Enhancing Requirement Clarity in Agile Teams: Investigating Communication Among Agile Team Members

EREN KODAL, University of Twente, The Netherlands

Agile development methods emphasize adaptability and self organizing teams to manage evolving software requirements. However, communication issues remain a key challenge, often leading to clarity issues in requirements which leads to products undesirable to the stakeholders. This study explores how communication practices within self organizing agile teams can be improved to maintain clarity of requirements. Through semi-structured interviews with five agile team members experienced in software development, qualitative data was gathered and analyzed using open, axial, and selective coding. The exploratory research revealed that effective communication practices is a central theme which interconnects agile team dynamics and requirements clarity. A robust communication framework is mandatory for agile principles to be utilized effectively.

Additional Key Words and Phrases: Agile development, self organizing teams, communication practices, team dynamics, requirements clarity, qualitative study, semi-structured interviews, software development, grounded theory

1 INTRODUCTION

Agile development methods have been leading various industries, especially for software development [8]. Agile is a methodology which emphasizes iterative development cycles through collaboration within the team which works in an autonomous manner [6, 7]. The rationale for using agile methods is the fact that it is increasingly challenging to fully specify requirements prior to the start of implementation [12]. Agile methods differ from plan-driven traditional methods and instead focuses on adaptability to requirements [8]. With the proposition that Agile is capable of enabling adaptation, where the requirements are meant to be iterated over and changed between each sprint, it has become the industry standard [20].

Agile development methods put a high emphasis on the agile team to function autonomously [22]. While this autonomy is meant to be the key factor in enabling the adaptive nature of the agile methodology, allowing teams to implement evolving software requirements, it also introduces several challenges. Among these challenges, miscommunication is particularly prominent, often leading to budget overruns and delayed product delivery. The less rigid structure of agile teams, compared to traditional approaches, can further amplify communication challenges [5].

It is crucial to examine the root issue of communication to enhance the effectiveness of autonomous Agile teams. "The process of designing, supporting and coaching autonomous agile teams is still not adequately addressed and understood in the context of software development organizations" [22]. Research is needed to understand how agile teams can communicate with high clarity, avoiding ambiguities both within the team and with stakeholders, to ensure the intended development goals are achieved. Since agile teams operate autonomously, it is essential to ensure that communication remains clear. Furthermore, as requirements are dynamically managed and frequently revised during development, maintaining clarity of requirements across agile team members is critical. However, it must be highlighted that while requirements clarity is a common challenge in software development, the autonomous nature of Agile teams, intended to mitigate this issue, adds further complexity to communication due to the reliance on high quality collaboration.

For this purpose, this paper aims at exploring "how communication practices within the autonomous agile team be improved to maintain the clarity of requirements". A case study has been completed through conducting semi-structured interviews with five agile members in this regard.

2 BACKGROUND

This section reviews relevant literature on agile self organizing teams and clarity of requirements in software development. It aims to establish a foundational understanding of these two core topics and explore their unique intersection, which remains underexplored in current literature.

2.1 Self organizing Teams in Agile Setting

Self organizing teams are the foundation of the agile methodology [8]. Traditional teams work within a rigid hierarchical structure "with a clear separation of roles". In contrast, self organizing teams rely less on external management and are capable of acting independently as a cohesive unit, who share interdependent tasks [18, 23]. In software development, a self organizing team is one that navigates the development process autonomously, demonstrating the ability to adapt and address challenges as they emerge [4].

Takeuchi and Nonaka [23] established self organizing teams as a pillar of product development, inspiring Agile methodologies like Scrum [14, 18].

To establish a self organizing team, specific criteria have been identified.

- "Place team members physically, together and closer."
- "Replace extensive documentation with talking in person and at whiteboards."
- "Improve the team's amicability so that people are more inclined to relay valuable information quickly."

Effective self organization in agile teams relies on various criteria, with a key intersection being the importance of fostering effective communication among team members [4]. The positive correlation between communication and self organizing teams is readily apparent.

Current state of the art has already established the positive effects of self organizing teams. From a study consisting of 111 teams being

TScIT 43, July 4, 2025, Enschede, The Netherlands

investigated, it was concluded that they are both more proactive and productive [16]. Another survey spanning 477 respondents also concluded that the team work quality of the self organizing team has large effects on success, where communication is a key factor [17].

2.2 Clarity of Requirements in Software Development

Clarity of requirements, where clarity corresponds to the clear understanding of what the requirement implies is one of the leading problems within the development phase. The ambiguity of the requirements contribute to the following issues [13, 19].

- Growing Technical Debt: Unclear initial requirements can lead into inappropriate architecture choices which result in excessive refactoring and sometimes complete rewrites.
- Imprecise Effort Estimates: Software development needs an overall roadmap. If requirements are unclear, it can lead into excess roadmap deviations which also lead into going over the estimated budget for the development phase.
- Reliance on tacit knowledge: Due to communication deficits and improper documentation practices, some requirements remain known only to certain team members, rather than being shared across the entire team.

For the purpose of increased clarity within the development phase appropriate planning phases are vital [11]. Without rigid management structure in agile teams, it becomes even more important to have planning phases at the start of each sprint, where design choices are made collaboratively as the whole team, and tacit knowledge issues are mitigated. These planning sections are vital for the agile team to function, and require optimal communication practices [2, 11].

Current state of the art has established that requirements engineering faces multiple problems [15]. In a study covering 354 organizations from 10 different countries, it was noticed that “communication flaws within the project team” and “communication flaws between the project team and the customer” are two leading factors in requirements engineering [9]. These two leading factors contribute to the issues on clarity of requirements.

The issue on requirements clarity is also rooted back to the self organizing nature of agile teams. While research has identified inherent difficulties in requirements engineering related to clarity, the additional reliance on communication among agile team members, as seen in self organizing teams, introduces a further layer of complexity.

3 RESEARCH QUESTION

Self organizing teams in agile methodology and requirements clarity are both extensively researched in the literature [2, 4, 8, 11]. However, these studies primarily focus on identifying the issues behind self organizing teams and requirements engineering, respectively. Although the keyword *communication* appears frequently in both areas of literature, it has rarely been explored conjunctively [9, 17]. Despite communication being a recurring theme in studies of both self organizing teams and requirements engineering, research seldom investigates the specific communication practices agile teams currently use to maintain clarity of requirements dynamically.

This research addresses this gap by examining how communication practices influence requirements clarity in agile teams. Maintaining clarity dynamically is essential in fast-paced, evolving agile environments, where misinterpretations, delays, and misaligned priorities can significantly undermine the methodology’s effectiveness.

It is unclear how agile teams can effectively communicate due to their unique self organizing nature, to minimize clarity issues related to requirements. The research question aims at an initial examination of understanding the communication practices utilized within self organizing agile teams, and understanding clarity issues unique to agile teams, in contrast to software development teams as a whole.

Research Question

- What communication practices between Agile team members contribute to maintaining clarity of requirements?

Sub Research Questions

- What are the primary causes of ambiguity in requirements during Agile iterations?
- How do team members communicate between each other to mitigate clarity issues?

4 RESEARCH METHODOLOGY

This section explains the research process and participant selection criteria, offering a structured overview of the methodological framework. The chosen framework, grounded theory, is explained in terms of how it was operationalized, along with relevant information about the interviewees who form the basis of this research.

4.1 Process

The research focuses on gathering qualitative data to explore agile teams through in-depth interviews [16, 18]. The study focuses on a narrow research goal and serves an exploratory purpose, hence five people from various agile software development teams have been chosen and deemed sufficient.

This qualitative approach is well-suited for exploring complex behavioral processes such as communication in self organizing teams. Grounded theory was chosen for this research as it provides a systematic framework to identify key insights and organize them into a narrative through utilizing the experiences of the interviewees, which is then utilized to investigate the phenomena [3, 21]. In this case, identifying the core communication practices within the agile team which contributes to clarity of requirements.

The research has been conducted through semi-structured interviews with key stakeholders within the Agile team: including developer and product owners. Semi-structured interviews are made through utilizing both closed and open ended questions. The open-ended questions are crucial for the qualitative analysis of the codes that will be derived from recurring themes within the answers [1]. The semi structured nature of the interview also allows the questions to potentially adapt to the ongoing direction of the answers which helps the exploratory nature of the interview [1].

The data was analyzed using grounded theory through a systematic three-step coding process aimed at generating themes and understanding their intersections. This approach aligns with the core principle of grounded theory, where themes emerge inductively from qualitative data [3, 21]. The process was operationalized starting with open coding, where interview transcripts were reviewed line by line to identify initial concepts and recurring patterns across the five interviewees. Codes were assigned to key phrases and insights, and closely related codes were grouped into themes that emerged inductively. Axial coding followed, connecting these themes by examining causal relationships and identifying how tangentially related codes influenced one another, as indicated by participant responses. Finally, selective coding was employed to develop and refine a central theme, integrating findings into a cohesive narrative that emphasizes the most interconnected and significant aspects of the results [10, 21].

The findings were then analyzed in comparison to the systemic findings documented in the existing literature. This comparative approach seeks to highlight areas of convergence and divergence while modeling these insights to enhance understanding of the studied phenomenon.

4.2 Participant Selection

Participants were invited through existing personal networks, as well as additional acquaintances connected to these networks. Choosing from this network has enabled more open and effective interviews. Through this process, five participants across different agile teams have been gathered for the study. The participants selected for the study include developers with diverse roles, ranging from full-stack development to specialized front-end and back-end development. Additionally, a product owner, responsible for leading an agile team and overseeing product development, was included. The analysis was conducted without consideration of job-specific differences among participants, as these distinctions were deemed outside the scope of the study and not central to its core objectives.

The participants were required to have at least two years of experience with the agile methodology in software development to ensure their familiarity with its principles and practices. Each interview lasted between 50 and 60 minutes on average, resulting in approximately five hours of total interview data collected across all participants. Table 1 below summarizes the key demographic and role related information for each interviewee.

Table 1. Interviewee Demographic Data

ID	Agile Role	Industry Experience	Agile Experience	Country
1	Developer	6 years	5 years	Turkey
2	Developer	6 years	6 years	Turkey
3	Developer	5 years	2 years	Turkey
4	Product Owner	29 years	10 years	Turkey
5	Developer	3 years	2 years	Turkey

5 RESULTS

This section presents the key findings from the qualitative analysis conducted through grounded theory. The analysis is segmented through the emerging themes within each overarching theme. The results section itself also has been segmented into the specified operationalization of grounded theory: open coding, axial coding and selective coding.

5.1 Open Coding

Through this process, two overarching themes were identified. Each overarching theme encompasses multiple themes, and each theme is further associated with specific codes [10].

Overarching Theme: Team Dynamics

Effective team dynamics are crucial for the success of agile teams, and this overarching theme explores the various factors that shape how the team operates and interacts. Figure 1 below is added to help visualize the overarching theme as a whole, where the themes and the corresponding codes are shown.

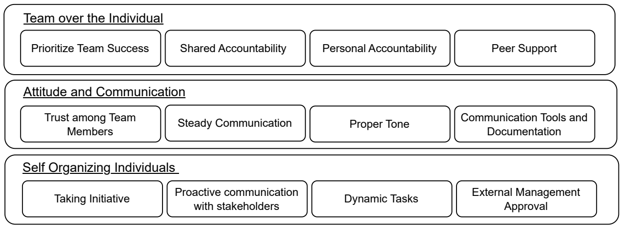

Fig. 1. Overview of Overarching Theme - Team Dynamics

Team Over The Individual

The first theme emphasizes the importance of prioritizing the team as a whole over individual wants and needs.

Some requirements are subjective by definition, and how to implement them more often than not depends on the preferences of the developer. It is important to discuss with team members and focus on the needs of the team and the project, instead of the wants and preferences of the specific developer (Interviewee 3 and 5). Prioritization of the successful delivery of the product was a common sentiment between the interviewees.

Building on the collective focus on team success, another two themes that have directly emerged were shared accountability and personal accountability. One participant explained, *"Within the sprints, we have an initial roadmapping and tracking progress for the team's goals"* (Interviewee 3). This highlights how the team collectively plans and monitors their progress, reinforcing the principle of shared accountability. However, while shared accountability is an important aspect of team dynamics, conflicts can arise when individuals are reluctant to take personal accountability for decisions made collectively by the team. *"Some people can be self-centered and try to frame the problem on others instead of themselves"* (Interviewee 4). These types of behaviour can jeopardize the integrity of the team, and lower morale. It seems to be a potential issue that agile teams can face, where it can be difficult to balance and clearly define individual responsibility within a shared accountability framework.

One factor that seemed to increase team morale and emphasize team cooperation was peer support. *"What is important is being in a* *healthy team environment, when needed more experienced members help others, we also implement pair programming"* (Interviewee 1). Relying on other team members and supporting one another seems to be a core factor of team dynamics in agile development. Facilitating strong peer support systems, especially for developers, can result in more mutual trust which could implicitly result in better productivity and achieving intended team results.

Attitude and Communication

The second theme highlights the crucial role of attitude and communication in fostering effective collaboration, trust, and clarity within agile teams.

One of the initial codes derived for this theme was trust among team members. Agile teams rely heavily on collaboration, and a lack of trust can disrupt this dynamic. If a team member is perceived as uncollaborative, unreliable, or disengaged in their work, it can lead to communication breakdowns within the team (Interviewee 1).

Steady communication was highlighted to be a key factor for the work environment in an agile team. *"Irregular communication is the biggest problem [that an agile team can face]"* (Interviewee 2). Where non steady communication would lead into clarity issues for requirements. However this communication should be conducted in a manner which facilitates good team behaviour. One of the unexpected results came from this section, where all five interviewees mentioned that they are usually with a casual tone when it comes to communication. A participant mentioned that *"The casual tone is way more prevalent when we are talking with other developers, as we are more often than not, always together"* (Interviewee 1). This casual tone results in more clear and to the point communication, and results in managing requirements with higher clarity.

Regarding the communication method and documentation, it was a common answer that tools such as Jira, Slack or Microsoft Teams were utilized. In addition, it was stated that *"Nothing that hasn't yet been added to the documentation can be added to the software"* (Interviewee 4), highlighting the importance of keeping all changes documented. Ensuring that updates are consistently tracked and accessible to the entire team is crucial for maintaining clarity and alignment throughout development. This emphasizes that without proper documentation, new features or changes risk being overlooked or misunderstood, which can lead to misalignment with the intended deliverable.

Self Organizing Individuals

This third theme delves into the self organizing nature of agile teams, a cornerstone of the agile methodology.

An initial code that quickly emerged was team members taking initiative. The dynamic nature of agile, compared to waterfall, demands greater proactivity as requirements often change during development (Interviewees 1, 2, 4). Poor time estimates or client-requested changes require developers to adapt their tasks and support the team, emphasizing the importance of initiative in managing urgent tasks. Another participant highlighted, *"It is never truly possible to be fully sure on the [implementation] of the requirements"* (Interviewee 3). This sentiment was also shared by Interviewee 1, and similar to an extent with also the rest of the participants. *"The developers creating the product might have a different expectation than the stakeholders meant to use the product."* (Interviewee 4). Indicating the difficulty in understanding product expectations, even though there is already a paramount amount of importance given to requirements elicitation within the agile team.

"Feedback sessions are crucial" (Interviewee 4), without being proactive with the client and stakeholders, and actually getting feedback for the implementation; it was seen as very hard to get tangible response on the development. So this taking initiative by the self organizing team happens similarly when it comes to contact and feedback sessions by the relevant stakeholders. Whether it be an early prototype or simple wireframes it seems urgent to make sure the implementations are on the right track with what the stakeholders truly need.

Both of these codes that revolved around need for initiative and need for communication both within the team and with the stakeholders resulted in the specific code for dynamic tasks. It was a common fact between all of the participants that due to many reasons listed before, such as client need changes, misunderstanding requirements prior and finding a better fitting implementation tasks could change dynamically within the sprint.

Lastly, the participants had an interesting answer when asked to evaluate how much of a reliance they had on external management and if they had viewed their agile team as self organizing. While all five participants viewed their team as self organizing, they had varying amounts of reliance on external management. One participant mentioned, we have to get an approval from the product owner and come into a consensus from within the agile team (Interviewee 5), which was a good definition of self organizing team. However, while the rest of the interviewee's still viewed themselves as highly self organizing, they had to rely on external management for approvals before certain changes. Which showed a mismatch of what was viewed as self organizing, and that the majority of the participants relied on external oversight and weren't fully autonomous.

Overarching Theme: Requirements Clarity

Clear and well-defined requirements are fundamental to the success of Agile teams. This overarching theme examines the challenges and strategies related to understanding, refining, and aligning requirements in a dynamic development environment. Figure 2 below is added to help visualize the overarching theme as a whole, where the themes and the corresponding codes are shown.

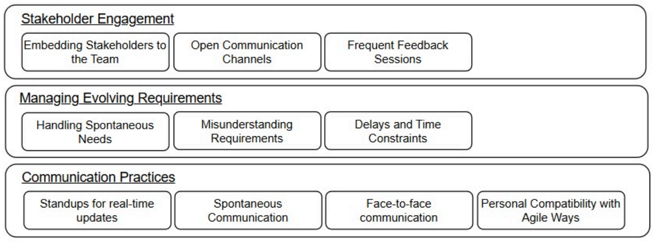
Stakeholder Engagement
Embedding Stakeholders to the Team
Open Communication Channels
Frequent Feedback Sessions
Managing Evolving Requirements
Handling Spontaneous Needs
Misunderstanding Requirements
Delays and Time Constraints
Communication Practices
Standups for real-time updates
Spontaneous Communication
Face-to-face communication
Personal Compatibility with Agile Ways

Fig. 2. Overview of Overarching Theme - Requirements Clarity

Stakeholder Engagement

This first theme emphasizes the crucial role of actively involving stakeholders to ensure alignment with intended product goals.

Embedding stakeholders to the team was a common occurrence for all five interviewees. According to one of the participants, *“The first point of contact is always the Product Owner [in terms of clarity issues in requirements].”* (Interviewee 2). The product owner, for the agile team, is the voice of the stakeholders. They make sure development is on track, and that there are no tangible issues with the implementation of the requirements. With the insights from the participants, the product owner could be seen as the backbone of the agile team.

Maintaining open communication channels with stakeholders was regarded as essential. Gathering feedback was identified as one of the most effective ways to ensure that product development not only adheres to timelines but also aligns correctly with the intended implementation of the requirements. These feedback sessions play a crucial role in validating and clarifying the product requirements, ensuring alignment with stakeholder expectations.

As much as the way of communicating and engaging with the stakeholders, for agile practices the frequency is also very important. The product owner was always available for all five of the participants' agile teams. As each sprint revolves around new additional requirements, it is important to get new feedback from the product owner. If there are problems with clarity of requirements, a new meeting with the product owner would be organized (Interviewee 3).

Managing Evolving Requirements

This second theme explores the challenges of navigating evolving requirements within the agile team.

With the requirements not being set to be definitive, it had become a common occurrence in all five of the agile teams to handle spontaneous needs. A participant indicated, *“If a requirement was too vague and we had implemented it poorly, we in a new meeting decide how to move forward with it, can we fix it or should it be redone in a completely new direction. . . ”* (Interviewee 1). If something goes wrong within the sprint, which it seems something going wrong is almost part of software development [which has an intended specific use case, where the stakeholders are different from the developer], there comes a point where the focus of the agile team changes into fixing this mistake.

One of the main reasons for clarity issues within the requirements was explained as misunderstanding the requirements. Misunderstandings could occur due to either understanding stakeholder needs poorly (Interviewee 1) where product requirements weren't understood clearly enough. Another reason was understanding stakeholder needs wrongly (Interviewee 3), where implementations weren't in the way the client had expected. These reasons in return cause the need for fixing prior misunderstood requirements.

Another participant explained that, *“Misunderstanding the time cost of each requirement can result in delays within the development cycle”* (Interviewee 5). So it could be understood that other than the clarity of requirements themselves, there was a clarity of time issues. Where developers, especially not accustomed to the current product being developed, can estimate time costs poorly resulting into potential time delays in deliverables.

Communication Practices

“Spontaneous communication becomes a necessity when issues emerge” (Interviewee 3). Even though the mass majority of communication happens with pre-planned stand-ups, spontaneous communication was a common sentiment (Interviewee 3, 4, 5). The frequency of spontaneous communication also depended on their work environment (Interviewee 1). This suggests that agile adaptation varies significantly between companies and that perfectly implementing agile practices can be challenging.

The stand-ups either happened weekly (Interviewee 1, 3) or daily (Interviewee 2, 4, 5). This was simply due to a difference in company structure. Indicating that there isn't a set perfect amount of meetings with the teams to be had. These stand-ups consisted of real-time updates on what new requirements were being handled and how the progress has been. Requirements clarity issues could also come up in these stand-ups, where if there was any ambiguity, relevant stakeholders and the product owner would come into place for assistance.

All of these stand-ups were conducted physically, in person, face to face for all five participants. An interesting information relayed was, the occurrence of recording stand-ups and making written notes (Interviewee 4). This was for the purpose of making sure key information within the meetings wouldn't get forgotten, which would potentially cause clarity issues later on.

Lastly, in regards to communication practices, while never mentioned explicitly by any of the interviewees, it was derived that there was a general need for “Personal Compatibility with Agile Ways”. Where if any agile team member wasn't communicative enough, relaying important information to other team members, and making sure they do not have any ambiguities when it comes to requirements; it would lead into problems for the whole team.

"Everyone's effort matters [within the agile team]" (Interviewee 5), and a single individual could cause the whole team to have problems. *"Lack of dedication [to the team] and lack of communication are two big problems"* (Interviewee 1). Where a commitment to the agile team, the product and good communication skills can be seen as vital for successful products.

5.2 Axial Coding

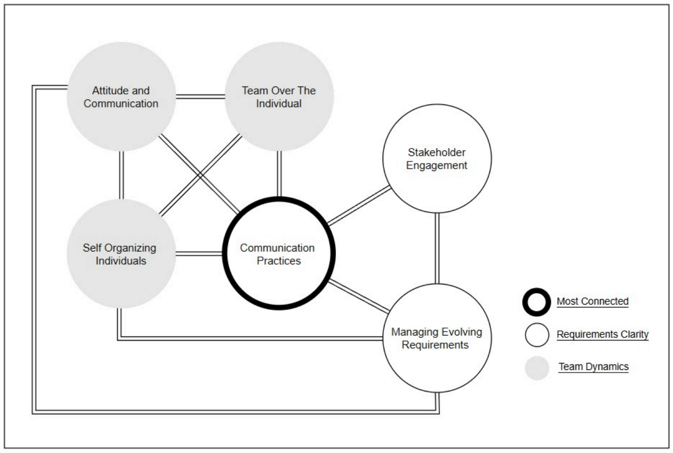

Fig. 3. Undirected Dependence Graph of Themes

Inter-dependencies between themes were identified throughout the open coding process and have been mapped as seen in Figure 3. These inter-dependencies were considered undirected, as themes influence each other reciprocally, without a clear hierarchy or primary direction of dependence . On average, one theme is interlinked with 3.5 other themes. This value indicates that the derived themes are frequently interconnected, even when different codes are associated with different themes.

The three themes derived within the first overarching theme were observed to be strongly connected to each other. Specifically, **Attitude and Communication**, **Team Over The Individual**, and **Self Organizing Individuals** all influenced each other significantly. These three aspects collectively form the foundation of an effective agile team, one that is capable of co-existing harmoniously among its members while working autonomously. Such teams thrive on strong communication skills, the right attitude, and prioritizing the benefit of the team over individual interests, as explained in the open-coding section prior.

Similarly, the themes within the second overarching theme were seen as strongly connected with each other, where **Communication Practices, Stakeholder Engagement** and **Managing Evolving Requirements** were all dependent on each other. **Attitude and Communication** and **Managing Evolving Requirements** were seen to be connected due to steady communication, documentation and communication tools being strong factors in being able to **Manage Evolving Requirements**.

The most interesting insight from the axial coding process was deriving that **Communication Practices** was the foundation of this graph, where it affected all other themes. Without the building blocks of **Communication Practices**, which were stand-ups for real-time updates, spontaneous and face-to-face communication together with a unique aspect that we had derived, which required team members to be personally compatible with agile ways. Without these building blocks, none of the other themes would be maintainable within the agile team. Hence **Communication Practices** has been identified as the most connected and the central theme of the analysis.

5.3 Selective Coding

The axial coding step has revealed that **Communication Practices** acts as the central theme, unifying and being the building block of the agile team. Effective communication practices such as preplanned stand-ups together with spontaneous meetings form the communication layer of the team. This information flow is vital for the stakeholder needs to be met. As a poor communication layer results in ambiguity in requirements, as highlighted by the overarching theme, requirements clarity. This communication should preferably happen face to face, physically; as it results in information being relayed more efficiently, facilitating less loss in clarity of requirements.

These communication practices don't only affect requirements clarity but also are a deep part of team dynamics. Where the casual tone, and sense of prioritizing the team over the individuals, creates a sense of cohesion and unity within the agile team. This results in being much more efficient as a team, which enables tackling the dynamic tasks within the agile development much more effectively. Peer support and a shared accountability within the team were seen as important factors within this team dynamic.

This communication layer also is aggregated to the feedback loop from the relevant stakeholders, which results in better clarity of requirements, and delivery of the intended product with minimum delay within the process. Ultimately, without robust communication practices, the foundational principles of agile such as self organization, responsiveness to change, and alignment with stakeholder goals would falter.

6 DISCUSSION

This section synthesizes the key findings, aligning them with existing literature and the research question. The section is segmented in relation to the prior defined themes. Limitations and avenues for future work are also discussed.

6.1 Key Insights and Comparison with Literature Review

6.1.1 ***Team Over The Individual.*** The findings reinforce prior established principles such as prioritizing team goals over individual preferences as critical, which is one of the foundations of self organizing teams [18, 23]. Interviewees emphasized shared accountability and collective planning through sprints, which aligns with the literature on self organizing teams valuing team cohesion and handling tasks together [18]. However, the challenge of balancing shared accountability with personal responsibility emerged as a potential risk to team morale and effectiveness, highlighting a nuanced tension that agile teams must manage carefully. This echoes prior tangential findings on the difficulties of clearly defining individual roles within self organizing frameworks and delegating clear boundaries within the team. This is an inherent characteristic of agile teams, as they function as a cohesive unit, this in return requires maturity and good communication skills by the agile members. These insights show how prioritizing team cohesion supports communication practices that keep requirements clear, directly answering the research question about communication's role in maintaining requirements clarity in agile teams.

6.1.2 *Attitude and Communication.*

Steady communication and trust were highlighted as key aspects of the team dynamics, confirming the importance of these factors in agile settings [4, 17]. Steady communication revolves around having open communication channels at all times, which is crucial as within agile methodology the requirements are ever evolving. The prevalence of a casual tone in developer interactions was a noteworthy insight, suggesting that informality may foster clearer, more direct exchanges and potentially cultivating a more healthy team dynamic. This aspect seems to be a common occurrence within agile methodology in practice but seems to be highlighted less commonly in existing research. Consistent with prior research, tools like Jira and communication tools such as Slack and Microsoft Teams play a vital role in documenting decisions and maintaining requirements clarity through open communication channels [17]. These tools and the informal, open communication practices directly support the research question by enabling clear, fast information flow, which is crucial to maintaining requirements clarity.

However the need for extensive documentation as highlighted by the interviewees introduces a tension with the theoretical framework of agile [4]. Agile is meant to replace the slow paced documentation process with fast-paced in person communication, however the interviewees mentioned that while they utilized the in-person communication they were still heavily relying on documentation. This had indicated some agile practices weren't being utilized appropriately, where a middle ground approach has been taken between traditional and agile methodologies. These insights highlight the existing framework of need to combine informal, face-to-face communication which is inline with agile practices, however structured and extensive documentation to mitigate clarity issues in requirements seemed to have been employed within these agile teams. This approach appears to be a deliberate strategy to prevent information loss during communication, thereby helping to maintain clarity of requirements even though it fundamentally slows down the fast-paced agile work environment.

6.1.3 *Self Organizing Individuals.*

The self organizing nature of agile teams was evident through all five participants, particularly regarding taking initiative and adapting to dynamic tasks during development. This aligns with prior studies which highlight self organizing individuals within product development [23]. However, the majority of the participants, while mostly autonomous, were stating reliance on external management for various needs for approvals within the process. Interestingly enough, when asked if they had viewed themselves as self organizing; they all rated their agile team highly. This highlights that the majority of participants aren't aware of theoretically correct applications of agile, and only are aware of the agile practices they have witnessed through their work environment. The observed reliance on external management approvals reveals a partial disconnect between theoretical self organization and practical realities. These findings also differ from literature which focuses on theoretical applications instead of practical realities that can occur such as within this participant pool [18, 22]. This means communication practices must balance individual autonomy with management input, highlighting how agile teams maintain clarity in real-world settings, giving a key insight related to the research question.

6.1.4 *Stakeholder Engagement.*

The presence of a dedicated product owner who acted as a communication bridge between the development team and the stakeholders was a recurring theme. This resulted in the development team having a first point of contact in times of requirements clarity issues, which in turn resulted in accelerating time losses that might occur when the development team isn't aware of how to progress with the implementation that would fit the requirement specification that would be with what the client and stakeholders expect. Actively engaging with relevant stakeholders was seen as a crucial way of validating that requirements were well understood by the development team similar to previous research findings [9, 11].

However, interviewees noted challenges such as stakeholders' occasional inability to articulate their needs effectively or their own inability of understanding what the client might necessarily expect, as requirements tend to be vague and not set in stone in early development phases. Frequent feedback sessions from relevant stakeholders, perhaps with relevant wireframing or minimal prototypes would accelerate the process of validating requirements clarity and whether the agile team is on the right track. This engagement is part of the vital communication pathways which helps maintain requirements clarity, directly addressing the research question's focus on effective communication and constant feedback for the iterative development cycles inherent to agile.

6.1.5 *Managing Evolving Requirements.*

Handling evolving requirements was consistently identified through all of the participants as a key issue within development. Requirements could be seen initially clear by the developers, but they might realize later on that they actually didn't fully grasp how it should be implemented. There were also issues with time estimates for requirements, which could result in time constraints and potential delays. The need to revise ambiguous requirements confirms the iterative basis of agile development [13]. These findings reinforce the critical need for ongoing requirement validation throughout sprints and making sure that while requirements evolve, the understanding of the requirements by developers should promptly follow. Effectively managing these continuous validation cycles with minimal time loss demonstrates that communication is essential for maintaining clarity during evolving requirements, directly addressing the research question.

6.1.6 ***Communication Practices**.* The amount of preplanned stand-ups supplemented by spontaneous ways of communicating with the team highlights the importance of communication and more so the need for fast paced communication channels. Requirements evolve throughout the sprint, and the new information should be both documented and also relayed to everyone without ambiguity through group meetings, preferably physically. All participants mentioned their agile teams utilized face to face meetings, highlighting the importance of being available in person, physically. This was similar to relevant literature which puts emphasis on working in close proximity to each other, where ambiguities could be resolved through approaching each other, which is only possible through working physically together [4]. The practice of recording stand-up key information to preserve information and combat requirements clarity issues that might arise from loss of knowledge. These practices complement available documented best practices in the field [17]. The implicitly derived need for personal compatibility and commitment to the team also highlighted the human factors critical to communication effectiveness which would result in higher clarity of requirements, which are less emphasized in formal models [16]. Such formal and informal communication practices directly support the research question by ensuring timely and clear information sharing to maintain requirements clarity. These communication practices also directly affect the other themes discussed as shown prior in Figure 3 during axial coding and explained in selective coding, showing that good communication practices are fundamental to answering the research question outlined.

6.2 Answering Research Question

6.2.1 Research Question: What communication practices between Agile team members contribute to maintaining clarity of requirements?

The communication practices within the agile team revolve around creating a steady communication line. The agile team aims to be able to spontaneously ask each other questions in times of ambiguity in requirements. To maintain clarity in requirements the agile team needs to ensure a fast paced in-person communication channel which is inline with the agile framework. A divergence from the theory was found where documentation is given priority and no action is taken without making it common knowledge for the whole team. This ensures there is no loss of key information, that could result in loss of clarity in requirements later on. This only addresses the aspect of maintaining existing information within the team. To obtain high clarity of requirements, the agile team similar to within the agile team also maintains open communication channels and embeds stakeholders, such as through the product owner. This ensures a steady channel of communication if loss of clarity occurs, it can be efficiently resolved by obtaining clarification from relevant stakeholders.

6.2.2 Sub Research Question 1. What are the primary causes of ambiguity in requirements during Agile iterations?

The primary causes of ambiguity in requirements stems from ever evolving and ambiguous stakeholder needs as well as misunderstandings that can happen within the agile team. Participants indicated that it is extremely hard to be sure that the current list of requirements are actually the requirements desired by the stakeholders. Misunderstandings can also occur if the team isn't diligent in communicating steadily and effectively. Communication issues is the primary issue that is behind requirements clarity issues. This communication issue is multifaceted. It can stem from the stakeholders needs not being properly conveyed to the agile team, or the agile team might have difficulties in properly understanding the requirements. The bigger issue is, it isn't always simply to even be aware of the fact that requirements might be misunderstood, as these misunderstandings might surface later on when stakeholders or product owners give negative feedback to the development team.

6.2.3 Sub Research Question 2. How do team members communicate between each other to mitigate clarity issues?

To mitigate clarity issues, the agile team employs a combination of preplanned and spontaneous communication methods which is also supplemented by a rigorous documentation process. The preplanned stand-ups are a result of steady communication being a requirement of the agile methodology, To keep up with ever evolving requirements the agile members should have open communication channels at all times, which is what enables the spontaneous communication layer. Aside from simple mitigation, the product owner plays a bridge like role between relaying stakeholder needs to the agile team members. It was indicated that casual tone is the preferable communication tone. This relaxed attitude positively effects team dynamics to become more cohesive and also efficient. As there is need for everyone within the team to relay information to each other within this dynamic development ecosystem and mitigate clarity issues.

6.3 Limitations and Future Work

The quantity of participants, while deemed sufficient for the exploratory work, limits the generalization of the insights due to potential overlap and biases inherent within the chosen participants. It should be noticed that this study is behavioral in essence and work culture of the country and the company affect the collected data. For future work, it is advised to increase the participant pool and introduce new variables within the study, such as examining differences between different company cultures and also effects of geographical location as it is a big factor in work environment and inherent culture. While also not utilized within this study, it would be interesting to analyze the communication differences between different roles within the agile team and depending on the industry which they work in. Within this study, most of the participants were of developer background the results primarily reflect the developers' perspective within the team. These variables, if utilized, would compound the length of the analysis. It would be beneficial to explore these additional variables separately to gain clearer exploratory insights into their effects on communication. These can in future be synthesized into larger models.

7 CONCLUSION

For the purpose of exploring the role of communication in clarity of requirements within Agile software development, five semi-structured interviews have been conducted. The data from the interviews have been utilized through grounded theory. It has been validated that communication is a core aspect of Agile development and it has various dimensions in mitigating clarity issues in requirements.

Some of these are the importance of real-time updates for managing evolving requirements. Stand-ups are one of the steady communication layers and are supplemented by spontaneous communication that can occur anytime clarity issues arise within the development phase. Communication should ideally occur physically, with developers present together during work to enable face-to-face interactions. Personal compatibility with Agile practices was also highlighted, where Agile individuals need to be open to working within this communication-heavy method. The findings indicate that a casual tone is more appropriate for relaying information efficiently. Documentation was also seen as crucial for mitigating the loss of information, which could result in a loss of clarity in requirements.

While these findings highlight the importance of communication and suggest the need for a broader analysis, including a model for applying these findings in practice within agile teams, it is also important to recognize the need for further work, including the incorporation of external variables such as company culture and geographical culture into the framework.

REFERENCES

[1] William C. Adams. 2015. Conducting Semi-Structured Interviews. In *Handbook of Practical Program Evaluation* (1 ed.), Kathryn E. Newcomer, Harry P. Hatry, and Joseph S. Wholey (Eds.). Wiley, 492–505. https://doi.org/10.1002/9781119171386.ch19

[2] Marco A. Boschetti, Matteo Golfarelli, Stefano Rizzi, and Elisa Turricchia. 2014. A Lagrangian heuristic for sprint planning in agile software development. *Computers & Operations Research* 43 (March 2014), 116–128. https://doi.org/10.1016/j.cor.2013.09.007

[3] Kathy Charmaz. 2014. *Constructing grounded theory* (2. ed ed.). SAGE, Los Angeles, Calif.

[4] A. Cockburn and J. Highsmith. 2001. Agile software development, the people factor. *Computer* 34, 11 (Nov. 2001), 131–133. https://doi.org/10.1109/2.963450

[5] A. De Meyer, C.H. Loch, and M.T. Pich. 2002. Managing project uncertainty: from variation to chaos. *IEEE Engineering Management Review* 30, 3 (2002), 91–91. https://doi.org/10.1109/EMR.2002.1032403

[6] Tore Dybå and Torgeir Dingsøyr. 2008. Empirical studies of agile software development: A systematic review. *Information and Software Technology* 50, 9-10 (Aug. 2008), 833–859. https://doi.org/10.1016/j.infsof.2008.01.006

[7] John Erickson, Kalle Lyytinen, and Keng Siau. 2005. Agile Modeling, Agile Software Development, and Extreme Programming: The State of Research. *Journal of Database Management* 16, 4 (Oct. 2005), 88–100. https://doi.org/10.4018/jdm.2005100105

[8] Ismael Edrein Espinosa-Curiel, Josefina Rodríguez-Jacobo, Erika Vázquez-Alfaro, José Alberto Fernández-Zepeda, and Daniel Fajardo-Delgado. 2018. Analysis of the changes in communication and social interactions during the transformation of a traditional team into an agile team. *Journal of Software: Evolution and Process* 30, 9 (Sept. 2018), e1946. https://doi.org/10.1002/smr.1946

[9] D. Méndez Fernández, S. Wagner, M. Kalinowski, M. Felderer, P. Mafra, A. Vetrò, T. Conte, M. T. Christiansson, D. Greer, C. Lassenius, T. Männistö, M. Nayabi, M. Oivo, B. Penzenstadler, D. Pfahl, R. Prikladnicki, G. Ruhe, A. Schekelmann, S. Sen, R. Spinola, A. Tuzcu, J. L. de la Vara, and R. Wieringa. 2016. Naming the Pain in Requirements Engineering: Contemporary Problems, Causes, and Effects in Practice. (2016). https://doi.org/10.48550/ARXIV.1611.10288 Publisher: arXiv Version Number: 1.

[10] Barney G. Glaser and Anselm L. Strauss. 2017. *The Discovery of Grounded Theory: Strategies for Qualitative Research* (1 ed.). Routledge. https://doi.org/10.4324/9780203793206

[11] Matteo Golfarelli, Stefano Rizzi, and Elisa Turricchia. 2013. Multi-sprint planning and smooth replanning: An optimization model. *Journal of Systems and Software* 86, 9 (Sept. 2013), 2357–2370. https://doi.org/10.1016/j.jss.2013.04.028

[12] Petra Heck and Andy Zaidman. 2014. A Quality Framework for Agile Requirements: A Practitioner's Perspective. https://doi.org/10.48550/ARXIV.1406.4692 Version Number: 1.

[13] Ville T. Heikkilä, Daniela Damian, Casper Lassenius, and Maria Paasivaara. 2015. A Mapping Study on Requirements Engineering in Agile Software Development. In *2015 41st Euromicro Conference on Software Engineering and Advanced Applications*. 199–207. https://doi.org/10.1109/SEAA.2015.70 ISSN: 2376-9505.

[14] Rashina Hoda, James Noble, and Stuart Marshall. 2013. Self-Organizing Roles on Agile Software Development Teams. *IEEE Transactions on Software Engineering* 39, 3 (March 2013), 422–444. https://doi.org/10.1109/TSE.2012.30

[15] Irum Inayat, Siti Salwah Salim, Sabrina Marczak, Maya Daneva, and Shahaboddin Shamshirband. 2015. A systematic literature review on agile requirements engineering practices and challenges. *Computers in Human Behavior* 51 (Oct. 2015), 915–929. https://doi.org/10.1016/j.chb.2014.10.046

[16] B. L. Kirkman and B. Rosen. 1999. BEYOND SELF-MANAGEMENT: ANTECEDENTS AND CONSEQUENCES OF TEAM EMPOWERMENT. *Academy of Management Journal* 42, 1 (Feb. 1999), 58–74. https://doi.org/10.2307/256874

[17] Yngve Lindsjørn, Dag I.K. Sjøberg, Torgeir Dingsøyr, Gunnar R. Bergersen, and Tore Dybå. 2016. Teamwork quality and project success in software development: A survey of agile development teams. *Journal of Systems and Software* 122 (Dec. 2016), 274–286. https://doi.org/10.1016/j.jss.2016.09.028

[18] Nils Brede Moe, Torgeir Dings, and Tore Dyb. 2008. Understanding Self-Organizing Teams in Agile Software Development. In *19th Australian Conference on Software Engineering (aswec 2008)*. IEEE, Perth, Australia, 76–85. https://doi.org/10.1109/ASWEC.2008.4483195 ISSN: 1530-0803.

[19] Eva-Maria Schön, Jörg Thomaschewski, and María José Escalona. 2017. Agile Requirements Engineering: A systematic literature review. *Computer Standards & Interfaces* 49 (Jan. 2017), 79–91. https://doi.org/10.1016/j.csi.2016.08.011

[20] R. Steegh, K.Van De Voorde, J. Paauwe, and T. Peeters. 2025. The agile way of working and team adaptive performance: A goal-setting perspective. *Journal of Business Research* 189 (Feb. 2025), 115163. https://doi.org/10.1016/j.jbusres.2024.115163

[21] Anselm L. Strauss and Juliet M. Corbin. 1998. *Basics of qualitative research: techniques and procedures for developing grounded theory* (2nd ed ed.). Sage Publications, Thousand Oaks.

[22] Viktoria Stray, Nils Brede Moe, and Rashina Hoda. 2018. Autonomous agile teams: challenges and future directions for research. In *Proceedings of the 19th International Conference on Agile Software Development: Companion*. ACM, Porto Portugal, 1–5. https://doi.org/10.1145/3234152.3234182

[23] Hirotaka Takeuchi and Ikujiro Nonaka. 1986. The New New Product Development Game. *Harvard Business Review* (Jan. 1986). https://hbr.org/1986/01/the-new-new-product-development-game