

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/329301740>

User Involvement in Software Development: The Good, the Bad, and the Ugly

Article in IEEE Software · November 2018

DOI: 10.1109/MS.2018.4321252

CITATIONS

6

READS

1,561

3 authors:



Muneera Bano

Deakin University

57 PUBLICATIONS 874 CITATIONS

[SEE PROFILE](#)



Didar Zowghi

University of Technology Sydney

191 PUBLICATIONS 4,895 CITATIONS

[SEE PROFILE](#)



Francesca da Rimini

University of Technology Sydney

18 PUBLICATIONS 78 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



Evaluation and design of IoT architectures [View project](#)



Software Outsourcing Relationship Trust (SORT) [View project](#)

User Involvement in Software Development: The Good, The Bad and The Ugly

Muneera Bano
Swinburne University of Technology
Melbourne, Australia
mbano@swin.edu.au

Didar Zowghi
University of Technology Sydney
Sydney, Australia
Didar.Zowghi@uts.edu.au

Francesca da Rimini
University of Technology Sydney
Sydney, Australia
francesca.darimini-1@uts.edu.au

User involvement (UI) is one of the basic principles of agile software development. However, the *agile manifesto* is not the first *Magna Carta* of user involvement in software development, as the concept dates back to Information Systems literature of the 1970s [1]. Back then it was axiomatic that user involvement leads to system success. However, over the years researchers have found that the concept of user involvement is not so simple and straightforward [2]. It is multifaceted, complex and convoluted, as there are confounding variables that impact the outcomes of any project when users get involved.

A 'User' is considered as a class of stakeholders, defined as someone who will either be the actual end user of the system or is someone whose work will be affected by the system in some way. Involving the right kind of users in requirements engineering should improve the quality of the elicited requirements and the resulting system, because users are assumed to have reasonable knowledge of the domain and the existing systems. However, the empirical research literature does not provide a convincing answer as to whether effective user involvement in requirements engineering activities eliminates the need for their further engagement in the rest of software development [3].

In 2012 we embarked on a longitudinal study to explore user involvement and system success, starting with a systematic literature review. The aggregated results of empirical literature of over three decades (1980-2012), aiming to investigate the relationship between user involvement and system success (UI-SS), indicated that 68% of the studies showed positive results and the rest were either negative or uncertain [4]. This was then followed by performing two separate case studies of this complex phenomenon.

The case studies were conducted in two financial institutions in Australia. In both case studies we observed that there was a mutually constituted relationship between user satisfaction with their *involvement (process)* and their satisfaction with the delivered *system (product)*. The mutual constitution in this context refers to the reciprocal way in which the satisfaction of the users with the process of involvement could be influenced by their satisfaction with the delivered system and simultaneously their satisfaction with the delivered system could contribute to their satisfaction with their involvement.

In the first case study [5], the data was collected over a period of three years, thus allowing us to analyse the patterns of user satisfaction over the various stages of software development. The system was considered successful due to the high level of "*user satisfaction*" even though it was delivered much later than expected and exceeded the agreed budget.

In the second case study, that spanned over two years [6], the "*power and politics*" of user involvement played a crucial role in "*user dissatisfaction*". The tug of power between users and development team to influence the decision-making process produced a negative impact on UI-SS relationship. The dissatisfaction of the users with their involvement eventually lead to their dissatisfaction with the delivered system.

From our exploratory longitudinal study and analysis of large amount of qualitative and context-rich data we learned that merely involving the users in software development is not enough to guarantee system success. UI is a complex and multi-faceted phenomenon and there is a Good, a Bad and an Ugly side to it.

• The Good

The good side of UI comes in the form of benefits that management hopes to achieve by involving the users. The empirical literature has enumerated a long list of benefits of UI from various perspectives, classified as *Psychological, Managerial, Methodological, Political* and *Cultural* [4]. These benefits are in some cases considered as surrogate measures for system success. The top most cited benefit of UI was '*user satisfaction*', followed by other benefits such as '*better understanding of user requirements*', '*better communication with users*',

|

'system acceptance' and 'increase system usage'. Concurring with the results of systematic review, in the first case study, 'user satisfaction' was observed to be the critical success factor for the project [5].

The users who are involved in software development are expected to have positive attitudes and emotions towards the system. From the analysis of the data collected it was observed that the level of user satisfaction evolved throughout the stages of software development. We conceptualised the notion of user satisfaction as a reinforcement relationship between user *satisfaction with their involvement (Process)* and their *satisfaction with the delivered system (Product)* [5]. The users reported a significant number of problems and challenges in early phases of requirements and scoping of the system; these were eventually dealt with in later stages of development through effective user representation and dynamic management practices. User satisfaction with the system after installation resulted in positive reflections of their past experiences of involvement during software development process, even though the same users had previously perceived it as challenging, frustrating and time-consuming.

'User satisfaction' was observed to be more critical for system acceptance and usage in comparison to budget and schedule. In an attempt to unwrap the concept of user satisfaction, researchers have found that it is also multi-faceted, is impacted by several factors, and evolves throughout the stages of software development [7]. User satisfaction in this context is a psychological state and is based on the attitudes and cognitive responses of the users towards a phenomenon in their workplace. However, if the management involves the users only in requirements-related activities, the users may not develop the sense of belonging to the project and the resulting system.

- **The Bad**

The bad side of UI comes in the form of challenges that prevent effective user involvement and hence undermine its benefits. Although we found that there were challenges related to project logistics in both case studies, such as '*budget and resources*', '*time constraints*', '*insufficient training for users*' and '*level of expertise of users*' [8], ultimately the top challenges that hindered effective UI were largely '*human-related*'. These corresponded to the personality of the involved users, such as '*lack of motivation*' for their involvement, and negative '*attitude*' or '*behaviour*' towards the new system [4].

In the first case study [5], the attitude of the user representative, also known as the subject matter expert (SME), played a very important role in the outcomes of the project. The SME had a positive attitude towards the work in general, and was motivated about her involvement in the project in particular. The SME was also equipped with excellent personal attributes and throughout software development her role evolved to '*product owner*' and '*champion user*'. The SME's positive attitude and interpersonal skills were appreciated by her peers and thus the SME's opinions also shaped the opinions of other users. In the second case study [6], however, the SME was passive and ineffective as a representative of users. We heard many criticisms and negative comments about the roles played by, and influence of, the project manager and SME in the project. Too much trust was afforded by the project manager to this SME and his ability to represent the consolidated opinions of the users. The SME thus had too much power to navigate and determine the design of the system to what he alone judged to be needed. We found no evidence of this SME holding any meaningful consultation with any users' groups. Our observation of the SME's attitude and behaviour in this project suggested that his involvement did not represent good requirements engineering practice.

Selecting the "right" person to represent the users and be involved in the project is critical, as their attitude towards their involvement and the system can greatly influence the project outcome.

- **The Ugly**

The ugly side of UI comes in the form of organisational power and political problems that are caused by user involvement in software development. Misunderstandings and conflicts between the development team and users are frequently highlighted problems related to the political aspect of UI in the literature [4]. In the second case study [6], the project was impacted negatively by internal organisational politics. Due to the ineffective role and undemocratic approach adopted by the SME and project manager, other people were able to influence decisions and therefore caused misunderstandings and confusion. Most of these problems originated in communication channels which were not transparent to all users. Organisational politics are inevitable with multiple stakeholders with different levels of expectation [9] and power to influence decision-making processes [10].

Nip the evil in the bud!

If you don't handle the '*bad*' effectively during requirements analysis in timely fashion, it will get '*ugly*' rather than '*good*'.

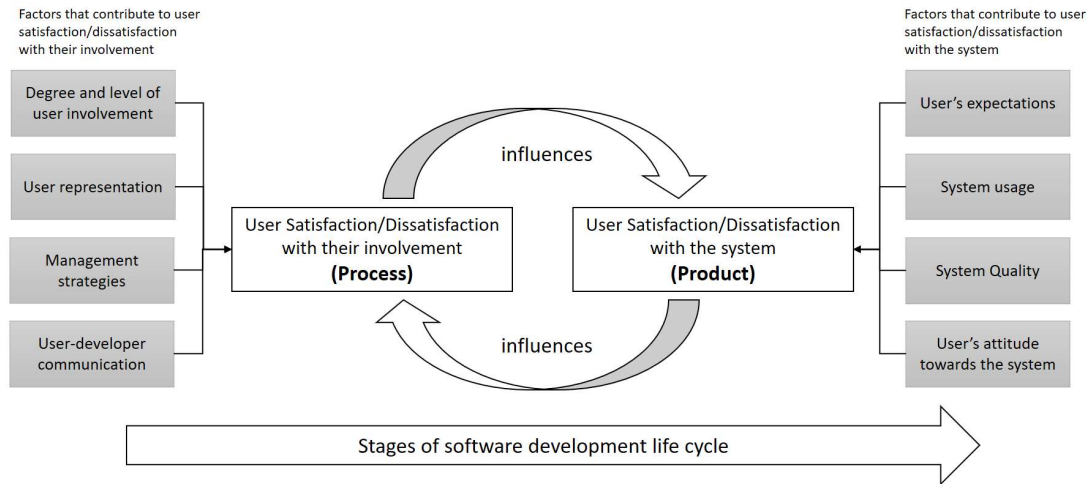


Figure 1: User satisfaction/dissatisfaction with their involvement and the system [5, 6]

UI does not begin with scoping and requirements elicitation, then disappear, only to reappear at User Acceptance Testing and again in training. From our observations we posit that users are to be involved throughout the software development lifecycle for achieving the intended *good*. However it is vital to have effective management strategy of UI in the early stages of software development [3]. User satisfaction or dissatisfaction with their involvement is not static and is subject to change throughout the software development process and beyond. If the users are satisfied or dissatisfied with their involvement (*process*) during the early phases of software development, it will contribute to their eventual satisfaction/dissatisfaction with the system (*product*) [5, 6]. The issues and challenges (*bad*) that are reported during user involvement (e.g. communication problems, misunderstandings, conflict etc.), are mostly those that are faced in the requirements analysis phase of development. In early stages of development the big picture may not be clear to all the users and hence the confusions are inevitable. If all stakeholders can see the big picture, every software project will be a success. However, if the management adopts a democratic process, and with the passage of time, users can witness the results of their voices being heard, and the benefits of their involvement could become visible to them. In hindsight, if the issues and challenges of UI are not addressed early on, they can escalate into the *ugly* side of UI.

We believe that a dynamic management strategy and effective user representation are critical for dealing with the issues and challenges during the early parts of software development.

Effective Management of User Involvement

From our longitudinal study we are able to conceptualise the five critical questions that the management should consider before involving users in product development.

1. **WHY to involve:** The management needs to have a very clear vision of the goals and objectives that they need to achieve through user involvement. The vision and bigger picture needs to be communicated clearly to the users to motivate them regarding their involvement.
2. **WHO to involve:** Not every user has the right attitude and expertise to be involved in the project. It is crucial to select the key relevant people with right expertise, motivation and communication skills for user representation. The motivation and opinions of the user representative can have a strong psychological impact on user satisfaction or dissatisfaction with the resulting system.

3. **WHEN to involve:** Users should be involved throughout the software development lifecycle. However their involvement in the requirements analysis stage is the most critical to deal with the inevitable issues and challenges of UI.
4. **HOW MUCH to involve:** User involvement has been considered as a democratic step towards empowering users, by giving them the ability to influence decisions and get a sense of control and ownership of the software system. It is important to determine at the outset the exact degree and level of influence the users can have on the final product, in order to avoid conflicts and misunderstandings in later stages.
5. **HOW to involve:** Depending on the level and degree of user involvement, various techniques are available to guide the user involvement depending on their roles during software development. If users are merely involved to provide information related to the end product, interviews, focus groups, and surveys are utilised. For actual participation of users where they have some power to influence decision making, agile methods, and participatory design techniques are recommended. For mere consultation, where users have to provide feedback or reviews, meetings could be arranged.

Involving users during software development requires time, cost and effort and necessitates careful management and planning. Their mere involvement does not guarantee project success. The management must develop a full appreciation of the complex nature of user involvement. Better and deeper understanding of the good, bad and ugly of user involvement can assist project managers to develop responsive strategies for increasing the effectiveness of user involvement.

References

1. Barki, H. and J. Hartwick, *Rethinking the concept of user involvement*. MIS quarterly, 1989: p. 53-63.
2. Abelein, U., H. Sharp, and B. Paech, *Does involving users in software development really influence system success?* IEEE software, 2013. **30**(6): p. 17-23.
3. Bano, M. and D. Zowghi. *Users' involvement in requirements engineering and system success*. in *Empirical Requirements Engineering (EmpiRE), 2013 IEEE Third International Workshop on*. 2013. IEEE.
4. Bano, M. and D. Zowghi, *A systematic review on the relationship between user involvement and system success*. Information and Software Technology, 2015. **58**: p. 148-169.
5. Bano, M., D. Zowghi, and F. da Rimini, *User satisfaction and system success: an empirical exploration of user involvement in software development*. Empirical Software Engineering, 2017. **22**(5): p. 2339-2372.
6. Bano, M., D. Zowghi, and F. da Rimini. *Power and Politics of User Involvement in Software Development*. in *Proceedings of the 22nd International Conference on Evaluation and Assessment in Software Engineering 2018*. 2018. ACM.
7. Briggs, R.O., B.A. Reinig, and G.-J. de Vreede, *The yield shift theory of satisfaction and its application to the IS/IT domain*, in *Information Systems Theory*. 2012, Springer. p. 185-217.
8. Zowghi, D., F. da Rimini, and M. Bano. *Problems and challenges of user involvement in software development: an empirical study*. in *Proceedings of the 19th International Conference on Evaluation and Assessment in Software Engineering*. 2015. ACM.
9. Buchan, J., et al. *Alignment of Stakeholder Expectations about User Involvement in Agile Software Development*. in *Proceedings of the 21st International Conference on Evaluation and Assessment in Software Engineering*. 2017. ACM.
10. Maiden, N., *Politics Are Inescapable*. IEEE software, 2012. **29**(5).