

RESEARCH

Open Access



Identifying components existing in Agile software development for achieving “light but sufficient” documentation

Basit Habib^{1*} , Rohaida Romli² and Malina Zulkifli³

*Correspondence:
basithabib36@gmail.com

¹ Department of Gender Studies,
Bahauddin Zakariya University,
Multan, Pakistan

² School of Computing, College
of Arts and Sciences, Universiti
Utara Malaysia, Kedah, Malaysia

³ School of Quantitative Sciences,
College of Arts and Sciences,
Universiti Utara Malaysia, Kedah,
Malaysia

Abstract

The Agile nature allows changes within the development procedure which can sometimes be non-predictable and affect the cost and time of the software. Face-to-face communication appears to be a faster process in Agile Software Development (ASD) in practice, but issues such as understandability and communicativeness cause ambiguity between the customer and the developer. Among the existing works, achieving light but sufficient documentation in ASD remains a challenge. Most previous studies have simply highlighted the issues associated with less documentation, its limitations, and challenges, but no proper solution has yet been proposed. As an initial step in our theoretical study, we conducted a systematic literature review (SLR) to identify the related components and subcomponents that are applicable in supporting Agile documentation. As such, the fundamental concepts related to software documentation have been studied in order to align with documentation in ASD. The identification of these components can be useful in achieving the concept of “light but sufficient” documentation in Agile, which will aid in achieving customer satisfaction.

Keywords: Process documentation, Product documentation, Software quality, User requirement, Customer satisfaction, Agile software development

Introduction

Agile software development model has been in popularity for the last two decades and is mostly used among many practitioners [1]. The Agile software process model illustrates the relationship between iterative and incremental process models and emphasis on product flexibility, customer satisfaction through communication between the customer and developer and by having less documentation [2]. The main concept of Agile is to overcome the traditional software process models to save time by breaking the software in set of modules and to develop the modules through teamwork [1, 2]. These modules are commonly developed parallel by involving customer in every sprint of the module so that documentation can be kept low and requirements can be understood face-to-face for rapid development [3, 4]. Due to the face-to-face verbal approach, Agile shows a lack of predictability that explains both developer and customer ignorance because factors concerning user requirements and customer satisfaction are basics in a software process

and rely on documentation which is less or not even seen in Agile [5, 6]. Therefore, with the passage of time, developers have understood that without proper documentation the process of sprint has become retrospective and commonly one module has to be sprinted more than once [2]. This practical output varies from the theoretical concept of Agile causing a deep effect on time and cost and sometimes even the failure of the software [7].

By understanding the Agile manifesto, it is seen that Agile focuses on less documentation but Agile does not say “no” to documentation [6, 7]. The current practices of documentation in ASD are mostly based on a temporary approach in which documentation is done in the perspective of key point, use case diagrams or data flow process on white boards, or writing pads mostly during the sprint process, but there is no such template, format, procedure, or model for Agile which covers the main aspects of process and product documentation so that every phase of ASD can be documented [8]. Therefore, due to the ambiguity between the minds of the customer and the developer, again and again redesigning of modules occurred [7, 8].

Being a part of human nature, most of the developers avoid documentation as they think of it as lethargic burden [2]. That is why, in the past years, software developers using different Agile methodologies like Scrum, XP, Crystal, Lean, and DSDM have misunderstood the concept of documentation in Agile from less documentation to no documentation [9]. They have focused on face-to-face communication with the customer, which has many problems in describing and understanding the requirements, causing ambiguity which further reflects software delays or even failures [9].

The importance of documentation in any software development cannot be denied [1]. Documentation is a part of software quality, so it is important to have a functional documentation model within Agile for achieving customer satisfaction and user requirement in ASD [1, 9].

Further sections of this study highlight Agile process, phases of Scrum methodology, components related to process documentation and product documentation, components of user requirements, and components related to customer satisfaction which can be used in ASD in terms of documentation to ease both the customer and developer. Along with this, these components will keep the documentation light but sufficient so that it cannot be a burden on the developers and can be easily managed.

Related work

Agile is one of the famous widespread software development models used in software development which aims to have a real-time relation between the customer and developer. Being a blend of incremental and iterative techniques that puts the customer's needs first by concentrating on user requirements, Agile has different methodologies which are chosen by the ease of the developer for giving a high-quality software product [1]. By satisfying rapidly changing needs within constrained timeframes, these methodologies have surpassed traditional software development [2]. Agile methodologies theoretically offer increased customer satisfaction, lower error rates, increased usability, and a quicker reaction to changing requirements [1, 9]. The popularity of the most used Agile methodologies is seen in Fig. 1 [10].

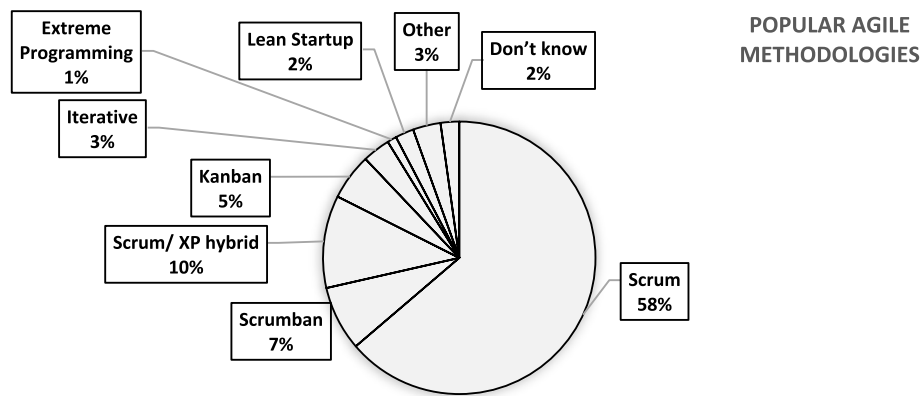


Fig. 1 Popularity of Agile methodologies [10]

Table 1 Role of documentation in Agile methodologies [8]

Methodology	Documentation Role	Document template	Documentation format	Information gathering method
XP	Minimally managed, with little temporary documentation	No	As a < type user >, I want < some goal > so that < reason >	Little docs, verbal
Lean	There is less documentation used only in for feedback and testing,	No	Not available	Face-to-face communication
Crystal	Moderate	Design notes	Not available	Short, rich, formal communication
FDD	As per need, light	Model use cases, design, functional requirement	< action > the < result > < by/for/ of/ to > < object >	Combined team
DSDM	As per need, light	Early instance/working software modeling and prototyping), system overview, context diagram	Not available	Not available
ASD	Less	User manual, white board	As a < role >, I want < features > so that < reason >	Not available
Scrum	Moderate	Report template,	Not Available	Face-to-face communication
Kanban	Less	White board/design notes	Not Available	Not available

By comparison, it is seen that Scrum is the most popular Agile methodology used by software developers. Scrum style focuses more on its efficiency and simplicity. Scrum is a well-liked Agile methodology because of the teamwork and coherence of its elements [11]. Most Agile methodologies show less documentation, mostly done on temporary basis [12]. They mostly consist of graphical data, which makes the requirements understandable and glancing over the documentation for giving a quick overview of the project [10, 12]. The existence of documentation on the basis of its role as a template, format, and information gathering is illustrated in Table 1 [8].

Based on the study of Shafiq and Waheed [8] in Table 1, the role of documentation in Agile methodologies is very less, and there is no such format or template used in

documentation which can be particular for Agile, whereas the importance of documentation in software development cannot be denied, and its less existence issues in Agile software process model practically causes an effect on the success of the software. This less existence of documentation causes the software to go off the track regarding time and cost. Therefore, if documentation will be proper and comprehensive regarding process and product documentation, then it will be easy to understand the user requirements properly.

Related issues concerned with less documentation in Agile

Agile focuses on software flexibility and customer satisfaction through less documentation and increased developer-client communication [7–9]. But due to the ignorance of both developers and customers, a lack of predictability, understandability, and ambiguity is often seen which causes software delay or failure [7].

Agile performance is achieved when the risks are managed involving costs of negotiations, time constraints, implementation, knowledge gaps about existing tools, and communication gaps for understanding the user requirements. But frequent changes and timely delivery are critical to predict in order to reduce major risks in the cost of software development due to insufficient documentation causing ambiguity between the customer and developer [12–14].

Apart from software predictability, one other reason of software failure in Agile is the communication issues causing understanding problems [14]. To determine the overall picture of the software is by generalizing the requirements and the development procedure by involving documentation [1, 9]. Literature on Agile documentation does not provide complete solutions but explains certain aspects of documentation, as there are very few empirical studies on Agile documentation.

Documentation in software process helps to understand the users' requirements which further helps in minimizing the ambiguity in the mind of the customer [1, 9, 14]. In Agile, to have "light but sufficient" documentation means to have less but comprehensive amount of documentation which can highlight the requirements properly [1]. Documentation typically gives a comprehensive overview of designing the software, which helps to achieve a clear picture of the software in the customer and developer's mind [9, 14, 15].

Related components in ASD

Through the theoretical study, the components used in this research related to Agile and documentation cover process documentation, product documentation, quality factors and their criterion, general requirements, user activities, requirement engineering (RE) activities, focus components, and documentation types used for customer satisfaction. The studies highlighting the components and their subcomponents are shown in Table 2.

Table 2 describes the different researches which have highlighted different components related to process documentation, product documentation, quality factor and criterion, user requirements, and customer satisfaction.

From the studies of Subih, Malik, and Suleman [6], Lee [17], and Sirshar and Arif [15], the software quality criterion highlighted in Table 1 is communicativeness which shows the software's capacity to intuitively explain its functionality to the user for making it

Table 2 Identifying components in ASD

Studies	Methodology	Domain	Components
[16]	Multiple case studies	Components concerned with quality requirement gathering through documentation in Agile	-Technical writers -Story cards -Management Document -Mitigation Strategy
[7]	Empirical Study	Evaluation of process and product documentation	-User document -Design document -Project overview -Requirement document -Support document -Operations document -Vision statement -System documentation -Contract model
[9]	Empirical study	Evaluation of General user requirements and product documentation	-Product requirement -Organizational requirements -External requirements -Vision statement -System documentation -Contract model
[8]	Empirical	User activities involved in gathering user requirements	-Use cases -Technical writers -Test-driven Development
[12]	Multiple case studies	Determine the components of RE activities	-Req. elicitation -Req. analysis -Req. document -Req. validation -Req. management
[15, 17]	Empirical	Existence of software quality factors in ASD	-Reliability -Correctness -Usability -Flexibility -Maintainability -Reusability -Efficiency -Interoperability -Functionality -Testability -Communicativeness -Completeness -Traceability -Operability -Training -Expandability -Scalability -Modularity -Integrity -Portability -Modularity -Performance
[11]	Empirical	Existence of customer satisfaction, environmental and organizational working regarding SCRUM based software	-Customer location -Customer knowledge -Customer requirement -Customer feedback

more usable. Completeness helps to identify the fundamental principles which must be followed to complete any module of a software. Traceability is the capacity to track something as it progresses through a process. Operability is the capacity to employ an action of a module in the software. Training teaches the skills to handle the software. Expandability increases the size of the software in terms of modules, storage, or any

other general requirement. Scalability measures the working and development of a module of a software. Integrity gives a good and meaning full software source code. Portability identifies how to install the module or software in certain sources like change of OS, etc. Modularity distinguishes the software in chunks so that they can be easily handled and Performance identifies the working of a software.

Behutiyea and Rodríguezb [16], Shafiq and Waheed [8], and Ebega and Mnkandla [13] highlight that user requirement is based on thirteen subcomponents in which general requirements consist of project requirements which are the elements, tasks, and needs that must be fulfilled for a project. Organizational requirements establish guidelines for how users should carry out their responsibilities, and external requirements are those requirement elements that have a realization link connecting them to the current policies/software house rules.

User activates consist of user stories which are an informal, all-encompassing description of a software feature written from the viewpoint of the client or end user. Story cards is the process used in Agile/Scrum projects to record project scope, quality requirements, estimations, and deliverable priority. Use cases are shown as an oval shape with labels. Stick figures depict the actors in the process, and a line connecting the actor and use case models their involvement in the system. Technical writers explain complicated and technical information; technical communicators should create instruction manuals, how-to guides, journal articles, and other supporting publications more effectively; and test-driven development creates unit test cases prior to writing actual code which is the core of the test-driven development (TDD) methodology.

Requirement engineering (RE) activities consist of requirement elicitation which is the process of investigating and learning what a system needs from developers and customers. Requirement analysis is the act of figuring out what users will expect from a new or changed product. Requirement document is utilized to make sure all stakeholders understand the project's goals in a clear, succinct manner. Requirement validation is the process of making that the system performs as designed and achieves its goals, and requirement management is the method used by businesses to define, manage, check, and validate ideas and satisfy the needs of stakeholders at each stage of the product life-cycle, from idea generation through development and commercialization.

Ali and Anjum [11] and Lee [17] in their research highlight that customer satisfaction consist of two main components in which focus components consist of four subcomponents which are customer location which identifies the availability and accessibility of the customer, customer knowledge which identifies the awareness of the customer regarding software development, customer requirements identifying the demand/needs of the customer, and customer feedback which gives an overview from the customer regarding the fulfillment of the requirements.

Documentation types have four subcomponents which are management document showing the papers that business managers and owners use to help them with their day-to-day business management duties, specification document which outlines the project participants, modules, and other activities expected results or deliverables, design document showing the compilation of materials covering every facet of your product design, and mitigation strategy offering the foundation for actions to be identified, prioritized, and carried out to lower risk of hazards.

Furthermore, the relation of these components and their subcomponents are highlighted in Fig. 2 in description.

Methodology

A systematic literature review (SLR) methodology was conducted in this study to identify the components involved in ASD documentation [18]. SLR is an efficient methodology of observing, evaluating, and analyzing various research to explore research questions.

The attributes of SLR are as follows:

i. Research questions:

The research questions (RQs) are as follows:

- RQ1: What are the components and subcomponents existing in Agile which are related to documentation practices in software development?
- RQ2: How can these components and subcomponents be used to achieve light but sufficient documentation in ASD?

ii. Search string:

In order to formulate a search string, the criteria provided by Kitchenham is employed [18]. Besides that, the asterisk symbol "*" is used to expand the research scope and retrieve all possible search terms. The search string for this study that is shown as follows:

("Agile Process" OR "Agile Methodologies") AND
("Evaluate*" OR "Validate*" OR "Assess" OR
"Measure*") AND ("Factor" OR "Dimension" OR
"Criteria*" OR "Metric")

The search string covers three main elements that represent the study objective. The first concept is designed to retrieve studies focused on the Agile and Agile methodologies, while the second concept is to retrieve studies concerning with components and subcomponent related to documentation, and the third concept is to elaborate the components and subcomponents.

iii. Conduct search:

Conduct search is a step to select the basic studies involved. It consists of two steps: (1) using search string to retrieve information from electronic resources and (2) databases selections. For the first step, keywords like Agile software development, Agile methodologies, Agile documentation, user requirement, customer satisfaction, and software quality were used. In the second step, well-known electronic databases were selected like ACM Digital Library, Google Scholar, IEEE Xplore, Scopus, Research Gate, ETHOS (e-thesis online services), and Science Direct.

iv. Screening of the papers:

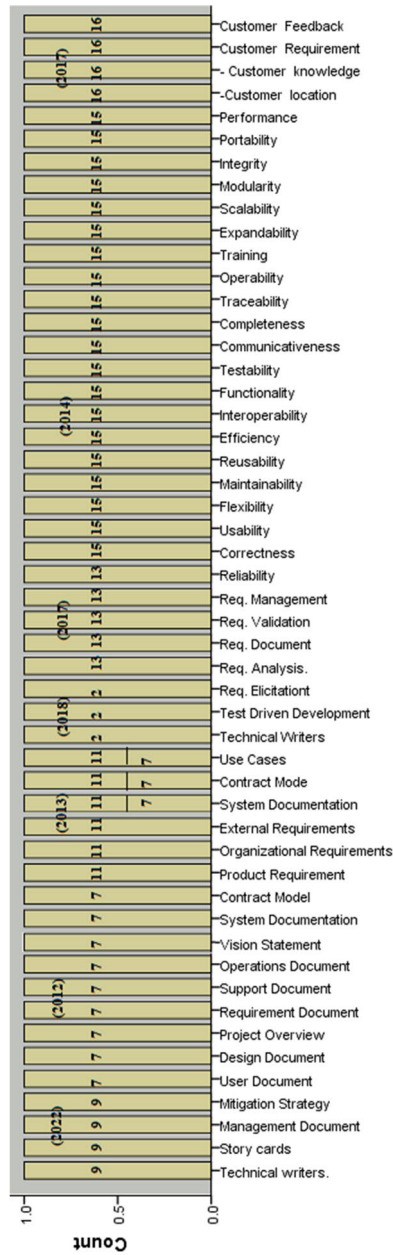


Fig. 2 Existence of documentation-related components in sources

This stage is related to screening the basic studies selected to the topic of this study such that the studies that are not related to answer the RQs can be excluded. The inclusion and exclusion criteria applied in this study are as follows:

- a) Inclusion criteria: papers are written in English, papers are related to agile or ASD, papers are related to software documentation, papers are related to documentation in agile or ASD.
- b) Exclusion criteria: papers that are not related to software process models, papers that are not related to software documentation. Further details can be seen in Table 3.

v. Keywording using abstracts:

Keywording was performed in two steps: (1) the abstract was read with its keywords, and (2) concepts that reflected the contribution of the paper were identified. The main focus of this section is to gain those papers which are based on ASD, documentation practices in ASD, and those that highlight the components which are involved to produce a light but sufficient documentation model, template, or format for documentation in ASD.

vi. Data extraction:

Data extraction is the final stage of SLR process. The data extraction procedure was conducted by collecting the information about the related papers to address the RQ literature review by presenting them in a flow in Fig. 2. In Fig. 2, it is seen that the extracted components from process documentation, product documentation, software quality factor and criterion, components of user requirement, and customer satisfaction related to documentation in software development are segregated from the latest researches from 2017 to 2022.

Results and discussion

Agile does not say “No” to documentation; instead, its emphasis is on “light but sufficient” documentation. As seen through the literature, Agile does not provide any practice, procedure, model, template, or format of documentation. So, keeping documentation, “light but sufficient” is a still a challenge. For the solution, components involved in software documentation are highlighted in the next sections.

Table 3 Inclusion and exclusion criteria

Inclusion criteria	-Studies that describe evaluation methods used in evaluating ASD -Studies which address the evaluation of Agile approaches -Studies using empirical results
Exclusion criteria	-Studies not relevant to the research questions -Studies not focusing on ASD -Studies not related to Agile evaluation -Studies which are not primary -Studies that are not complete or unavailable

RQ1: What are the components and subcomponents existing in Agile which are related to documentation practices in software development?

This section describes that there are six components of process documentation which involve user document which includes documentation for customer and developers during software development like a reference guide, a user manual, a support instructs, and training materials which are included. Design document is a process documentation type which describes the software's operation, working, module development procedure along with the functioning of the software. Project overview is a document used to identify the outcomes of the software. Requirement document elaborates the requirements by the customer and developer. Support document is useful in the process of developing the modules of the software and their synchronization. And operation document clarifies the installation, storage, and other general perspectives of the software. Product documentation involves vision statement which is a type of documentation regarding the time and cost estimated for the software. System documentation is the overall documentation of the software on contract basis like installation and training, and contract model is a software document and deal between the customer and developer.

The software quality factor, highlighted in Table 1, is the reliability which identifies the degree of accuracy that can be expected from a measurement, calculation, or specification. Correctness identifies if the fundamental principles have been followed. Usability identifies if the function is beneficial in other situations, is resistant to human mistake, and does not require a lot of core memory. Flexibility sees the modifications in the operational environment and the need of the simplicity of making changes. Maintainability refers to how easily a product can be maintained in order to isolate and/or rectify flaws and/or their causes, fulfill new requirements, and adapt to a changing environment. Reusability is the software's ability to be reused in a variety of situations. Efficiency is what a software program should accomplish its aim without wasting resources. Interoperability is the time and effort it takes to connect the system to another system. Functionality is to ensure that the software is always completely functional in all its application areas, and testability shows the software testing facility to verify that the developed modules in every iteration are specified and working properly.

As shown in Table 2, it can be seen that there are seven researches since the last decade from 2012 to 2022 which have highlighted the issues of less documentation in Agile and have highlighted various components which are used in documentation practices of Agile. These researches are based on the multiple case studies collected from real-time development and empirical data. The existence of the researches along with the components can be seen in Fig. 2.

RQ2: How can these components and subcomponents be used to achieve light but sufficient documentation in Agile process?

Theoretically, to achieve "just enough" documentation, documentation strategies have been distinguished like documenting electronic backups, documenting changes, documenting business terminology used, documenting inter team work development,

and documenting feedback after every sprint [7]. But to understand how these strategies will be conceptually fulfilled, it is necessary to distinguish some components which are directly involved in the documentation and exist in different literatures of Agile.

Thus, the first RQ of this study is to identify all the components involved in Agile and have a direct impact on documentation practices in software development. These components are process documentation, product documentation, software quality factors and criterion, user requirement components, and components of customer satisfaction. Including all these six components, they further include forty-two subcomponents as described in Table 2. The main purpose of these components is to gather the user requirements for the satisfaction of customer.

Second is to achieve “light but sufficient” documentation in Agile. For this, the first phase is to identify the most widespread Agile methodology which can be seen in Fig. 1 as “Scrum.” Then, use the identified components in Table 2, under the phases of Scrum described in the “[Related work](#)” section, so that these components can be called stepwise in the software procedure to highlight the gathering of the user requirements, development of the requirements, and changes made in the requirements and their execution. This can further be seen in Fig. 3.

This relation of components involved in Agile which are related to documentation will be useful in terms of gathering user requirements through the traditional Agile face-to-face communication process, but with a certain type of documentation type which can be kept as a backup of preliminary requirement gathering, requirement changes, documenting the business terms and conditions, and the feedback of the requirement fulfillment by the customer.

Conclusions

Taking into account the existence of documentation in Agile is very blur, it is causing issues related to requirement gathering and having a deep effect on customer satisfaction. These issues eventually go deep on the time and cost of the software project. Agile is a platform which is unique in comparison of traditional software process models and still has better success rate, but with the passage of time, the misinterpretation of no documentation over less but sufficient documentation has caused many ambiguities. After almost two decades of no template, the procedure or model has yet been designed for documentation in Agile. This study has identified the components which are concerned with Agile, and if documented, then the success rate of Agile projects can increase by minimizing the retrospectiveness during the sprint process in Agile projects. For the understanding of the components and subcomponents highlighted in Table 1 from various studies, Fig. 2 elaborates the flow of the components described in Table 1, so that they can be seen in the perspective of documentation within the Scrum-based Agile methodology.

Furthermore, in the future, these components should be shared with the real-time Agile practitioners to see which components are useful and which have no concern in the Agile software process. Along with this, new components can also be explored which are used by practitioners in real-time practices.

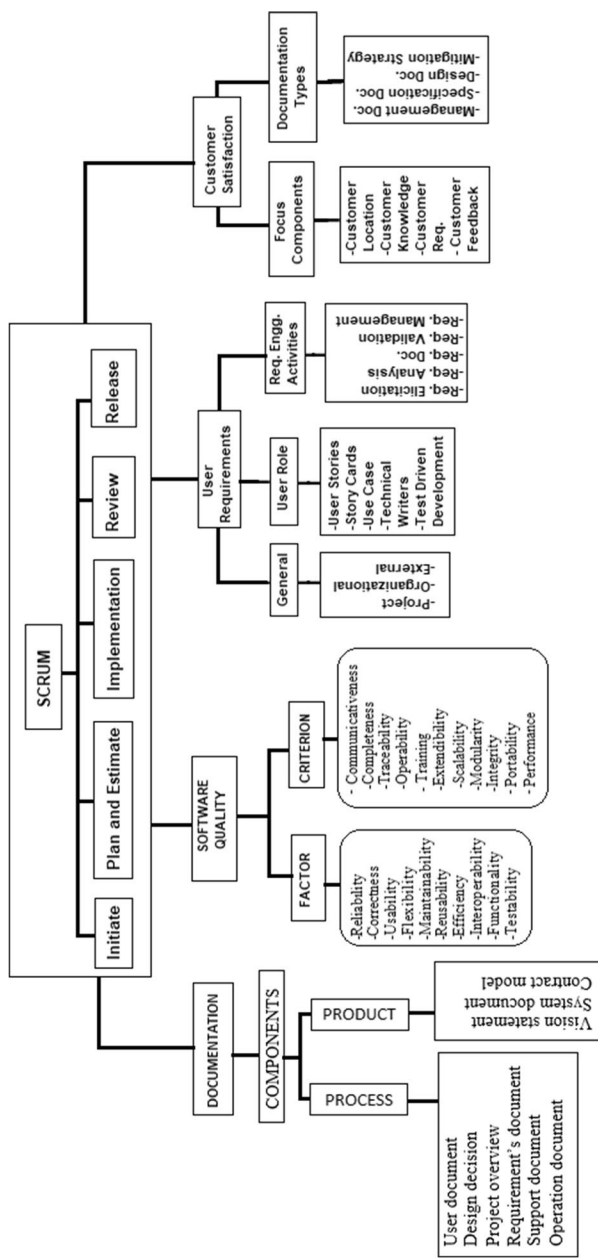


Fig. 3 Highlighted components and subcomponents

Abbreviations

ASD	Agile software development
RQ1	Research question 1
RQ2	Research question 2
DSDM	Dynamic system development method
XP	Extreme programming
SLR	Systematic literature review
RE	Requirement engineering

Acknowledgements

Not applicable.

Authors' contributions

This study is one step forward towards introducing a practice, template, procedure, or model of having documentation in Agile. This is a PhD work of BH, in which, being a practitioner himself in Agile environment, he has experienced that there is no such template, format, or model of documentation in Agile which can give light and sufficient documentation within the domain of Agile. BH has highlighted the components which are used while working in a real-time environment in ASD, whereas RR has indicated the concept of these components theoretically and have justified them through literature. MZ on the other hand has managed to arrange the components and develop a sequence between them so that they can be analyzed and can be used in future work for creating the concept of documentation in ASD. After going through all these contributions, all authors, BH, RR, and MZ, approved the manuscript.

Funding

Not applicable.

Availability of data and materials

From the previous related articles <https://ieeexplore.ieee.org/document/9522254>. All data analyzed during this study are based on the published article which highlights the issues and importance of documentation in Agile. Furthermore, the articles which show the relevance and components introduced in the study are also present in the reference section.

Declarations**Competing interests**

The authors declare that they have no competing interests.

Received: 25 April 2023 Accepted: 21 June 2023

Published online: 07 July 2023

References

- Behutiye W, Seppänen P, Rodríguez P, Oivo M (2020) Documentation of quality requirements in Agile software development. *ACM International Conference Proceeding Series*, pp. 250–259
- Pelclová J (2014) Documentation in Agile. MASARYKOVA UNIVERZITA, Brno Czechia. https://is.muni.cz/th/vi2bu/Pelcova_MastersThesis.pdf
- Martínez C, Alix R (2020) A Scrum implementation plan by phases and oriented by team members: a case study. Universidad Ean, Bogotá, Colombia. https://ceurws.org/Vol2714/icaiv_aiesd_1.pdf
- Heck P, Zaidman A (2014) A quality framework for agile requirements: A practitioner's perspective. Technical Report TUD-SERG-2014-006, Software Engineering Research Group, Delft University of Technology. https://www.researchgate.net/publication/263237832_A_Quality_Framework_for_Agile_Requirements_A_Practitioner's_Perspective/link/54b92f260cf2d11571a31c41/download
- Way T, Chandrasekhar S, Murthy A (2009) The Agile Research Penultimatum. *Proc Int Confer Softw Eng Res Pract* 2:530–536
- Subih MA, Malik BH, Mazhar I, Izaz-ul-Hassan Sabir U, Wakeel T, Ali W, Yousaf A, Bilal-bin-Ijaz Nawaz H, Suleman M (2019) Comparison of Agile method and Scrum method with software quality affecting factors. *Int J Adv Comput Sci Appl* 10(5):531–535
- Hoda R, Noble J, Marshall S (2012) Documentation strategies on Agile software development projects. *Int J Agile Extreme Softw Dev* 1(1):23–37
- Shafiq M and Waheed U (2018) Documentation in Agile development: a comparative analysis. *Proceedings of the 21st International Multi Topic Conference, IEEE* 2018
- Jan I, Tabrez S (2013) Documentation and agile methodology. Uppsala: Department of Informatics and Media. <http://www.diva-portal.org/smash/record.jsf?pid=diva2%3A678784&dsid=-7529>
- Budiman T, Suroso J (2017) Optimizing it infrastructure by virtualization approach. *IOP publications, International Conference on Electrical Engineering, Computer Science and informatics, Indonesia*, Vol. 190, pp. 1–7
- Ali A, Rehman M, Anjum M (2017) Framework for applicability of Agile Scrum methodology: a perspective of software industry. *Int J Adv Comput Sci Appl* 8(9):225–232
- Sebega Y, Mnkandla E (2017) Exploring issues in agile requirements engineering in the South African software industry. *Electron J Inform Syst Dev Countr* 8(1):1–18
- Buresh DL (2008) Customer satisfaction and agile methods. *IEEE Reliability Society Annual Technology Report*, pp. 1–8

14. Habib, B., Romli, R.: A systematic mapping study on issues and importance of documentation in agile. In: 2021 IEEE 12th International Conference on Software Engineering and Service Science (ICSESS), pp. 198–202. IEEE, August 2021
15. Sirshar M, Arif F (2012) Evaluation of quality assurance factors in Agile methodologies. *Int J Adv Comput Sci* 2(2):73–78
16. Behutiyea W, Rodríguez P, Oivo M (2022) Towards optimal quality requirement documentation in agile software development: a multiple casestudy. *J Syst Software* 183:1–17
17. Lee M-C (2014) Software quality factors and software quality metrics to enhance software quality assurance. *Br J Appl Sci Technol* 4(21):30–69
18. Kitchenham B, Pretorius R, Budgen D, Brereton OP, Turner M, Niazi M, Linkman S (2010) Systematic literature reviews in software engineering – a tertiary study. *Inf Softw Technol* 52(8):792–805

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Submit your manuscript to a SpringerOpen[®] journal and benefit from:

- ▶ Convenient online submission
- ▶ Rigorous peer review
- ▶ Open access: articles freely available online
- ▶ High visibility within the field
- ▶ Retaining the copyright to your article

Submit your next manuscript at ▶ [springeropen.com](https://www.springeropen.com)
