

Project Management issues while using Agile Method- Ology

Shariq Aziz Butt¹ (✉) [0000-0002-5820-4028], Piñeres-Espitia G², Paola Ariza-Colpas³, Muhammad Imran Tariq⁴.

^{1st}Department of computer science and information technology, The University of Lahore, Pakistan.

^{2nd & 3rd}Universidad De la Costa, CUC Barranquilla, Colombia.

^{4th}Superior University, Lahore, Pakistan.

Shariq2315@gmail.com, gpineres1@cuc.edu.co, parizal@cuc.edu.co, imrantariqbutt@yahoo.com.

Abstract. Software engineering has many software development life cycle (SDLC) models to develop a software application and the latest SDLC models have been provided by agile methods. The agile methodology has been introduced due to some existing lacks in software development. Now agile methodology is used to overcome these deficiencies and improve software development. The use of the agile methodology is increased within software industries due to its distinctive features such as enabling change requests from the client at any stage of a project, client satisfaction, iterative development, and client-developer interaction. Another reason for agile adoption is the methods that are being used for agile software development. These methods include Scrum, Feature drive development, Extreme programming, and Dynamic system development methods. However, the agile methodology has some issues for project development and management. In this study, we discuss all these issues which are related to agile methods and individuals (i.e. team and developer). Further, we suggest the possible improvements that need to be introduced in the agile methodology. We believe such improvements is to make the agile methodology more productive for development environments.

Keywords: Agile methodology, software development, methods of agile, and challenges.

1 Introduction

Software Engineering has principles, SDLC models, and systematic techniques that are used to develop software applications. The agile model was introduced with a complete manifesto in 2001 to overcome some deficiencies from productive software development [1]. Now the agile model is the trendiest and mainly in use model in Software Development. The reason for the popularity of agile is distinctive features

such as enabling change requests from the client at any stage of a project, client satisfaction, iterative development, and client-developer interaction [2]. Another reason is agile's methods such as Scrum, Feature driven development (FDD), Extreme programming (XP), Dynamic system development Method (DSDM) [3, 4]. All of them have been presented within the umbrella of agile and are investigated in this paper. They have added more productive value to agile development by their characteristics as mentioned in the Table.2 and 3.

Agile also facilitates the client to prioritize the user stories and can give change requests at any level of the project [3]. Agile methods are productive for small-scale projects, small-size teams, requirements welcomed, and group effort estimation for example a planning poker technique [9]. In the agile methodology, the project manager leads estimates the project effort with teamwork [40]. In the estimation, the team assesses the complexity of the user stories.

Agile's most distinctive features and individuals associated properties are productive for agile software development. These individual properties include the developer's capabilities, understanding of user stories, and code writing skills [5]. Although the specifics of all these methods have the same purpose of allowing teams to adjust to changes relatively. Because accommodating changes later throughout the project is costly [6].

Instead of wide use for software development agile methodology still has some limitations that make it less productive due to agile's methods capabilities [3, 7]. This study is identifying the risks associated with each agile's method, as well as with an individual, which are a key part of software project management [8]. Moreover, agile methods usually do not cover some areas of project management like budget and schedule management [10]. Agile methodology is also not pertinently productive for large-scale projects [11]. Therefore, there is a need to make some improvements and introduce new features in agile methodology to support its use for large-scale projects. Some of such features in agile methodology includes SAFe (Scaled Agile Framework), SoS (Scrum of Scrums), LeSS (Large Scale Scrum), DAD (Disciplined Agile Delivery), RAGE (Recipes for Agile Governance in the Enterprise) [41].

1.1. Problem statement

Agile is the most used model for software development, but still has some issues that directly impact its proficient productivity. The main issues with the development are due to methods used in agile software development [14]. One of the most valuable features is a change request from the client because the client is the owner of the product backlog [15]. In agile development due to small iteration development when any change comes from a client that is not listed initially in product backlog then might increase the cost of a project. Nevertheless, the cost is fixed at the start of the project by the team effort estimation i.e. a cost has decided with the client after the

team effort assessment for a project [16]. Normally changes suggested by the client are due to market revolutions [16, 19].

Another limitation with the agile methods is its application only for small-scale projects and for small development teams [4, 11]. Agile also should be applicable for all sizes of projects, large-scale organizations, and large teams. It is because of agile's methods that make it not applicable for the large-scale organization.

Moreover, the agile methodology also has some flaws with respect to individuals i.e. developer and team. All these make agile less productive for efficient software development. Therefore, the agile methodology needs some improvements in its methods to become suitable for all types of projects and needs to introduce a cost estimation process with some new team-based estimation procedure to make accurate estimation [9, 10, 11].

1.2. Objectives of study

This study is to highlight the main issues that directly affect the productivity of agile methodology. Such issues are related to shortcomings in the agile methods and constraints of individuals for proficient software development in agile. On the other hand, the effort estimation in agile is also a challenge. The study also suggests some parameters that need to be included in the effort estimation technique for an agile estimate. After such improvements, agile will be applicable for all sizes of projects. We can implement these contributions on real projects in the software development industries.

1.3. Significance of the study

The significance of the study is to improve agile software development for all scales of projects and scenarios. Suggested changes would be useful for the professionals, who are using the agile methodology in their software industries, research scholars who want to contribute scientific methods in the agile methodology. The suggested improvements will enhance agile software development in all diverse areas of development. The project management in agile software development will get a new positive direction.

2 Methodology

We have adopted the methodology for conducting the study and reviewing the related studies of agile software development. For the completion of the study, we have followed an agile development score rating methodology. It contributes to evaluating

the depth and accuracy of agile project management specifications, as well as providing a roadmap for experts in identifying flaws in agile development [14].

In Figure.1 we have defined the steps that followed for literature gathering [15, 16]. In the methodology steps, we defined the necessity of and purpose of study that we mentioned in sections 2, 3, and 4. In this section, we clearly explained the study objectives and significance. The second step is defining research questions that illustrate the overall picture of the study. We have developed two research questions that relate to finding the issues with the agile software development and division of these issues according to agile's methods and individuals (developer, team/s). These research questions are mentioned as follows:

RQ1: What are the shortcomings in agile development with respect to agile methods and individuals?

RQ2: How such issues are affecting productive agile software development?

The third step is the searching of literature related to agile software development and agile's methods and issues with them. This step further is divided into sub-steps such as:

2.1. Search strategy:

A search query is used to do systematic searches across the digital libraries including Google Scholar, IEEE, Science Direct, ACM, Springer, Hindawi, MDPI, Sage, and Hindawi [14, 16]. All reputed data search bases are used for the collection of related works

2.2. Search Strings:

We used different search strings to find the agile software development papers within the domain. These strings are ("Agile Methodology" OR "Agile Software Development" OR "Software Development Issues") AND ("Large Development Team in Agile" OR "Agile Project Management") AND ("Factors in Agile Issues" OR "Agile methods positively affects" OR "Issues with key Methods") AND ("Individual Practices" OR "Issues").

The fourth step is application inclusion and exclusion criteria on the searched papers. We used inclusion and exclusion criteria to select studies published and eliminate

irrelevant publications. The paper had to match the keywords in the title with the keywords of the study to be considered for inclusion. The first process is to examine the title of every publication and determine if it met the inclusion/exclusion parameters. If the topic of the publication corresponded to key terms within the study objectives, the abstract is examined to check significance using the inclusion parameters. In contrast, we used exclusion criteria when the search strings outcomes did not meet simply a little with the primary terms in our study topic. We have gathered the papers between the years 2014 and 2021, we looked for agile project development and management. Related a total of 60 papers are obtained from the databases described above, and 50 are chosen based on respective titles. The 44 papers that are chosen are then filtered further depending on abstract and keyword, relevancy, getting 39 publications [14, 16].

The fifth and last step is the data extraction strategy, in which after the inter-rater reliability test, there is no conflict after extracting data. Date of study, the title of publication, databases, procedures, strategy applied within the paper for research are all gathered from each scientific publication. We used the reliability test to find the weightage of some topics as mentioned in the Table.1. As stated in the table.1 P_i represents the topic priority, R_i represents the topic ranking, and W_e represents the weight for every topic [14]. Here the weights are assigned based on the ranks to get a relative magnitude. To rank the validated elements, results of the survey were considered which gave the priority percentage of each element based on its importance in a scope definition.

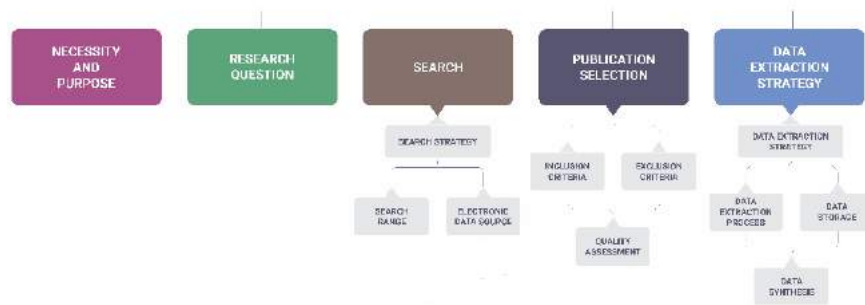


Figure 1. Methodology Steps

Table 1. Weightage, raking, and priority of agile studies [14].

Topics	P_i	R_i	W_g
T1. Agile methodology survey	2.01	11	2.10
T2. Agile methods surveys	2.91	12	4.02
T3. Agile review studies	3.18	13	5.12
T4. Agile and project management literature	2.14	14	7.52
T5. Agile methods applications	3.20	17	10.72
T6. Agile development limitations	3.78	22	15.30
T7. Agile Project management limitations	3.75	25	21.20
T8. Agile methods flaws	3.46	24	21.00
T9. Agile issues with individual	4.42	27	22.10

3 Literature Review

Agile is a well-regarded SDLC development approach and is becoming more well-known and widely used in software industries. The enhancement in agile development becomes well-known. It is adopted in the software houses due to its distinctive features that make agile more productive for efficient software development. Among other SDLC models, the agile method is the most in use model for development due to its unique methods and features [7, 8]. The agile methods include the (i) Scrum: is a much-known approach in agile software development. It enables the product development in sprints and establishes a concept of daily meeting with the team to project updates. (ii) DSDM: is an agile's project delivery process that works with rapid application development concepts and prioritizes the user stories based on user feedback (iii) FDD: is an iterative development process in agile software development and (IV) XP: is an agile process as it takes the development at an extreme level. It is used for small teams and small-sized projects to produce high-quality software products. The agile unique features include client satisfaction, change requests from the client at a time during the project development, user stories prioritized by the client, iterative development. Instead of so much fame and unique features, agile software development still has some issues that are discussed in the next section.

3.1. Factorization of issues with Agile's Methods

3.1.1. Scrum Iterations Issues

The agile model eliminates many issues from the software development at that time and allows the change request at any time and any level of the project. Scrum teams work in sprints which divide the project into small iterations. These sprints are easy to manage and the software project easily developed. These sprints send to the client to get his feedback. When there is no change request then the next sprints start and otherwise changes merge with the next iterations. To accommodate the change request is the main in a scrum within the agile model. Ultimately such changes increase the cost and time of the project [14, 15].

Other issues in the scrum are related to the team is daily based meetings for project updates. However, the daily meetings have limitations because not all the developers support the meeting and mostly are non-responsive to meetings. Due to this forced meeting and part of the agile development does not meet the requirements [16].

3.1.2. Extreme Programming (XP) Iterations Issues

eXtreme Programming (XP) has the ultimate objective of completing the project at hand. Exploration, Planning, Iterations to Release, Productionizing, and Maintenance are the 5 phases of the XP product lifecycle. Instead of so much use XP still has some issues that directly impact agile software development. Firstly, XP is not applicable for large-scale projects because it takes development to an extreme level. The reason is that XP does not measure the code for quality and complexity that cause code defects at the initial level. XP is also not feasible for global software development, due to different geographical locations [16, 18]. XP also supports the less documentation that causes defects in documentation, identical defects may arise in the future [37].

3.1.3. DSDM Issues

As agile welcomes the changes at any stage of the project then always the requirements fluctuates while development. Thus in the DSDM primary issue is requirements are not fixed and always add to the product backlog. Due to these continuous adding of new requirements/user stories project's cost and time increased. Cause of direct involvement of client throughout the agile development [19, 20]. Such issues are making the DSDM less productive for agile development.

3.1.4. FDD

Using the FDD in agile development main issue is less documentation. Secondly, it results in a high level of dependence on a single person. The Chief Programmer serves as a coordinator, main designer, and instructor, among other things. Multiple

responsibilities in a large project is a problem since it raises the risks of human error. In addition to the aforementioned drawbacks, the structure of this approach is unlike other agile methodology, sprints are not very well described within the process. These are project-specific and adapted to the project's needs. As a result, there is no standardized process for iterations [21, 22].

Table 2 Comparison of Agile methods.

<i>Characteristics</i>	<i>XP</i>	<i>Scrum</i>	<i>FDD</i>	<i>DSDM</i>	<i>References</i>
Methodology for development	Incremental Improvements are a type of modification.	Incremental Iteration	Iterative	Incremental	3, 4, 5
The time between iterations is required.	One to six weeks	Two to four weeks	Two days to two weeks	Depending on the method belong to the family	6,7,8,9
Project team	fewer than 20 individuals of team/s	All sizes (concept of scrums)	Many people are members of multiple teams	All sizes are available based on the methodology of the family.	9,10,11, 13
Collaboration inside the team	Regular meetings of team members	Regular meetings of team members	Depends on documentation	face to face Informal	11,14,15,17, 38
The scope of the project	Small projects	Various types of projects	Projects that are more difficult to complete	Projects of all kinds Based on the methodology, could be a member of a family	18,20

Participation of Customers	Involvement of Customers	The responsibility to the Product Owner is performed by the Client	Customer-generated reports	Customers will benefit from incremental updates.	19,21,23
Project Documentation	Minimum documentation	Less Documentation	Documentation's Significance	simple Basic documentation	22,24
Skills	Refactoring, User stories DD	scrum master, for example plan-	Diagrams of UML	Family's adaptive methodology	25, 26, 28

Table 3 Summary of agile methods features

Condition	XP	Scrum	FDD	DSDM
Small Team/s	Yes	Yes	No	unclear
Requirements that are very changeable	Yes	Yes	Yes	unclear
Teams that are dispersed	No	Yes	Yes	No
High Ceremony Culture	No	No	unclear	unclear
Systems with a High Severity	No	unclear	unclear	Yes
Multiple Customers / Stakeholders	No	Yes	unclear	unclear

3.2. Factorization of issues with Individuals

3.2.1. People

A software project might include a wide spectrum of individuals, including developers, testers, and project managers, to mention a few. The final product is frequently required by a client or end-user. Top executives (company managers and development department heads) are particularly concerned with costs, investment

returns, and human resources. In agile development, every one of these has a responsibility [22].

3.2.2. Developers

The developers are maybe the most affected by agile processes. Agile methodologies rely on good programmers who are skilled, experienced, and willing to interact with clients productively. Developers have to be willing to work as part of a team, be capable to deal with frequent change, and also be innovative in their problem-solving abilities. Agile processes seem to be very flexible approaches that do not require developers to adopt rigorous standards and practices. But it is a problem for a software house as some programmers may not be able to work in an agile environment. In an agile context, the "5" rank of developer shown in the Table.4 would be challenged. "Hand-holding" takes resources even for "4" programmers. As a result, the agile development team's base is made up of the top 3 ranks. Rank "1" programmers may or may not be required for any projects, depending on how rare it seems. Agile Approaches may be challenging to implement within a typically staffed software house due to the high degree of skill required. Highly talented professionals are consistently in demand, and developing a long-term human development approach may be challenging without integrating 4 rank programmers. Long-term projects provide a considerable risk for Agile Methodologies for several reasons [23, 25, 27].

Table 4 Summary of agile methods features

Rank	Features
1	abilities to develop solutions under bizarre circumstances
2	capable of changing solutions to meet a new, yet previously encountered circumstance
3	programmer capable of implementing functionality, estimating effort, and refactoring code
4	capable of implementing basic functionality, running tests, and completing tasks
5	Reluctant or unable to collaborate in a team environment.

3.2.3. Project Leaders

Project managers and team leads are the two most important Project Leader responsibilities throughout software development. As leadership under such an agile methodology varies from previous approaches, it has its own set of difficulties. This difference is well defined as controlling process resources and leadership effectiveness. A leadership strategy is particularly efficient when agile teams include skilled professionals with significant responsibilities. Team leaders should be willing to pro-

vide team members the freedom to take initiative. Rather than using the central command, collaboration is used as a method of leadership. For some, it will be a psychological adjustment, since they will have to share decision-making power. A team leader's responsibility is to make it easier for the team to make decisions. On the other hand, project managers are responsible for monitoring performance and approving business decisions within agile approaches. Project managers must make a more significant adjustment [23, 24].

The attention is on reacting to changes rather than adhering to a strict schedule. It is a problem because they are typically looked upon to provide updates about the project's progress. In addition, project managers play a considerably more active role. In scrum, the project management interacts with the team every day and supervises the daily scrum. For the agile team, regular team member gatherings are the standard procedure. Rather than emphasizing defining the milestones and contracts, project managers are much more engaged in building and maintaining customer relationships

3.2.4. Customers

Agile methodologies have a much greater effect on customers as compared to traditional methods. Clients are involved from the beginning of the project, participating with defining the requirements and contract responsibilities, and at the ending, and acceptability testing under relatively traditional techniques. However, under agile methodologies, clients are engaged even more frequently and have more control over user stories [25].

Clients might be unwilling to participate in software development. Clients may be unfamiliar with startups because the market is still to be determined. When using an agile methodology, the presence of client representatives should be considered. In agile clients should be "decided to commit, skilled, cooperative, representative, and empowered through the development. They should be aware of the requirements for end-users. Furthermore, as decisions regarding which functionality will be included in which releases should be decided, the representative should have the ability to do so. Agile Approaches may not be applicable including all sizes and types of projects because a client representative might not have been available [29].

3.2.5. Team

The team is important to effectiveness in agile processes because they rely significantly on interaction and coordination. A sole skilled developer, who can't work well together, and a client who doesn't interact with the team all have the potential to undermine a team's feasibility and effectiveness. The chemistry of the team poses a substantial risk for agile development. Another important human aspect to consider within an agile team is turnover. High turnover in a project might result in the loss of key

skills if there is no formal documentation. Although code inspections and having developers alternate working on various functional areas can help prevent this, losing the main member of a team can be catastrophic. When determining whether the team is suitable for the agile approach, the project leader should consider this situation. Recognizing that one of the main principles of XP is to keep current knowledge through keeping skilled staff [26, 27, 30].

4. Findings and Suggestive Measures

This section of the study is covering the findings of the research and suggestive measures that can overcome issues in the agile software development and project management mentioned in the Table. 5

Table 5. Findings of study.

Contents	Research Findings	Recommendation	Impact of recommendations
Scrum	Daily meetings agitate the developers because they are responsible to update them about their tasks in front of all others.	Split the meetings into different sessions rather than arrange them on daily basis. The meeting can also be done in the digital environment means online.	It will remove issues from the scrum and make the developer and teams more comfortable to up- date. On other hand, it will enhance trust among them. It will reduce pressure from developers to work productively.

DSDM	Requirements are not fixed and also not pertinent for distributed team/s.	Defined all the requirements clearly at the initial stage of a project and reduce the level of control of the client on the product backlog but did not remove control completely. Thus, whenever the client gives a change request then the team or chief developer should examine whether do able to or not.	Define the project core parameters at the initial stage of a project and fix them to not extend it. Digital processes in agile software development will enhance the development and team coordination.
------	---	--	---

FDD	The main issue is less documentation and highly rely on the chief developer for requirement specification and effort estimation.	Increase the documentation and make all the specifications clearly defined in the document. Establish a team effort estimation mechanism rather than support the time chief programmer or team lead as in planning poker technique as example.	Proper documentation will stop the frequent change from the client because when you defined all the specifications then ultimately the project will develop as per the requirements. In case the client suggested more additional changes than before welcome changes should be validated with the document. Team/s or developer/s individual suggestions about the effort estimation will reduce the dependency on the team lead. Additionally, it will reduce the biased nature of estimation in a team.
XP	Some issues are similar with the FDD like documentation but another issue is not applicable for large projects.	Need to increase the team size.	Increase the team size and then split the team into 2 parts as per their expertise. Because 2 small groups of the same team will work more productively.
Developers	Developer expertise for a user story.	Always give a user story to develop who has the expertise for it.	It will directly impact on accurate estimation of user stories. It will also eliminate the biased nature of effort estimation from agile. Thus, the estimation technique should be more predictive based and should estimate the user stories by the

			individual developer rather than follow the process as ex- ample planning poker. Estimation by the Indi dual as per their skills and knowledge, team leaders can have the most accurate judgments about the efforts of each developer for the particular user story. This would make the estimation accurate and safe from bi-ased estimation.
Client	Control on the product backlog	Document the re-quirements and then overcome the control of the client on the product backlog.	When reducing the con-trol of client from product backlog but supporting client satisfaction, then it will control the changes from the client. Some-times a client is also am-biguous related to the user stories.
Team/s	Team's abilities	As we stated that se-lect developers as per the user story and ex-pertise. Thus, the se-lect team as per their previous project's his-tory and working skills.	Selection of team will im-prove the productivity much more because in the agile team needs to ar-range daily meetings, work pressure, and client changes so the team abili-ties for the current pro-ject

5. Conclusion

Software engineering has a lot of models to develop software projects. All models have some flaws and to overcome these from the efficient software development agile models were introduced. Agile methods are more viable for adoption in the software industry. The agile main goal and feature ae to support the client who can send a change request at any time and any level of the project. On the other hand, it is a limi-tation in the agile model. Another limitation in investigated agile methods is that they are applicable for only small-scale projects and do not give fruitful results when use

for large-scale projects. We have studied all the factors as categorization for the agile software project's development and management. We have done extensive study on both factors agile's methods and developer, and client, etc. We found that both factors need major improvements to make it more proficient for development. Such improvements precisely regarding the developers, large projects, daily meetings, and client direct much involvement throughout the project. The study's findings and suggestive measures are pertinent for the followers of the agile methodology. These are providing new directions for agile practitioners to amend the agile and introduce some new features. Researchers can improve agile methodology by following the listed suggestions and making the agile method an advanced version for future use.

References

1. Steghöfer, J. P., Knauss, E., Alégroth, E., Hammouda, I., Burden, H., & Ericsson, M. (2016, May). Teaching Agile-Addressing the Conflict Between Project Delivery and Application of Agile Methods. In 2016 IEEE/ACM 38th International Conference on Software Engineering Companion (ICSE-C) (pp. 303-312). IEEE.
2. Noteboom, C., Ofori, M., Sutrave, K., & El-Gayar, O. (2021, January). Agile Project Management: A Systematic Literature Review of Adoption Drivers and Critical Success Factors. In Proceedings of the 54th Hawaii International Conference on System Sciences (p. 6775).
3. Trier, K. K., & Treffers, T. (2021, May). Agile Project Management in Creative Industries: A systematic literature review and future research directions. In 2021 IEEE Technology & Engineering Management Conference-Europe (TEMSCON-EUR) (pp. 1-8). IEEE.
4. Khalid, A., Butt, S. A., Jamal, T., & Gochhait, S. (2020). Agile Scrum Issues at Large-Scale Distributed Projects: Scrum Project Development At Large. *International Journal of Software Innovation (IJSI)*, 8(2), 85-94.
5. Vishnubhotla, S. D., Mendes, E., & Lundberg, L. (2018, February). An insight into the capabilities of professionals and teams in agile software development: A systematic literature review. In Proceedings of the 2018 7th International Conference on Software and Computer Applications (pp. 10-19).
6. Martin, A., Anslow, C., & Johnson, D. (2017, May). Teaching agile methods to software engineering professionals: 10 years, 1000 release plans. In *International Conference on Agile Software Development* (pp. 151-166). Springer, Cham.
7. Al-Saqqa, S., Sawalha, S., & AbdelNabi, H. (2020). Agile Software Development: Methodologies and Trends. *International Journal of Interactive Mobile Technologies*, 14(11).
8. Fuchs, C. (2019). Adapting (to) agile methods: Exploring the interplay of agile methods and organizational features.
9. Gandomani, T. J., Faraji, H., & Radnejad, M. (2019, February). Planning Poker in cost estimation in Agile methods: Averaging vs. Consensus. In 2019 5th Conference on Knowledge-Based Engineering and Innovation (KBEI) (pp. 066-071). IEEE.
10. Noll, J., & Beecham, S. (2019, November). How agile is hybrid agile? an analysis of the helena data. In *International Conference on Product-Focused Software Process Improvement* (pp. 341-349). Springer, Cham.

11. Wińska, E., & Dąbrowski, W. (2020). Software Development Artifacts in Large Agile Organizations: A Comparison of Scaling Agile Methods. In *Data-Centric Business and Applications* (pp. 101-116). Springer, Cham.
12. Tessem, B. (2017). The customer effect in agile system development projects. A process-tracing case study. *Procedia computer science*, 121, 244-251.
13. Dingsøy, T., Falessi, D., & Power, K. (2019). Agile development at scale: the next frontier. *IEEE Software*, 36(2), 30-38.
14. Amjad, S., Ahmad, N., Saba, T., Anjum, A., Manzoor, U., Balubaid, M. A., & Malik, S. U. R. (2017). Calculating completeness of agile scope in scaled agile development. *IEEE Access*, 6, 5822-5847.
15. Ruk, S. A., Khan, M. F., Khan, S. G., & Zia, S. M. (2019, December). A survey on Adopting Agile Software Development: Issues & Its impact on Software Quality. In *2019 IEEE 6th International Conference on Engineering Technologies and Applied Sciences (ICETAS)* (pp. 1-5). IEEE.
16. Abdalhamid, S., & Mishra, A. (2017). Adopting of agile methods in software development organizations: systematic mapping. *TEM Journal*, 6(4), 817.
17. Schuh, G., Prote, J. P., Gützlaff, A., Ays, J., & Donner, A. (2019). Fixed cost management as an enabler for agile manufacturing networks. *Procedia Manufacturing*, 39, 625-634.
18. Raza, S., & Waheed, U. (2018, November). Managing Change in Agile Software Development a Comparative Study. In *2018 IEEE 21st International Multi-Topic Conference (INMIC)* (pp. 1-8). IEEE.
19. Al-Saqqa, S., Sawalha, S., & AbdelNabi, H. (2020). Agile Software Development: Methodologies and Trends. *International Journal of Interactive Mobile Technologies*, 14(11).
20. Goel, S., Hawi, S., Goel, G., Thakur, V. K., Agrawal, A., Hoskins, C., ... & Barber, A. H. (2020). Resilient and agile engineering solutions to address societal challenges such as coronavirus pandemic. *Materials Today Chemistry*, 17, 100300.
21. Gablas, B., Ruzicky, E., & Ondrouchova, M. (2018). The Change in Management Style during then Course of a Project from the Classical to the Agile Approach. *Journal of Competitiveness*, 10(4), 38-53.
22. B. Ratner, "The Correlation Coefficient: Definition," *DM Stat-1 Articles*, vol. 11a, 2007.
23. Kuusinen, K., Balakumar, V., Jepsen, S. C., Larsen, S. H., Lemqvist, T. A., Muric, A., ... & Vestergaard, O. (2018, August). A large agile organization on its journey towards DevOps. In *2018 44th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)* (pp. 60-63). IEEE.
24. Anwer, F., Aftab, S., Waheed, U., & Muhammad, S. S. (2017). Agile software development models tdd, fdd, dsdm, and crystal methods: A survey. *International journal of multidisciplinary sciences and engineering*, 8(2), 1-10.
25. Przybyłek, A., & Zakrzewski, M. (2018). Adopting collaborative games into agile requirements engineering.
26. Kumar, R., Singh, K., & Jain, S. K. (2019). Agile manufacturing: a literature review and Pareto analysis. *International Journal of Quality & Reliability Management*.
27. Patel, A., Seyfi, A., Taghavi, M., Wills, C., Na, L., Latih, R., & Misra, S. (2012). A comparative study of agile, component-based, aspect-oriented and mashup software development methods. *Tehnicki Vjesnik*, 19(1), 175-189.
28. de la Barra, C. L., Crawford, B., Soto, R., Misra, S., & Monfroy, E. (2013, June). Agile software development: It is about knowledge management and creativity. In *International*

- Conference on Computational Science and Its Applications (pp. 98-113). Springer, Berlin, Heidelberg.
29. Pham, Q. T., Nguyen, A. V., & Misra, S. (2013, June). Apply agile method for improving the efficiency of software development project at vng company. In *International Conference on Computational Science and Its Applications* (pp. 427-442). Springer, Berlin, Heidelberg.
 30. Mundra, A., Misra, S., & Dhawale, C. A. (2013, June). Practical scrum-scrum team: Way to produce successful and quality software. In *2013 13th International Conference on Computational Science and Its Applications* (pp. 119-123). IEEE.
 31. Correia, A., Gonçalves, A., & Misra, S. (2019, July). Integrating the Scrum Framework and Lean Six Sigma. In *International Conference on Computational Science and Its Applications* (pp. 136-149). Springer, Cham.
 32. Habib, B., & Romli, R. (2021, August). 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.
 33. Fernández-Sanz, L., Gómez-Pérez, J., Díez-Folledo, T. I., & Misra, S. (2016). Researching human and organizational factors impact for decisions on software quality. In *Proceedings of the 11th International Conference on Software Engineering and Applications* (pp. 283- 289).
 34. Khan, R.A.; Abrar, M.F.; Baseer, S.; Majeed, M.F.; Usman, M.; Ur Rahman, S.; Cho, Y.-Z. Practices of Motivators in Adopting Agile Software Development at Large Scale Development Team from Management Perspective. *Electronics* 2021, 10, 2341. <https://doi.org/10.3390/electronics10192341>.
 35. Niederman, F., Lechler, T., & Petit, Y. (2018). A research agenda for extending agile practices in software development and additional task domains. *Project Management Journal*, 49(6), 3-17.
 36. Usman, M., Mendes, E., Weidt, F., & Britto, R. (2014, September). Effort estimation in agile software development: a systematic literature review. In *Proceedings of the 10th international conference on predictive models in software engineering* (pp. 82-91).
 37. Szewc, A., Karovič, V., & Veselý, P. (2022). The Documentation in the Project of Software Creation. In *Developments in Information & Knowledge Management for Business Applications* (pp. 361-441). Springer, Cham.
 38. Przybyłek, A., Albecka, M., Springer, O., Kowalski, W. (2022). Game-based sprint retrospectives: Multiple action research. In: *Empirical Software Engineering*, vol. 27. <https://doi.org/10.1007/s10664-021-10043-z>
 39. Jones, A., & Thoma, V. (2019). Determinants for successful agile collaboration between UX designers and software developers in a complex organisation. *International Journal of Human-Computer Interaction*, 35(20), 1914-1935.
 40. Weflen, E., MacKenzie, C. A., & Rivero, I. V. (2022). An influence diagram approach to automating lead time estimation in Agile Kanban project management. *Expert Systems with Applications*, 187, 115866.
 41. Kalenda, M., Hyna, P., & Rossi, B. (2018). Scaling agile in large organizations: Practices, challenges, and success factors. *Journal of Software: Evolution and Process*, 30(10), e1954.