Sustainability-centric integration software requirements validation

Prabhakar Kandukuri[1*], *Dasari. N. V.* Syma Kumar[2], *G.* Ramesh[3] and *Prathima* Kativarapu[4]

[1]Department of Artificial Intelligence and Machine Learning, Chaitanya Bharathi Institute of Technology, Hyderabad, India
[2]Department of CSE, Bapatla Engineering College, Bapatla, AP, India.
[3]Department of CSE, Gokaraju Rangaraju Institute of Engineering and Technology Hyderabad, India.
[4]Department of Computer Science and Engineering, SRK Institute of Technology, Enikepadu, Vijayawada, India.

Abstract. Software Development Life Cycle (SDLC) is enriched with different phases. Requirement Analysis is predominant in designing and developing the software application with proper functionality and to produce expected outcome. The client expects the great deal of functionality and ease of use application and outstanding performance from the application developed. If the application is appropriately demonstrating the outcome what the client expected the requirements have to be gathered once again and re do the application until the client satisfaction. Understanding the client specific requirements from the users' point of view and smooth running of the operations with proper integration between the modules of the application. The requirements from the client would be vague and taken in the user's perception. The requirements taken from the client's user requirements should be integrated with realistic expectations and plan according to the project schedule for each stage of SDLC. The requirement gathering technique should replicate the software requirement engineering with proper integration oriented methods. So that the project development will be completed without facing the challenges and issues arise in the different stages of SDLC. This research is proposing the integration oriented scrutiny on the software requirement engineering mechanism in the requirement analysis phase of SDLC.

1 Introduction

A good software requirement analysis will give well-functioning software application. Requirement analysis is regarded as the most crucial phase to develop any software application [1]. The usage of automatic requirements tracers will also give a good results in requirements validation but finding a good tool become a bottleneck [2]. The client specific requirement should be understood properly. The classification of client specific requirements can be business requirements, functional requirements and user requirements.

[*] Corresponding author: prabhakarcs@gmail.com

The functional requirements can replication the functionality of the project what it is meant for. The business requirements will replication the business operations what the client is performing [3]. The user requirements will illustrate the user of the business operations with the customers of the client. The requirements gather should be done meticulously to avoid the ambiguity and confusion. While taking the user requirements, the requirement engineering should be applied. The specific requirements should be gather properly. The users are not aware of the other user requirements and business operations what they are performing. As a software engineering while taking all the requirements for different users of the client should be obviously understood. Based on the client's specific requirements the requirement engineering process should be applied and remove the missing links between the user's specific functionalities. This can be possible by implementing the integrated oriented scrutiny in software requirement engineering. The software requirement engineering specified by IEEE should gather the system objective, the expected software behavior, properties exhibited by the system and any constrains while working the software application in the working condition [4][5][6]. The present paper is introducing the requirement engineering process should be enriched and included with the integration oriented scrutiny process to avoid the ambiguity and confusion in designing and developing the proposed software application [7]. The proposed model should be redesigned with the new component. The new component is integration oriented requirement analysis for the modules of the application as a mandatory requirement analysis in requirement engineering [8].

1.1 The Existing Requirement Engineering Process

According to the Institute of Electrical and Electronics Engineering (IEEE) the requirement engineering is regarded as in Fig.1.

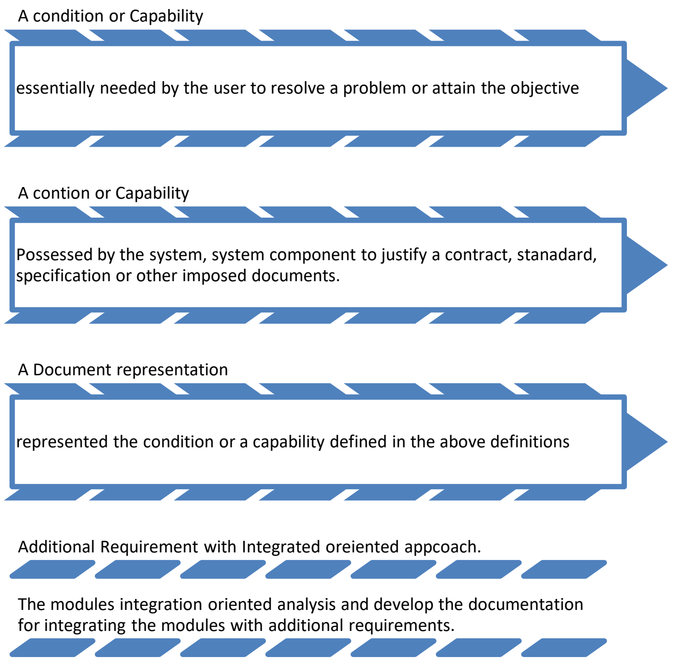
A condition or Capability
essentially needed by the user to resolve a problem or attain the objective
A contion or Capability
Possessed by the system, system component to justify a contract, stanadard, specification or other imposed documents.
A Document representation
represented the condition or a capability defined in the above definitions
Additional Requirement with Integrated oreiented appcoach.
The modules integration oriented analysis and develop the documentation for integrating the modules with additional requirements.

Fig.1. IEEE standard requirement engineering components

In the above figure the first three requirement engineering components are defined by IEEE. The final requirement is proposed in this paper. The requirement engineering should also include with the integrated oriented approach requirements in the requirement analysis of the SDLC and traceability of requirements [9].

1.2 The Challenges in Requirement Engineering

1 The user specifications should be taken from all users of the clients place.
2 The users are not aware of the other users operations and specifications. So they can't declare the requirements obviously.
3 The gaps of requirement analysis between the users will create the ambiguity and confusion what data transparency is needed in the module definition and functionality.
4 The requirement engineering is not specifying any design of the modules. This will create the gaps in integration points and exact requirement specifications missing.
5 A well requirement scrutiny is essentially needed to find the additional requirement specifications for data transparency and data availability.
6 The data transparency requirement are mentioned in the requirement engineering. This will create the integration of modules with specific data availability requirements.

1.3 Ambiguity Vs Requirement Engineering

1. Requirement engineering aims at functional requirements, business requirements and user requirements gathered from the users of the client place
2. Requirement verification aims at collection of information from user's perspective, functional perspective and business operational perspective.
3. Requirement Management is emphasizes to track the changes in requirements and ensures the changes should meet the requirements of the stakeholders of the software application which is going to be developed.
4. The ambiguity can be raised from the module integration and the user requirements integration and functionality.
5. The collection of the requirements can be taken from individual users who are not aware of fully with the other user requirements. This will create the ambiguity of data availability needed for specific user of the application.

2 The Proposed View of The Requirement Engineering Process

The system requirement engineering process is defined with the requirement collection, analysis, specification, verification and management. The proposed requirement engineering process should be embedded with the integration oriented requirements in addition to the above specification. This will avoid the ambiguity and confusion among the modules functionality and operations [10]. The transparency of the data from one module to other module is need to be demonstrated in specific modules. The limitations of the data transparency according to the user rights and users functionality is need to be designed and well documented in the requirement engineering. The following diagram can illustrate the requirement engineering process with the new specification.

The new specification related to the integration of the modules should be added in the requirement engineering process. The data transparency and data availability within the modules should be analyzed and well written in the additional requirement to avoid the ambiguity and confusion while the application is running. The integration oriented scrutiny is essentially required to define the data availability within the user modules to be used to enable the specific user to perform the limitations and restrictions to meet the overall functionality without creating the confusion and ambiguity. The integration oriented scrutiny can be well described in the scenario.

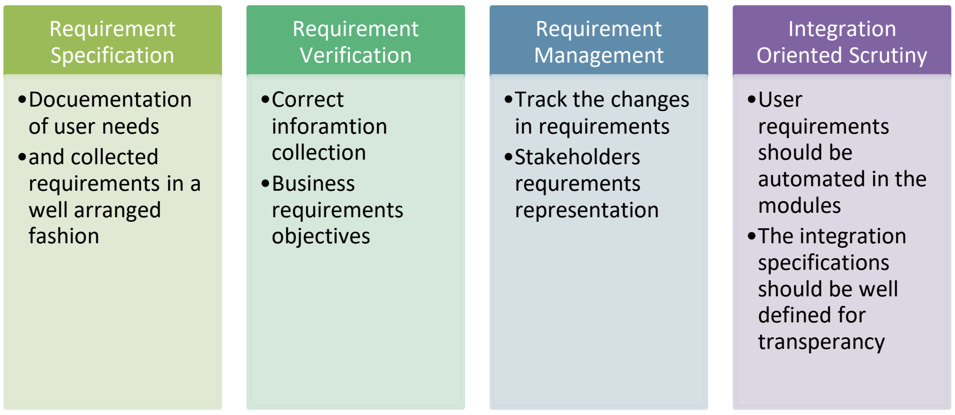
Requirement Specification
•Docuementation of user needs
•and collected requirements in a well arranged fashion
Requirement Verification
•Correct inforamtion collection
•Business requirements objectives
Requirement Management
•Track the changes in requirements
•Stakeholders requrements representation
Integration Oriented Scrutiny
•User requirements should be automated in the modules
•The integration specifications should be well defined for transperancy

Fig. 2. Inclusion of integration oriented security

For example the software development is done for a bank the banking application has users like clerk, cashier, officer and manager. Each user will have a module and configured with the specific functionality and business operations. The clerk module is creating the new accounts of the new customers and issue the withdrawal slip for the customers from their accounts. The customers' accounts are flooded with the heavy amount. The operation of withdrawal should be within the limits of the cash availability of the cashier of the bank. While issuing the withdrawal slip and accepting them to withdraw certain amount from their accounts the limit should be indicated by the system from the other module. That means the amount available in the cashier should warn the clerk to create the limit of withdrawal for the customer. This integration points are need to be analyzed and specified in the requirement analysis. This type of integration requirement need to be analyzed from the user requirements, business requirements and functional requirements collected from the users. The essence of data transparency can be estimated from the scrutiny of the business requirements and functional requirements of the requirements engineering only.

An independent survey conducted by the software engineers association in India has revealed the following facts and figures related to the requirement engineering and analysis.

Table 1. Failure Percentage of Requirement Engineering Phases

S.No.	**Issues of Requirement Engineering**	**Percentage of Failures**
1	Project Requirement Management	35
1	Problems faced by the end users	18
2	Lack of requirement verification	17
3	Rollback out of procedures	10
4	lack of integration between the modules	20

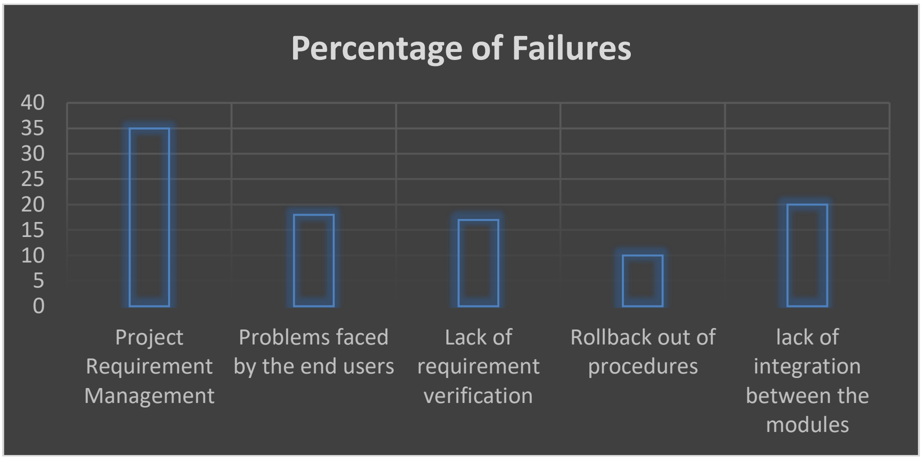

Fig. 3. Requirement Analysis Failure Percentage

The results acquired from the independent survey conducted by the software engineers association has revealed that the importance of the requirement generated from the integration and scrutiny of the requirement engineering process. The lack of integration has given well impact on the software application and forced them to do the rework in the application functionality and operations. The issues raised from the lack of integration between the modules and data transparency and data availability with in the modules has drawn high attention in the software development life cycle. The lack of integrity has generated the serious problems for the operations in the live environment. Hence the application development process should be enriched with the software requirement engineering associated with the integration oriented requirement scrutiny generated requirement for data availability and data transparency processes [11].

3 Achievements of Integration Oriented Scrutiny on The Requirement Engineering

The complete success of the software project without re-work can be achieved by including the integration oriented scrutiny on the requirement engineering to achieve the following

1. Correct requirement gathering
2. Unambiguous application running
3. Complete and fully functional application
4. Consistent and ease of use application
5. Ranked for stability in all operations and functionalities
6. Scrutiny of the requirements engineering
7. Modifiable application for additional users interface
8. The realistic expectations in modules functionality
9. The integrated data availability and obviousness
10. Requirement gathering related to the module integration point of view
11. Project planning with all the stages of software development life cycle stages

The scrutiny of the requirement engineering analysis will generate the additional requirement related to the data availability and data transparency. This additional requirements documentation should be considered in designing the project. This will stop the ambiguity and confusion in the modules functionality and reduce the rework of the project development [10].

4 Conclusion

Different phases of Software Development Requirement Life cycle is basically depends on the requirement analysis phase. This phase is well equipped with the requirement engineering process. The requirement engineering process should be enriched with the integration oriented scrutiny to find the essence of the data availability and integrity. The data availability and transparency points will not be enacted by the user's requirements gathering. The users are not so obvious to illustrate the business requirements and functional requirements and user requirements of other users. This will create ambiguity and great confusion. Hence the integration oriented scrutiny should be done and the requirement analysis should develop the additional requirements related to the data availability and transparency.

References

1. Merugu, Gopichand. "Five layered model for identification of software performance requirements." International Journal of Software Engineering & Applications (IJSEA) 3.5 (2012).
2. Tizard, James, et al. "A Software Requirements Ecosystem: Linking Forum, Issue Tracker, and FAQs for Requirements Management." IEEE Transactions on Software Engineering (2022).
3. Madapudi, Rudra Kumar, A. Ananda Rao, and Gopichand Merugu. "Change requests artifacts to assess impact on structural design of SDLC phases." Int'l J. Computer Applications 54.18 (2012): 21-26.
4. Goldman, James L., George Abraham, and IlYeol Song. "Generating Software Requirements Specification (IEEE Std. 830 1998) document with Use Cases." no 215 (2007): 1-12.
5. John Doe (2011) Recommended Practice for Software Requirements Specifications (IEEE) Author: John Doe Published in Copyright © 2011 Midori. All rights reserved IEEE
6. IEEE Guide for Developing System Requirements Specifications sponsored Software Engineering Standards Committee of the IEEE Computer Society published in 2011.
7. I. Atoum et al., IEEE Access, vol. 9, pp. 137613-137634, 2021, doi: 10.1109/ACCESS.2021.3117989.
8. K. El Emam and A. Birk, IEEE Transactions on Software Engineering, vol. 26, no. 6, pp. 541-566, June 2000, doi: 10.1109/32.852742.
9. P. Rempel and P. Mäder, IEEE Transactions on Software Engineering, vol. 43, no. 8, pp. 777-797, 1 Aug. 2017, doi: 10.1109/TSE.2016.2622264.
10. M. S. Farooq, M. Ahmed and M. Emran, " IEEE Access, vol. 10, pp. 48193-48228, 2022,
11. C. Tao, J. Gao and T. Wang, in IEEE Access, vol. 7, pp. 120164-120175, 2019,