Improving Requirement Prioritization process in Product line using Artificial Intelligence technique

Wasi Haider[1] Yaser Hafeez[2] Sadia Ali[3]

M Azeem Abbas[4] M Numan Rafi[5] Abdul Salam[6]

Abstract

Product families emerged a new and useful development technique in the field of software development. In Software Product Line (SPL) there are some core assets and some variants so using these assets anyone can build their desired product in very short time and effort. While working in product family's environment we must keep an eye on the requirement prioritization and ranking because that requirement are very important because these requirement lay the foundation of the core and variants assets which are the building blocks of SPL. So there are some major issues which we face are the more human interaction, ambiguous requirements and wrong or no requirement ranking. In this paper we proposed a framework for the ranking of stakeholders' requirements for the SPL's variant and core assets using the case base reasoning CBR if available in previous use or assign them new ranking according to their requirement and their assign ranking for software product line. We evaluated our framework by empirical study. The results prove that the considerable improvement for different parameters is achieved by our framework as compared to conventional approaches of requirement prioritization.

Keyword: Software Product Line (SPL); Requirement Prioritization (RP); Case Base Reasoning (CBR); Artificial intelligence (AI)

1 Introduction

Software product family is a interrelated software systems, sharing a common and managed collection of features to accomplish the wants of a suitable market section [1]. The main goal of SPL is reuse in an effort to enhance the quality and production while reducing cost as well as time to market. SPL engineering has become an efficient and minimizes time-to-develop approach for providing a common model for developing product families. The central concept at the back of SPL is to provide a stage with common and distinct components of a software system identified in order to build a consistent line of products [2]. Software product variants are often develop from an early product development. These product variants are generally share some common but they are also different from each other due to upcoming change request to fulfill the specific demand and requirement of the end user [3]. As a number of features and the number of products increase, it is significance re-engineering product variants into a SPL

[1] PMAS-Arid Agriculture University, Rawalpindi | wasihaider734@gmail.com
[2]PMAS-Arid Agriculture University, Rawalpindi | yasir@uaar.edu.pk
[3]PMAS-Arid Agriculture University, Rawalpindi | sadiaalief@gmail.com
[4]PMAS-Arid Agriculture University, Rawalpindi | azeem.abbas@uaar.edu.pk
[5]PMAS-Arid Agriculture University, Rawalpindi | miannuman10437@gmail.com
[6]PMAS-Arid Agriculture University, Rawalpindi | abdulsalam2301@gmail.com

for systematic reuse. The first step of SPLE is to extract a feature model. This further suggested recognizing the common and variant features. Manual reverse engineering of feature model for the available software variants is time and effort taking [4].

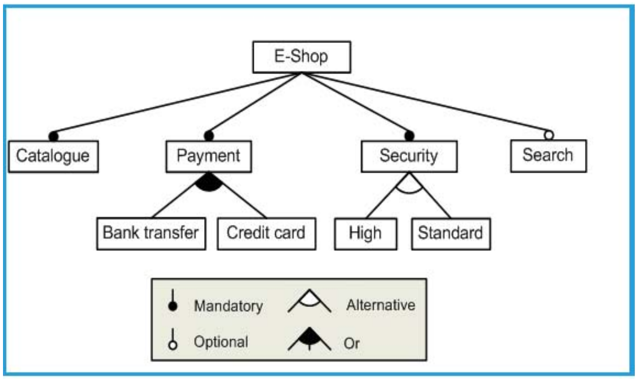


Figure 1: SPL Feature Model

When developing software, Requirements Engineering is field of defining, documenting and maintaining software requirements, mostly described in natural language [5]. This information motivated some proposals to use Natural Language Processing (NLP) to minimize the uncertainty, identify omitted information, and even enhance traceability with remaining stage of process [6].

Requirement prioritization (RP) is a main part in the requirement engineering phase. RP plays a vital part in the RE process, particularly, regarding vital tasks like requirements negotiation and software release [7]. Outstanding RP is necessary to any well-run project. It ensures that project concentrate on the main parts first, and that everybody perceived and conforms about what the project's most important parts are. There are many techniques, which are helpful for specification and prioritization of requirement according to stakeholder's time, cost, nature of the project etc. When developer used any requirement prioritization technique and find out the priority or ranking of requirements in any system, they save the ranking of the requirement with it all information and stored it in database for reuse purpose in future. For knowledge management and reuse of previous knowledge, researcher adapted AI technique called case based reasoning (CBR). CBR retrieve previous solutions for current problem solving base on expert knowledge intelligently in different scenarios [8].

In this paper, we have presented a comprehensive framework for the requirements ranking in which we extract the commonalities Cs and variabilities Vs of the software product line from the requirement document using J48 Decision algorithm. It initiates the rules for the calculation of the target variable. With the assistance of J48 classification algorithm [9] the significant distribution of the data is easily understandable. After finding the Cs and Vs apply the CBR and find the previous ranking if available then assign them else assign their ranking and find out the sorted prioritized requirement list.

The rest of this paper is structured as follows: Section 2 present literature review. In Section 3, we present our framework. In Section 4, we present evaluation and discussion.

2 Literature Review

The growing complication and cost of software-intensive systems has led developers to find the alternatives of reusing software parts in development of systems. One approach to increasing re-usability is to develop a SPL. Existing research has paying attention on techniques that create a configuration of an SPL in a single step. First, they present a formal model of multi-step SPL. Second, present the solutions to these SPL configuration problems can be automatically derived with a constraint. In future work, they plan to investigate Real-time configuration process monitoring [10].

The analysis of the requirements artifacts (SRS document, use case models) is a time taking process when performed manually. There is also required for creating consistent and complete collection of NFRs from user-specific individual projects in SPL. Therefore, they [11] propose a method to create Domain NFRs from Product NFRs using model driven approach.

It is essential for an organization to boost value creation for a given investment. The principle RE activities are to add business value that is considered for in terms of return on investment of a software product. This [12] paper provides insight into the release planning processes used in the software industry to create software product value. It presents to what degree the significant stakeholders' viewpoints are spoken to in the basic decision-making process.

SPLE strengthened high-level constructive software reuse by exploiting commonality and managing variability in a product family. To overcome the complexity of the modeling, it is divided into two views a feature tree and a dependency view [13].

In the development of a SPL, any project requires to grow core assets according to the change in environment, market, and technology. In order to successfully grow core assets, it is critical for the project to get ready and use a standardized strategy for prioritizing requirements. In paper [14], authors examine the evolution of foundation assets. Tacit knowledge for prioritizing requirements was extracted. Such knowledge was made explicit and clear to develop a way for prioritizing.

Reusing of software varies from operational, ad-hoc and short-term to strategic, planned and long-term. They [15] present and compare two different requirements-led approaches. The first deals with requirements reuse and re-usability in context of product line engineering and second in context of CBR. To assist large-scale development they proposed a Feature-Similarity model.

Requirements assurance seeks to maximize confidence in the quality of requirements through audit and review. Authors of [16] present a method that applies well-established text-mining and statistical methods to minimize this effort and increase traceability matrix assurance. The method is new, that it utilizes both requirements similarity and dissimilarity.

Prioritizing requirements focus on stakeholders' feedback brings a noteworthy cost because of time elapsed in a large number of human interactions. A Semi-Automated Framework has been presented in paper [17]. It predicts appropriate stakeholders' ratings to reduce human interactions. Future work of this research is to cluster requirements.

A prioritization method called Case-Based Ranking (CB Rank), presented in [18] which integrate project's stakeholder's desires with requirements ordering approximations calculated through AI techniques.

3 Methodology

In this segment, we present our proposed framework for ranking of stakeholders' requirements using the case base reasoning CBR if available in previous use or assign them new ranking according to their requirement and their assign ranking for software product line.

A Proposed Approach

Our framework gives an inclusive model for the requirement ranking of software product line using the CBR. Our proposed framework consists of the following layers which are:

1) Description Layer:

In this first layer we performed profiling of the system, it include two main steps first is requirement elicitation which is the process of extracting the information from stakeholders. We also get the initial ranking from the stakeholders against each requirement. As the outcome of this layer we get the requirement document along with requirement initial ranking.

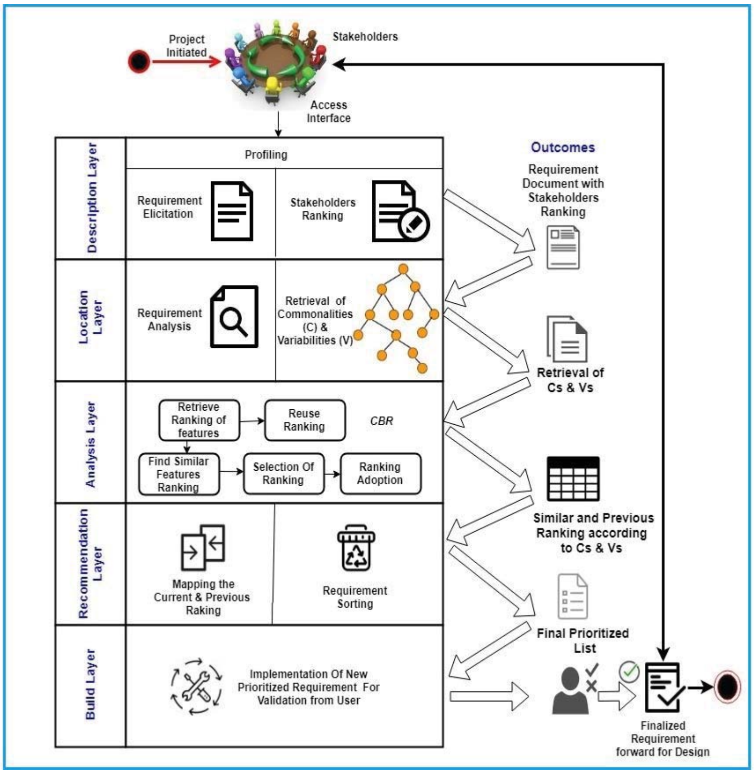


Figure 2: Proposed Framework

2) *Location Layer:*

In this second layer, we find the commonalities and variabilities of product line from the document using the J48 classification algorithm. It generates a binary tree. This approach is helpful in classification problem. Using this technique, a tree is constructed to model the classification process. [19]

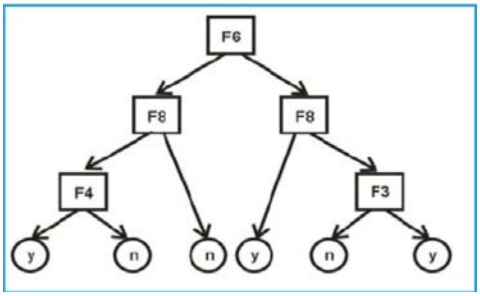


Figure 3: J48 Working

3) *Analysis Layer:*

In analysis Layer we apply the CBR, it is an AI technique that work on expert knowledge and previous experiences with less time, effort and cost. It works on the concept of reuse the solution of previous cases like new case and stores the cases in the database for later use. [8]

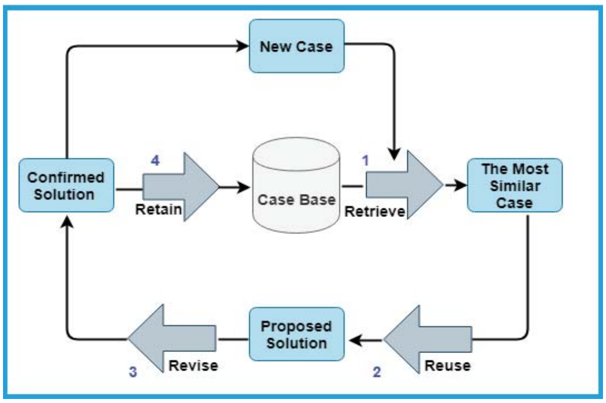


Figure 4: Case-Based Ranking (CBR)

4) *Recommendation Layer:*

In this layer we map the ranking of the stakeholder's requirements and the ranking find out from the CBR if we found the better result against the applied query we adopt the best available ranking and then apply the sorting on that list and we get the sorted prioritized list as the outcome of this layer.

5) *Build Layer:*

At this last layer we send the prioritized list to the stakeholders if they accept it and approved it then we forward it to the prototype and design of the product

4 Results and Discussion

For practical implementation of our proposed work in real world context we developed an intelligent requirements prioritization recommendation (IRPR) tool using steps of proposed work. Therefore, to evaluate IRPR we performed an empirical study. For this matter of fact, we technologies development organization which work on different projects both nationally and globally, but company not allow us to disclose any information about company. From large bulk of projects pool we selected two projects (P) i.e. LMS system (P-A) and card swipe machine (P-B).

For the elicitation and prioritization of projects user requirements before implementations uses different applications. Hence, the traditional tools/techniques (TT) they adopted increase the challenges that mention in literature review section i.e. more human interaction, ambiguous requirements etc. To resolve these issues company agreed to use IRPR tool to attain higher user satisfaction and product quality. Consequently, for IRPR implementation we conducted experiment and divided participants of company employees i.e. 21 in total for experiment into two groups' i.e. experimental treatment (ET) and non-experimental treatment (NET). The participants of ET used to develop both P-A and P-B using IRPR whereas NET participants adopted TT for implementation both projects. While participants consist of project manager (PM), requirement engineers (RE), requirement analysis (RA), developers (D) and stakeholders (S). The overall working of IRPR prototype show in figure 5-9 to illustrate the interfaces of IRPR.

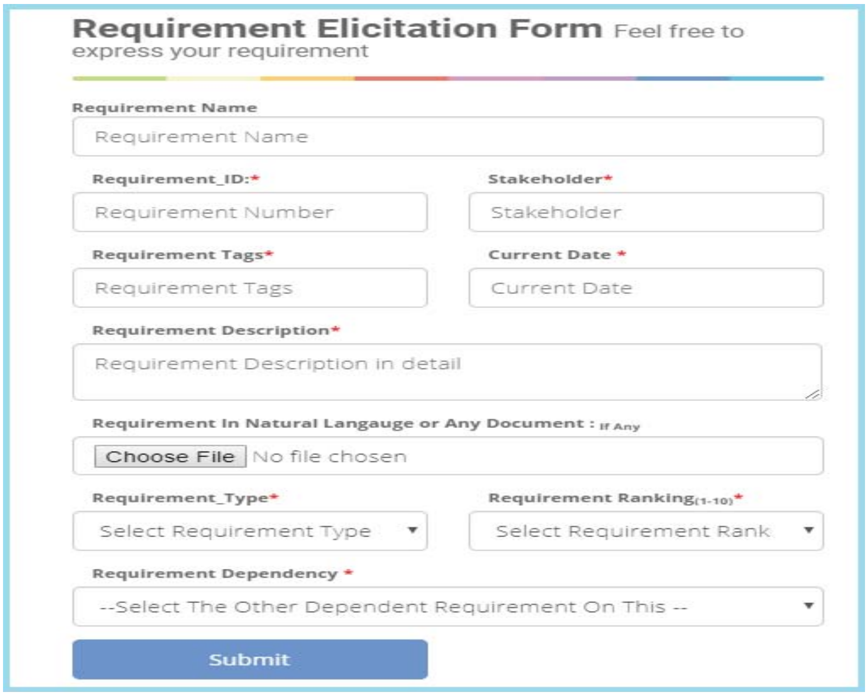


Figure 5: Requirement Elicitation & Stakeholders Ranking

When the project initiated the working of IRPR started; therefore, S connected to PM and the form open for elicitation of requirements as show in figure 5 screen shot of form interface. In the form user enter their requirements with ranking and profile of all users maintained in the database for future use. After the evaluation of profiling RE and RA analysis the requirements because these projects are SPL based. Therefore, then using j48 algorithm retrieve commonalities and variabilities in the form of decision tree for the CBR mapping to extract previous ranking as depicted in figure 6.

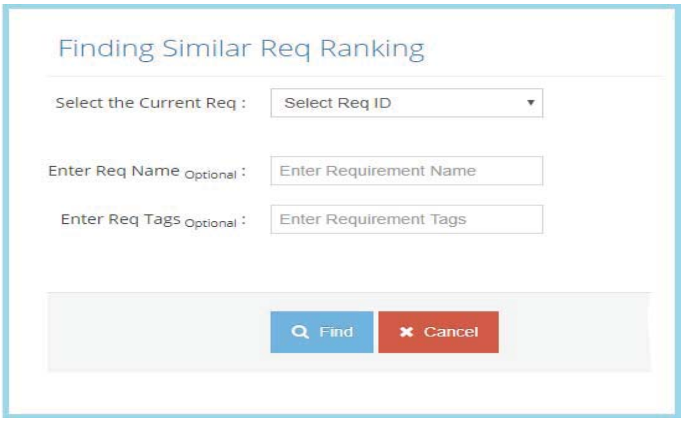
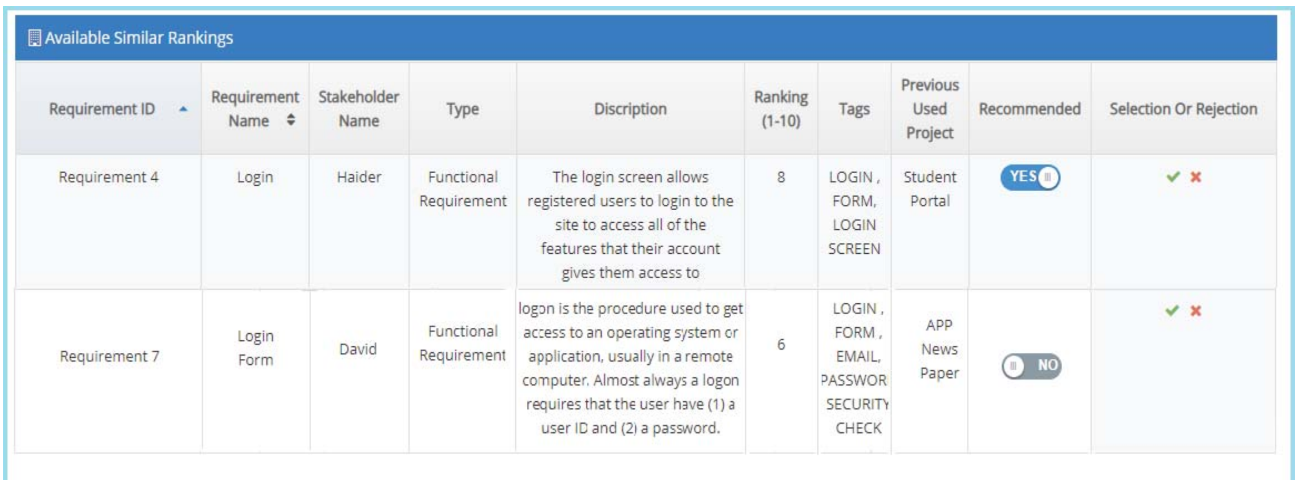


Figure 6: Finding Similar Ranking Query (CBR)

In CBR when we apply a query for finding similar ranking we will get the list (shown in figure 7) of the previous cases which are similar to the current case with the ranking. We will accept and adopt the case which is high rank amongst them.

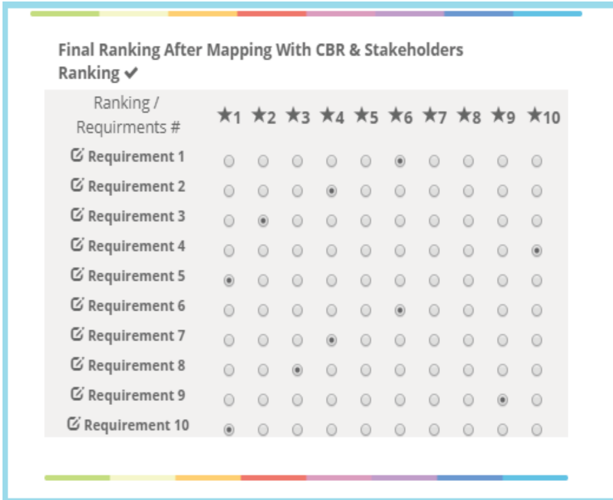


Available Similar Rankings

Requirement ID	Requirement Name	Stakeholder Name	Type	Discription	Ranking (1-10)	Tags	Previous Used Project	Recommended	Selection Or Rejection
Requirement 4	Login	Haider	Functional Requirement	The login screen allows registered users to login to the site to access all of the features that their account gives them access to	8	LOGIN , FORM, LOGIN SCREEN	Student Portal	YES	✔ ✖
Requirement 7	Login Form	David	Functional Requirement	logon is the procedure used to get access to an operating system or application, usually in a remote computer. Almost always a logon requires that the user have (1) a user ID and (2) a password.	6	LOGIN , FORM , EMAIL, PASSWOR SECURITY CHECK	APP News Paper	NO	✔ ✖

Figure 7: Selection and adoption of similar ranking

When we adopt some cases from CBR and mapped the current and the previous ranking we will get the prioritized list of the requirement with the ranking from 1-10 in a unsorted order shown in Figure 8 below.

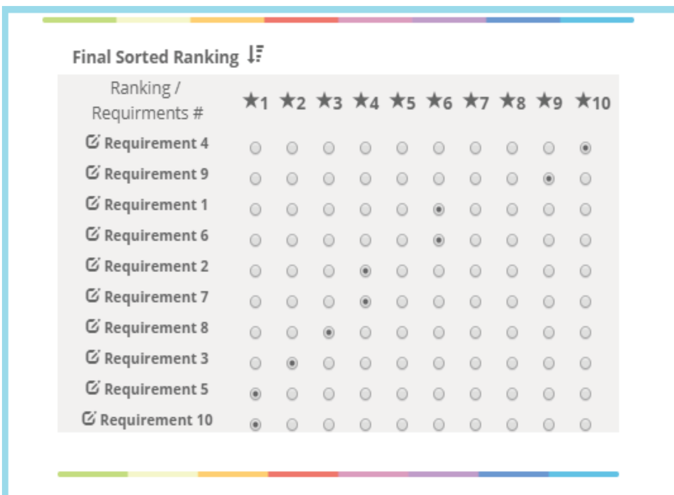


Final Ranking After Mapping With CBR & Stakeholders Ranking ✔

Ranking / Requirements #	★1	★2	★3	★4	★5	★6	★7	★8	★9	★10
Requirement 1	○	○	○	○	○	●	○	○	○	○
Requirement 2	○	○	○	●	○	○	○	○	○	○
Requirement 3	○	●	○	○	○	○	○	○	○	○
Requirement 4	○	○	○	○	○	○	○	○	○	●
Requirement 5	●	○	○	○	○	○	○	○	○	○
Requirement 6	○	○	○	○	○	●	○	○	○	○
Requirement 7	○	○	○	●	○	○	○	○	○	○
Requirement 8	○	○	●	○	○	○	○	○	○	○
Requirement 9	○	○	○	○	○	○	○	○	●	○
Requirement 10	●	○	○	○	○	○	○	○	○	○

Figure 8: Final Prioritized Requirements after mapping stakeholders and CBR Ranking

Apply any sorting technique with respect to their ranking we will get the final sorted prioritized ranking of the requirements (shown in figure 9) which will decide the education order of the requirement in development phase



Final Sorted Ranking

Ranking / Requirments #	★1	★2	★3	★4	★5	★6	★7	★8	★9	★10
Requirement 4	○	○	○	○	○	○	○	○	○	●
Requirement 9	○	○	○	○	○	○	○	○	●	○
Requirement 1	○	○	○	○	○	●	○	○	○	○
Requirement 6	○	○	○	○	○	●	○	○	○	○
Requirement 2	○	○	○	●	○	○	○	○	○	○
Requirement 7	○	○	○	●	○	○	○	○	○	○
Requirement 8	○	○	●	○	○	○	○	○	○	○
Requirement 3	○	●	○	○	○	○	○	○	○	○
Requirement 5	●	○	○	○	○	○	○	○	○	○
Requirement 10	●	○	○	○	○	○	○	○	○	○

Figure 9: Sorted Prioritized Requirements

For the assessment of experiment performance, we conducted questioner based review from both ET and NET members. The review based on parametric analysis which based on existing literate i.e. user friendly (UF), usability (U), learnability (L), efficient (E), high effectiveness (HE), less human interaction (LHI), proficient knowledge management (PKM), efficient knowledge identification and retrieval (EKIR), requirements priority accuracy (RPA), enhance elicitation and prioritization (EEP), high productivity (HP) and higher user satisfaction (HUS).

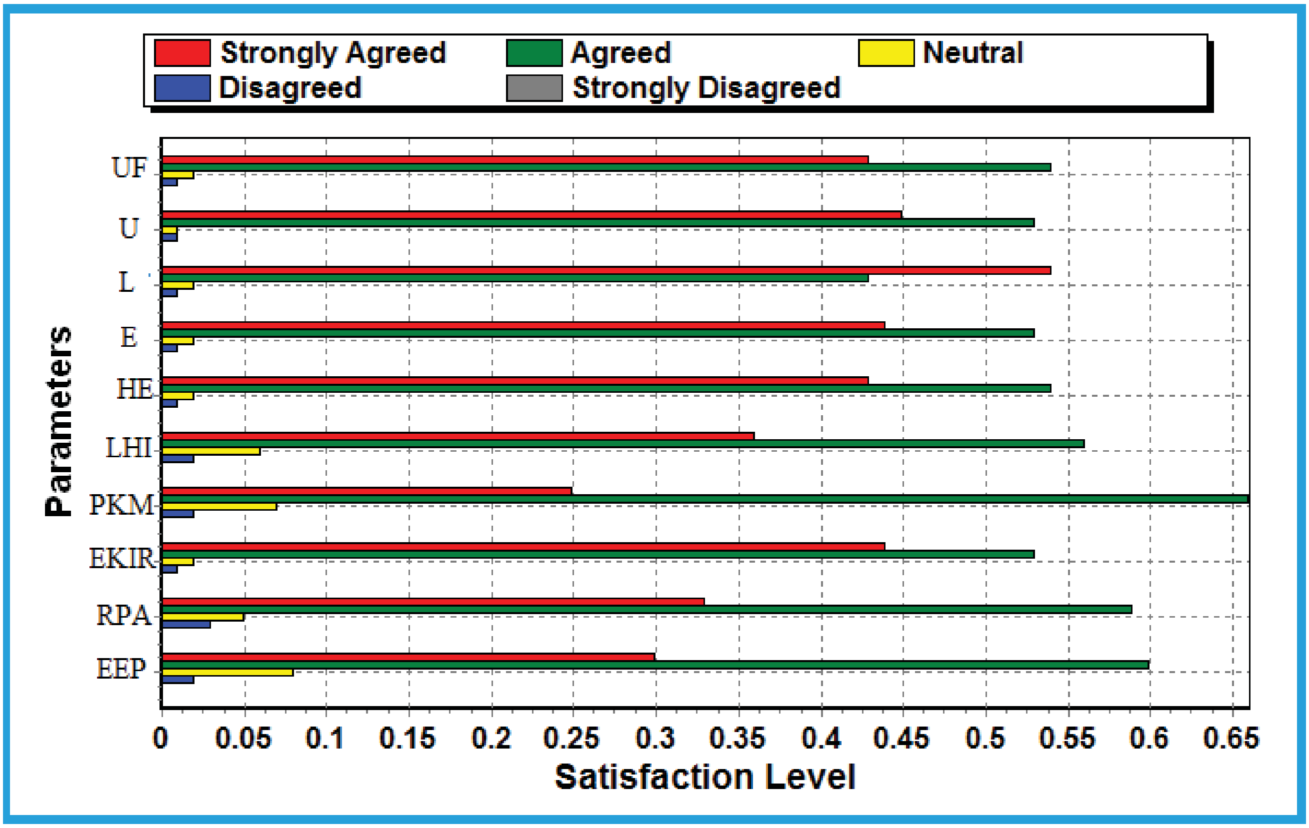


Figure 10: Review Analysis

The overall analysis result on the parameters implementing both tools i.e. IRPR and TT as demonstrate in figure 10. The figure 10 shows the satisfaction ratio of users on left side vertically with more than 50 percent satisfaction ratio and parameters review on the Y-axis.

Table 1: Comparative Analysis

Techniques	Participators				
	PM	TL	RA	Ds	QE
Experimental Treatment of P-A (ET P-A)	0.7	0.69	0.63	0.86	0.76
Non- Experimental Treatment of P-A (NET P-A)	0.32	0.36	0.45	0.27	0.38
Experimental Treatment of P-B (ET P-B)	0.80	0.70	0.60	0.76	0.86
Non-Experimental Treatment of P-A (NET P-B)	0.35	0.40	0.45	0.37	0.28

The users of project A in which experimental treatment (ET) is applied, are more satisfied and gained better results than the participants of non-experimental treatment (NET). Whereas; same is the case with participants of project B. The members of experimental treatment (ET) of B give better quality and competence.

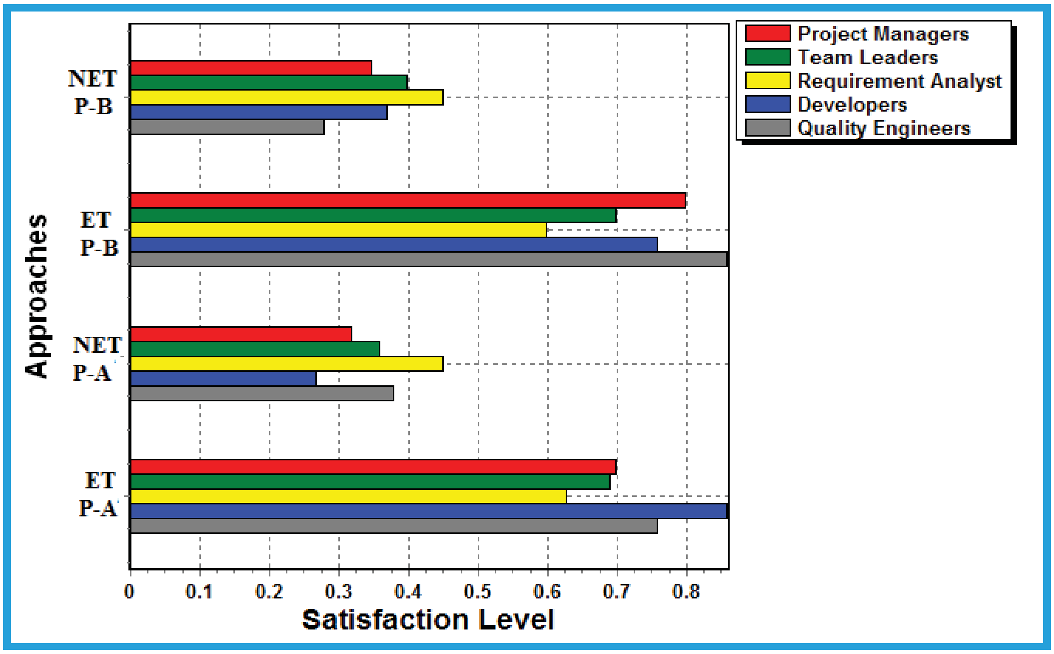


Figure 11: Comparative analysis results

We also illustrate the comparative analysis of both projects with experimental treatment (ET) and non-experimental treatment (NET) in Figure 11. Figure 11 represent the participants' satisfaction level. The y-axis labels each project development approaches while x-axis explains the satisfaction level of each user. The results present that our proposed framework's performance and satisfaction for quality and customer needs.

5 CONCLUSION

In this research, we proposed a framework for requirement ranking for software product line using CBR. The proposed framework uses J48 to find out the Cs and Vs from requirement document and then apply CBR on these requirements to find their final ranking. We have performed a tool based evaluation to evaluate our framework. Our results show noteworthy improvement in terms of satisfaction level for various parameters as compared to traditional approaches of ranking in SPL. The proposed research provides direction to industry and researchers to manage software prioritization.

References

[1] Bhushan, Megha, Shivani Goel, and Karamjit Kaur. "Analyzing inconsistencies in software product lines using an ontological rule-based approach." Journal of Systems and Software 137 (2018): 605-617.

[2] Khalique, F., Butt, W.H. and Khan, S.A., 2017, December. Creating domain non-functional requirements software product line engineering using model transformations. In 2017 International Conference on Frontiers of Information Technology (FIT) (pp. 41-45). IEEE.

[3] Xue, Y., Xing, Z., Jarzabek, S.: Feature location in a collection of product variants. In: IEEE 19th RE Conference, pp. 145–154 (2012)

[4] Ra'Fat, A., Seriai, A., Huchard, M., Urtado, C., Vauttier, S. and Salman, H.E., 2013, June. Feature location in a collection of software product variants using formal concept analysis. In International Conference on Software Reuse (pp. 302-307). Springer, Berlin, Heidelberg.

[5] D. Zowghi and C. Coulin, Requirements Elicitation: A Survey of Techniques, Approaches, and Tools, pp. 19–46. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005.

[6] Arias, M., Buccella, A. and Cechich, A., 2018. A Framework for Managing Requirements of Software Product Lines. Electronic Notes in Theoretical Computer Science, 339, pp.5-20.

[7] Hasan, M. S., Mahmood, A. Al, Alam, J., & Hasan, S. N. (2010). An Evaluation of Software Requirement Prioritization Techniques. International Journal of Computer Scienceand Information Security, 8(9), 83–94.

[8] Ali, S., Iqbal, N. and Hafeez, Y., 2018. Towards Requirement Change Management for Global Software Development using Case Base Reasoning. Mehran University Research Journal of Engineering and Technology, 37(3), pp.639-652.

[9] Kaur, G. and Chhabra, A., 2014. Improved J48 classification algorithm for the prediction of diabetes. International Journal of Computer Applications, 98(22).

[10] White, J., Galindo, J.A., Saxena, T., Dougherty, B., Benavides, D. and Schmidt, D.C., 2014. Evolving feature model configurations in software product lines. Journal of Systems and Software, 87, pp.119-136.

[11] Khalique, F., Butt, W.H. and Khan, S.A., 2017, December. Creating domain non-functional requirements software product line engineering using model transformations. In 2017 International Conference on Frontiers of Information Technology (FIT) (pp. 41-45). IEEE.

[12] Barney, S., Aurum, A. and Wohlin, C., 2008. A product management challenge: Creating software product value through requirements selection. Journal of Systems Architecture, 54(6), pp.576-593.

[13] Ye, H. and Liu, H., 2005. Approach to modelling feature variability and dependencies in software product lines. IEE Proceedings-Software, 152(3), pp.101-109.

[14] Inoki, M., Kitagawa, T. and Honiden, S., 2014, August. Application of requirements prioritization decision rules in software product line evolution. In Requirements Prioritization and Communication (RePriCo), 2014 IEEE 5th International Workshop on (pp. 1-10). IEEE.

[15] Kaindl, H. and Mannion, M., 2018, August. Software Reuse and Reusability based on Requirements: Product Lines, Cases and Feature-Similarity Models. In 2018 IEEE 26th International Requirements Engineering Conference (RE) (pp. 510-511). IEEE.

[16] Port, D., Nikora, A., Hayes, J. H., & Huang, L. (2011, January). Text mining support for software requirements: Traceability assurance. In System Sciences (HICSS), 2011 44th

Hawaii International Conference on (pp. 1-11). IEEE.

[17] Asif, S. A., Masud, Z., Easmin, R., & Ul, A. (n.d.). arXiv : 1801 . 00354v1 [cs . SE] 31 Dec 2017 SAFFRON : A Semi-Automated Framework for Software Requirements Prioritization, 1–21

[18] Perini, A., Susi, A., & Avesani, P. (2013). A Machine Learning Approach to Software Requirements Prioritization, 39(4), 445–461.

[19] Patil, T.R. and Sherekar, S.S., 2013. Performance analysis of Naive Bayes and J48 classification algorithm for data classification. International journal of computer science and applications, 6(2), pp.256-