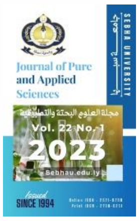
SEBHA UNIVERSITY JOURNAL OF PURE & APPLIED SCIENCES VOL.22 NO. 2 2023
DOI: 10.51984/JOPAS.V22I2.2599

مجلة جامعة سبها للعلوم البحتة والتطبيقية
Sebha University Journal of Pure & Applied Sciences

Journal homepage: www.sebhau.edu.ly/journal/index.php/jopas



Leveraging Latent Natural Language Processing Techniques for User Story Management in Agile Software Development

Hisham A. A. Imhmed[1], Khamiss M. S. Ahmed[2], Yousif I. E. Salem[2], Hazura Zulzalil[3]

[1]Department of Information Systems, Faculty of Information Technology, Sebha University- Libya
[2]Department of Computer Science, Faculty of Information Technology, Sebha University- Libya
[3]Department of Software Engineering and Information Systems,Faculty of Computer Science & Information Technology,University Putra Malaysia, 43400, Serdang- Malaysia

Keywords:
Agile Software Development
Artificial Intelligence
Behaviour Driven Development
Domain Specific Language
Gherkin Testing Agile
User Story Management
Natural Language Processing (NLP)

ABSTRACT

User story management is a critical aspect of agile software development, as it involves understanding and prioritizing the needs of stakeholders, and translating them into actionable tasks for development teams. Furthermore, the identification of new activities within products is crucial for enhancing software quality assurance, and user story management is a crucial component of agile software development and software testing. Reusing manually written test steps would be a waste of time and effort for the testers because requirements in agile software development are continually changing and eventually becoming out-dated. Therefore, developers need to implement the necessary functions and write test steps for user stories to determine the desired behaviour or desired result of the program. In an agile world, the user story is the link between the customer and the development team, as well as the main pillar on which the development team relies to understand the product requirements. Therefore, developers need to implement the necessary functions and write test steps for user stories to determine the desired behaviour or desired result of the program. Since agile testing can detect defects early in the software life cycle and deliver a high-quality product, we propose our approach to generating test cases using natural language processing to analyse the user story. Therefore, Neural Language Processing (NLP) techniques help the development team to obtain clear data and achieve customer satisfaction.

الاستفادة من تقنيات معالجة اللغة الطبيعية الكامنة لإدارة قصة المستخدم في تطوير البرمجيات الرشيقة

هشام ابوبكر عبدالرحمن امحمد[1] وخميس مسعود سالم احمد[2] ويوسف اسويسي امحمد[2] و هازورة زولزاليل[3]

[1]قسم نظم المعلومات ، كلية تقنية المعلومات ، جامعة سبها، ليبيا
[2]قسم علوم الحاسب - كلية تقنية المعلومات - جامعة سبها،ليبيا
[3]قسم هندسة البرمجيات ونظم المعلومات ، كلية علوم الحاسب وتكنولوجيا المعلومات ، جامعة بوترا ماليزيا

الملخص

تعد إدارة قصة المستخدم جانبًا مهمًا من جوانب تطوير برمجيات رشيقة، حيث إنها تتضمن فهم احتياجات أصحاب المصلحة وتحديد أولوياتها، وترجمتها إلى مهام قابلة للتنفيذ لفرق التطوير. علاوة على ذلك ، يعتبر تحديد الأنشطة الجديدة داخل المنتجات أمرًا بالغ الأهمية لتعزيز ضمان جودة البرامج، وتعد إدارة قصة المستخدم عنصرًا حاسمًا في تطوير البرمجيات الرشيقة واختبار البرامج. قد تكون إعادة استخدام خطوات الاختبار المكتوبة يدويًا مضيعة للوقت والجهد للمختبرين لأن المتطلبات في تطوير البرمجيات الرشيقة تتغير باستمرار وتصبح في النهاية قديمة. لذلك ، يحتاج المطورون إلى تنفيذ الوظائف الضرورية وكتابة خطوات اختبار لقصص المستخدم لتحديد السلوك المطلوب أو النتيجة المرجوة من البرنامج. في عالم رشيقة، تعتبر قصة المستخدم هي الرابط بين العميل وفريق التطوير، فضلاً عن الركيزة الأساسية التي يعتمد عليها فريق التطوير لفهم متطلبات المنتج. لذلك، يحتاج المطورون إلى تنفيذ الوظائف الضرورية وكتابة خطوات اختبار قصص

الكلمات المفتاحية:
اختبار غيركين الرشيق الذكاء
الاصطناعي
إدارة قصة المستخدم
تطوير البرمجيات الرشيقة
معالجة اللغة العصبية

*Corresponding author:
E-mail addresses: His.imhmed@sebhau.edu.ly, (K. Ahmed) km.ahmed@sebhau.edu.ly, (Y. Salem) cyou.salem@sebhau.edu.ly, (H. Zulzalil) hazura.zulzalil@gmail.com

Article History : Received 27 May 2023 - Received in revised form 05 October 2023 - Accepted 15 October 2023

المستخدم لتحديد السلوك المطلوب أو النتيجة المرجوة من البرنامج. ونظرًا لأن الاختبار السريع يمكنه
اكتشاف العيوب في وقت مبكر من دورة حياة البرنامج وتقديم منتج عالي الجودة، فإننا نقترح نهجنا لإنشاء حالات
اختبار باستخدام معالجة اللغة الطبيعية لتحليل قصة المستخدم. حيث أن تقنيات معالجة اللغة العصبية
(NLP) تساعد فريق التطوير في الحصول على بيانات واضحة وتحقيق رضا العملاء.

Introduction

In one of the four items of the "agile manifesto" the work programs must be comprehensive (Molokken-ostvold & Furulund, 2007). The twelve agile principles behind the agile manifesto helps us elicitation a clear concept of how it will be agile development software. Where, principles of agile include early and continuous delivery of the product, changing requirements any time throughout development, frequent delivery, Collaboration between and within the developers and customer throughout the development, trust, motivate and Support the people interested, effectual communication, concentrate on working software, reinforcement sustainable development, continual attention to simplicity, technical distinction, perfect design and self-organizing teams, at regular intervals (Robey et al., 2001). There have been many promising studies suggest that agile methods can improve productivity, job satisfaction, and customer satisfaction. The main reason for the emergence of agile methods is to overcome the challenges and weaknesses of traditional methods. The user's story expresses the customer's point of view. More specifically, the customer collaborates with agile team by specifying the User Stories, setting their priority and answering questions about them in order to clarify the real requirements by having development and testing go hand-in-hand from the start. Testing is continuously integrated into Agile from early developmental stages to ensure defect-free continuous deployment and that all requirements are met.

Agile software development works with constantly changing requirements and business environments and makes networking in agile projects a critical factor for a successful project. Agile Software Methodology with Scrum stated that choosing an agile development methodology with Scrum is very appropriate, and the results showed that the Agile Software methodology with Scrum is the right way to increase effectiveness and efficiency in developing QAS applications. In order to produce high-quality products that meet customer expectations, the involvement of customers through all stages of the project is very necessary. However, due to a lot of factors such as the different geographic location of the software company (Kevin Sungkur & Ramasawmy, 2014) and the customers (Hoda et al., 2011), as well as the lack of availability of customers to make such a visit on daily basis as an agile emphasis; It is almost impossible to adopt an agile method with the existence of such circumstances. The lack of having customers and team members, and teams' together collaboration and communication with each other would result in a loose process of managing customer requirements and handing agreement/sign off of each finished requirement. Moreover, Developers need to implement the necessary functions and write test steps for user stories to determine the desired behavior or desired outcome of the software. Test steps that are similar to a number of stories must be reused rather than re-implemented. Nevertheless, the conventional process of finding out reusable test steps is monotonous and time-consuming (Landhäußer & Genaid, 2012).

The major artifact produced in ASD projects is the code; however, it is usually very difficult to discuss requirements with the product owner in terms of code.

In this study, the tool has been designed to retrieve test steps for user stories using Gherkin which provides a lot of time and effort to the testers. This tool has given all participants in the project a common understanding of what is being developed, delivered, or tested. However, the tool supports internal communication between team members, as the testers write a test script describing the features to be implemented and then systematically translate them into code stubs attached for later use by developers to fill and execute them using a high-level language. The main aim of this study is a tool that uses agile user stories to automatically generate test steps.

The tool can be used by testers who may or may not be experienced, to achieve the following improvements in testing within ASD. The tool:

- Reduces the time taken to create test steps.
- Reduces the effort required to create test steps.
- Generates test steps for multiple user stories of a feature, further reducing the time and effort for the tester.

Natural Language Processing (NLP) technology can be used in different stages of user story management. In the initial stage of creating user stories, NLP can be used to extract requirements from natural language text. NLP techniques were used to extract user stories from customer reviews on e-commerce websites. The study showed that NLP techniques could extract requirements with an accuracy of 82%.

NLP is a subfield of artificial intelligence (AI) that focuses on enabling computers to understand and process human language. It involves the development of algorithms and computational models that can analyze, interpret, and generate human language data.

The purpose of this study is to use Natural Language Processing (NLP) techniques to improve the way communication software requirements are in agile software development projects, to ensure that the intended product is being developed and tested correctly. As they convey end-user requirements in a clear and understandable manner, user stories are a crucial part of agile software development. Throughout the software development lifecycle, user stories are collected, prioritized, and improved through the process of user story management. Yet, handling vast and different stakeholder groups can make managing user stories a time-consuming and challenging process. The user story management process can be streamlined using NLP approaches, which will also increase the effectiveness of agile software development overall.

Related Work

User stories are used to record needs from stakeholders as part of story management, which is a crucial component of software development. Creating, prioritizing, estimating, and tracking user stories over the course of the development process are all tasks that are involved in managing user stories. In recent years, there has been a lot of interest in the application of NLP technology to user narrative management. This study of the literature provides an overview of the pertinent research on the application of NLP technology to user story management.

NLP techniques can be used to automate several aspects of user story management, including requirements gathering, prioritization, and refinement. For example, NLP can be used to automatically extract key information from user feedback and convert it into user stories. These user stories can then be automatically prioritized based on their relevance and importance. NLP can also be used to identify and resolve conflicts between user stories, ensuring that the requirements are consistent and aligned with the overall project (Gunes & Aydemir, 2020).

Another way in which NLP can be leveraged in user story management is by automating the process of story refinement. NLP techniques can be used to identify common patterns and themes in user stories, which can then be used to generate new stories or refine existing ones. This can help to reduce the amount of manual effort required to refine user stories, thereby increasing the efficiency of the agile development process Almanaseer et al., 2022).

NLP techniques are used in a wide range of applications, including machine translation, chatbots, sentiment analysis, speech recognition, and text summarization. NLP involves several stages, including:

- Tokenization: Breaking down the text into individual words, phrases, or sentences.

- Part-of-speech (POS) tagging: Assigning a grammatical label to each token, such as a noun, verb, or adjective.
- Named entity recognition (NER): Identifying named entities such as people, places, and organizations within the text.
- Parsing: Analyzing the grammatical structure of the text and identifying the relationships between words.
- Sentiment analysis: Determining the emotional tone of the text, such as positive, negative, or neutral.

NLP techniques rely on machine learning algorithms that are trained on large datasets of human language data. These algorithms can learn patterns and relationships within the data and use this knowledge to analyze and generate human language.

NLP has several challenges, including ambiguity, context sensitivity, and the complexity of human language (Chowdhary, 2020). NLP algorithms must be able to understand the nuances of human language, including idioms, metaphors, and sarcasm.

Despite these challenges, NLP has made significant progress in recent years and has enabled many applications that were previously impossible, such as voice assistants and chatbots. NLP is expected to continue to advance and play a critical role in enabling computers to understand and interact with humans in more natural and intuitive ways.

a. Agile Software Development and User Story Management

Agile software development approaches are becoming more and more well-liked because they place a strong emphasis on adaptation, flexibility, and continual improvement. The use of user stories to record and rank the needs of stakeholders is one of the tenets of agile development. User stories are succinct, casual summaries of a desired feature or functionality, usually written from the viewpoint of an end user. Making sure that development efforts are kept concentrated on providing value to stakeholders requires effective management of these user stories.

Agile Software Development (ASD) works with requirements that change from time to time during the product life cycle. Also, one of the four items of the agile statement is "customer cooperation on contract negotiations" (Molokken-Ostvold & Furulund, 2007). Effective communication in development projects is therefore an important factor in identifying all requirements and ensuring that the product will be delivered in a timely manner. The aim of communication in development projects to reduce the time needed to develop the project and the cost of changing software (Bhalerao & Ingle, 2010)(Korkala et al., 2006).

Agile testers are part of the development team who collaborate well with entrepreneurs, developers and technicians. They need to understand what is required of stakeholders as well as understand the perspective of developers. An agile test encourages internal and external communication of projects, as it aims to provide value to customers. Agile testing can detect defects very early in the software life cycle and deliver a higher quality product.

A language of domain specific is an important language for describing specific domain problems and it can also be defined as an executable language or programming language. Domain experts use a DSL to express the problem in a way that is understandable and clear, and allows validation at the domain level instead of at the implementation level. DSL focuses on domain knowledge, increasing possibility of re-use at the domain level. It also increases the reliability, readability, and moderation of all participants in the software development team (Deursen et al., 2000).

b. Behaviour Driven Development (BDD)

The goal of BDD is to focus on development based on the value of business, using a language common to business, developers and testers referred to as a language everywhere. A ehavior-Driven Development Scenario consists of the actual scenario, feature title, and associated user story that defines the behavior of the system (Gupta et al., 2023). The basis of BDD is the user story written in the template of as a [role],

- I want to [means]
- So that [ends]
- Given [context],
- When [some event occurs]
- Then [outcome]

This is also used for testing. Tests ensure that the expected behaviour and function of the system meets customer standards. Functional testing scenarios can be written by Gherkin in BDD (Gupta et al., 2023), making it easier for us to use the automated test framework such as Cucumber (Landhäußer & Genaid, 2012). In BDD, the scenario describes the software requirements that are written in DSL. The scenarios written in DSL are used as a communication tool between business analysts, testers and software developers to create better quality in the software produced and involve everyone in the project during the software development process (Rahman & Gao, 2015).

c. Natural Language Processing (NLP) in User Story Management

Neural Language Processing (NLP) is a branch of artificial intelligence and computer science (Gunes & Aydemir, 2020). There is great interest from practitioners and researchers in NLP applications to solve real-world problems, due to the recent phenomenon of collecting unstructured text data and using it in machine learning.

NLP can be used for prioritizing user stories. In a study by (Izmailov et al., 2017), NLP techniques were used to analyze the sentiment of user stories to prioritize them. The study showed that sentiment analysis could be used to prioritize user stories more efficiently than traditional methods.

Estimating the effort required for implementing user stories is another critical aspect of user story management. NLP can be used to estimate the complexity of user stories by analyzing their natural language text. For example, in a study by (López-Martínez et al., 2017)

, NLP techniques were used to extract features from user stories and estimate their complexity. The study showed that NLP techniques could accurately estimate the complexity of user stories.

All advanced natural language-based applications contain three tasks: natural language understanding, natural language processing, and natural language generation (Kumar & Solanki, 2023) as shown in figure (1).

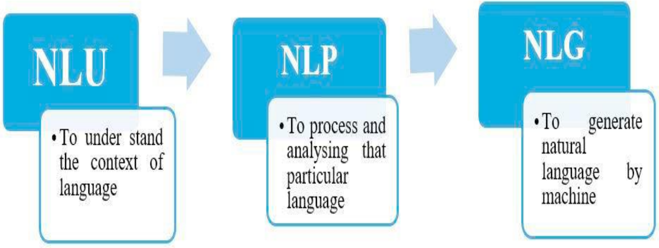

NLU
•To under stand the context of language
NLP
•To process and analysing that particular language
NLG
•To generate natural language by machine

Fig. 1: A Natural Language Processing System using (Kumar & Solanki, 2023)

Tracking user stories throughout the development process is another essential aspect of user story management. NLP can be used to analyze the natural language text of user stories to identify their status. NLP techniques were used to analyze the status of user stories in a software development project. The study showed that NLP techniques could accurately identify the status of user stories.

Methods & Materials

a. Latent Natural Processing Techniques

This study develops a web-based management tool to facilitate communication and collaboration between stakeholders and the agile development team. Where, artificial intelligence has become rapidly developing, and innovative methods can be developed to determine test priorities. Since natural language processing is a branch of artificial intelligence in order to reduce the communication gap between the computer and the human being (Gunes & Aydemir, 2020), (Regression Test cases selection using Natural) mentioned that natural language processing can be applied to obtain high test effectiveness and achieve customer satisfaction. Whereas, most studies use Stanford

CoreNLP, NLTK, or SpaCy for NLP (Almanaseer et al., 2022) and (Kumar & Solanki, 2023). In this study, the SpaCy tool will be chosen as it has a library that supports encoding and is compatible with Python, and this tool can be used to analyze texts as a single unit.

Also, recently, software engineers have started to get more and more interested in using artificial intelligence to test user stories. The Scrum Master takes the brunt of the unseen work of the team to enable them to be more focused and productive on the job. The Scrum Master leads the team and manages the entire business in terms of coaching, team orientation, teaching, and removing obstacles (Ereiz & Music, 2019). This approach makes sense for translating user stories and acceptance criteria to hierarchical tier generation.

b. Proposed Framework for User Story Management

The proposed model is to create communication and collaboration between stakeholders and the development team, where we can use neural programming techniques to analyze the user story to help the development team to obtain clear data. As the user story can be divided into three parts, no more, first the part of the role in the user story and the next part is the action of the user story, and the last part is the benefit of user awareness.

Refereed to (Landhäußer & Genaid, 2012), A proposed model for eliminating nonfunctional requirements in, we designed Framework for User Story Management as shown in figure (2).

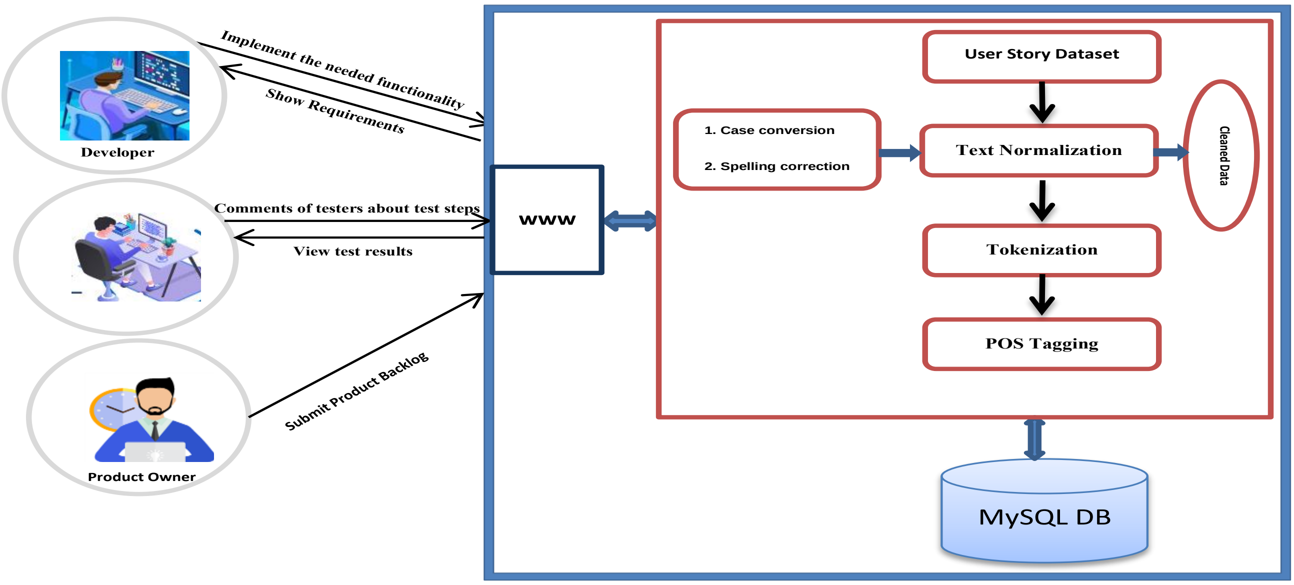


Fig. 2: Proposed Framework for User Story Management

The first step in our model is for the product owner to go to the website to add a backlog containing a list of user stories. For example "As an owner, I want to click on the details, so that Show me all information". After submitting the requirements, the Product Owner and the team hold a planning meeting to decide which task should start first and when each task should finish.

Results and discussion

Manual prioritization and estimating approaches are just two examples of the existing user story management strategies that have some drawbacks. For instance, they frequently rely on team members' subjective assessments, which can bring bias and inconsistent results. Furthermore, as the quantity and complexity of user stories increase, these techniques may be time-consuming and challenging to scale. This study uses a novel user narrative management strategy that makes advantage of latent natural processing (LNP) to overcome these difficulties. This study suggests a novel user narrative management strategy that makes use of latent natural processing (LNP) techniques to overcome these difficulties.

There may be some cases in which unclear specifications lead to increased informal contacts that increase trust among project team members (Pikkarainen et al., 2008). Additionally, the development team along with the owner of the product receives changes and observations until quality assurance is implemented at the end of each race after development (Beck et al., 2001).

NLP researchers focus on the use and understanding of the language so that they can develop appropriate tools to help computer systems deal with and understand natural languages. Often tokenization and text normalization are used together. Therefore, normalization involves removing redundant words, correcting spelling, and changing the case of words. We notice that there are keywords that are repeated in each user story using regular expressions, so it is the first step we must remove those keywords ("as a", "want to" and "so that"). We use Python for this task.

In the next step, we use the SpaCy tool to apply a Part of the Speech (POS) tag. The aim of this step is to build the goal model from the information collected for each user story. After the step of a Part of Speech, the result will store in the database. Thus, the most important factor to increase cooperation and communication in agile software development projects is that all users in the project are able to discuss the same function without misunderstandings, as stated by one agile principle that developers and entrepreneurs must work together to create work programs (Beck et al., 2001).

However, it is recognized that examiners consider that the time and effort they spend to understand product requirements is critical to providing this source code to developers clean and flawless.

The use of NLP techniques in user story management can provide several benefits to agile software development teams. Firstly, it can help to reduce the time and effort required to manage user stories, allowing teams to focus on other critical aspects of software development. Secondly, it can help to ensure that the requirements are consistent and aligned with the overall project goals, reducing the risk of rework and delays. Finally, NLP can help to improve the quality of user stories by identifying and resolving conflicts, ensuring that the requirements are accurate and complete.

This study concludes that (90%) of the respondents agree that cooperation and communication between the development team and stakeholders increases product quality. While respondents who have a neutral opinion (8%) or those with a contrary opinion are (2%). Thus, we can say that cooperation and communication between them during product development will increase product quality.

The overall percentage of participants who agree that the automatic test of the user story is useful for developers is (79%) and the overall percentage of respondents with neutral opinion is (18%). While (3%) who had a different opinion is that the test of the user's story is not necessarily automatic.

From the points we discussed earlier, we can say that most participants feel that cooperation and networking gives stakeholders and development team increased confidence between them. Developers also feel that automatic testing helps reduce time and effort.

Conclusion

The use of NLP technology in user story management has the potential to improve the efficiency and accuracy of different tasks involved in managing user stories. However, more research is needed to explore the use of NLP technology in different aspects of user stories. In this paper, we propose a new approach using NLP techniques to generate clear, misspelled requirements as well as remove redundant words in the user story. The proposed model will help the development team in managing time and obtain customer satisfaction. When customers' requirements are in the hundreds, it is likely that there will be duplication of user stories. One of the challenges facing the development team is the difficulty of knowing the repeated user stories that will be reused and tested. In conclusion, NLP techniques have the potential to significantly improve user story management in agile software development. By automating several aspects of the user story management process, NLP can help to reduce the time and effort required to manage user stories, improve the quality of requirements, and ensure that the requirements are consistent and aligned with the overall project goals. As NLP technology continues to evolve, it is likely that it will become an increasingly important tool for agile software development teams.

Future Works

Since test steps are directly related to testing step applications, the proposed study plan is to evaluate API calls and source code comments. Comments may be submitted to the team of testers to assess whether the recommended test steps are reusable.

References

[1]- Almanaseer, A. M., Alzyadat, W., Muhairat, M., Al-Showarah, S., & Alhroob, A. (2022). A proposed model for eliminating nonfunctional requirements in Agile Methods using natural language processes. 2022 International Conference on Emerging Trends in Computing and Engineering Applications, ETCEA 2022 - Proceedings. https://doi.org/10.1109/ETCEA57049.2022.10009796.

[2]- Beck, K., Beedle, M., Bennekum, A. Van, Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Hunt, A., Jeffries, R., Kern, J., Marick, B., Martin, R. C., Mellor, S., Schwaber, K., Sutherland, J., & Thomas, D. (2001). Manifesto for Agile Software Development. The Agile Alliance.

[3]- Bhalerao, S., & Ingle, M. (2010). Analyzing the Modes of Communication in Agile Practices. 391–395.

[4]- Chowdhary, K. R. (2020). Fundamentals of artificial intelligence. In Fundamentals of Artificial Intelligence. https://doi.org/10.1007/978-81-322-3972-7

[5]- Deursen, A. Van, Klint, P., & Visser, J. (2000). Domain-specific languages: an annotated bibliography. ACM Sigplan Notices, 35(6), 26–36. https://doi.org/10.1145/352029.352035

[6]- Ereiz, Z., & Music, D. (2019). Scrum Without a Scrum Master. 2019 IEEE International Conference on Computer Science and Educational Informatization, CSEI 2019, 325–328. https://doi.org/10.1109/CSEI47661.2019.8938877

[7]- Gunes, T., & Aydemir, F. B. (2020). Automated Goal Model Extraction from User Stories Using NLP. Proceedings of the IEEE International Conference on Requirements Engineering, 2020-Augus, 382–387. https://doi.org/10.1109/RE48521.2020.00052.

[8]- Gupta, A., Poels, G., & Bera, P. (2023). Generating multiple conceptual models from behavior-driven development scenarios. Data and Knowledge Engineering, 145(January), 102141. https://doi.org/10.1016/j.datak.2023.102141.

[9]- Hoda, R., Noble, J., & Marshall, S. (2011). The impact of inadequate customer collaboration on self-organizing Agile teams. Information and Software Technology, 53(5), 521–534. https://doi.org/10.1016/j.infsof.2010.10.009

[10]- Izmailov, A. Y., Liskin, I. V., Lobachevsky, Y. P., Sidorov, S. A., Khoroshenkov, V. K.

[11]- Mironova, A. V., & Luzhnova, E. S. (2017). Simulation of soil-cutting blade wear in an artificial abrasive environment based on the similarity theory. Russian Agricultural Sciences. https://doi.org/10.3103/s1068367417010104.

[12]- Kevin Sungkur, R., & Ramasawmy, M. (2014). Knowledge4Scrum, a novel knowledge management tool for agile distributed teams. Vine, 44(3), 394–419. https://doi.org/10.1108/VINE-12-2013-0068.

[13]- Korkala, M., Abrahamsson, P., & Kyllönen, P. (2006). A case study on the impact of customer communication on defects in Agile software development. Proceedings - AGILE Conference, 2006. https://doi.org/10.1109/AGILE.2006.1.

[14]- Kumar, S., & Solanki, A. (2023). A Natural Language Processing System using CWS Pipeline for Extraction of Linguistic Features. Procedia Computer Science, 218, 1768–1777. https://doi.org/10.1016/j.procs.2023.01.155

[15]- Landhäußer, M., & Genaid, A. (2012). Connecting User Stories and code for test development. 2012 3rd International Workshop on Recommendation Systems for Software Engineering, RSSE 2012 - Proceedings, 33–37. https://doi.org/10.1109/RSSE.2012.6233406.

[16]- López-Martínez, J., Ramírez-Noriega, A., Juárez-Ramírez, R., Licea, G., & Jiménez, S. (2017). User stories complexity estimation using bayesian networks for inexperienced developers. Cluster Computing. https://doi.org/10.1007/s10586-017-0996-z.

[17]- Moløkken-Østvold, K., & Furulund, K. M. (2007). The relationship between customer collaboration and software project overruns. Proceedings - AGILE 2007, 72–83. https://doi.org/10.1109/AGILE.2007.57.

[18]- Pikkarainen, M., Haikara, J., Salo, O., Abrahamsson, P., & Still, J. (2008). The impact of agile practices on communication in software development. Empirical Software Engineering. https://doi.org/10.1007/s10664-008-9065-9.

[19]- Rahman, M., & Gao, J. (2015). A reusable automated acceptance testing architecture for microservices in behavior-driven development. Proceedings - 9th IEEE International Symposium on Service-Oriented System Engineering, IEEE SOSE 2015, 30, 321–325. https://doi.org/10.1109/SOSE.2015.55.

[20]- Robey, D., Welke, R., & Turk, D. (2001). Traditional , Iterative , and Component-Based Development : A Social Analysis of Software Development Paradigms. Information Technology and Management, 2, 53–70. https://doi.org/10.1023/A:1009982704160.