User Story Clustering using K-Means Algorithm in Agile Requirement Engineering

Conference Paper · May 2022

DOI: 10.1109/CISES54857.2022.9844390

4 authors, including:

Umesh Tiwari
Graphic Era University
36 PUBLICATIONS **93** CITATIONS
SEE PROFILE

Dinesh Dobhal
Graphic Era University
8 PUBLICATIONS **17** CITATIONS
SEE PROFILE

Harendra Negi
Graphic Era University
6 PUBLICATIONS **2** CITATIONS
SEE PROFILE

1st International Conference on Computational Intelligence and Sustainable Engineering Solution(CISES-2022)

User Story Clustering using K-Means Algorithm in Agile Requirement Engineering

Bhawnesh Kumar
Department of Computer Science and Engineering,
Graphic Era Deemed To Be University
Dehradun, India
bhawneshmca@gmail.com

Umesh Kumar Tiwari
Department of Computer Science and Engineering,
Graphic Era Deemed To Be University
Dehradun, India
umeshtiwari22@gmail.com

Dinesh C. Dobhal
Department of Computer Science and Engineering,
Graphic Era Deemed To Be University
Dehradun, India
dineshdobhal@gmail.com

Harendra Singh Negi
Department of Computer Science and Engineering,
Graphic Era Deemed To Be University
Dehradun, India
mail.harendrasinghnegi @gmail.com

2022 International Conference on Computational Intelligence and Sustainable Engineering Solutions (CISES) | 978-1-6654-8004-8/22/$31.00 ©2022 IEEE | DOI: 10.1109/CISES54857.2022.9844390

***Abstract*— In Agile software development user story plays an important role. User story is the concept of requirement engineering that manages requirements in the form of text as software requirement specification maintained in traditional development models. Proper format is used to write the user story which is easily understandable. The impact of poor user story design may lead to modifications in later phases of development. It also places negative impression on the customer, project and development team as well. Therefore the analysis of these user stories is necessary before they are assigned to the development team. In this paper we propose an approach of user story clustering using k-means algorithm to analyze these user stories. Through this approach similar user stories are clustered on the basis of similarity measures. Our experimental results show that as the value of *k* increases the quality of resulting cluster improves. To generate balanced clusters our approach uses tf-idf which is comparatively better than count vectorizer. This approach reduces the implementation time of requirements and hence the overall development life cycle of the software. To illustrate our approach we use a case study considering various user stories.**

Keywords*— *K-means Algorithm, User Story, Silhouette Coefficient, Agile Requirement Engineering, Text Clustering

I. INTRODUCTION

Agile is a methodology that promotes involvement of customer in software development to achieve the requirement engineering. Agile has the capability to perform testing along with coding throughout the software development [1]. In waterfall model after requirements (software requirement specification) are freeze no customer involvement is there. The Agile software development methodology [2] is one of the simplest and effective processes to turn a business requirement *vision* into software productive solutions in an iterative manner. Agile Requirements Engineering (ARE) is a phase where expertise is applicable in various situations and processes. Agile RE practices are those that support an iterative development with emergent requirements [3]. ARE follows the distinguish phases for collecting requirement such as elicitation, analysis and negotiation, modeling, validation and management [4]. To make agreement between developers and customers, it is required to prepare textual data of requirements following the principles of agility that is known as User Story. To fulfill the needs of the requirement phase, we have documentations in waterfall model as well as in other traditional models that is known as SRS. In agile software development we have collection of user stories. Agile teams prefer to capture requirements in a way that strongly supports interaction with stakeholders and the desired flexibility. For example, in agile framework the following format is used to cover user stories: As a/an [type of user, say admin] I want [some feature] so that [some reason]). These user stories may be on sticky notes or may be recorded on index cards, story maps or vision boards. Most common techniques to record user stories are role-feature-reason template, the 3Cs-card, conversation, confirmation and INVEST rule where I denotes Independent, N denotes Negotiable, V is Valuable, E means Estimable, S stands for Small, and T is used to denote Testable [5].

These user stories are assigned to development team for further development and achieve the desired requirements. Poor user story designs are responsible for unnecessary modifications and may put negative impact on the customer, project and development team as well [8]. Therefore analysis of user stories is very much required before assigned to the development team. The concept of text clustering can also be implemented to form the similar user stories to the corresponding clusters. Many authors [6, 7] suggested the need of exploring user story clustering on the basis of similarity measures.

This paper in divided into six sections which are as follows: section II and III represents the background and related work respectively in detail. The proposed approach and flow chart are explained in Section IV. Section V discusses the case study of user stories which are clustered using similarity measures. Finally conclusion is described in section VI.

II. BACKGROUND

A. Text clustering

Text clustering is used to identify the groups of similar documents. The basic means of the document clustering is to identify similarity measures on the basis of which clusters are formed [9]. The main goal of document clustering technique is to minimize the intra-cluster distance among documents and maximize the inter-cluster distances among documents by using similarity measure. The analysis of text clustering algorithm is focused on the occurrence of text in given documents. Clustering technique can be categorized into two ways; hierarchical or partitioned [10].

B. Vector space model

The document is initially converted into numerical format and represented in the form of metric which is represented by vector space model (VSM) [11]. VSM represents a document d with the set of terms $T = \{t_1, t_2, t_3 \dots \dots t_n\}$.Let the set of documents represented as $D = \{d_1, d_2, d_3 \dots \dots d_n\}$ and distinct terms as $T = \{t_1, t_2, t_3 \dots \dots t_n\}$ in document d. Therefore the tf-idf is equal as shown in equation (1):

$$tf - idf(t_i, d_j, D) = tf(t_i, d_j)\, log \frac{|D|}{df(t_i)} \quad (1)$$

where $tf(t_i, d_j)$ is weight of term t_i in a document d_j. The total number of documents is represented as $|D|$. $df(t_i)$ is used to denote document frequency of term t_i.

C. Preprocessing

Tasks that are performed in preprocessing phase are: tokenization, stop word, stemming and lemmatization. All these tasks are implemented to filter the textual requirements. Tokenization is used to divide the text into tokens by delimiters (white spaces, comma and similar). Stop word is used for removing prepositions, pronouns and irrelevant words from the document. Stemming and lemmatization succeed with root and core word [12].

D. The K-means algorithm

It is an iterative and partitioned based clustering approach where data points partitioned into k clusters using centroid. In a cluster, mean value of data points work as centroid. K-means steps are as follows[13]:

(a) randomly select k data points as centroids.

(b) similarity of data points for each cluster centroid is computed.

(c) based on newly assigned data points k centroids are updated.

(d) repeat steps (a) and (b) till all data points covered.

E. Silhouette Coefficient

Silhouette tells how well data point is placed in a cluster that it also indicates the consistency of data point with in cluster [14]. The value of silhouette lies between -1 and +1 where -1 indicates worst value and +1 is for best value. If value is zero then it shows the overlapping clusters. It is calculated by the mean of intra-cluster distance $a(d_i)$ and mean of nearest cluster distance $b(d_i)$ for each sample. The equation (2) for silhouette coefficient as follows:

$$S(d_i) = \frac{b(d_i) - a(d_i)}{max\{a(d_i), b(d_i)\}} \quad (2)$$

III. RELATED WORK

Most of the software projects fail due requirement engineering challenges and the poor management of these challenges [15]. Studies show that the project development challenges are 37% (13% poor user input, 12% incomplete requirements and 12% changing requirement) in the requirement phase as illustrated by [16, 17]. Poor requirements gathering may cause the improper formation of user stories that ultimately leads to faulty development decisions. Due to these reasons project completion cannot be done on time and increases the delay time which directly impacts the budget. R. Barbosa et al. [18] (2015) introduces an approach based on textual requirements clustering and data dictionary. They suggested a sequence for implementation of user stories. Their work calculates silhouette coefficient for cluster size 10, 15, 20 and 25 and user stories distribution done by K-medoids algorithm. R. Barbosa et al. [19] (2016) presented a semantic similarity measure (Jaccard Index and Cosine Similarity) based approach to suggest possible cases of duplication between original and duplicate user stories. Y. Sebega and E. Mnkandla [20] (2017) gives issues regarding project failures in the context of South African software industry. This work reported that lack of proper validation tools/techniques which is approximately 73.9% as a major problem of project failure. Approximately 68% issues due to elicitation, 64% due to lack of proper documentation, and 60.90% due to prioritization. Niels Bik et al. [21] (2017) constructs a reference method through examining and consolidating the life cycle user story. There are six case studies discussed in this work and indicate that need of minor changes and suggestions are required to get improvements with reference methods. In future organizations are ready to optimize user story usage. Angela Cristina Pereira [22] (2018) proposes an approach to generate user stories using natural language processing techniques. This work considers three groups and the third group has the best an average accuracy. Two texts tested in between 76% and 88.23% for the average precision, 78.95% for recall and 83.33% for average F-one measure, lowest error rate for this group presented is 24%. In future research can extended to improve the identification of system entities based on the lemmatization of their core word. Indra Kharisma Raharjana et al. [6] (2019) proposes a conceptual model for creating user stories from online news. This model converts the online data into user story template such as “who, what and why”. This work can also extended by implementing the clustering to form the similar user stories. Takwa Kochbati et al. [7] (2021) proposed a machine learning based approach to divide the whole system into subsystems. This approach initially computes the similarity of requirements using word2vec model and later apply the Hierarchical Agglomerative Clustering to make the clusters of similar user stories of requirements. Indra Kharisma Raharjana et al. [23] (2021) gives an SLR of natural language processing (NLP) on user stories. Mostly research has been done to extract the user story phrase in the form of “who, what and why” from sentences through NLP. It is really a challenging task to implement the NLP to generate the software artifacts.

IV. PROPOSED WORK

To analyze software requirements similarity is used as a key factor. This proposed approach basically works for text clustering to recognize the homogeneity of requirements and classify the clusters. This approach follows steps as shown in fig 1.

A. Preprocessing

Initial step is taken to implement tokenization, removal of stop-words, and stemming/lemmatization processes on requirements. Later on terms assigned by count and tf-idf vectorizer.

B. Clustering

This step is used to implement k-means clustering technique on count and tf-idf vectorizer with cosine similarity to identify the clusters of requirements.

C. Validation

Now we need to find the best result of clustering. The number of clusters created with different k-values and verifies the silhouette coefficient value (which lies between -1 to +1) that indicates the best cluster. Also find out that how many user stories assigned to each cluster.

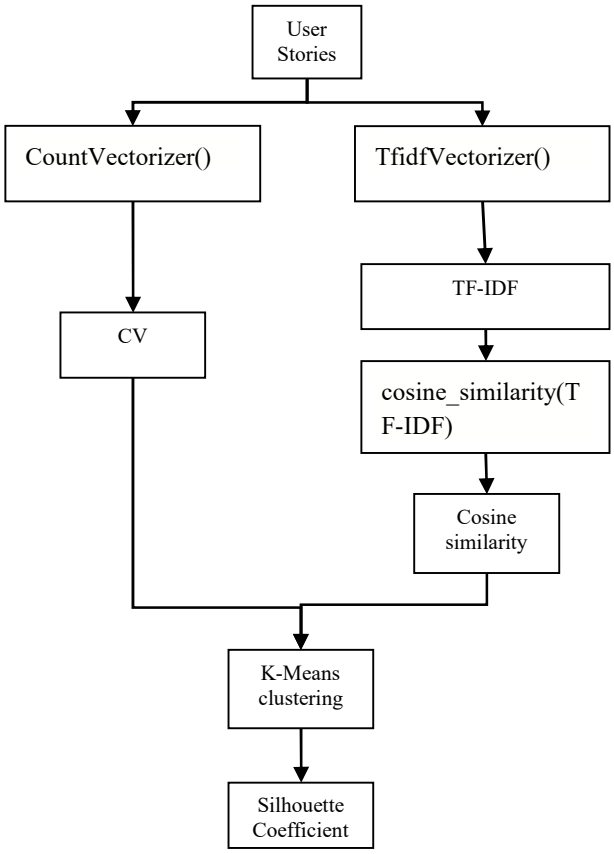

Fig 1. Steps for proposed work

V. CASE STUDY AND RESULT DISCUSSION

To implement our clustering approach 98 user stories are clustered according to rules and regulation by Cohn[5]. These user stories are from open access (mendeley data) and publisher is Data Archiving and Networked Services. These user stories numbered as User story 0 to 97 i.e US0 to US97. In 98 user stories there are who (98), what (97) and why (49). In this proposed work cluster size starts from 2 to 14. In this paper 14 clusters are considered for user story distribution. User stories distribution as count vectorizer shown in fig 2. In the graph each point represents a user story and colored circle indicates the cluster they belong. User story distribution as count vectorizer cluster 3 has more cohesion as user stories 1, 2, 3, 4, 5, 9, 10, 21, 23 and 24 come under cluster 3 which is represented in upper left portion of the graph. User story 50 is closer to user stories 12, 62 and 85 still belong to different clusters due to the distance measures.

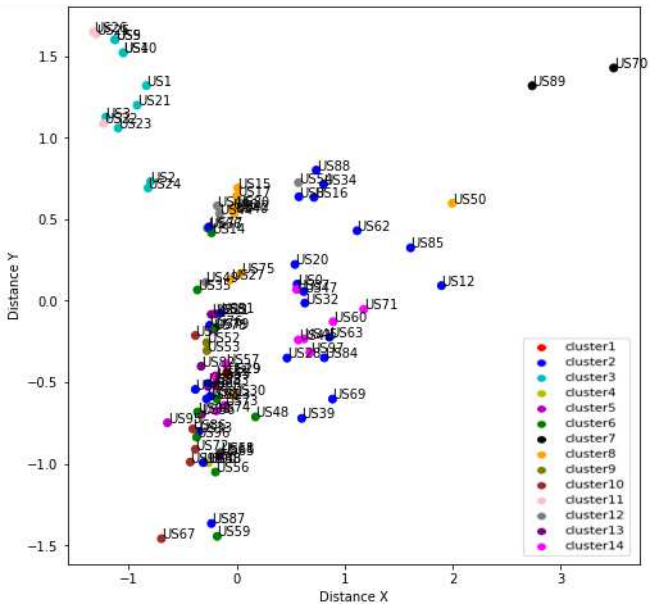

Fig 2. User story distribution as per count vectorizer

Fig 3 shows the user story distribution as per tf-idf vectorizer. Principal component analysis [24] is used to reduce the dimensions so that the graphical resentation can be possible in two dimensional graph. User story distribution as tf-idf vectorizer cluster 3 has more cohesion. User stories 3, 4, 5, 9 and 10 are shown in right portion of the graph.

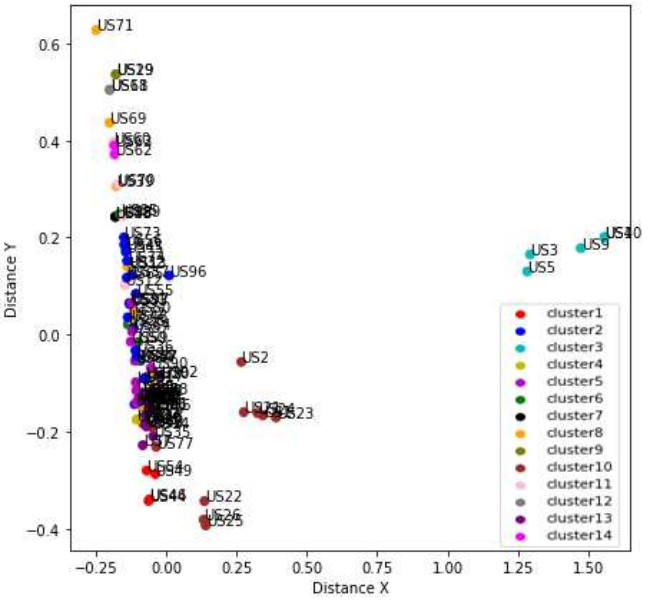

Fig 3. User story distribution as per tf-idf vectorizer (cosine similarity)

R. Barbosa et al. [18] considered 50 user stories were clustered using K-mediods algorithm with having stopword and stemming process which removed 183 words later calculates the silhouette coefficient values. In our approach considered 98 user stories clustered using K-means algorithm along with stopword, stemming and lemmatization process as well implemented. Silhouette coefficient values represented in fig 4 according to the count-vertorization and tf-idf user story distribution. In case of tf-idf as the cluster size increases the correspodning silhouette value also increase. We observe that the tf-idf performance is better than count vectorizer.

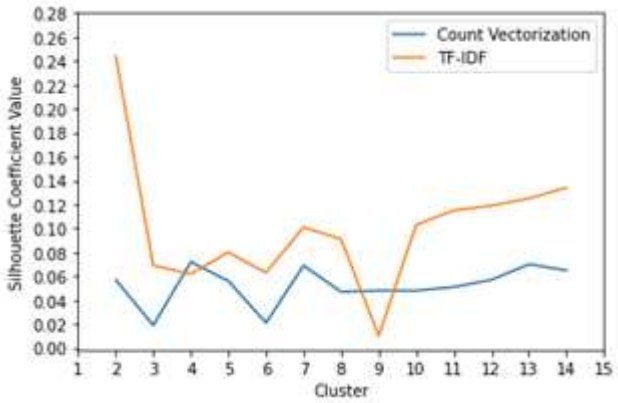

Fig 4. Silhouette coefficient values

Fig 5 shows the comparison graph between count and tf-idf vectorizer. It depicts that how many user stories come under the cluster. Balanced user story distribution is done by using tf-idf in comparison to count vectorizer. Overall the best performance is obtained using K-means with cosine similarity.

The contribution of this work is the clustering of multiple user stories by using k-means clustering as per the similarity measures. As the cohesion among user stories increases the implementation of requirements becomes easier, identification and understanding of features becomes simpler to implement in agile software development.

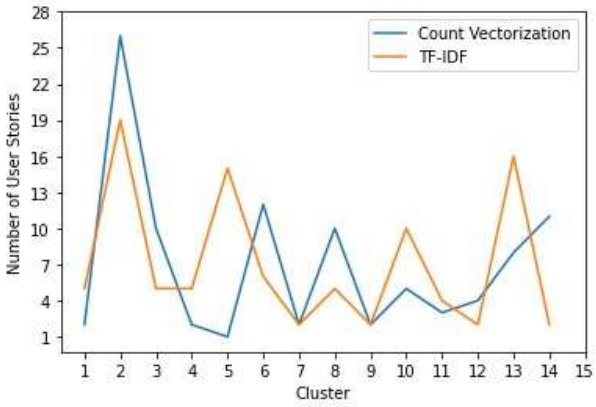

Fig 5. User story clusters

This approach can be helpful to make a common system that features as per the sequencing of functional requirements. In this work we have discussed tools and techniques with implementation. This work is not only helpful in reducing the time of implementation of requirements and better identification but it also helps to understand the features that are to be implemented in software.

VI. Conclusion

This paper introduces an approach for user story clustering using k-means algorithm with cosine similarity. Proposed approach has three steps namely preprocessing, clustering and validation. In preprocessing step tokenization, stop word, stemming and lemmatization followed. Clustering creates clusters of user stories on the basis of similarity measures. The final step is to find out the cohesion level of user story clusters through silhouette coefficient values. Case study discussed in this work contains 98 user stories. Our approach creates 2 to 14 clusters of user stories. Similar user stories clustered on the behalf of similarity measures. Our experimental results show that as the value of *k* increases the quality of resulting cluster improves. Our approach shows that tf-idf values are better to generate balanced clusters in comparison to count vectorizer. Technique proposed in this work helps to reduce the time of implementation of requirements due to which software development can be done on time.

In future more clustering algorithms with other similarity measures can also be considered to enhance the agile software development.

References

[1] P. Alessandra and N. Scarpato, “Affective agile design a proposal for a new software development model,” J. Theor. Appl. Inf. Technol., vol. 96, no. January, pp. 68–79, 2018.

[2] P. Abrahamsson, N. Oza, and M. T. Siponen, “Agile Software Development Methods: A Comparative Review,” Int. Conf. Softw. Eng., vol. 07/80, no. 2, 2003, [Online]. Available: https://arxiv.org/pdf/1707.06526.pdf%0Ahttps://www.yrpri.org%0Ahttp://weekly.cnbnews.com/news/article.html?no=124000%0Ahttps://www.fordfoundation.org/%0Ahttp://bibliotecavirtual.clacso.org.ar/Republica_Dominicana/ccp/20120731051903/prep%0Ahttp://webpc.cia.

[3] M. Ochodek and S. Kopczyńska, “Perceived importance of agile requirements engineering practices – A survey,” J. Syst. Softw., vol. 143, pp. 29–43, 2018, doi: 10.1016/j.jss.2018.05.012.

[4] E. Rantanen, “Requirements Engineering in Agile Software Development,” University of Oulu Faculty of Technology, 2017.

[5] M. Cohn, User Stories Applied for Agile Software Development, 13th ed. Addison Wesley Longman Publishing Co., Inc., 2004.

[6] I. K. Raharjana, D. Siahaan, and C. Fatichah, “User Story Extraction from Online News for Software Requirements Elicitation : A Conceptual Model,” 2019 16th Int. Jt. Conf. Comput. Sci. Softw. Eng., pp. 342–347, 2019.

[7] T. Kochbati, S. Li, S. Gérard, and C. Mraidha, “From user stories to models: A machine learning empowered automation,” Model. 2021 - Proc. 9th Int. Conf. Model. Eng. Softw. Dev., pp. 28–40, 2021, doi: 10.5220/0010197800280040.

[8] M. Tanner and M. L. Pan, “The Impact of Impromptu User Story Modifications on the Project , Customers and Team Members During a Sprint,” in Joint International Conference, 2015, no. January 2015, pp. 111–120.

[9] A. Hotho, A. Nurnberger, and G. Paass, “A brief survey of text mining,” Gld. J. Comput. Linguist. Lang. Technol., 2005.

[10] A. Casamayor, D. Godoy, and M. Campo, “Functional grouping of natural language requirements for assistance in architectural software design.,” Knowledge-Based Syst., no. 30(1), pp. 78–86, 2012.

[11] G. Salton and C. Buckley, “Term-weighting approaches in automatic text retrieval,” Inf. Process. Manag., vol. 24, pp. 513–523, 1988.

[12] S. Al-anazi, H. Almahmoud, and I. Al-turaiki, “Finding Similar Documents Using Different Clustering Techniques,” Procedia - Procedia Comput. Sci., vol. 82, no. March, pp. 28–34, 2016, doi: 10.1016/j.procs.2016.04.005.

[13] J. A. Hartigan, Clustering Algorithms, 99th ed. John Wiley & Sons, Inc., 1975.

[14] K. R. Shahapure and C. Nicholas, “Cluster Quality Analysis Using Silhouette Score,” 2020, doi: 10.1109/DSAA49011.2020.00096.

[15] S. Arun Kumar and T. Arun Kumar, “Study the impact of Requirements management Characteristics in global software development projects: An Ontology based approach,” Int. J. Softw. Eng. Appl., vol. 2, no. 4, pp. 107–125, 2011, doi: 10.5121/ijsea.2011.2410.

[16] A. De Lucia and A. Qusef, “Requirements engineering in agile software development,” J. Emerg. Technol. Web Intell., vol. 2, no. 3, pp. 212–220, 2010, doi: 10.4304/jetwi.2.3.212-220.

[17] A. Hussain, E. O. C. Mkpojiogu, and F. M. Kamal, “The role of requirements in the success or failure of software projects,” Int. Rev. Manag. Mark., vol. 6, no. 7Special Issue, pp. 305–310, 2016.

[18] R. Barbosa, D. Januario, A. E. Silva, R. Moraes, and P. Martins, “An Approach to Clustering and Sequencing of Textual Requirements,”

Proc. - 2015 45th Annu. IEEE/IFIP Int. Conf. Dependable Syst. Networks Work. DSN-W 2015, pp. 39–44, 2015, doi: 10.1109/DSN-W.2015.20.

[19] R. Barbosa, A. E. A. Silva, and R. Moraes, "Use of Similarity Measure to Suggest the Existence of Duplicate User Stories in the Srum Process," Proc. - 46th Annu. IEEE/IFIP Int. Conf. Dependable Syst. Networks, DSN-W 2016, pp. 2–5, 2016, doi: 10.1109/DSN-W.2016.27.

[20] Y. Sebega and E. Mnkandla, "Exploring issues in agile requirements engineering in the South African software industry," Electron. J. Inf. Syst. Dev. Ctries., vol. 81, no. 1, pp. 1–18, 2017, doi: 10.1002/j.1681-4835.2017.tb00597.x.

[21] N. Bik, G. Lucassen, and S. Brinkkemper, "A reference method for user story requirements in agile systems development," Proc. - 2017 IEEE 25th Int. Requir. Eng. Conf. Work. REW 2017, no. September, pp. 292–298, 2017, doi: 10.1109/REW.2017.83.

[22] ANGELA CRISTINA PEREIRA, "USING NLP TO GENERATE USER STORIES FROM SOFTWARE SPECIFICATION IN NATURAL LANGUAGE," Universidade Federal do Paraná, 2018.

[23] I. K. Raharjana, D. Siahaan, and C. Fatichah, "User Stories and Natural Language Processing : A Systematic Literature Review," IEEE Access, vol. 9, pp. 53811–53826, 2021, doi: 10.1109/ACCESS.2021.3070606.

[24] I. T. Jollife and J. Cadima, "Principal component analysis: A review and recent developments," Philos. Trans. R. Soc. A Math. Phys. Eng. Sci., vol. 374, no. 2065, 2016, doi: 10.1098/rsta.2015.0202.