

Requirements Engineering in Agile Projects

Focusing

“Requirement Engineering Process For Non-Functional Requirements In Agile Projects”

Summary & Reference Research Papers for
Research Paper to be Submitted for
fulfillment Of Semester Project
For

Software Requirement Engineering

by

BSEF19M004 - Hurmain Javaid

BSEF19M007 - Anns Shahbaz

BSEF19M010 - Nimra Haq

BSEF19M012 - Iqra Sarwar

To

Ma'am Amna Mirza



Faculty of Computing &
Information Technology

CHAPTER 20

AGILE PROJECTS



SOFTWARE REQUIREMENTS

KARL WIEGERS AND JOY BEATTY

Summary

Agile development is based on iterative and incremental approaches to software development. It focuses on the continuous collaboration of stakeholders and delivering the software in small increments with useful functionality. Different methods of agile development are Scrum, Extreme Programming, Lean Software Development, Feature-Driven Development, and Kanban. Agile development follows the **adaptive change-driven** approach rather than the traditional plan-driven approach. In the traditional approach, the risk is minimized by extensive planning and documentation before the development. It works well if the requirements don't change during development. In agile, the change is accommodated during development and it works well for projects with changing and volatile requirements.

Limitations of the waterfall

Software development is not a sequential process as thought by organizations. In theory, it is a process of requirement specification, system design, code, and system testing. It has a lot of advantages like budget, resource, cost, and time estimation. But in practice development is not sequential. Development has to face a lot of changes and it's never straightforward. There lie several software development methodologies between waterfall and agile extremities. Waterfall strictly follows a plan-driven approach whereas agile follows an uncertain, adaptive approach. Other methodologies lie between these two extremes, and they all vary in the amount of time elapsed between development and requirement elicitation.

Real-world projects are based on the market and business needs that vary over time. Stakeholders change their requirements as their vision changes and the waterfall model fails to accommodate changes as it relies on the preplanned requirements. This results in failure in meeting user requirements, late delivery, and poor quality of the delivered product ultimately leading to project failure. That's why often large-scale projects following the waterfall model fail. **Winston Royce (1970)** presented the traditional model for the first time as a risky and failure-inviting model. He identified that errors are caught in testing, all of the development activities overlap and we need to iterate between them.

The agile approach breaks development into iterations. Functionality is added gradually in the span of iterations according to the priorities of stakeholders. Testing and validation are conducted at the end of each iteration to correct defects before they get too far in development. Essential aspects of agile development are the following:

- **Customer Involvement** increases the chances of project success. In the waterfall model, the customer is majorly involved in requirement gathering and documentation before the start of project development. But customer involvement is also required in the testing and validation phase for the success of the project. The lack of involvement in later phases of the waterfall makes it difficult to adapt to changing user requirements while in agile development customers are involved in all phases of development. Customers help to write user stories, gather requirements, testing and validation of the system. Experts and

business analysts analyze and improve user stories before providing them to the development team.

- Extensive **documentation** is required in the waterfall to record the agreed set of requirements as the development is based on these requirements. But in agile minimum documentation is carried out to avoid the effort wastage as there is enough customer collaboration. The most critical details are recorded in the form of acceptance tests.
- Agile projects maintain a **product backlog** that contains a list of requested features that might be developed. The backlog contains user stories and defects to be corrected in the form of story cards or managed by tools. **Prioritization** is carried out over the course of iterations, the items in the backlog are prioritized and discarded according to the requirements of the next iteration.
- Agile projects require traditional requirements but the **timing** is different. Instead of eliciting all requirements upfront before the start of development, only high-level requirements are specified in the form of user stories. These stories are then assigned to different iterations for detailed inspection. Non-functional requirements are specified at the start to align the system with required usability, performance, and security, etc.
- **User Stories** are concise statements of requirements that serve as a starting in agile. **Epics** are user stories that are too large to be implemented in a single iteration. So, they are further divided into epics that are divided into user stories. **Features** are system capabilities that might encompass several user stories or epics.
- **Change** is inevitable in the real world. Agile projects embrace the change and try to reduce its negative impact. The team should pay attention to what change can occur in the future to create robust and expandable architecture. Change might include removing items from scope due to other high-priority requirements or time constraints.

For making the transition to agile from traditional, development teams don't need to change the set of practices they follow, only the scope and time of these changes. All team members should focus on the goals of the project, identify best practices for your organization, collaborate with existing teams, and implement a small agile first. Seek guidance from an expert to avoid reverting to the traditional model. Don't follow all or unnecessary agile practices for the sake of being an agile purist.

About Our Research Paper

Agile is the most successful development methodology nowadays but it has to face challenges as other methodologies. After our initial study, we concluded that among other challenges, the negligence of non-functional requirements is worth considering. As functional requirements are the major focus in agile there is a need for an effective, less time-consuming, and practical requirements engineering process for non-functional requirements. In our paper, we will use the existing literature to conduct a review and present a model for the requirement engineering of NRFs. Papers to be reviewed and referenced are listed below.

Research Papers

1. [YMjDgIkR17] Younas, Muhammad & Jawawi, Dayang & Ghani, Imran & Kazmi, Rafaqat. (2017). **Non-Functional Requirements Elicitation Guideline For Agile Methods**. Journal Of Telecommunication, Electronic And Computer Engineering. 9. 137-142.
2. [YMsMaJDi20] Younas, Muhammad & Shah, Muhammad Arif & Jawawi, Dayang & Ishfaq, Muhammad & Awais, Muhammad & Wakil, Karzan & Mustafa, Ahmad. (2020). **Elicitation Of Nonfunctional Requirements In Agile Development Using Cloud Computing Environment**. Ieee Access. Pp. 1-1. 10.1109/Access.2020.3014381
3. [AjPw21] A. Jarzębowicz And P. Weichbroth, "A Qualitative Study On Non-Functional Requirements In Agile Software Development," In Ieee Access, Vol. 9, Pp. 40458-40475, 2021, Doi: 10.1109/Access.2021.3064424.
4. [ksOmN20] Kopczyńska, S., Ochodek, M., Nawrocki, J., 2020. **On Importance Of Non-Functional Requirements In Agile Software Projects—A Survey**, In: **Studies In Computational Intelligence**. Pp. 145–158. https://doi.org/10.1007/978-3-030-26574-8_11
5. [FCC18] Franco (Frank) Curtolo, CEng **Requirements Management applied in an agile Project Management environment** Copyright © 2018 by Principal Consultant, Program Planning Professionals Ltd, Frank.Curtolo@pcubed.com.
6. [YaTaMi17] Y. Aziz , T. Aziz , M. I. Malik , M. K. Baig , M. Z. Ali , M. Baqer **Non Functional Requirement in Agile Software Development**. Technical Journal, University of Engineering and Technology (UET) Taxila, Pakistan Vol. 22 No. I-2017
7. [LBKRF17] López, L., Behutiye, W., Karhapää, P., Ralyté, J., Franch, X., Oivo, M., 2017. **Agile Quality Requirements Management Best Practices Portfolio: A Situational Method Engineering Approach**, https://doi.org/10.1007/978-3-319-69926-4_45
8. [PMR19] Oriol, M. *et al.* (2019). **Data-Driven Elicitation of Quality Requirements in Agile Companies**. In: Piattini, M., Rupino da Cunha, P., García Rodríguez de Guzmán, I., Pérez-Castillo, R. (eds) *Quality of Information and Communications Technology*. QUATIC 2019. Communications in Computer and

Information Science, vol 1010. Springer, Cham. https://doi.org/10.1007/978-3-030-29238-6_4

9. [KJR18] Franch, X. *et al.* (2018). **Data-Driven Elicitation, Assessment and Documentation of Quality Requirements in Agile Software Development.** In: Krogstie, J., Reijers, H. (eds) Advanced Information Systems Engineering. CAiSE 2018. Lecture Notes in Computer Science(), vol 10816. Springer, Cham. https://doi.org/10.1007/978-3-319-91563-0_36
10. [AjPw21] A. Jarzębowicz And P. Weichbroth, "A Qualitative Study On Non-Functional Requirements In Agile Software Development," In Ieee Access, Vol. 9, Pp. 40458-40475, 2021, Doi: 10.1109/Access.2021.3064424.
11. [RvDk19] Raghu Veer, Deepak Kumar Singh, **Extracting Non-Functional Requirements Based On Agile Software Development(Asd).** (2019). International Journal Of Innovative Technology And ExploringEngineering,8(11),3413–3415.<https://doi.org/10.35940/Ijitee.K2542.0981119>
12. [KiSaMan16] Kulsoom Iftikhar, Saqib Ali, Md Asri Ngadi, “**Enhancement Of Non Functional Requirements In Agile Software Development**” International Journal Of Computer Science And Information Security (Ijcsis), Vol. 14, No. 12, December 2016
13. [TsGr16] Trupti Suryawanshi, Mtech, Computer Engineering, Bvducoe, Pune, India & Gauri Rao, Associate Professor, Computer Engineering, Bvducoe,Pune, India, “**Modeling Of Nonfunctional Requirements For Agile Development Processes**” Vol.04 Issue-02, (February, 2016) Ijite - International Journal In It And Engineering, Impact Factor- 5.343
14. [BrLcRb07]B. Ramesh, L. Cao, And R. Baskerville, “**Agile Requirements Engineering Practices And Challenges: An Empirical Study,**” Inf. Syst. J., Vol. 20, No. 5, Pp. 449–480, Nov. 2007.
15. [VhDdClMp15] V. T. Heikkilä, D. Damian, C. Lassenius, And M. Paasivaara, “**A Mapping Study On Requirements Engineering In Agile Software Development,**” In Proc. 41st Euromicro Conf. Softw. Eng. Adv. Appl., Aug. 2015, Pp. 199–207.
16. [ISsSmMdSs15] I. Inayat, S. S. Salim, S. Marczak, M. Daneva, And S. Shamshirband, “**A Systematic Literature Review On Agile Requirements Engineering Practices And Challenges,**” Comput. Hum. Behav., Vol. 51, Pp. 915–929, Oct. 2015.

17. [SwDmMkMf18] S. Wagner, D. Méndez Fernández, M. Kalinowski, And M. Felderer, “**Agile Requirements Engineering In Practice: Status Quo And Critical Problems**,” Clei Electron. J., Vol. 21, No. 1, P. 15, Apr. 2018.
18. [JmDcAvCsEw15] J. Medeiros, D. C. Alves, A. Vasconcelos, C. Silva, And E. Wanderley, “**Requirements Engineering In Agile Projects: A Systematic Mapping Based In Evidences Of Industry**,” In Proc. Ibero-Amer. Conf. Softw. Eng. (Cibse), 2015, Pp. 460–476.
19. [WaMdRw17] W. Alsaqaf, M. Daneva, And R. Wieringa, “**Quality Requirements In Large-Scale Distributed Agile Projects—A Systematic Literature Review**,” In Proc. Int. Working Conf. Requirement Eng., Found. Softw. Qual. Cham, Switzerland: Springer, 2017, Pp. 219–234.
20. [IbLkMm20] Iqra Batool , Lareb Kosar , Marrium Mehmood **Non-Functional Requirements As Constraints And Their Values In Software Development: A Review** Department Of Software Engineering, Bahria University Islamabad, Pakistan , 2016 - 2020
21. [RSBJm21] Rahy, S., & Bass, J. M. (2021). **Managing non-functional requirements in agile software development**. *IET Software*, 16(1), 60–72. <https://doi.org/10.1049/sfw2.12037>
22. [DMpHp18] Dave, Ms & Patel, Hitesh & Prajapati, Mr. (2018). **Non-Functional Testing In Agile Development**.
23. [VaEbAmAt20] Vera, Antonio & Espinosa Bedoya, Albeiro & Manager, Ams & Tech, Algar. (2020). **Applying Verification and Validation in Agile Methods: a tough road discipline**.
24. [Bn15] Baruah, N. (2015). **Requirement Management in Agile Software Environment**. *Procedia Computer Science*, 62, 81–83. <https://doi.org/10.1016/j.procs.2015.08.414>
25. [BwKpCOM17] Behutiye, W., Karhapää, P., Costal, D., Oivo, M., Franch, X. (2017). **Non-functional Requirements Documentation in Agile Software Development: Challenges and Solution Proposal**. In: Felderer, M., Méndez Fernández, D., Turhan, B., Kalinowski, M., Sarro, F., Winkler, D. (eds) Product- Focused Software Process Improvement. PROFES 2017, vol 10611. Springer, Cham. https://doi.org/10.1007/978-3-319-69926-4_41
26. [WbPsPrMo20] Woubshet Behutiye, Pertti Seppänen, Pilar Rodríguez, Markku Oivo. (2020) **Documentation of Quality Requirements in Agile Software Development**.

Books

1. [KwJb13] Karl Wiegers And Joy Beatty **Software Requirements**, Copyright © 2013
Karl Wiegers And Seilevel
2. [AaCw05] Aybüke Aurum · Claes Wohlin (Eds.) 123 **Engineering And Managing Software Requirements**

State Of Agile

1. [Soar15] “15th **Annual State Of Agile Report**,”2021 (Accessed: 13.06.2022)
<https://stateofagile.com/>