Knowledge Mobilization in Agile Information Systems Projects: A Literature Analysis

Eli Hustad[1], Makoto Nakayama[2], Norma Sutcliffe[2] and Merri Beckfield[2]
[1]University of Agder, Norway
[2]DePaul University, Chicago, US
eli.hustad@uia.no
mnakayama@cdm.depaul.edu
nsutcliffe@cdm.depaul.edu
mbeckfie@cdm.depaul.edu

Abstract: This study focuses on how knowledge is mobilized in agile information systems (IS) projects. One crucial success factor of those projects is to mobilize knowledge through different knowledge management processes. It is vital to establish efficient knowledge management (KM) processes to generate a knowledge culture based on transparency and communication. Communication channels, digital tools, and platforms are essential for establishing a KM infrastructure supporting the knowledge work of the project organization. Thus, each IS implementation team should maintain a knowledge base and a knowledge potential at some level. However, this is not always the case. We conducted a literature review to survey the extant research on the role of KM in agile system development projects. The agile approach is often associated with the networking model and tacit knowledge. The findings indicate that the agile approach is supposed to promote KM. While tacit knowledge is rooted in the analogue process of continuous actions and informal communication, explicit knowledge is captured in digital records of documentation and databases. In KM, the *personalization* model (behavioural, networking) and the *codification* (technocratic, repository) model is central. The choice of system development method (agile versus plan-driven) influences how knowledge is mobilized in the project organization. An agile approach heavily relies on informal communication, tacit knowledge sharing, and light documentation. In contrast, the plan-driven methods such as the waterfall approach generate more explicit knowledge through documentation. Communities of practice are important structures for transforming from plan-driven to agile approaches. We present a framework showing specific challenges the literature identifies concerning the efficient mobilization of knowledge in the agile context. For large-scale agile projects, informal coordination mechanisms were important. This study identifies several measures for overcoming barriers and risks for knowledge sharing in the agile context.

Keywords: knowledge management, knowledge sharing, knowledge mobilization, agile software projects, large-scale information systems (IS) projects

1. Introduction

In the last two decades, the agile software development approach has gained much attention since the introduction of the agile manifesto (Fowler and Highsmith 2001). The emphasis has been on faster deliveries of working software to customers who are actively engaged with developers in an iterative process of guiding and shaping the end-product of an information system (Zykov and Singh 2020). This contrasts the traditional plan-driven methods with heavy documentation, pre-defined requirement specifications, and less involvement of customers after the design process (Dingsøyr et al. 2012; Li et al. 2010). The agile system development approach requires self-organizing and well-functioning teams and efficient collaboration and communication routines (Batra et al. 2017). This is not always easy to establish, especially when agile projects increase in scope and complexity involving distributed coordination of large-scale information systems (IS) projects (Santos et al. 2015).

It is an increasing interest among scholars and practitioners regarding the role of knowledge management (KM) in agile software projects (Ghobadi 2015; Maudal and Dingsøyr 2021; Ouriques et al. 2019). Software development is a highly innovative and knowledge-intensive activity, and KM is therefore an important perspective for understanding the knowledge processes going on during the system development life cycle (Nakayama et al. 2021). However, the ways that tacit and explicit knowledge is transferred and how knowledge is optimally utilized in an agile software setting are still not fully understood (Napoleão et al. 2021). Tacit knowledge and especially sharing of knowledge during stand-up meetings, pair programming, post-mortems, and informal communication in general, are embedded in the nature of the agile approach (Ouriques et al. 2019). By comparing rigorous documentation in plan-driven methods, the agile approach seeks to develop "just enough" documentation, and working software is ranked higher than documentation (Marinho et al. 2019). One KM perspective combines the personalization (behavioural, networking) and the codification (technocratic,

repository) to develop relevant practices, strategies, and appropriate tools to support knowledge work (Alavi et al. 2005). The personalization model pays attention to tacit knowledge and seeks to support knowledge-sharing and rich communication activities in different types of groups, teams, and communities of practice (e.g., system development teams, professional networks) by providing them with interactive communication tools, such as collaboration platforms, email, videoconference, and social software. The personalization model focuses on networking, collaboration, communication, and informal relations in organizations (Wenger et al. 2002). In contrast, the codification model is more technologically oriented and pays attention to explicit knowledge and how to codify knowledge by creating documents for reuse purposes through repositories. Regardless of knowledge forms, knowledge transfer relies on individual motivations, formal and informal networks, and social cohesion (Sarker et al. 2005).

This study seeks to understand how knowledge can be efficiently *mobilized* in an agile software setting comprising IS development. We develop the concept of *knowledge mobilization* to understand how different knowledge processes can come into play both within agile intra-team and inter-team settings. To develop this concept of knowledge mobilization, we conducted a literature review to integrate extant research to get an overview of empirical studies focusing on different knowledge management processes in an agile software setting. The following research question has guided our research study:

How can knowledge be mobilized in agile information systems projects?

Thus, we systematically reviewed studies of relevance to understand how previous research has addressed the role of KM in agile software development settings. We organize our findings into recurring themes and patterns identified in the literature to understand knowledge processes and barriers to knowledge sharing. We also aim to clarify how the agile context enables knowledge mobilization.

The paper is organized as follows. First, we present our research method for conducting a literature review. Next, we present our findings organized into six recurring themes. After that, we discuss implications for future research. Finally, we make some concluding remarks and present a framework for understanding knowledge mobilization in an agile context for IS development.

2. Method

A literature review provides an overview of existing knowledge by identifying and combining prior research studies and recommends a future research agenda. We followed the procedure and practical guidelines of a systematic literature review developed by Kitchenham (2004). The review conducted followed these steps: 1) Identify important literature reviews already existing in the field, develop a protocol with relevant search phrases, and define inclusion and exclusion criteria; 2) Conduct the review including empirical research studies, select relevant studies, perform quality evaluation, and integrate related topics and patterns; and 3) Summarize the review results and prepare a write-up. The results of our literature study include research studies published during the last decade (2012-2022). In addition, a backward search is performed not to overlook important studies in the last decade.

To identify relevant research studies, we used search phrases related to knowledge management and agile software development (Table 1). We used the Scopus database as the search engine. The search was performed on “title” and “keywords” on the specified literature topics. The inclusion criteria were empirical research studies published in outlets comprising research articles, conference papers, and book chapters. Especially studies focusing on the role of KM and knowledge processes related to teamwork, collaboration, coordination, and communication in agile software projects were included. In addition, studies focusing on specific organizational context/structure and culture facilitating knowledge mobilization were of interest. We excluded studies on computer science topics related to specific agile methods, programming languages, and algorithms. However, technical solutions and infrastructures supporting knowledge work and processes in the agile environment were of interest. The primary focus was to understand the socio-technical processes that are unfolding in agile software projects and how knowledge is mobilized in these settings to enhance team performance.

Table 1: Overview of search strings

Search engine	Scopus
Search phrases in combinations (KEYWORDS or TITLE)	*Knowledge-related* "knowledge management" OR "knowledge sharing" OR "knowledge transfer" OR " knowledge application" OR "knowledge creation" OR "knowledge process" OR "knowledge mobilization" AND *Agile-related* "agile software development" OR "agile system development" OR "agile software engineering" OR "agile software project"
Inclusion criteria	Peer-reviewed, empirical studies, English language, research articles, conference papers, book sections Topics: agile software development contexts, socio-technical issues, team dynamics, collaboration, coordination, communication, culture, structure, knowledge processes, facilitating knowledge work, supporting technologies
Exclusion criteria	Books, conference reviews Topics: Specific programming languages and code, algorithm development and pure technological focus (e.g., in computer science, and part of software engineering)

Research in knowledge management and agile system development is multidisciplinary. Therefore relevant topics recurred in several subject areas comprising information systems, business, management and accounting, social sciences, knowledge management, software engineering, and computer science. The initial search provided 88 publications encompassing 65 conference papers, 22 journal articles, and one book section.

The abstracts of these papers were read to select relevant studies for the research question. The inclusion/exclusion criteria were used to select relevant studies or remove studies that did not match the topics of interest. After this step, we ended up with a shortlist of 54 publications, including 37 empirical studies, eight literature reviews, and nine conceptual papers. The empirical studies comprised 15 journal papers and 22 conference papers. We decided to focus on the journal papers to ensure the richness of the empirical studies.

The contents of these journal publications were completely checked, and both the inclusion and exclusion criteria guided this selection step. In addition, the quality of the research studies reported was carefully reviewed, and especially the research methods applied were evaluated. Only one of the journal papers at this stage did not match our topic of interest. After this step, we had a final body of 14 journal publications to analyse in-depth (Table 2).

These empirical studies comprise rigorous studies with rich material and are the subject of deeper analysis. The literature review studies and the conceptual studies identified were utilized to get an overview of the field. The limited number of empirical studies published in highly acclaimed journals indicates that the research area of knowledge management in relation to agile software development is an immature research area, and conference papers are dominant per se.

Table 2: List of selected empirical publications

#	Reference
	Empirical studies
1	Annosi, M. C., Magnusson, M., Martini, A., and Appio, F. P. 2016. "Social Conduct, Learning and Innovation: An Abductive Study of the Dark Side of Agile Software Development," *Creativity and Innovation Management* (25:4), pp. 515-535.
2	Batra, D., Xia, W., and Zhang, M. 2017. "Collaboration in Agile Software Development: Concept and Dimensions," *Communications of the Association for Information Systems* (41:1), pp. 429-449.
3	Dingsøyr, T., Moe, N. B., Fægri, T. E., and Seim, E. A. 2018. "Exploring Software Development at the Very Large-Scale: A Revelatory Case Study and Research Agenda for Agile Method Adaptation," *Empirical Software Engineering* (23:1), pp. 490-520.
4	Ghobadi, S., and Mathiassen, L. 2017. "Risks to Effective Knowledge Sharing in Agile Software Teams: A Model for Assessing and Mitigating Risks," *Information Systems Journal* (27:6), pp. 699-731.
5	Ghobadi, S., and Mathiassen, L. 2016. "Perceived Barriers to Effective Knowledge Sharing in Agile Software Teams," *Information systems journal* (26:2), pp. 95-125.
6	Gregory, P., Barroca, L., Sharp, H., Deshpande, A., and Taylor, K. 2016. "The Challenges That Challenge: Engaging with Agile Practitioners' Concerns," *Information and Software Technology* (77), pp. 92-104.
7	Heeager, L. T., and Nielsen, P. A. 2017. "Intrafirm Knowledge Transfer of Agile Software Practices: Barriers and Their Relations," *Journal of Information Technology Case and Application Research* (19:4), pp. 199-224.
8	Heredia, A., Garcia-Guzman, J., Amescua-Seco, A., and Serrano, A. 2014. "Study of Factors Influencing the Adoption of Agile Processes When Using Wikis," *International Journal of Software Engineering and Knowledge Engineering* (24:6), pp. 859-885.
9	Khalil, C., and Khalil, S. 2020. "Exploring Knowledge Management in Agile Software Development Organizations," *International Entrepreneurship and Management Journal* (16:2), pp. 555-569.
10	Khoza, L. T., and Bwalya, K. J. 2021. "An Insider's Perspective of Knowledge Sharing in Software Development Projects," *Journal of Information and Knowledge Management* (20:3), pp. 1-17.
11	Paasivaara, M., and Lassenius, C. 2014. "Communities of Practice in a Large Distributed Agile Software Development Organization - Case Ericsson," *Information and Software Technology* (56:12), pp. 1556-1577.
12	Santos, V., Goldman, A., and de Souza, C. R. B. 2015. "Fostering Effective Inter-Team Knowledge Sharing in Agile Software Development," *Empirical Software Engineering* (20:4), pp. 1006-1051.
13	Singh, A., Singh, K., and Sharma, N. 2014. "Agile Knowledge Management: A Survey of Indian Perceptions," *Innovations in Systems and Software Engineering* (10:4), pp. 297-315.
14	Wendling, M., Oliveira, M., and Maçada, A. C. G. 2013. "Knowledge Sharing Barriers in Global Teams," *Journal of Systems and Information Technology*, pp. 239-253.

3. Findings

Based on the keyword analysis, the findings from the literature review were classified into six main recurring themes of KM in agile software projects comprising (1) *Agile practices promoting KM*, (2) *Agile practices require efficient knowledge sharing and transfer,* (3) *Barriers and risks to knowledge sharing and knowledge transfer in agile projects,* (4) *Important conditions to resolve barriers and risks for knowledge sharing and knowledge transfer in agile projects,* (5) *Scaling up agile projects – inter-team structures and the need for coordination,* (6*) Dark side of agile projects.* These themes are mapped into a framework showing important conditions, characteristics, enablers, and inhibitors for mobilizing knowledge in an agile context (Table 3).

3.1 Agile practices promoting KM

The focus on tacit knowledge sharing and the networking model is a dominating research perspective on KM in agile software development. Studying agile practices and KM in Indian firms, Sing and colleagues (2014) show evidence that more than 60 % of the KM practices resulted in positive responses. This indicates that agile practices themselves promote KM. Those practices were related to KM environment, KM culture and policies, knowledge capture and organization, knowledge sharing, training, mentoring, and KM technologies. The most commonly used practice was learning and sharing through discussion forums. The least accepted practice was dependence upon documents for knowledge transfer, which indicates that explicit knowledge and the repository model of KM are less supporting. Overall, their study provides evidence that tacit knowledge dominates in agile software development and that the agile approach depends on tacit knowledge sharing. However, the Indian software industry with agile practices emphasized the need for more formality regarding KM. The formal role in handling KM was often lacking (e.g., employment of a chief knowledge officer). In addition, it was a need for a scrum master role, but this was seldom established. Reliance on authority that provides guidelines for using agile methods could be significant for the outcome of a project. Moreover, a

healthy KM environment and appropriate technologies are essential to establish. Most of the findings in this study agreed with the agile manifesto.

An informative workspace, knowledge repositories, engineering practices and cloud infrastructures, were all important elements of the agile environment leading to KM processes of knowledge capitalization, knowledge sharing, and knowledge creation. Knowledge repositories were, in some cases, critical formal structures besides networking. This was illustrated in a model depicting agile practices and cloud services supporting KM in large organizations (Khalil and Khalil 2020). The authors conclude that agile practices lead to a knowledge-intensive culture, and collaborative practices consisting of daily meetings, retrospective meetings with clients, and pair programming will promote knowledge capitalization.

Table 3: Framework to understand mobilization of knowledge in an agile context of IS development

Main themes	**Sub-themes**
Agile practices promoting KM	Tacit knowledge, the networking model of knowledge management
Agile practices require efficient knowledge sharing and transfer	Useful knowledge to share, tools to support knowledge sharing; Wikis, chat rooms, team networking, personal communication, social networking, social software, customer collaboration communication
Barriers and risks to knowledge sharing and knowledge transfer in agile projects	Different organizational roles have different barriers, lack of communication, lack of rewards, lack of trust, job security, motivational and psychological factors, organizational culture, time and resource, knowledge strategy, system methods, professional skills, cost, technology. Factors are interrelated. Risks are team diversity, team capabilities, team perception, project communication, low quality of collaboration technologies, complex and domain specific projects.
Important conditions to resolve barriers and risks for knowledge sharing and knowledge transfer in agile projects	Allocated time, supportive culture, organizational structure enhancing collaboration, pair programming, sprint planning, project retrospective, bridge communication gap, positive learning environment, avoid risks by engaging experienced team members, assign project full-time workers, allocate substantial time and resources, tailor the knowledge strategy, choose skilled and motivated people, consider the organizational context
Scaling up agile projects – inter-team structures and the need for coordination	Informal coordination; experience forums, lunch seminars, instant messaging, open work arena, formal coordination; Scrum of Scrums, MetaScrum, communities of practice for learning and innovation, for scaling up.
Dark side of agile projects	Time pressure – inhibiting team performance, wide scope of knowledge activities, formal and informal controls are important

3.2 Agile practices require efficient knowledge sharing and transfer

Several agile software studies examined knowledge processes regarding knowledge sharing and transfer in the agile environment. They report how contextual factors can promote efficient knowledge processes and avoid risks (Ghobadi and Mathiassen 2017) and barriers (Ghobadi and Mathiassen 2016; Heeager and Nielsen 2017) for knowledge flow.

Seventeen useful knowledge types encompass knowledge related to methodology and database design, software testing, communication skills, system integration, and writing business reports (Khoza and Bwalya 2021). Ways of sharing knowledge include wikis, chat rooms, team networking, personal communication, social

networking, and customer collaboration. At the same time, codified knowledge was also noted as essential means of supporting knowledge sharing. Social software, particularly the use of Wikis, provided understanding for learning and adoption of agile processes and was reported as beneficial for experience sharing and collaborative learning (Heredia et al. 2014). The learning of agile software processes was strongly related to actively being involved in tacit knowledge sharing during the collaborative learning process. Efficient collaboration in teams is crucial in agile software projects, and understanding different dimensions of the concept is a must. According to Batra and colleagues (2017), collaboration comprises mutual benefits, engagement, coordination, communication, and knowledge sharing. Thus, collaboration is much more multifaceted than presumed in the Agile Manifesto.

3.3 Barriers and risks to knowledge sharing and knowledge transfer in agile projects

Different organizational roles have different views of knowledge-sharing barriers. Ghobadi and Mathiassen (2016) identified 37 specific barriers. Causal maps for different roles (project managers, developers, testers, and users) were presented to get an overview of the barriers related to team, process, and context to understand barriers to effective knowledge sharing in agile software teams. Project setting barriers are crucial for project managers, project communication, and project organization. Also, team capabilities barriers are vital for developers, testers, and users. In addition, Khoza and Bwalya (2021) identified 33 factors influencing sharing of useful knowledge. They highlighted job security, motivational factors, time, psychological factors, lack of communication, lack of rewards, and lack of trust.

Heeager and Nielsen (2017) developed a conceptual model with barriers to knowledge transfer comprising organizational culture, time and resources, knowledge strategy, and motivation and willingness. These barriers should be looked at as interrelated and not in isolation. Concurrent with this last-mentioned study, Wendling and colleagues (2013) also showed the importance of understanding the relationships between barriers to knowledge sharing in global teams. For instance, cultural differences were negatively related to absorptive capacity. Some barriers were also enablers for knowledge sharing; time zone differences can be challenging but also positive since this increases time for activities and knowledge sharing in a global team. They presented a framework of enablers and barriers for knowledge sharing in global teams. The study confirmed existing barriers described in previous literature but also identified new knowledge-sharing barriers related to technology, professional skills, cost, and software development methodology.

Identified risks to knowledge sharing are team diversity (language, time, place, discipline), team capabilities (insufficient understanding of business domain, context, collaboration technologies, requirements, social skills), team perceptions (e.g., lack of motivation), project communication, low quality of collaboration technologies and development processes, complex and domain-specific projects (Ghobadi and Mathiassen 2017).

Gregory and colleagues (2016) aimed to map the landscape of agile practitioner challenges and identified seven main themes and 27 sub-themes. Sustainability was an important theme that included knowledge sharing, process improvement, documentation, and contracts. The challenge was also related to skills – and the need for developers to be “masters of all trades,” including social skills and business knowledge.

Important conditions to resolve barriers and risks for knowledge sharing and knowledge transfer in agile projects

Allocated time, supportive culture, the organizational structure supporting team collaboration, pair programming, sprint planning, and project retrospectives were important factors supporting knowledge sharing (Khoza and Bwalya 2021). Software development teams must bridge communication gaps and build a shared understanding to overcome barriers. In addition, the organizational roles’ diversity must be considered (Ghobadi and Mathiassen 2016). In addition, a positive learning environment was crucial to increase motivation and individual commitment to enable effective knowledge sharing (Gregory et al. 2016). Moreover, several knowledge-sharing solutions were suggested to avoid risks, engage experienced team members, assign full-time project workers, do mentoring, and create and develop shared goals (Ghobadi and Mathiassen 2017). Heeager and Nielsen (2017) developed some heuristics to handle barriers to knowledge sharing. They are: be aware of potential barriers and the relationships between them, choose skilled and motivated people, tailor your knowledge strategy, allocate substantial time and resources, and be mindful of your organizational context.

3.4 Scaling up agile projects – inter-team structures and the need for coordination

Coordination is crucial for agile software projects, especially when they scale up to larger projects requiring an inter-team structure (Batra et al. 2017; Dingsøyr et al. 2018; Santos et al. 2015). Different competencies and business languages exist in such complex settings, and knowledge sharing between client and vendor is part of the collaboration leveraging expertise and know-how. Business knowledge is shared from client to vendor, and technical knowledge is shared from the vendor to the client (Batra et al. 2017).

According to Dingsøyr and colleagues (2018), large-scale software projects require informal and formal coordination of inter-team knowledge sharing. Informal coordination comprises experience forums, lunch seminars, and instant messaging, and formal arenas are Scrum of Scrums (on sub-project level) and Metascrum (on project level). In this study, the informal arenas became more dominating over time, and an open work area supported efficient coordination and knowledge sharing. The organizational culture changed gradually towards more trust and collaboration. Customer collaboration involved knowledge sharing between people with business skills from the customer organization, and people with construction skills from the developers. Santos and colleagues (2015) provide evidence that more iterations that were undertaken improved coordination and knowledge sharing by the most agile companies, and knowledge sharing mechanisms proved to support the co-creation of business value. They developed a conceptual model explaining knowledge sharing across agile teams, including purposeful practices, organizational conditions, and stimuli. Important work practices were face-to-face conversations, informative workspaces, rotations among teams and projects, collective meetings, pair programming between different teams, internal seminars with technical presentations, marathons (competitions), coding Dojos (e.g., at lunch time). Important conditions were strategy, structure, culture and individual behavior, top management and leadership support, communication flow, integration among teams and projects, and successful agile method adoption.

In addition, communities of practice (CoPs) which are groups that share and create knowledge to enhance learning and innovation in organizations (Brown and Duguid 1991), generated valuable mechanisms for knowledge sharing related to inter-team coordination and communication (Paasivaara and Lassenius 2014). In this study conducted by Paasivaara and Lassenius (2014), CoPs played different roles in the transformation phase from plan-driven to agile software development and in the scaling-up phase from intra- to inter-team settings. They played a role in continuous improvement and showed it to be a valuable mechanism for knowledge sharing, inter-team coordination, and communication. They were necessary to implement a large-scale agile approach supporting the transformation process successfully. In this study, 20 CoPs were active, four of which were most important for the transformation process. Successful criteria for the CoPs and how they evolved were also elaborated. Some characteristics of CoPs were an interesting topic with concrete benefits to participants, a passionate leader, decision-making authority, supporting tools to create transparency, and cross-site participation when needed. The CoPs became important for knowledge sharing and learning in other projects after the agile project was completed.

3.5 Dark side of agile projects

One study reported on the dark side of agile projects regarding innovation and learning in organizations (Annosi et al. 2016), which implies knowledge-sharing problems. The study showed that applying agile approaches (e.g., Scrum) provided negative effects. Time pressure (deadline pressure) was found to be negative for self-regulated teams and inhibited team performance and engagement in learning and innovation activities. Other challenges were related to lack of support for technical implementation and product innovation, and that too broad knowledge was damaging expertise, difficulty in understanding the big picture of the product, and the teams had problems handling the broad scope of knowledge activities. Managerial practices and good work routines, including formal and informal controls, were necessary conditions for the teams' self-regulating learning and innovation. It is also dependent upon the teams' internal pressure. A paradigm model for time pressure was developed in this study. While not mentioned explicitly, this study has implications for KM. Both formal and informal controls are a critical part of a knowledge management strategy.

4. Implications for future research

Future research should assess the influence of different software development methodologies on knowledge sharing and how agile methods can facilitate knowledge sharing (Wendling et al. 2013). Moreover, there is a need to understand factors influencing knowledge sharing in different contexts (Ghobadi 2015; Ghobadi and Mathiassen 2016; Ghobadi and Mathiassen 2017; Khoza and Bwalya 2021) to understand factors related to inter-team knowledge sharing (Dingsøyr et al. 2018; Santos et al. 2015), to address strategies for knowledge transfer

between agile teams, and to develop measures against knowledge sharing barriers (Heeager and Nielsen 2017). Future studies should investigate incentives and mechanisms that may encourage knowledge sharing, compare its processes across different industries, and study knowledge sharing in different organizational roles (Ghobadi and Mathiassen 2016; Khoza and Bwalya 2021).

Moreover, future research should focus on high-performing and low-performing projects concerning knowledge-sharing risk profiles (Ghobadi and Mathiassen 2017) and on customer collaboration and inter-team coordination. The influence of culture, context, and the coordination mechanisms in larger projects should be focused (Batra et al. 2017; Dingsøyr et al. 2018), how practices are adopted to scale, and how the context of development programmes may influence the up-scaling approach (Dingsøyr et al. 2018). Future research should also determine antecedents of collaboration and study how collaboration affects capability, agility, and project success, in addition to team autonomy, competence, and trust (Batra et al. 2017).

Future research should continuously focus on the networking model of KM to get additional data on challenges and success factors regarding the role of CoPs for agile adoption (Paasivaara and Lassenius 2014). Especially the focus could be on contextual factors, organizational culture, structure, and national culture regarding the value of CoPs. Future studies should involve a diversity of industry sectors and large organizations to increase the understanding of the value of CoPs in large-scale agile IS projects. Further studies are needed to understand the agile practitioner's challenge landscape, measure the agile value, and understand the cultural change (Gregory et al. 2016). Finally, future research should focus on the dark side of agile projects and include teams from multiple firms to understand the impacts on team learning and innovation (Annosi et al. 2016).

5. Conclusion

This study examined the role of KM in agile software development projects. Our findings build on a literature study to overview how previous research has addressed this topic. The aim has been to understand how knowledge can and ought to be mobilized in the agile context. Table 3 shows a framework including elements enabling/inhibiting the mobilizing of knowledge in an agile context of IS development. First, the previous body of research clearly illustrates that agile software development practices and the agile manifesto promote KM, tacit knowledge sharing, and the networking model of KM dominate in front of the repository model and explicit knowledge. To understand how knowledge is efficiently mobilized in the agile software development context, we need to consider specific challenges, barriers, and risks to overcome them. Findings illustrate that mobilizing knowledge in the agile software development context embeds complexity, making knowledge sharing and knowledge transfer challenging. And the complexity increases when agile projects are scaling up into inter-team structures. Informal coordination mechanisms and communities of practice were important initiatives for dealing with large-scale agile projects. We recommend several measures for efficient knowledge sharing in the agile context.

Overall, this literature study revealed specific research gaps, providing evidence that the role of KM in agile software projects represents an immature research field. Future research recommendations have also been outlined.

References

Alavi, M., Kayworth, T. R., and Leidner, D. E. 2005. "An Empirical Examination of the Influence of Organizational Culture on Knowledge Management Practices," *Journal of Management Information Systems* (22:3), pp. 191-224.

Annosi, M. C., Magnusson, M., Martini, A., and Appio, F. P. 2016. "Social Conduct, Learning and Innovation: An Abductive Study of the Dark Side of Agile Software Development," *Creativity and Innovation Management* (25:4), pp. 515-535.

Batra, D., Xia, W., and Zhang, M. 2017. "Collaboration in Agile Software Development: Concept and Dimensions," *Communications of the Association for Information Systems* (41:1), pp. 429-449.

Brown, J. S., and Duguid, P. 1991. "Organizational Learning and Communities-of-Practice: Toward a Unified View of Working, Learning and Innovating," *Organization Science* (2:1), pp. 40-57.

Dingsøyr, T., Moe, N. B., Fægri, T. E., and Seim, E. A. 2018. "Exploring Software Development at the Very Large-Scale: A Revelatory Case Study and Research Agenda for Agile Method Adaptation," *Empirical Software Engineering* (23:1), pp. 490-520.

Dingsøyr, T., Nerur, S., Balijepally, V., and Moe, N. B. 2012. "A Decade of Agile Methodologies: Towards Explaining Agile Software Development," *Journal of Systems and Software* (85:6), pp. 1213-1221.

Fowler, M., and Highsmith, J. 2001. "The Agile Manifesto," *Software development* (9:8), pp. 28-35.

Ghobadi, S. 2015. "What Drives Knowledge Sharing in Software Development Teams: A Literature Review and Classification Framework," *Information & Management* (52:1), pp. 82-97.

Ghobadi, S., and Mathiassen, L. 2016. "Perceived Barriers to Effective Knowledge Sharing in Agile Software Teams," *Information systems journal* (26:2), pp. 95-125.
Ghobadi, S., and Mathiassen, L. 2017. "Risks to Effective Knowledge Sharing in Agile Software Teams: A Model for Assessing and Mitigating Risks," *Information Systems Journal* (27:6), pp. 699-731.
Gregory, P., Barroca, L., Sharp, H., Deshpande, A., and Taylor, K. 2016. "The Challenges That Challenge: Engaging with Agile Practitioners' Concerns," *Information and Software Technology* (77), pp. 92-104.
Heeager, L. T., and Nielsen, P. A. 2017. "Intrafirm Knowledge Transfer of Agile Software Practices: Barriers and Their Relations," *Journal of Information Technology Case and Application Research* (19:4), pp. 199-224.
Heredia, A., Garcia-Guzman, J., Amescua-Seco, A., and Serrano, A. 2014. "Study of Factors Influencing the Adoption of Agile Processes When Using Wikis," *International Journal of Software Engineering and Knowledge Engineering* (24:6), pp. 859-885.
Khalil, C., and Khalil, S. 2020. "Exploring Knowledge Management in Agile Software Development Organizations," *International Entrepreneurship and Management Journal* (16:2), pp. 555-569.
Khoza, L. T., and Bwalya, K. J. 2021. "An Insider's Perspective of Knowledge Sharing in Software Development Projects," *Journal of Information and Knowledge Management* (20:3), pp. 1-17.
Kitchenham, B. 2004. "Procedures for Performing Systematic Reviews," *Keele, UK, Keele University* (33:2004), pp. 1-26.
Li, J., Moe, N. B., and Dybå, T. 2010. "Transition from a Plan-Driven Process to Scrum: A Longitudinal Case Study on Software Quality," *Proceedings of the 2010 ACM-IEEE international symposium on empirical software engineering and measurement*, pp. 1-10.
Marinho, M., Noll, J., Richardson, I., and Beecham, S. 2019. "Plan-Driven Approaches Are Alive and Kicking in Agile Global Software Development," *2019 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)*: IEEE, pp. 1-11.
Maudal, O., and Dingsøyr, T. 2021. "14 Steps toward Lean Knowledge Management," in: *IFIP Advances in Information and Communication Technology*. pp. 291-298.
Nakayama, M., Hustad, E., and Sutcliffe, N. 2021. "Agility and System Documentation in Large-Scale Enterprise System Projects: A Knowledge Management Perspective," *Procedia Computer Science* (181), pp. 386-393.
Napoleão, B. M., de Souza, É. F., Ruiz, G. A., Felizardo, K. R., Meinerz, G. V., and Vijaykumar, N. L. 2021. "Synthesizing Researches on Knowledge Management and Agile Software Development Using the Meta-Ethnography Method," *Journal of Systems and Software* (178), p. 110973.
Ouriques, R. A. B., Wnuk, K., Gorschek, T., and Svensson, R. B. 2019. "Knowledge Management Strategies and Processes in Agile Software Development: A Systematic Literature Review," *International Journal of Software Engineering and Knowledge Engineering* (29:3), pp. 345-380.
Paasivaara, M., and Lassenius, C. 2014. "Communities of Practice in a Large Distributed Agile Software Development Organization - Case Ericsson," *Information and Software Technology* (56:12), pp. 1556-1577.
Santos, V., Goldman, A., and de Souza, C. R. B. 2015. "Fostering Effective Inter-Team Knowledge Sharing in Agile Software Development," *Empirical Software Engineering* (20:4), pp. 1006-1051.
Sarker, S., Sarker, S., Nicholson, D. B., and Joshi, K. D. 2005. "Knowledge Transfer in Virtual Systems Development Teams: An Exploratory Study of Four Key Enablers," *Communication, IEEE Transactions on* (48:2), pp. 201-218.
Singh, A., Singh, K., and Sharma, N. 2014. "Agile Knowledge Management: A Survey of Indian Perceptions," *Innovations in Systems and Software Engineering* (10:4), pp. 297-315.
Wendling, M., Oliveira, M., and Maçada, A. C. G. 2013. "Knowledge Sharing Barriers in Global Teams," *Journal of Systems and Information Technology*), pp. 239-253.
Wenger, E., McDermott, R., and Snyder, W. M. 2002. *Cultivating Communities of Practice*. Boston: Harvard Business School Press.
Zykov, S. V., and Singh, A. 2020. "Agile Knowledge Management," in: *Smart Innovation, Systems and Technologies*. pp. 37-53.