# Investigating organizational structures and means for effective knowledge sharing and coordination in large agile organizations

Anil Can Kara

17.06.2024, Master's Thesis Final Presentation

Chair of Software Engineering for Business Information Systems (sebis)
Department of Computer Science
School of Computation, Information and Technology (CIT)
Technical University of Munich (TUM)
wwwmatthes.in.tum.de

# Agenda

Motivation

Research Questions

Methodology

Results

Key Findings

Future Research

# Motivation (1/3)

Agile methodologies, initially designed for small teams, have gained popularity in large organizations to enhance flexibility and to increase performance.

However, scaling agile practices for large organizations involves addressing communication, coordination, and knowledge sharing challenges.

In large-scale agile organizations, knowledge sharing and coordination play a crucial role for the success.

[1][2][12]

# Motivation (2/3)

Existing literature investigates the topic of knowledge exchange and coordination in large-scale agile organizations by mainly concentrating on specific organizations and certain aspects of the topic.

However, a broad overview of mechanisms, their usage contexts, challenges and facilitators is limited.

# Motivation (3/3)

To fill this research gap, our research sheds light on:

**Broad overview** of the mechanisms used for knowledge exchange and coordination in large-scale agile organizations.

**Hindering and facilitating factors** for effective knowledge exchange and coordination.

The **contexts** that the mechanisms, barriers and facilitators tend to occur.

# Agenda

Motivation

Research Questions

Methodology

Results

Key Findings

Future Research

# Research Questions

RQ1

How do knowledge exchange and coordination take place in large agile organizations, and which mechanisms are used for this purpose?

RQ2

What are the barriers to and facilitators for effective knowledge sharing and coordination in large agile organizations?

RQ3

What are the benefits, trade-offs, and application contexts of knowledge exchange and coordination mechanisms in large agile organizations?

# Agenda

Motivation

Research Questions

Methodology

Results

Key Findings

Future Research

# Methodology - Overview

## Literature Review

Scopus, IEEE Xplore, ACM Digital Library

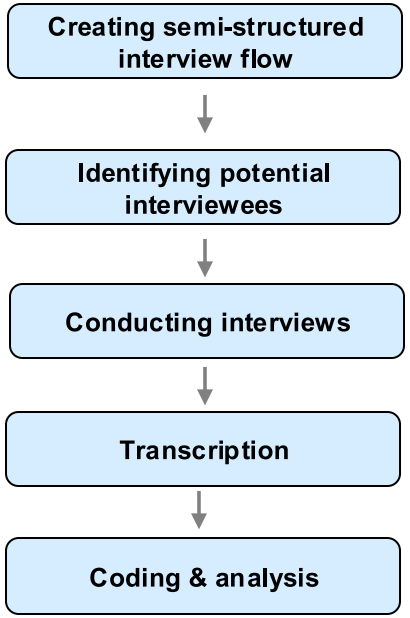
**Inclusion criteria:**

- Full-text accessible
- In English
- Search string matching the title or the keywords
- Published after 2001

**Exclusion criteria:**

- Content not relevant to answer research questions
- Duplicates from different databases

## Interview Study



Creating semi-structured interview flow
Identifying potential interviewees
Conducting interviews
Transcription
Coding & analysis

# Methodology – Literature Review

**Search string:** ("agile" OR "scrum") AND ("large" OR "large scale" OR "large-scale" OR "scaling" OR "scaled" OR "inter-team" OR "multiteam" OR "distributed") AND ("knowledge sharing" OR "knowledge exchange" OR "knowledge management" OR "coordination")

| Database | Query Hits | Inclusion Criteria not Fulfilled |
|---|---|---|
| Scopus | 91 | 2 |
| IEEE Xplore | 26 | 0 |
| ACM Digital Library | 44 | 1 |

After applying the exclusion criteria, **59 publications** have been identified and reviewed.

# Methodology – Interview Study

- In total, **11 experts** have been interviewed in **10 sessions**, which have various levels of experiences and backgrounds.

- **16 distinct roles** covered: Software developer, technical project lead, development lead, solution train engineer, quality assurance manager, scrum master, project manager, product owner, chapter lead, lead architect, business analyst, requirements engineer, rollout manager, change manager, machine learning engineer, software architect

# Interview Partners

| Roles | Companies | Experience in Large-Scale Agile | Experience Overall |
|---|---|---|---|
| Software Developer, Technical Project Lead, Development Lead, Solution Train Engineer, Quality Assurance Manager | ConsultCo1 | 20 years | 30+ years |
| Scrum Master | SoftwareCo1 | 10+ years | 10+ years |
| Software Developer, Project Manager | TelecomCo, FoodCo | 4 years | 4 years |
| Product Owner | InsurCo, ConsultCo2 | 5 years | 5 years |
| Chapter Lead | InsurCo | 3 years | 10 years |
| Lead Architect ,Scrum Master | ConsultCo2 | 8 years | 8 years |
| Business Analyst, Requirements Engineer, Rollout Manager | TransportCo | 4 years | 4 years |
| Scrum Master, Product Owner | TransportCo | 9 years | 35+ years |
| Change Manager, Product Owner | ConsultCo2, EnergyCo | 1 year | 10 years |
| Machine Learning Engineer | ConsultCo3 | 2 years | 5 years |
| Scrum Master, Product Owner, Development Lead, Software Architect | SoftwareCo2, SoftwareCo3, EnergyCo2 | 20+ years | 20+ years |

# Agenda

Motivation

Research Questions

Methodology

Results

Key Findings

Future Research

# Results - Mechanisms

- The interview study reveals **41 mechanisms** for knowledge sharing and coordination that are classified under three categories:
  - ➢ 30 meetings
  - ➢ 4 organizational structures
  - ➢ 7 categories of tools

- Although each mechanism might have its own initial goals of supporting knowledge sharing or coordination, our results suggest that these two practices often tend to occur simultaneously via the mechanisms identified.

- All mechanisms are presented with their initial purpose in the upcoming tables.

# Meetings

| Meeting | Knowledge Sharing | Coordination |
|---|---|---|
| Sprint Planning | | ✓ |
| Sprint Retrospective | ✓ | |
| Sprint Review | ✓ | |
| Sprint Demo | ✓ | |
| Daily Scrum | | ✓ |
| Scrum of Scrums Daily | | ✓ |
| PI Planning | | ✓ |
| PI Demo | ✓ | |
| PI Retrospective | ✓ | |
| Backlog Refinement | | ✓ |
| PO Alignment | | ✓ |
| Meetings for Certain Roles | | ✓ |
| Future Workplace | ✓ | |
| Kick-Off | | ✓ |
| Hackathons | ✓ | |

# Meetings (Ctd.)

| Meeting | Knowledge Sharing | Coordination |
|---|---|---|
| Release Kick-Off | | ✓ |
| Scrum Master Sync | ✓ | |
| Technical Internal Meetings | | ✓ |
| Level-Specific Meetings | ✓ | |
| Internal Coordination Meetings | | ✓ |
| Lessons Learned | ✓ | |
| Intra-Team Knowledge Sharing Sessions | ✓ | |
| Internal Knowledge Sharing Sessions | ✓ | |
| Lunch Talks | ✓ | |
| Agile Coffee Break | ✓ | |
| All-Staff Meetings | ✓ | |
| Town Hall Meetings | ✓ | |
| Developer Conference | ✓ | |
| Trainings | ✓ | |
| Forums | ✓ | |

# Organizational Structures

| Organizational Structure | Knowledge Sharing | Coordination |
|---|---|---|
| Story-Specific Coordination Role | | ✓ |
| Communities of Practice (COPs) | ✓ | |
| Mentoring Programs | ✓ | |
| Guilds | | ✓ |

# Story-Specific Coordination Role

- It is a role ensuring everything regarding a story is successfully maintained and completed.
- It is not a separate, independent role in the classical sense but is taken by the existing team members, specifically developers.

- The experts having this role are responsible for creating all necessary subtasks and allocating the right tasks to the right people.
- By the end, this role ensures that the final story is fully functional, all the requirements are met, and deployed on the integration environment before being presented to the customer.

# Story-Specific Coordination Role

- The role aims to fill the following coordination gaps arising in large-scale agile development environments regarding the management of tasks:

  - ✓ No surveillance on the integrated individual feature branches.
  - ✓ Unawareness of what to do with the open tasks.
  - ✓ Unawareness of which additional team members could be allocated to the tasks to finish them earlier.

# Communities of Practice (COPs)

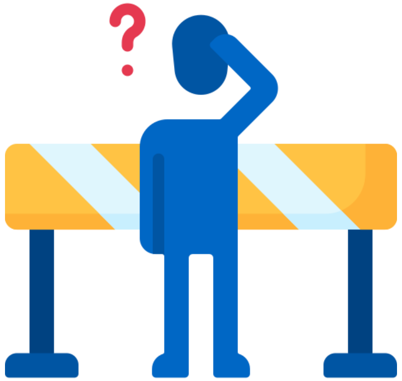
- Communities of Practice are observed to provide certain benefits regarding the efficient usage of the knowledge that team members have, and they are widely used for knowledge sharing purposes in large-scale agile contexts.
- In COPs, people talk about various topics such as coding guidelines, new tools, new frameworks, and best practices. They exchange ideas and experiences, and learn from each other while potentially teaching their colleagues certain topics that they are knowledgeable about.
- Some organizations implement multiple COPs that focus on different topics, such as frontend development, backend development, architecture, and security. In this setup, one member from each team whose expertise suits one of the COPs joins the respective COP.
- Although the main objective of COPs is to share knowledge, they also serve as an indirect coordination mechanism in certain contexts. For instance, participation of one team member from all teams in COPs enables teams to coordinate the dependencies and the topics that affect all teams.

# Tools

| Category | Tools | Knowledge Sharing | Coordination |
|---|---|---|---|
| Communication | Microsoft Teams<br>Slack<br>Outlook<br>Zoom | ✓ | ✓ |
| Documentation | Jira<br>Confluence<br>SharePoint | ✓ | ✓ |
| Version Control and Repository | Git<br>GitHub<br>Bitbucket<br>Nexus | ✓ | ✓ |
| Requirements Management | Doors | ✓ | ✓ |
| Design | Figma | ✓ | |
| Visual Workspace | Miro | | ✓ |
| Online Whiteboard | ConceptBoard | ✓ | ✓ |

# Results - Barriers

- Our interview study identifies **20 barriers** to effective knowledge sharing and coordination in large agile organizations.
- Results show that barriers often affect knowledge sharing and coordination practices together, either directly or indirectly.
- Upcoming tables list the identified barriers and their relation to knowledge sharing and coordination practices.



# Barriers

| Barrier | Knowledge Sharing | Coordination |
|---|---|---|
| Personality Traits | ✓ | ✓ |
| Language Differences | ✓ | ✓ |
| Cultural Differences | ✓ | ✓ |
| Lack of Experience | ✓ | ✓ |
| Limitation of Ad-Hoc Exchanges | ✓ | ✓ |
| Inability to Use Physical Tools | ✓ | ✓ |
| Inability to Overhear | ✓ | indirectly |
| Inability to Observe Colleagues | ✓ | ✓ |
| Inability to Use Body Language and Facial Expressions | ✓ | ✓ |
| Time Zone Differences | ✓ | ✓ |

# Barriers (Ctd.)

| Barrier | Knowledge Sharing | Coordination |
|---|---|---|
| Insufficient Documentation Efforts | ✓ | ✓ |
| Involvement of External Parties | ✓ | indirectly |
| Lack of Transparency | ✓ | ✓ |
| Misinterpretation of Agile | ✓ | indirectly |
| Unawareness | ✓ | indirectly |
| Lack of Management Support | ✓ | ✓ |
| Project Kick-Off |  | ✓ |
| Inadaquate Tools | ✓ | ✓ |
| Reduntant Meetings |  | ✓ |
| Lack of Clear Responsibilities | ✓ | ✓ |

# Facilitators

| Facilitator | Knowledge Sharing | Coordination |
|---|---|---|
| Documentation Efforts | ✓ | ✓ |
| Efficient Usage of Tools | ✓ | ✓ |
| Appropriate Participant Selection | ✓ | ✓ |
| Emotional Bonding | ✓ | ✓ |
| Flat Hierarchy | ✓ | |
| Ad-Hoc Exchanges | ✓ | ✓ |
| Appropriate Meeting Selection | ✓ | ✓ |
| Encouraging Knowledge Sharing | ✓ | |
| Clear Definition of Roles | | ✓ |

# Agenda

Motivation

Research Questions

Methodology

Results

Key Findings

Future Research

# Key Findings

- Knowledge sharing and coordination are tightly coupled, and a barrier or a facilitator affecting one of them is very likely to affect the other.
- Contrary to the findings of the existing literature, ad-hoc meetings are not necessarily more efficient than scheduled ones. Especially when a thorough preparation of the presenters is required to enable more effective knowledge sharing and coordination during a meeting, it is more beneficial to schedule the meeting with clear goals defined beforehand so that the presenters can use their time more effectively.
- The implementation details of the online tools used for knowledge sharing and coordination are as important as the tools themselves, and the exact way they are used decides whether they work as effective mechanisms for knowledge sharing and coordination.
- Although an online tool is not initially designed to be used for knowledge sharing or coordination purposes, it can still act as an important mechanism for knowledge sharing or coordination practices in large agile organizations.

# Key Findings

- Certain mechanisms whose primary goal is to facilitate knowledge sharing can indirectly enable effective coordination activities even if they are not intended to be doing so.
- Certain barriers and facilitators are inter-connected, and one of them can potentially create a snowball effect on the others, and consequently, can favor or hinder effective knowledge sharing and coordination even more than it would do just by itself. (External Colleagues -> Lack of Transparency)
- Besides the identified mechanisms and facilitators, subtle procedures and habits are also used to enable more effective knowledge sharing and coordination.

# Agenda

Motivation

Research Questions

Methodology

Results

Key Findings

Future Research

# Future Research

- The study can be conducted with a larger number of participants to have more extensive data to analyze and to identify more mechanisms inherent to the large scale agile organizations.

- Identified barriers, facilitators, application contexts and trade-offs can be verified by assessing their applicability in more organizations.

- Connections between the barriers and the facilitators can be investigated in more depth as well as the relationship between knowledge sharing and coordination.

Thank you!

# Bibliography

[1] Edison, H., Wang, X., & Conboy, K. (2021). Comparing methods for large-scale agile software development: A systematic literature review. *IEEE Transactions on Software Engineering*, *48*(8), 2709-2731.

[2] Razzak, M. A., & Ahmed, R. (2014, September). Knowledge sharing in distributed agile projects: Techniques, strategies and challenges. In *2014 Federated Conference on Computer Science and Information Systems* (pp. 1431-1440). IEEE.

[3] Gervigny, M. L. I., & Nagowah, S. D. (2017, December). Knowledge sharing for agile distributed teams: A case study of Mauritius. In *2017 International Conference on Infocom Technologies and Unmanned Systems (Trends and Future Directions)(ICTUS)* (pp. 413-419). IEEE.

[4] Paasivaara, M., & Lassenius, C. (2019). Empower your agile organization: Community-based decision making in large-scale agile development at Ericsson. *IEEE Software*, *36*(2), 64-69.

[5] Kitchenham, B., & Charters, S. "Guidelines for performing systematic literature reviews in software engineering". In: (2007), Zhang, H., Babar, M. A., & Tell, P. (2011). Identifying relevant studies in software engineering. *Information and Software Technology*, *53*(6), 625-637.

[6] Myers, M. D., & Newman, M. (2007). The qualitative interview in IS research: Examining the craft. *Information and organization*, *17*(1), 2-26.

[7] Miles, M., Huberman, A., & Saldaña, J. (2014). Qualitative data analysis: A methods sourcehook Sage publications.

[8] Saldaña, J. (2021). The coding manual for qualitative researchers. *The coding manual for qualitative researchers*, 1-440.

[9] Seaman, C. B. (1999). Qualitative methods in empirical studies of software engineering. *IEEE Transactions on software engineering*, *25*(4), 557-572.

[10] Santos, V., Goldman, A., & De Souza, C. R. (2015). Fostering effective inter-team knowledge sharing in agile software development. *Empirical Software Engineering*, *20*, 1006-1051.

[11] Almeida, F., Miranda, E., & Falcão, J. (2019). Challenges and facilitators practices for knowledge management in large-scale scrum teams. *Journal of Information Technology Case and Application Research*, *21*(2), 90-102.

# Bibliography

[12] Dingsøyr, T., Moe, N. B., Fægri, T. E., & 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*, 490-520.

[13] Ipe, M. (2003). Knowledge sharing in organizations: A conceptual framework. *Human resource development review*, *2*(4), 337-359.

[14] Navimipour, N. J., & Charband, Y. (2016). Knowledge sharing mechanisms and techniques in project teams: Literature review, classification, and current trends. *Computers in Human Behavior*, *62*, 730-742.

[15] Dahlqvist, M., & Forsberg, J. (2018). Inter-team knowledge sharing: A case study on co-located teams’ drivers and barriers for KS.

[16] Dorairaj, S., Noble, J., & Malik, P. (2012, August). Knowledge management in distributed agile software development. In *2012 Agile Conference* (pp. 64-73). IEEE.

[17] Melnik, G., & Maurer, F. (2004, June). Direct verbal communication as a catalyst of agile knowledge sharing. In *Agile Development Conference* (pp. 21-31). IEEE.

[18] Nyrud, H., & Stray, V. (2017, May). Inter-team coordination mechanisms in large-scale agile. In *Proceedings of the XP2017 scientific workshops* (pp. 1-6).

[19] Boden, A., Avram, G., Bannon, L., & Wulf, V. (2009, July). Knowledge management in distributed software development teams-does culture matter?. In *2009 Fourth IEEE International Conference on Global Software Engineering* (pp. 18-27). IEEE.

[20] Bjørnson, F. O., & Vestues, K. (2016, May). Knowledge sharing and process improvement in large-scale agile development. In *Proceedings of the Scientific Workshop Proceedings of XP2016* (pp. 1-5).

[21] Paasivaara, M., & Lassenius, C. (2014). Communities of practice in a large distributed agile software development organization–Case Ericsson. *Information and Software Technology*, *56*(12), 1556-1577.

[22] Paasivaara, M., Lassenius, C., & Heikkilä, V. T. (2012, September). Inter-team coordination in large-scale globally distributed scrum: Do scrum-of-scrums really work?. In *Proceedings of the ACM-IEEE international symposium on Empirical software engineering and measurement* (pp. 235-238).

[23] Berntzen, M., Hoda, R., Moe, N. B., & Stray, V. (2022). A taxonomy of inter-team coordination mechanisms in large-scale agile. *IEEE Transactions on Software Engineering*, *49*(2), 699-718.

[24] A LeSS Adoption at RBS. https://less.works/case-studies/rbs

# Bibliography

Icons: https://www.flaticon.com/

https://www.flaticon.com/free-icons/change Change icons created by Eucalyp - Flaticon

https://www.flaticon.com/free-icons/buildings Buildings icons created by Freepik - Flaticon

https://www.flaticon.com/free-icons/research Research icons created by rsetiawan - Flaticon

https://www.flaticon.com/free-icons/lack-of-concentration Lack of concentration icons created by Freepik - Flaticon

https://www.flaticon.com/free-icons/across Across icons created by gravisio - Flaticon

https://www.flaticon.com/free-icons/overview Overview icons created by Freepik - Flaticon

https://www.flaticon.com/free-icons/challenge Challenge icons created by Eucalyp - Flaticon

https://www.flaticon.com/free-icons/feedback Feedback icons created by Ning Nong - Flaticon

https://www.flaticon.com/free-icons/books Books icons created by Freepik - Flaticon

https://www.flaticon.com/free-icons/job-interview Job interview icons created by Freepik - Flaticon

https://www.flaticon.com/free-icons/plus Plus icons created by Pixel perfect - Flaticon

https://www.flaticon.com/free-icons/job-search Job search icons created by Flat Icons - Flaticon

https://www.flaticon.com/free-icons/connect Connect icons created by Eucalyp - Flaticon

https://www.flaticon.com/free-icons/worker Worker icons created by ultimatearm - Flaticon

https://www.flaticon.com/free-icons/coordination Coordination icons created by zero_wing - Flaticon

https://www.flaticon.com/free-icons/obstacle Obstacle icons created by Freepik - Flaticon

https://www.flaticon.com/free-icons/analysis Analysis icons created by Uniconlabs - Flaticon

https://www.flaticon.com/free-icons/big-data Big data icons created by xnimrodx - Flaticon

https://www.flaticon.com/free-icons/search Search icons created by Freepik - Flaticon

https://www.flaticon.com/free-icons/mechanism Mechanism icons created by Uniconlabs - Flaticon

.

**Anil Can Kara**

Technical University of Munich (TUM)
TUM School of CIT
Department of Computer Science (CS)
Chair of Software Engineering for Business
Information Systems (sebis)

Boltzmannstraße 3
85748 Garching bei München

+49.89.289.17132
anil.kara@tum.de
wwwmatthes.in.tum.de