Wadi AlShatti University Journal for
Pure and Applied Science

مجلة جامعة وادي الشاطئ
للعلوم البحتة والتطبيقية

Volume 1, No. 1, July-December 2023

المجلد 1، الاصدار 1، يوليو - ديسمبر 2023

COMPUTER SCIENCE

A Kanban Framework as a Systematic Approach for Reducing the Human Tendency to Resist Change

Fatma H. Fazzani [1,*], Khamisa A.Yousef [2]

[1] Department of Computer Science Faculty of Education University of Omar El-Mokhtar, AL-Beda, Libya
[2] Department Computer Science Faculty of Science University of Tobruk, Tobruk, Libya

ARTICLE HISTORY

Received 17 August 2023
Revised 26 September 2023
Accepted 28 September 2023
Online 29 September 2023

KEYWORDS

Requirements Engineering;
Human aspects;
Agile Methodologies;
Kanban Framework;
UML Models.

ABSTRACT

The real success of developing any system is that the customer is satisfied and agrees with the system requirements specifications that were extracted from the requirements engineering stage. Requirements Engineering (RE)-related activities require high collaboration between various roles in software engineering (SE), such as requirements engineers, stakeholders, developers, etc. The development of requirements engineering methodologies has been of great importance in overcoming the problem of not accepting change for a customer when developing the system. Among these advanced methodologies the agile methodology that involves constant collaboration with stakeholders and continuous improvement at every stage the kanban process is one of agile methodologies that structured to address the human tendency to resist change. The main objective of this paper is to provide a kanban framework as a systematic approach that helps to reduce the human tendency to resist change.

مخطط كانبان كطريقة منهجية منظمة لتقليل الميول البشري لمقاومة التغيير

فاطمة حمد محمد الفزاني [1،*] ، خميسة عوض يوسف [2]

الكلمات المفتاحية

هندسة المتطلبات
الجوانب الانسانية
المنهجبات الرشيقة
مخطط كانبان
نماذج لغة النمذجة الموحدة

الملخص

المفتاح الحقيقي لنجاح تطوير أي نظام هو أن يكون العميل راضيا وموافقا على مواصفات متطلبات النظام التي تم استخراجها من مرحلة هندسة المتطلبات. تتطلب الأنشطة المتعلقة بهندسة المتطلبات (RE) تعاونًا عاليًا بين الأدوار المختلفة في هندسة البرمجيات (SE)، مثل مهندسي المتطلبات وأصحاب المصلحة والمطورين وما إلى ذلك. لذا فأن لتطوير منهجيات هندسة المتطلبات أهمية كبيرة في التغلب على مشكلة عدم قبول التغيير للعميل عند تطوير النظام. ومن بين هذه المنهجيات المتقدمة، تعد المنهجية الرشيقة التي تتضمن التعاون المستمر مع أصحاب المصلحة والتحسين المستمر في كل مرحلة، تعد عملية كانبان إحدى المنهجيات الرشيقة التي تم تصميمها لمعالجة الميل البشري لمقاومة التغيير. الهدف الرئيسي من هذه الورقة هو استخدام مخطط كانبان بطريقة حديثة ومنظمة لتساعد المطور ي يساعد تقليل الميل البشري لمقاومة التغيير أثناء تطوير النظام.

Introduction

The stage of preparing requirements for systems development is considered the most difficult and important stage, due to the difficulty of satisfying the customer and his lack of acceptance of developing and changing his system in the future. Therefore, developers need more advanced methodologies and tools to use at this advanced stage of systems development. Requirements engineering is the most important stage in software engineering, as the analyst collects requirements from a group of different sources, these sources help developers know what the system is so that, in turn, how the system can work in the future [1]. Requirements engineering is considered one of the most important and most difficult stages in software development, because it has a significant impact later on every stage of the software development process [2]. Requirement engineering is a comprehensive process for gathering requirements, providing documentation, and monitoring requirements. Gathering requirements serves as the starting point for all other requirement reengineering activities. The project's requirements, features, and functionalities are discussed in a meeting with stakeholders and developers. The following task involves analyzing the demands and engaging in negotiations with interested parties to resolve any disputes. Furthermore, these specifications are written down, created, and presented as text or via graphical models. Last but not least, managing the requirements and ensuring that they are being followed correctly are crucial [3]. Since, requirements engineering depends primarily on the performance of the development team, the specialized team in the RE phase, this team has the most important responsibilities. In order for this team to be able to accomplish the development task, there must be an appropriate environment for cooperation among its members and to benefit from the skills of each of its members. Emphasis must also be placed on the importance of understanding the role of the term “human aspects,” which has a significant impact on improving the work between

*Corresponding author: fatma.fazzany@omu.edu.ly

software developers, requirements engineers, to work effectively together and with their stakeholders [4].

Because, the importance of the impact of different human aspects on various requirements engineering activities. Since software engineering is considered a socio-technical activity that involves all the people involved in developing the system, attention to the human aspects of all the development team will reflect positively on their performance and achievements [5]. As we have noted, the role of the human side in requirements engineering activities has become clear and important with the continuous development of requirements engineering methodologies, which has resulted in the emergence of agile methodologies.

During the past decade, the use of agile software development has greatly increased, replacing the use of traditional life cycle models [6]. The agile methodology aims to deliver the product more quickly by placing a high value on customer satisfaction. Agile is an ongoing iterative process where customers and developers continuously communicate throughout the entire software development life cycle. Agile is accommodating of requirements changes throughout the life cycle. Due to this, the use of Agile in software systems has increased [7].

Kanban is considered one of the important agile methodologies that have made progress in software management and development. Kanban also has an impact on improving the means of interaction and cooperation between all the different members of the development team, which helps in improving their tasks and managing the time and cost of development [8]. The concept of the kanban appeared in 1940 by the Japanese automobile company Toyota. It focuses on delivering work on time. It integrates the customer into the development process so that he sees the completion of the required and desired work, which helps in reducing differences and wasting time and cost in completing the required work [9].

This paper aims to shed light on overcoming the human feeling of development and change by developing a developed mechanism for using kanban in a way that helps us solve this problem because of its impact on the future of work development, which results in reduced cost and time. The paper is divided as follows: Section 2 provides literature review, Section 3 introduces our work is represented in the proposed kanban framework, Section 4 presents the conclusion.

Literature Review

Actually, the focus has been important become on human tendencies and desires in the development process of any system or product, as it has become highly dependent on focusing on the extent to which the system stakeholder's accept what the new system will be like in the future, and it is considered the main factor for the success of any system [10]. Considerate as that requirements engineering is the initial stage of software development engineering and the main weight of the human role in participating in the development of the system rests on it. At this stage, stakeholders and developers work closely together [11]. Many of researches presented human tendency to resist change. In [12] they proposed a review and listed all the tools, steps and templates that have an effect of human aspects on the requirements engineering process. Also, [13] they presented hierarchical classification of errors according to the cognitive level at which people operates when performing various tasks. They classified into skill-based slips and lapses, and rule-based and knowledge-based mistakes. Moreover, [14] they defined human aspects in software engineering as human-related aspects that can become make-or-break issues in software projects by investigate various human aspects in various SE contexts – such as personality, emotions, motivation, gender, culture, communication issues, human errors, attitude, team climate and others – and identified their impact on SE, for better or worse. [15] they conducted a study on the impact of the humanitarian aspects on more than one institution in more than one different country, in which it focused on the importance of the impact of the humanitarian aspects and their advantages and disadvantages in the process of developing systems, but the importance of this study was on the extent to which these advantages and disadvantages influence the process of transforming traditional systems into Agile only without establishing a clear mechanism to integrate these systems with the human aspects of developing these systems.

As is clear, most previous research does not provide a clear and simplified mechanism to reduce the innate idea of “human resistance to change and development.” Considering that the requirements engineering stage is the basic stage for building a convenient and simple environment that brings together all the people involved in developing the system with the main developers, in this work we have tried to present a simple and easy proposal to help everyone reach the final form of the system requirements specification model in an acceptable manner, and it was agreed upon by all system owners.

A proposed kanban framework

The proposed kanban framework as shown in figure 1 enable teams to visualize and manage their workflows effectively and it suggests the use of the physical board. In proposed kanban board, we will divide the board into three main columns and then split it into a few sub columns that show different work stages, the three main columns are following as:

1. Preparing for work
2. Works in progress:
   - In implementing - In validating
3. Delivering system work

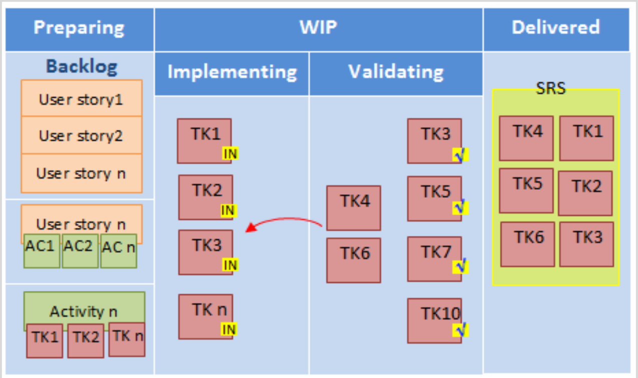

Figure 1: A proposed kanban framework

Preparing for work

The key to overcoming human tendency to resist change is to include all people involved in system development during the kanban board discussion. In the initial stages of requirements engineering the stage of eliciting what he needs whether the customer, user or any individual of the stakeholders, at the initial step we have advised by using user story mapping as tool to represent their needs. Each user story is associated with one user, customer or stakeholders, after collecting all their stories we will prioritize and document into proposed user's stories document. Table 1 can be used to document these information's.

At the end of this stage, the last column of the previous table 1 will be collected and considered as a summary of the backlog, which is comprehensive of what the people of the system need and the specifications of the new system. These backlogs will be in the form of tasks to be accomplished and moved to the next stage for the purpose of implementation by the development team.

Works in progress

This stage is the most important stage in our work which the development team will work together in of both two directions, first to implement these needs according to what the owners of the system see, and then test each task that was implemented with the customer to ensure that his needs are correct. The main focus in both the two under phases that fall under this phase is by using the Actor model and term "message-passing communication" between members of development team which helps to facilitate the progress of communication between them as shown in figure 3.

In implementing

Actually every actor represents a member of the development team and the messages sent between them that is every task of your mind that has not been extracted from the previous stage, the task is in the form of implementation from the point of view Each of the team members is replaced by each other through messages where each actor reviews the received task and re -sends it in the event of amendment or sends admission to it. In figure 2 an example of the process of interaction and communication between team members is shown, so that the tasks that are being worked on are exchanged in the form of messages that they exchange among themselves.

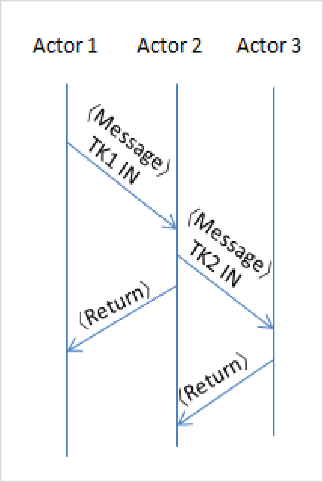

Figure 2: An actor message passing model

At the end of this step, all team members have completed the work on implementing all the required needs according to the required work mechanism and the customer's vision. In the end, all the tasks that are processed will be placed in a list, and each task will be marked in a card bearing a label (IN) to classify it as having been processed within the board.

In validating

The purpose of this step is to ensure that all the tasks that were accomplished in the previous stage will be approved by the decision makers who have a relationship with the system. In this step, we suggest using use case diagram to test system states and ensure the required functionality of the new system. The validation steps can be as follows:

- Design a use case diagram based on list of tasks that extracted from the previous stage, this use case diagram can summarize the details of your system's users and their interactions with the system as shown in Figure 3.

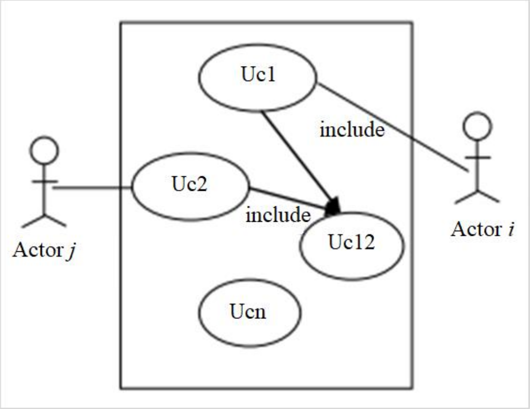

Figure 3: Use case diagram

- Draw a paper prototype for each use case diagram this paper prototype can be used to represent a use case action in a visual representation to increase the understanding of stakeholders. Also, this paper used to show that their needs are cleared understood by the team developers.
- Distribute the papers to the system stakeholder's in order for them to verify that all their needs and desires have been fulfilled and that the evolutionary change has proceeded in accordance with their desires and needs, as each task that has been approved is placed in a card and this card is marked with a sign (✓) meaning that it is in accordance with the required and that any task has a difference will return to the implementation column for the purpose of adjusting it as required.

Delivering system work

This step is considered the conclusion of the kanban Board work, where all the requirements and needs are documented according to what the owners of the system desire, and there is no doubt that the valuable work must be documented according to an official and approved document, which is the SRS document (System Requirements Specifications).

The SRS document focuses on what the system needs to do and how it performs. It lays the important foundation so that all system stakeholders understand the most important details. It also specifies the behaviors, functionality, and capabilities required of the system, along with any potential limitations. Functional and non-functional requirements are included. Design suggestions and information outside of client's requirements are not included [16]. Because of, in accordance with the need to overcome human inclinations to embrace change as well as to ensure traceability that consent is received from all necessary stakeholders, demonstrating that they clearly understand the requirements of the system and that everyone agrees. In a sense, the SRS acts as an insurance policy that either party can fall back on in case of uncertainty. In this work we use the Use case specification that one of SRS document standard as shown in Table 2.

Table 1: User's stories document

Priority	Type of User	Title	Set of activities	Set of tasks	Backlog
US #number	Who is responsible it	As a [*type of user*], I want [*a goal or objective*], So that [*customer benefit or value*].	Features of the map.	Features of the activity.	Features of the prioritize tasks.

Table 2: Use case specification

Use Case Name:	
Actor(s):	
Summary Description	
Priority:	
Post-Conditions(s):	
Basic Path:	
Alternative Paths:	
Business Rules:	
Non-Functional Requirements:	

Conclusion

In this paper, we presented a mechanism to extract the needs and desires of the customer, the user, and stakeholders in a manner that aims to reduce humanity's fear of change in development. In this work, we have improved the kanban framework in a way that makes it appropriate the principles of requirements engineering, such as traceability and reliability to ensure that everything requested by the customer is accomplished by using same of tool such as a user story map that helps to improve your understanding of your customers and to prioritize work. a Use case diagram is used to visualize the different types of roles in a system and how those roles interact with the system. A paper prototype technique is used as a validation technique to present the functionalities of the system to its stakeholders. At the end of this work we used standard system requirements speciation to document all needs of stakeholder's system. As future work, this work needs more of development to improve its functionality, and it can also be made in the form of a software tool to simplify its use.

Author Contributions: "Conceptualization, methodology, Fazzani; writing—original draft preparation, review and editing, Fazzani and Yousef. All authors have read and agreed to the published version of the manuscript."
Funding: "This research received no external funding."
Data Availability Statement: "The data are available at request."
Acknowledgments: "The authors would like to express their appreciation to Dr. Mohamed A. Hagal (faculty member Benghazi University) for his encourage, feedback and comments."
Conflicts of Interest: "The authors declare no conflict of interest."

References

[1] **Pandey, D., & Pandey, V.** (2012). Impor**tance of requirement management**: A requirement engineering concern. *International Journal of Research and Development - A Management Review (IJRDMR)***, 1:** 2319–5479.

[2] Chakraborty, A., Baowaly, M.U., Arefn, A., & Bahar, A. N. (2012). The role of requirement engineering in software development life cycle. *Journal of Emerging Trends in Computing and Information Sciences*, 3: 723–729.

[3] Chemuturi, M. (2012) Requirements Engineering and Management for Software Development Projects. pp, 288. Kindle, Springer 2013.

[4] Wick, T. (1999). The importance of team skills for software development, Faculty of Graduated Studies, Simon Fraser University, Canada.

[5] Hazzan, O., & Tomayko, J. (2004). Human aspects of software engineering: The case of extreme programming. *in Extreme Programming and Agile Processes in Software Engineering*. 303–311. Springer Berlin Heidelberg.

[6] Babar, M., Brown, A. & Mistrík, I. (2013). Agile Software Architecture: Aligning Agile Processes and Software Architectures, Newnes.

[7] Inayat, I., Salim, S., Marczak, S., Daneva, M., & Shamshirband, S. (2015). A systematic literature review on agile requirements engineering practices and challenges. *Computers in Human Behavior*, *51* :915-929.

[8] Omar, M., Alaidaros, H., & Romli, R. (2020). An Improved Software Project Monitoring Task Model of Agile Kanban Method: A Practitioners' Perspective. *International Journal on Advanced Science Engineering and Information Technology*, *10*(2) :548.

[9] Banijamali, A., Ahmad, M. O., Simil, J., Oivo, M., & Liukkunen, K. (2017). Empirical investigation of Scrumban in global software development, in: Hammoudi, S. P. L., Selic B., & Desfray P. (Ed.), Model-Driven Engineering and Software Development. (pp. 229–248) MODELSWARD 2016. Communications in Computer and Information Science. Rome, Italy: Springer, Cham.

[10] John, M. Maurer, F., & Tessem, B. (2005). Human and social factors of software engineering: workshop summary. *ACM SIGSOFT Software Engineering Notes*, *30*(4): 1–6.

[11] Gregoriades, A. Jae-Eun, S., & Sutcliffe, A. (2004). Human-centred requirements engineering. *in Proceedings 12th IEEE International Requirements Engineering Conference*. 154–163. September 2004, Kyoto, Japan.

[12] Hidellaarachchi, D., Grundy, J., Hoda, R., & Madamp, K. (2021). The Effects of Human Aspects on the Requirements Engineering Process: A Systematic Literature Review. *IEEE Transactions On Software Engineering*, 1(1): 99.

[13] Viller, S, Bowers, J., & Rodden, T. (1999). Human factors in requirements engineering: A survey of human sciences literature relevant to the improvement of dependable systems development processes. *Interacting with Computers*, *11*(6): 665-698.

[14] Hidellaarachchi, D. Grundy, J, Hoda, R., & Mueller, I. (2022). The Influence Of Human Aspects On Requirements Engineering-Related Activities: Software Practitioners Perspectiv. *ACM Transactions on Software Engineering and Methodology*, *1*(1): 37-45.

[15] Gandomani, T., Zulzalil, H., AbdulGhani, A., AbuBakar, S., & Sharif, K. (2014). How Human Aspects Impress Agile Software Development Transition and Adoption. *International Journal of Software Engineering and Its Applications*, *8*(1): 129-148.

[16] Coleman, D. (1998). A Use Case Template: Draft for Discussion. *Computer Science*.