

The impact of user involvement in software development process

Nouf Bin Saif, Mashael Almohawes, Nor Shahida Mohd Jamail

College of Computer and Information Sciences, Prince Sultan University, Riyadh, Saudi Arabia

Article Info

Article history:

Received Apr 14, 2020

Revised Jul 15, 2020

Accepted Aug 1, 2020

Keywords:

Extreme programming

Scrum

V-Model

Waterfall

ABSTRACT

In software development process, user can take part in any phase of the process, depending on what model is being applied. Lack of user involvement can result in a poorly designed solution, or even a solution that conflicts with user's needs. This review paper presents the impact of user involvement in software development process. In this study, different software development processes will be reviewed, show where the user usually gets involved in different models such as: structural (waterfall, V-model) and incremental (scrum-extreme programming XP). As each model differs from the other, each of them has a different perspective of where user should take part and where they should not. This can be an asset that helps project managers, and leaders to develop suitable strategies to follow in their projects.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Corresponding Author:

Nouf Bin Saif

College of Computer and Information Sciences

Riyadh, Prince Sultan University

66833 Rafha Street, Riyadh 11586 Saudi Arabia

Email: nouf08@gmail.com

1. INTRODUCTION

Several definitions were assigned to the term Software Engineering, mainly, all definitions cover both technical and management aspects of software development life cycle. A major part of software engineering is choosing a proper and suitable model to be applied and throughout the software development life cycle. An example of those models is the software iterative and incremental models, where requirements get prioritized, and the requirement with the highest priority always comes first. Also, they support the efficient utilization of resources with well-planned scheduling to come up with an early deployment [1]. On the other hand, structured processes rely on previously defined steps including detailed documentation and solid management [2]. Success in something that depends on many activities to be completed in the right way so that it can be achieved. Software development requires some effort to be put in place, where it calls for the use of several resources in the technical field, and even the stakeholders, who will have to do more through giving different pieces of mind inputs so that some success can be experienced [3]. The impact of involving users in the process of software development requires the software developers to play their part as well as the users who will appreciate the developed product.

The main area of interest comes when the software is required to work successfully, where it needs to fulfill the demands and needs of the users. This becomes an important point in the development process [4]. However, through the combination of various ideas, then a better way needs to be taken into consideration where coordination and collaboration is required to bring the results in terms of developing successful software. The potential of success of a software development project or any project that affects

someone increases when users that are affected know about it through sharing and communication, therefore, when users share and get to the point of use, it will bring very affordable and more time saving solutions. User involvement throughout all phases of software development process is essential cause of project's success and should be encouraged [5]. Even though the user can misunderstand the software, or the technology itself which leads to an inefficient and ineffective work [6]. On the other hand, software engineers face different obstacles associated with analyzing user's demands and detailed specifications, in addition to requirements evolution and new technologies adoption [7].

This paper presents The Impact of user involvement in the software development process by reviewing the negative and positive perspectives of user's participation in software development process based on prior researches. Also, user involvement framework has been designed to show which software development model involves the users on each phase of the software engineering process. Thus, a result is shown to evaluate the best and suitable model for user involvement. Finally, the paper concludes with future work. The paper aims to looking at the effects that can be experienced when users take part in the software development process.

2. LITRATURE REVIEW

The researchers [8] have strongly demonstrated where the user has reported and given information about how users try to get their chance to take part in the development process. It leads to societal development in the long run. In addition, using collaborative communication as a way of looking at things brings an approach which mainly entails giving the best in terms of decisions with given assumptions that the users have an influence on what gets decided about. On a larger extent, user's involvement was linked with quality. Bano and Zowghi [9] explain that applying and utilizing the collaborative approach helps to bring unity among different users, which leads to the sharing of ideas leading to continuous growth. The method the authors proposed was linking user's involvement with success, and how it makes the user satisfaction and acceptance possible and reachable. It contains a total of four stages of data collection throughout the development of a product and utilized user-involvement while collecting the information for a customer relationship management system (CRM) and Portal project. Those four stages contained face-to-face meetings, workshops, phone calls, and several workshops, while taking into consideration that it was based on the waterfall model. User's involvement during implementation phase wasn't very much needed, instead, it caused so many unnecessary changes to workflows and processes, but then it got improved after the incorporation of some agile practices.

A methodology named CloudTeam was built to support collaborative development. It helps aligning software with preferences of the user in the software development process in order to avoid unnecessary costs and accelerate the development process by involving end-users at every phase of the software development process. The methodology is based on two existing models: the V-model and FITMAN verification and validation (v&v) model. The proposed method stresses the brainstorming phase and the opinions of customers, and encourages cooperation between customers and software teams. It focused on involving the user in software development at the early stages. Involving the user helps detecting and identifying flaws as early as possible which as a result effects the cost positively. The methodology consists of 7 steps which are: brainstorming, as it makes a connection between development teams and end-users. Secondly, gathering feedback from customer group led by a CloudTeam representative, which eventually leads to the complete software requirements specification document as well as user acceptance testing plan. Then, Functional and Non-functional requirements get defined as user stories. Then comes the development phase [10].

The researchers have studied the features of the software development life cycle with the main purpose of identifying potential bottlenecks, which hinder the development process. By performing a case study at one of Ericsson AB's development units. The findings follow a qualitative interview analysis using a streamline development system for one cross-function team. Using lean as a theoretical basis. The researchers found some of the possible obstacles that can raise the lead-time of a software project. The researchers arrive at results that Long contact chains, evaluation requirements, feedback, and inadequate documentation are some of the possible obstacles that can raise the lead-time of a software project [11].

3. RESEARCH METHOD

3.1. Positive impacts of user involvement

3.1.1 User Involvement to identify the challenges in the system

During software's development life cycle, the presence of users helps in creating an environment where they will be in a position to observe the existing challenges associated with the system construction and giving much clearer requirements for the needed system. The main challenge is when software engineers

are not familiar with business needs; this as a result will make them have a poor understanding of the hidden details associated with business needs. Therefore, user's involvement can help avoid those barriers that can happen due to lack of understanding of user's environment [12]. Having efficient communication with the end-user can be helpful to fully understand their hidden needs. Despite of the fact that software engineers are designing the solution and implementing it, users are experts in their given areas of work, and have a better understanding of what they require; software engineers can derive information by having a good communication channel and way of exchanging information with them on areas of interest [13]. Involving the user in software project leads to fully understanding the business process they are looking for, which as a result helps in having a well planned project as well as enhancing team experience in deriving and designing business processes [14].

3.1.2 User involvement impact on final result

Involving the user is a great method that leads to achievable and understandable system objectives which when developed, provides a positive influence on user's environment and provides the solution as requested. Software engineers need to have direct access to user's environment so that they can be in a position to evaluate or prioritize requirements and be able to design something that creates a noticeable, positive difference. In addition, it is worth mentioning that users' involvement is considered as an effortless method that helps to design, implement and validate the functionalities of a software system. Through the application of user involvement, a reliable way of realizing user's needs, having them validated will be offered including what needs to be taken into consideration in the existing system [15].

3.1.3 Transparency of requirements

When user is involved, they bring out the best in terms of realizing the different ways of having the knowledge and perception about the current domain. Especially when communication is needed to come up with the problems faced and their solutions [16]. Software engineers might not be familiar with natural language user is communicating with. However, the practice of communication with end-users can help them to get the expertise of understanding users' intentions, way of explaining problems and analyzing them which finally can lead to a good design and quality. Having direct communication with end-users with efficient usage of natural language and vagueness avoidance can boost the development efficiency.

3.2. Negative impacts of user involvement

3.2.1 Sustainability of user–software engineer relationship

Unsuccessful relationships with end-users can be caused by various events that occurred during software development life cycle. Having a constant and direct communication with end users requires both emotional and physical energies from software engineer, which sometimes can worsen the relationship if no efficient and professional way of handling communication was there. One of the main causes of a bad user–software engineer relationship is the situation where software engineer is in negotiation and disagreement while user is requesting for an additional feature or modification, software engineer might disagree to include it either design or technical reasons. In many circumstances, the blame game begins when the final result is achieved. As a result, user was involved but caused complicating the process and brought negative impact to the entire project. Yes, delivering the required product is important, however, having a solid, long-term relationship with users (who can be customers as well) is essential.

3.2.2 Requires user's commitment

The effort needed from user sometimes may not be easy or even impossible to fulfill when it comes to discussing the details during the development process, which can be a long duration. Users might not have enough time to go through such details or might be only interested to see the final product only while relying on requirements gathered at the beginning, without having to discuss every detail. Not having a prompt user response, or no response at all can be very challenging for software engineers. This can give a negative impact on the project by causing it to be on-hold, or get delayed, since it is pending with the end-user to give feedback or answer an inquiry. It might even lead to project failure. The requirements given by the users need to be evaluated, discussed in detail. During development, Software Engineers might come to a point where they need end-user to give some explanation about a specific part of the system and its possible scenarios. Therefore, end-user is required to dedicate some time and energy to give meaningful feedback to Software Engineers [9]. This is one of the real challenges of user involvement. The importance of the user's required effort and input cannot be underestimated. Comparison between negative and positive impact of user involvement in software development as shown in Table 1.

Table 1. Comparison between negative and positive impact of user involvement in software development

Positive Impacts of User Involvement	Negative Impacts of User Involvement
Enables users to recognize the challenges in the system.	The difficulties to maintain good relationship.
Gives developers the chance to have fast and comfortable methodologies.	Effort that need from customer
Enables developers of systems to understand the language of the user.	

3.3. User involvement framework

The involvement of user in Waterfall model occurs in requirements and design, where modifications are requested by the user in the form of change requests or as a result of deficiencies discovered after using the software [17]. In addition, waterfall model cannot easily adopt and react to changes, like some other incremental models [18]. While in V-model, the user is involved in the testing phase, where they do the acceptance test and give a feedback about the software [19, 20]. On the other hand, the agile approaches emphasize the simultaneous execution of a project's typically successive activities and the continuous collaboration of participants. One of agile's methods is extreme programming (XP), which shows the need to have the customer always available on-site; as it accelerates the collection of requirements and development. An empirical result reported that the percentage of customers that were present with the development team in XP projects was (83%), while only (21%) of customer work effort was required to assist the development team [21]. Another type of agile methods is Scrum, it focuses on addressing rising user requirements and prioritizing them efficiently [22]. User is never a part of implementation or maintenance phases [23]. Research papers have been taken into consideration to present a comprehensive comparison that is summarized in Table 1, where every model is compared to the other models based on every phase in software development life cycle, from a user involvement perspective. Table 2 show the comparison shows that both structural models have minimum user involvement, compared to incremental models which involve the user in much more stages, especially extreme programming, user is takes part in every stage of the process.

Table 2. Comparison between SDLC phases and software development models in user involvement

Author	Ref No.	Paper title	User Involvement				
			Technique/ Phase	Structural Waterfall	V-Model	Incremental Scrum Extreme programming	
S. M. Butte and W. F. W. Ahmad	[16]	An overview of Software Models with Regards to the users Involvement	Requirement engineering	✓	✓	✓	✓
S. Mathur and S. Malik P. Permana	[15]	Advancements in the V-model Scrum Method	Design Implementation	✓		✓	✓
	[19]	Implementation in a Software Development Project Management					
M. Shahriar, B. Nikkhahan and S. Sohrabi	[17]	Challenges of user Involvement in Extreme Programming Projects	Testing		✓	✓	✓
R. Yadav	[18]	Improvement in the V-Model	Maintenance	✓	✓		✓

4. RESULTS AND DISCUSSION

Software development is a field where people come from different backgrounds and work towards achieving the same goal. Therefore, it is suggested to have all parties involved and take responsibility to collaborate, share knowledge and expertise in a way that helps in developing the required software. Successful software development requires teamwork, communication skills, and the ability to acquire information from different sources. It includes continuous support and cooperation all parties to identify, and even make modifications and adjustment, until the objective is finally reached. We can satisfy users by having them involved and aware of the details. The solution a specific software to-be will provide mainly depends on the user's interests, input, and existing problem domain. Software development process requires sharing, and unifying ideas and sometimes prioritizing them; hence, this brings out the need for users to be involved during the software development life cycle. User's feedbacks are assets that are always valuable, and having them as a part of the software development process can reduce the time spent on adjustments, modifications and corrections that is spent after users' testing sessions, as those modifications can be done at earlier stages, where the impact of change is minimum.

The Pie chart in Figure 1 and Table 3 shows relative contributions of user involvement in different software development phases across four models, waterfall, V-model, Scrum [23, 24] and Extreme programming [25]. It is clear that Requirement Engineering phase represents the largest portion of where user is participating across several models whereas Implementation is undoubtedly the smallest.

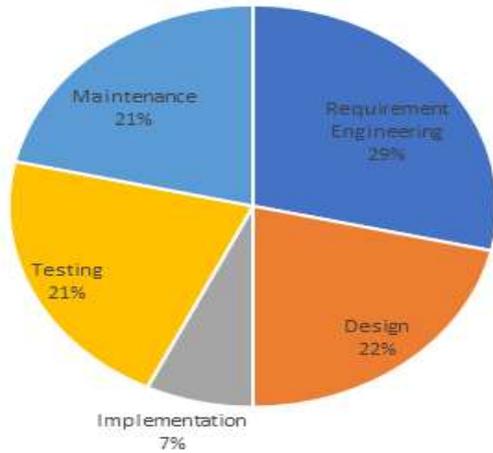


Figure 1. User involvement in SE models

Table 3. Number of phases with User involvement in software development models

Software development Models	Number of phases
V-model	Threere phases
Waterfall Model	Threere phases
V-model	Threere phases
Extreme Programming	All SDLC phases (five phases)

The phase that requires maximum user input is requirements engineering. It is where different ideas are discussed, objectives are set, and detailed requirements are gathered, therefore, it is the source of truth for the other phases and represents what the user is looking for. It also has the highest percentage of user participation (29%). While in maintenance (21%), design (22%), and testing (21%) the percentage of user involvement is almost the same, user involvement can be useful at those stages. However, it is not an essential part, as they can be passed without having the user a part of them. Lastly, implementation phase, where the system that was planned is getting developed. Implementation phase requires solid technical expertise especially software development skills and programming, which is an activity a user is never expected to be involved in, due to their lack of technical knowledge. Even if the user had the technical knowledge, by any chance, it is still not a part where a user is expected to get involved in. As a reflection of the above explanation, implementation has the lowest percentage of user involvement which is (7%).

5. CONCLUSION

This paper presents a review about the impact of user involvement in software development process. Different models have been discussed that present different forms of user involvement in planning, designing, implementing, testing or maintaining a software product while looking at the various ways in which customer relations and engagement can be handled. Understanding how the obstacles come along and how they can be coped is one of the key challenges. One solution to overcome those challenges is to involve the user and have them close to development team. This makes both user and software engineer have direct communication channel and prompt feedbacks are provided from one another. Currently, users are usually highly engaged in the requirements phase (29%), while not having too much user involvement in implementation phase (7%). This the need for user's engagement at early stages of every development project which helps sharing ideas and clarifying vagueness, as well as mid and final stages. As a future work, we will be considering measuring user's involvement impact on a project's time and cost.

ACKNOWLEDGEMENTS

This work was partially supported by the Artificial Intelligence Data Analytics Lab (AIDA), Prince Sultan University, Riyadh, Saudi Arabia.

REFERENCES

- [1] Petersen, Kai, Robert Feldt, Shahid Mujtaba, and Michael Mattsson. "Systematic mapping studies in software engineering." In *12th International Conference on Evaluation and Assessment in Software Engineering (EASE)*, vol. 12, pp. 1-10, 2008.
- [2] Estler, Hans-Christian, Martin Nordio, Carlo A. Furia, Bertrand Meyer, and Johannes Schneider. "Agile vs. structured distributed software development: A case study." *Empirical Software Engineering*, vol. 19, no. 5, pp.1197-1224, 2014.
- [3] U. Abelein and B. Paech, "Understanding the Influence of User Participation and Involvement on System Success- a Systematic Mapping Study," *Empirical Software Engineering*, vol. 20, no. 1, pp. 28-81, Dec. 2013.
- [4] P. Haake, S. Morana, S. Schacht, L. Zhou-Hartmann, and A. Maedche, "Designing an End User Participation and Involvement Assistant for Continuous IS Development," in *Thirty Seventh International Conference on Information Systems*, pp.1-14, 2016.
- [5] Mandal and S. C. Pal, "Investigating and Analysing the Desired Characteristics of Software Development Lifecycle (SDLC) Models," *International Journal of Software Engineering Research and Practices*. vol. 2, no. vol. 4, pp. 9-15, 2012.
- [6] M. Sudarma and N. W. Sri Ariyani, "Transformation of Information Technology Based on Human Computer Interaction Concept," *International Journal of Informatics and Communication Technology (IJ-ICT)*, vol. 2, no. 3, pp. 137 -143, 2013.
- [7] S. Pathak and P. Saxena., "Hybrid Methodology Involving Scrum and Waterfall Model towards the Software Project Development in Academic Knowledge Centers," *International Journal of Evaluation and Research in Education (IJERE)*, vol. 1, no. 1, p. 25-32, 2012.
- [8] K. Grønbaek, J. Grudin, S. Bødker, and L. J. Bannon, "Improving Conditions for Cooperative System Design: shifting from a product to a process focus," in *Participatory Design*, vol. 19, no. 331. Pp. 1-17, 2017.
- [9] Bano, Muneera, Didar Zowghi, and Francesca da Rimini. "User satisfaction and system success: an empirical exploration of user involvement in software development." *Empirical Software Engineering*, vol. 22, no. 5, pp. 2339-2372, 2017.
- [10] Zowghi, Didar. "" Affects" of User Involvement in Software Development." In *2018 1st International Workshop on Affective Computing for Requirements Engineering (AffectRE), IEEE*, pp. 13-13, 2018.
- [11] Katumba, Brian, and Andrei Antanovich. "Bottlenecks in the Development Life Cycle of a Feature-A case study conducted at Ericsson AB." In *7th Annual International Conference on Computing and ICT Research*, pp. 472-490. 2011.
- [12] Kushniruk, Andre, and Christian Nøhr. "Participatory design, user involvement and health IT evaluation." *Stud Health Technol Inform*, vol. 222, pp. 139-151, 2016.
- [13] Wing, Jeanette, Theo Andrew, and Doncho Petkov. "The changing nature of user involvement in information system development projects." In *2017 Conference on Information Communication Technology and Society (ICTAS), IEEE*, pp. 1-6, 2017.
- [14] M. Bhardwaj and A. Rana, "Key Software Metrics and its Impact on each other for Software Development Projects," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 6, no. 1, pp. 242-248, 2016.
- [15] Alvertis, S. Koussouris, D. Papaspyros, E. Arvanitakis, S. Mouzakitis, S. Franken, S. Kolvenbach, and W. Prinz, "User Involvement in Software Development Processes," *Procedia Computer Science*, vol. 97, pp. 73-83, 2016.
- [16] Lempp, H., S. Abayneh, D. Gurung, L. Kola, J. Abdulmalik, Sara Evans-Lacko, M. Semrau, A. Alem, G. Thornicroft, and C. Hanlon. "Service user and caregiver involvement in mental health system strengthening in low- and middle-income countries: a cross-country qualitative study." *Epidemiology and Psychiatric Sciences*, vol. 27, no. 1, pp. 29-39, 2018.
- [17] S. M. Butte and W. F. W. Ahmad, "An overview of Software Models with Regards to the users Involvement," *IJCSI International Journal of Computer Science issues*, vol. 9, no. 3, pp. 107-112, 2012.
- [18] Petersen, Kai, Claes Wohlin, and Dejan Baca. "The waterfall model in large-scale development." In *International Conference on Product-Focused Software Process Improvement*, Springer, Berlin, Heidelberg, pp. 386-400, 2009.
- [19] S. Mathur and S. Malik, "Advancements in the V-model," *International Journal of Computer Applications*, vol. 1, no. 12, pp. 29-34, 2010.
- [20] R. Yadav, "Improvement in the V-Model," *International Journal of Scientific & Engineering Research*, vol. 3, no. 2, pp. 1-8, 2012.
- [21] Koskela, Juha, and Pekka Abrahamsson. "On-site customer in an XP project: empirical results from a case study." In *European Conference on Software Process Improvement*, Springer, Berlin, Heidelberg, pp. 1-11, 2004.
- [22] S. Anwar, Y. Motla, Y. Siddiq, S. Asghar, M. Hassan and Z. Khan, "User-Centered Design Practices in scrum Development Process: A Distinctive Advantage?" in *17th IEEE International Multitopic Conference*, Karachi, pp.161-166, 2014.
- [23] P. Permana, "Scrum Method Implementation in a Software Development Project Management," *International Journal of Advanced Computer Science and Application*, vol. 6, no. 9, pp. 198-204, 2015.
- [24] R. Kulkarni, P .Padmanabham and M. Harshe, "Investigating Agile Adaptation for Project Development," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 7, no. 3, pp. 1278-1285, 2017.
- [25] M. Shahriar, B. Nikkhahan and S. Sohrabi, "Challenges of user Involvement in Extreme Programming Projects," *International Journal of Software Engineering and its Application*, vol. 3, no. 1, pp. 19-32, 2009.