, doi: 10.1109/ACCESS.2018.2834473,2018.
5. R. Mohanani, P. Ralph, B. Turhan and V. Mandic, "How Templated Requirements Specifications Inhibit Creativity in Software Engineering," in *IEEE Transactions on Software Engineering*, doi: 10.1109/TSE.2021.3112503,2021.
6. F. Hujainah, R. B. A. Bakar, M. A. Abdulgaber and K. Z. Zamli, "Software Requirements Prioritisation: A Systematic Literature Review on Significance, Stakeholders, Techniques and Challenges," in *IEEE Access*, vol. 6, pp. 71497-71523, 2018, doi: 10.1109/ACCESS.2018.2881755,2018.
7. M. U. Malik, N. M. Chaudhry, and K. S. Malik, "Evaluation of efficient requirement engineering techniques in agile software development," *Int. J. Comput. Appl.*, vol. 83, no. 3, pp. 24–29, Dec. 2013.
8. B. Christian et al., "Requirements for smart home applications and realization with WS4D-PipesBox," German Federal Ministry Educ. Res. BMBF, Univ. Rostock, Rostock, Germany, Tech. Rep. 01S080031, 2011
9. Rapid Value Solutions. (2016). *Mobile Usage Statistics and Trends 2016*. Accessed: Jul. 2017. [Online]. Available: www.rapidvaluesolutions.com "PDCA12-70 data sheet," Opto Speed SA, Mezzovico, Switzerland,2017.
10. J. Prasara and H. Tew, "Requirements elicitation to develop mobile application for elderly," in *Proc. Int. Conf. Digit. Arts, Media Technol. (ICDAMT)*, Mar. 2017, pp. 464–467,2017
11. J. Horkoff, "Keynote - Requirements Engineering for Machine Learning: Non-functional Requirements as Core Functions," 2022 IEEE 30th International Requirements Engineering Conference Workshops (REW), Melbourne, Australia, pp. 141-141, doi: 10.1109/REW56159.2022.00034,2022.