International Journal of Management, Knowledge and Learning | ISSN 2232-5697

On the Dynamics of Structure, Power, and Interaction in Innovative Processes: A Case Study from a Software Developing Company

Trygve J. Steiro
Norwegian University of Science and Technology
trygve.j.steiro@ntnu.no

Ragnar Vennatrø
Norwegian University of Science and Technology
ragnar.vennatro@ntnu.no

Johan Bergh
Oslo New University College
johan.bergh@oslonh.no

Glenn-Egil Torgersen
University of South-Eastern Norway
glenn-egil.Torgersen@usn.no

Purpose: In a dynamic world with high customer demands and volatile markets, organisations need to adapt to changes in context. Agile methods constitute a well-recognised framework for software companies. The current study looks into how these methods are applied and their perceived strengths. The problem formulation in the current study is: What are the experiences with agile methods?

Methodology: The data material is derived from two interviews with the department manager at the start and end of the project. In addition, 12 interviews with department members were conducted, and observations from four stand-up meetings and three department meetings were conducted.

Findings: Four themes emerged from the study concerning Empowerment in mastery and overview, Meaningfulness and job satisfaction as preconditions for learning, Transparency in processes facilitates psychological safety and learning, and Imbalance in power between the programmers and the product owners: Lack of transparency.

Originality: These results are discussed in the study and presented in a conceptual model. We argue that transparency and meaningfulness play a central role in adapting and learning. The current study also investigates power aspects as both an enabling and disabling factor.

Introduction

Hagen and Steiro (2001) described a work-life shift from mass production to more tailor-made production. Still, mass production is important, but more and more workplaces do not fit the traditional description of assembly lines. Likewise, is a shift from capital to knowledge (Hagen & Steiro, 2001). There is a shift toward using more project organising to be more flexible (Torgersen & Steiro, 2009). Further, project groups can accommodate organisations´ need for more flexibility while being highly effective vehicles for learning, problem-solving, and innovation (Brown & Duguid, 1991; Edmondson, 2002; Davies & Hobday, 2005; Janz & Prasarnphanich, 2009). Empowered teams are considered vital for software development (Katz, 1988; Cockburn, 2007; Nerur & Balijepally, 2007). Team research has demonstrated that group learning is largely influenced by how work group members collaborate; for example, in terms of leadership, mutual performance monitoring, balanced contributions from members, backup behaviour, adaptability, and team orientation (Hoegl & Gemuenden, 2001; Salas et al., 2005). These collaboration patterns are subsequently dependent on lower-level coordination

mechanisms such as shared mental models, closed-loop communication, and mutual trust (Rulke & Galaskiewicz, 2000; Salas et al., 2005).
In a similar vein, highlighting a need to understand work life within contexts of leading-edge technological organisations less in terms of modern tropes of mass production methods and more in terms of 'hands-on' methods of a craft-orientated workflow are findings within sociomaterial human-technological research such as contemporary archaeology (Vennatrø & Høgseth, 2021). Focused on cold-war technology and deindustrialisation are findings of a crucial reliance on hands-on production methods and the tacit expertise of specific teams of engineers (Moshenska, 2016). From the retro-engineering of the RAF nuclear capacity Avro Vulcan bombers during the Falkland conflict (White, 2007), the archaeological 'excavations' of the ground-breaking MOS 6502 microchip of the 1970s (Swaminathan, 2011), or digital archaeology of early software development (Sharwood, 2004), the one component that makes or breaks present-day attempts at retro-engineering these technologies is not a lack of either the hardware, blueprints, or technological know-how. Instead, it is the tacit factor of human-technology social interaction related to the highly skilled engineers and workers involved in the original production. As such, critically important social aspects of how humans interact with, optimise and make technology functional have never been absent from modern industrialised contexts. Zuboff (1988) argued early that new information technology would not only represent a significant change in itself but also challenge authority and power. Perhaps unsurprisingly, one arena where a tacit dimension of human-technological interaction has always played a crucial role is within high-end technological contexts (Graves-Brown, 2013; Law, 2002; Suchman, 2007).

Chmiel (2000) has argued that we need to research organisations typically at the forefront of technology to learn and understand more about an organisation. Such organisations may be software-developing companies, as Chmiel (2000) suggested. Adler (2005) suggests that software development is difficult because it involves combinations of uncertainty, complexity, and interdependence. The software has become the invisible glue that holds a multitude of systems together by ensuring faster, more accurate, and more extensive exchange and processing of information. Building software is hard since it involves high technological complexity and a wide range of demands from a multitude of stakeholders. Many software development problems are beyond the cognitive capacity of a single individual; their epistemic complexity is simply too high (Enberg et al., 2006; Fægri, 2011). This means that vast amounts of knowledge and skills are necessary to determine effective development actions. Development processes are dynamic and unpredictable (March & Simon, 1993). Poppendick and Poppendick (2003) enhance learning processes in agile methods. It is flexible and creates creative processes and collaboration in small teams as Scrum processes are set up. If we are not entirely sure where we are going and consider that we are learning something during the process, this is a much better approach (Poppendick & Poppendick, 2003).

Agile methods were initially developed when people realised that software development had to be dynamic and 'on demand' when the customer wants it and adaptive as markets appear to be changing. The agile principles emphasise frequent collaboration and continuous feedback (Abhishek & Frank, 2014; Younas, et al., 2018). Rolfsen and Wulff (2014) point out that the concept is parallel or simultaneous development within software development. It involves developing the concept, requirements, and guidelines together with the customer at the same time as you begin working with the product itself. This will be based on a circular rather than a sequential process. The key to getting fast deliveries, effective learning loops, and decisions late in the process is that the team has an overview of the entire task and has delegated enough authority and decisions that they can function as the flexible unit it can be (Rolfsen & Wulff,

2014). Furthermore, short deadlines, well-integrated teams, and good feedback routines are how to achieve this (Rolfsen & Wulff, 2014). It is also important to simulate and create a feeling that all are in the same room (Rolfsen & Wulff, 2014). Schwaber (2004) originated the term *scrum*. Nonaka & Takeuchi (1986) introduced this. They highlight a need to move away from a linear mindset to a more flexible way of working where the scrum team is responsible for making the entire product. Like in rugby, scrum encourages teams to learn through experiences, self-organise while working on a problem, and reflect on their wins and losses to continuously improve. Agile methods can be seen as a reaction to plan-based or traditional methods, which emphasise *a rational, engineering-based approach* (Dybå, 2000). The new focus is on methods that sort under the umbrella of Agile methods and leanness (Dingsøyr et al., 2010). The problem formulation in the current study is: What are the experiences with agile methods?

Theoretical framework

Williams and Cockburn (2003) state that agile software development is about feedback and change, where short feedback loops are necessary to achieve a desirable and predictive outcome. The agile manifesto prioritises the following:

- Individuals and interactions over processes and tools
- Working software over comprehensive documentation
- Customer collaboration over contract negotiation
- Responding to change by following a plan

These points originate from the 'Manifesto for Agile Software Development' (Beck et al. 2001). Broza (2015) writes that the Agile Manifesto can be seen as a theory Y derived from McGregor's (1960) perspective. Broza (2015) goes further, explaining the theory Y in terms of it being transparent (Steiro & Torgersen, 2018a), centred on respect (2018 b), and is about creating psychological safety (Edmondsson, 2002; 2011). This makes software development an interesting case to study. As Marks and Scholarios (2008) discovered, the ability to combine softer skills with technical skills may become increasingly important in software practice.

The unforeseen is a central factor in software development. Uncertainty is closely related to novelty, analysability, and ambiguity and is the difficulty of solving the problems posed or resolving the exceptions encountered in work (Adler, 2005). Uncertainty is challenging because we cannot rely solely on prior experience or routine. In situations of uncertainty, we make decisions and carry out actions under bounded rationality (March & Simon, 1993). March and Olsen (1976) challenged the rational actor model of decision-making by considering novel, poorly defined situations for which no clear goal or procedure exists for finding a solution. Bounded rationality limits our ability to adapt optimally to the environment (Simon, 1991) and also means that plans and procedures can only partially guide our work. Rational and deterministic strategies become less useful. In uncertainty, we make decisions and participate in problem-solving following pragmatic strategies, hunches, and intuition rather than adhering strictly to routines, programs, or models (March, 1988). Following Cherns (1976; 1987), this can be specified through the concept of *minimum critical specifications*. The minimum critical specification is a means to deal with complexity since all systems need the freedom to develop. Morgan (1986) sees this as necessary, rather than commanding and controlling to the smallest detail. Operating in a dynamic environment means that methods and applications must be dynamic, as first addressed by Burns and Stalker (1961). Steiro and Torgersen (2018a) put this forward, as did the seminal work by Bungay (2011), claiming that the focus in operations must be on *why* and for the executors on *how*. This is a view that is taken from Sinek (2014). However, all these thoughts can be studied from Helmut von Moltke the Elder's (1800-1891)

thoughts. This is reflected in the doctrine where the German 'Auftragstaktikk', also known as 'Mission Command', aims to address the discrepancy between plans and the chaos of war. This problem with warfare is what Clausewitz (1832/1976, 119) termed *friction*, a discrepancy between the plan and what was happening on the battlefield. Gross (2016, 29) writes that Moltke developed this thought further and is known to have said *'any plan, regardless of how meticulously it was laid out, would be offset when meeting the enemy'*. In a study on leadership by Volini *et al*., (2019), only 41 per cent thought their organisations were ready or very ready to meet their leadership requirements. They found that leaders are being pressured to 'step up and show the way forward'. This was before the Covid-19 global crisis. Stepping up, showing the way, and leading by example are not new, at least not in a military sense (e.g., Roberts, 2018). We find this somewhat peculiar for software developing companies. Such companies are prone to many challenges and/or risks (Conchúir et al., 2006, Verner *et al*., 2014; Sinha et al., 2020) and, therefore, must be strategically flexible. Strategic flexibility refers to the company's ability to respond to uncertainty by adjusting its objectives with the support of knowledge and superior capabilities (Fachrunnisa et al., 2020). This may also demand agile or flexible leadership (Yukl, 2008).

Agile leadership signifies an agile leader who can guide his/her team and continually influence the team's behaviour by defining, spreading, and maintaining organisational vision (Perker et al., 2015). Therefore, as stated above, there is reason to assume that agile methods also require agile or flexible leadership. Agile leaders discipline their inquiry processes by seeking and harnessing evidence of impact throughout the change process rather than waiting for a final evaluation (Breakspear, 2017). Agility in leadership is also fairly common in modern military organisations, especially when it comes to mission command (e.g., Ploumis, 2020). Agile leadership is, therefore, also relevant for software development since disruptive digital business models characterise the world, augmented workforces, flattened organisations, and an ongoing shift to team-based work practices (Volini et al., 2019). Reimagining leadership as a top-down approach (Senge, 1995, Kezar, 2012, Raelin, 2015) has become especially emergent during the ongoing Russian-Ukrainian war, in which uncertainty and complexity stress the need for flexible solutions. As the war rages on, we can see, at least in Western media, that there are significant differences in leadership between the Russian and Ukrainian armed forces. This has similarities with software development as an example of evolving sociotechnical systems (Clegg et al., 1996) that must embrace uncertainty, complexity, and interdependence (Adler, 2005). It has been suggested that society's increased complexity and faster pace also explain the demand for leadership (Storey, 2016). Fullan (2001) argues that the more complex society gets, the more sophisticated leadership must become. Therefore, strategic flexibility (Fachrunnisa et al., 2020) seems to be of the utmost importance as this refers to the ability to respond flexibly to uncertainty.

An agile or flexible leadership approach seems essential for most managers (Yukl, 2008). Similarly, *transparency*, or being completely open and honest in business, is considered a competitive tool (e.g., Baum, 2005). As we have seen, this also plays a role in learning. Transparency in leadership, for example, for software development companies, seems to contribute to enabling trust, openness, and tolerance to develop (Subramanian, 2017). Also, transparent organisational communication is a process that generates trust and credibility (Rawlins, 2009). Transparent organisations also find it much easier to foster open communications (Subramanian, 2017). In today's global and interdependent workforce cultures, agility, flexibility, and openness seem essential for success. Fægri (2011) found the following five characteristics for agile methods to succeed: (A1) collective, situated performance, (A2) purposeful, (A3) cooperative behaviour, (A4) coordinated action, and (A5) empowerment. There are five identified prerequisites: (B1) participation, (B2) opportunities

for learning, (B3) knowledge diversity, (B4) knowledge redundancy, and (B5) proximity, physical or virtual. Lastly, there are three identified considerations for improvement: (C1) a reorientation of governing values, (C2) creating and exploring opportunities for collaborative learning, and (C3) sociotechnical reconciliation. Organisations dealing with innovation may gain benefits from encouraging and enabling collaborative learning. Knowledge of great value is created to solve practical problems, be it customer support, product development, or maintenance (Brown & Duguid, 1998). In collaborations, the organisation may better exploit its skills' diversity and receive higher innovative performance from the cross-fertilisation of ideas and cognitive resources. Collaborations can also give benefits to the individual. For example, providing richer experiences allows for bridging different knowledge sources and better dealing with the world's complexity (Kolb, 1984) and improves double-loop learning in dealing with complexity (Argyris & Schön, 1978; 1996). The flexibility drives the speed of innovation in software; a competitive advantage rewards software organisations if they can engage in rapid innovation in terms of development processes (Baskerville et al., 2003) and products (Cusumano, 2004). Software development is highly knowledge-intensive because intellectual assets replace natural resources and physical inputs as the basis for value creation and economic evaluation (Hansen et al., 1999; Powell & Snellman 2004). Learning organisations create conditions for employees´personal fulfilment, improved motivation, and retention (Patuvakis & Bouranta, 2013). The enjoyment of work is an important facilitator of experiential learning (Matsuo, 2015). Enjoyment of work supports active experimentation and reflection (Matsuo, 2015).

Sociotechnical systems design theory embraces work organisation's combined social and technical complexity (Trist & Bamforth, 1951; Emery, 1959). Furthermore, it sets out with the explicit ambition to improve people's job satisfaction and productivity while simultaneously creating the conditions necessary for the adaptive and learning-centric organisation (Trist, 1981; Mumford, 2006) and focusing on humanisation, work democratisation, industrial relations, and focusing on humanisation, work democratisation, industrial relations, and industrial production (Trist, 1981; Mumford, 2006).). Sociotechnical theoretical perspectives can be valuable in exploring the relationships between organisational, process-related, technical, and human issues. However, it is claimed that the technological reality of software development is often given precedence over the social reality (Andelfinger, 2002). A central idea in sociotechnical theory is that local knowledge within the group should be used and developed collectively to improve coordination efficiency (Mumford, 2000). This is also reflected in the strong emphasis on user participation in sociotechnical system development (Mumford, 1983; Mumford, 2000).

From a sociotechnical perspective, collective learning can be understood as the collective accomplishments of autonomous work groups in their collaborations with each other and their relationships with technical systems. Coordination in software development work where people are not co-located or face other barriers to interaction faces additional challenges as communication is less direct and rich (Mumford, 2006; Tabaka, 2006). Because software development is an example of evolving sociotechnical systems (Clegg et al., 1996) that must embrace uncertainty, complexity, and interdependence (Adler, 2005), software development is a practice that is driven by knowledge creation. Software systems emerge as the necessary knowledge becomes available through cycles of design and use. In collaborative problem-solving, people must depend on emergent dialogue and mutual monitoring to build accountability rather than formal routines (Faraj & Xiao, 2006). Such dialogue can be effectively supported by artefacts (Bertelsen, 2000). Third, common understanding allows group members to anticipate how their work fits within the whole (Okhuysen & Bechky, 2009).

In projects involving high interdependence, collaboration is essential for efficient integration (Mark, 2002; Faraj & Xiao, 2006).

Methods and materials

A small/medium software development department case study was chosen to answer the research question. The number of employees was 25. They were part of a larger company of small and medium-sized companies with a total of 350 employees. The department manager invited the first author in for a brief talk on common interests. The first author was then invited to a department meeting to present the research and gain approval. The department manager was positive but wanted the department's approval. This can be seen as a form that analysis can take in all aspects of research (Kvale & Brinkmann, 2009). Empowerment was clearly seen here, and it, in turn, shaped the research approach.

The first author was invited to a department meeting to present the project. The organisation members were free to ask questions regarding the researcher's background and the research project itself. The project was approved collectively. This meant being able to participate in scrum meetings and department meetings. Interviews were arranged and begun after taking part in stand-up meetings. The data material is derived from two interviews with the department manager at the start and end of the project. In addition, 12 interviews with department members were conducted. Observations from four stand-up meetings and three department meetings were conducted. When data maturity was achieved, the last interview with the department manager was conducted. Data maturity is the objective when conducting qualitative research (Denzin & Lincoln, 2018). (Denzin & Lincoln, 2018). Qualitative methods provide insight into people´s actions as they unfold in a social context (Key, 2018). Therefore, different qualitative methods such as interviews and observations are recommended to gain insight into the field by triangulating (Denzin & Lincoln, 2018). The material was analysed by conducting a thematic analysis (Braun & Clarke, 2006; Clarke & Braun, 2014). The following themes were identified: Empowerment in mastery and overview, Meaningfulness and job satisfaction as preconditions for learning, Transparency in processes facilitates psychological safety and learning, and Imbalance in power between the programmers and the product owners: Lack of transparency.

Results and discussion

Before presenting the themes further, some generic background information might be needed. One aspect concerns the location of participants in various areas of the country or the ability of the product owners to see the work teams while the teams could not see the virtual team members. As we interpret it, this represents an imbalance in power and that social interaction is hampered within this domain. During the current study, there was a change of leadership. The initial department leader was strongly inspired by the idea of building on processes and interpersonal resources. This was expressed both normatively and in practice. It was expected that one should offer his or her competence, aim to go home at four p.m. and be ready to perform the next day, only occasionally work long hours of overtime. Humour, a good mood, and social activities were also emphasised. This particular leader was keen not to focus too much on individual performance or ego-boosting. The department leader appointed his successor. The new manager was more academically specialised in the field than his predecessor but shared the same general ideas regarding organising and would continue to follow the main essence. When the data was collected, there was no indication of any major changes. Minor adjustments were found, but they did not seem to unsettle the main features. Stand-up meetings were held inside the manager's office, where there was plenty of space for grids and charts that showed how they were performing and the challenges ahead. The room was bright and airy. Each employee was given a minute to explain what had been done, what

was coming up, and, and whether there were any challenges. This meant that each person was given a minute in the limelight, giving the manager a good idea of the current status. Typically, those who shared challenges were met by others who came to 'the rescue' either by giving tips and advice or putting their own tasks aside to help. One could use 'programming in pairs' for support, learning, and reducing stress.

Empowerment in mastery and overview

Several of the informants reported that they were more empowered than before. It was possible earlier to see challenges and then come up with solutions. Before applying agile methods, a manager told them what to do and when. By applying agile methods, they saw more of the overall processes. They were encouraged to come up with challenges. Fægri (2011) sees empowerment as a key feature of agile enabling success. In addition, Fægri (2011) also pointed to purposefulness. Several reported a better overview and a higher general feeling of mastery. A dialogue with the product owners also facilitated understanding and seeing the 'bigger picture'. Rolfsen and Wulff (2014) point out that the concept is parallel or simultaneous development within software development. It involves developing the concept, requirements, and guidelines together with the customer at the same time as you start working with the product itself. Furthermore, short deadlines, well-integrated teams, and good feedback routines are how to achieve this (Rolfsen & Wulff, 2014). The data shows that agile methods facilitate a larger understanding and are in line with mission command (Gross, 2016; Sinek, 2014; Bungay, 2011; Steiro & Torgersen, 2018a). Decisions are communicated more openly, and there is a better chance of challenging assumptions and staying ahead. As one programmer stated: *'Before, I could be asked to travel to another city the day before. This was a challenge when I had my son living with me. Now, I will know when a trip is needed, and I can also be in power to steer when it is needed to go. Before, this was the call of the project manager and the project manager alone'.*

Meaningfulness and job satisfaction as preconditions for learning

This theme is strongly linked to the previous point. We see from our data that the majority reported a more meaningful workday and increased job satisfaction. This is an important characteristic of the workplace. In addition, several of the informants claimed this impacted learning. They reported a higher degree of confidence in each other, which can be seen as high psychological safety (Edmondson, 2002, 2011). This can also be interpreted as their seeing the tasks they are responsible for as a part of a higher calling (Thorsrud & Emery, 1970). This can be seen as a psychosocial work condition. If someone had challenges, they could choose pair programming to solve the task and bring more work satisfaction. Pair programming is seen as one of the most significant aspects of collaboration within the agile framework. This was the main purpose. However, they also typically saw this as an important means of learning from each other. The reported meaningfulness created a platform and a direction for learning. It also seems that more meaningfulness creates more knowledge exchange (Clark, 2020).

Transparency in processes facilitates psychological safety and learning

The informants report more transparent processes. *'It is much easier now to see if one is struggling or hiding problems. Then we can assist or intervene.'* Another informant said, 'Now we know what the other is doing. If I have some slack, I can assist.' Psychological safety is necessary (Edmondson, 2002; 2011). Open communication on a daily basis helped put a safe psychological climate forward. However, as the manager put it: *'You must be able to see benefits for others, yourself, and for the whole team'.* Group learning is largely influenced by how work group members collaborate in terms of, for example, leadership, mutual performance

monitoring, balanced contributions from members, backup behaviour, adaptability, and team orientation (Hoegl & Gemuenden, 2001; Salas et al., 2005). However, what was observed in the stand-up meetings was mainly information exchange. When asking about this observation, it was said that the stand-up meetings were important, but much of the actual assisting behaviour happened outside the room. Broza (2015) writes that stand-up meetings should not last more than 15 minutes. Each participant should tell what they have done, what they will do, and whether there are some obstacles (Broza, 2015). Several informants said that the level of trust was increased. Mutual trust is important for learning (Rulke & Galaskiewicz, 2000; Salas et al., 2005). *As* can be derived from the data, transparency plays a significant role in learning within the team. Rolfsen and Wulff (2014) stress that team members must be delegated enough authority and latitude for decision-making to function as a flexible unit.

Imbalance in power between the programmers and the product owners: Lack of transparency

One central observation was that the stand-up meetings were held in a room where the product owners only could be heard but not seen. On the other hand, the product owners could see the stand-up meeting on camera. Being in the room provided an asymmetric feeling of being watched. The informants said it was not a direct problem because they had become used to it. The manager recognised that it was not optimal and that he would have that changed. The stand-up meetings at 11 a.m. were institutionalized. The informants said that this could not be missed. One informant said that, on one occasion, a product owner was obviously out of the office since, when asking for input, they heard unfamiliar noise from a public place. However, they had received proper feedback, and the programmers considered the product owner very competent. Still, the product owners' ability to see and hear everything, whereas the programmers could only hear the owners, had weaknesses representing an imbalance in power. An imbalance of power can pose a severe threat to collaboration (Dougherty & Hardy, 1996) and can be especially problematic for inter-organisational collaboration (Mintzberg et al., 1996). Scrum represents that the customer is part of the development team in the product owner role. Fægri (2011) stressed that proximity identification is a key prerequisite for successful agile methods. It is also important to stimulate that you are in the same room (Rolfsen & Wulff, 2014). This lack of proximity in the same room can reduce transparency if the technology works unequally. A central idea in sociotechnical theory is that local knowledge within the group should be used and developed collectively to improve coordination efficiency (Mumford 2000). Structuration (Giddens, 1984) can help to explain social processes in collectives. Hatch (1997) sees the role of structures as key prerequisites for interaction. Carlsen, Clegg, Pitsis, and Mortensen (2020) argue that more attention should be paid to processes of interactional framing in which people jointly attend to situations, reach new integrations, and produce new social realities.

Overall discussion and conclusion

The current study shows that processes and relations are made more transparent. This is very important. It means that members can read and react (Bjurwill, 1993) and that more social interaction is present. When the processes are transparent, it is easier to empower decisions (Steiro & Torgersen, 2018a). At the same time, when the processes are transparent, it is difficult to hide problems. This can also be seen as something that can hamper autonomy. People need to feel free to choose for themselves (autonomy), experience that they master activities they are a part of (competence), and feel they belong (belonging) (Ryan & Deci, 2017). This was not reported as such in the current study. When one of the programmers stated a problem, it was more felt as owned by the community. Typically, members would step up to aid. One

solution was to program in pairs. This could be instrumental support and ensure that the programmer did not feel alone with the problem. This is another example of the social interaction of something handled within the action (Steiro & Torgersen, 2018a). From the observation, the product owners could be seen as remote in physical distance and the role. Improved technology may change this.

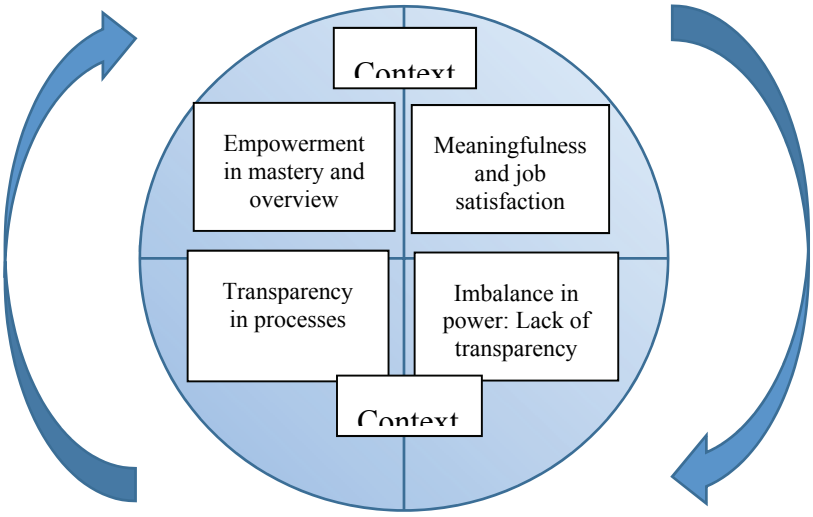

Figure 1. The interplay between the four dimensions seen concerning the context.

Agile methods in software development are an area where sociotechnical thoughts are of interest, as seen from the current study, both as a means of execution of the method and for the development and analysis of organisations. Further, the seminal work of Thorsrud & Emery (1970) focused on a good work environment. The good work environment and psychosocial work environment factors such as the need for content in work, the need to be able to make decisions, the need for recognition, the need to see the connection between the outside world and the work you perform, and finally, the need to see that the job you have and do is compatible with the desired future. Thorsrud & Emery's (1970) thoughts were developed decades ago. However, we find an interesting connection between Thorsrud and Emery (1970) and a present-day agile manifesto which can be explored further. We see from the current study that several of the informants reported increased mastery and more meaningful work. Antonovsky (2012) writes of the expression *sense of coherence* as an important feature of human life. Increased mastery is seen as empowered in decisions and power. Power is important because social interactions, and the knowledge created through them, are influenced by power. Power is part of the social situation where motives are negotiated. In addition, power relationships influence what knowledge to search for (Contu & Willmott, 2003). Because knowledge is dynamic in content and perceived value, the resulting power is also dynamic. Carlsen et al. (2020) argue that the word power is not very present when it comes to organisational creativity. They call for integrating power in analysis and seeing it as a dynamic factor of connection, abundance, and collective agency (Carlsen et al., 2020). Hence, power is pivotal when we discuss situated action (Mills, 1940), with the justification process determining what is true (Tell, 2004) and, therefore, the situated nature of learning. We can say that collective learning is a negotiation process (Brown & Duguid, 1998); collective learning entails negotiating meaning. Still, power has been only marginally discussed in the analysis of collective learning (Marshall & Brady, 2001; Roberts, 2006; Fenwick, 2008). Furthermore, key works in collective learning have been criticised for not discussing power but rather trying to conceal it (Contu & Willmott, 2003). Because power is such an influential aspect of learning, such a lack of transparency raises serious questions about the validity of empirical findings.

Among the most influential scholars on power is the French philosopher and social theorist Michel Foucault. Foucault (1980) holds a neutral position regarding power and argues for seeing power as either positive or negative. He further argues that power is present in all relations. Power embeds authority, and power is part of social interaction (Clegg, 2003). Power is, therefore, most usefully seen as a power relationship. Moving beyond this, researchers have identified a need for further investigations of power in learning. Giddens (1984) explains power as all forms of dependence, including how subordinates can influence the activities of their superiors.

Meaningfulness might be understood in the sense of seeing its role as a part of the whole. We see these as important means for learning and development. The informants also reported a high degree of psychological safety. Carl Rogers spoke of unconditional positive regard (Rogers, 1957). Douglas McGregor referred to non-physical safety needs (McGregor, 1960). Clark (2020) claims that psychological safety is a condition in which you feel: 1) included, 2) safe to learn, 3) safe t o contribute, and 4) safe to challenge the status quo. However, as seen in the current study, condition 4 in Clark's (2020) terminology could be more developed.

Organisational success in the global knowledge economy depends on the ability of individuals and organisations to experiment, extract knowledge, and implement learning (Chadwick & Rawer, 2015). Success leads to validation of knowledge, repetition of action, and confidence and initiatives (Chen et al., 2017). Continuous learning from positive and negative experiences is essential for human capital development (Noe et al., 2014). In a dynamic environment, the ability to learn from successful experiences and failures is significant for success (Aranda et al., 2017; Irgens & Hernes, 2013). Organisational learning is a complex, socially constructed, and context-specific process (Zhou et al., 2015). Success and failure are often considered dichotomous 'either/or' outcomes, but organisations may learn to appreciate that certain 'failures' create very positive learning potentials (Edmondson, 2011). Therfore psychological safety is of such importance. A failure can be seen as a hard-learned lesson or a fiasco depending on who is around. Therefore, mutual trust and transparency are of critical significance.

References

Abhishek, S., & Frank, M. (2014). A roadmap for software engineering in the cloud: results of a systematic review. IGI Global.

Adler, P. S. (2005). The evolving object of software development. Organization 12(3), 401-435.

Andelfinger, U. (2002, January). On the Intertwining of Social and Technical Factors in Software Development Projects. In Social Thinking-Software Practice (pp. 185-203).Antonovsky, A. (2012). Helsens mysterium. Den salutogene modellen. Gyldendal Akademisk.

Aranda, C., Arellano, J., & Davila, A. (2017). Organizational learning in target setting. Academy of Management Journal, 60(3), 1189-1211.

Argyris, C., & Schön, D. A. (1996). Organizational learning II: Theory, method, and practice. Reading, Massachusetts: Addison-Wesley.

Argyris, C. (1982). The Executive Mind and Double-Loop Learning. Organizational Dynamics 11(2), 5-22.

Baum, H. (2005). Transparent leadership. Leader to Leader, (37).41.

Baskerville, R., Ramesh, B., Levine, L., Pries-Heje, J., & Slaughter, S. (2003). Is internet speed software development different? IEEE Software, 20(6), 70-77.

Beck, K., Beedle, M., Van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., & Thomas, D. (2001). Manifesto for agile software development. https://moodle2019-20.ua.es/moodle/pluginfile.php/2213/mod_resource/content/2/agile-manifesto.pdf

Bertelsen, O. V. (2000). Design artifacts: Towards a design-oriented epistemology. Scandinavian Journal of Information Systems, 12(1), 15-27.

Bjurwill, C. (1993). Read and React. The Football Formula. Perpetual and Motor Skill, 76, 1983-86.

Braun, V., & Clarke, V. (2006). Using thematic analysis in psychology. Qualitative research in psychology, 3(2), 77-101.

Breakspear, S. (2017). Embracing Agile Leadership for Learning: how leaders can create impact despite growing complexity. Australian Educational Leader, 39 (3), 68-71.
Brown, J. S., & Duguid, P. (1998). Organizing knowledge. California Management Review 40(3): 90-111.
Broza, G. (2015). *The Agile Mindset: Making Agile Processes*. CreateSpace Independent Publishing Platform.
Brown, J. S., & Duguid, P. (1991). Organizational learning and communities-of-practice: Toward a unified view of working, learning, and innovation. Organization Science 2(1), 40-57.
Bungay, S. (2011). The art of action: how leaders close the gaps between plans, actions, and results. Hachette
Burns, T., & Stalker, G. M. (1961). The Management of Innovation. Tavistock.
Carlsen, A., Clegg, S. R., Pitsis, T. S., & Mortensen, T. F. (2020). From ideas of power to the powering of ideas in organizations: Reflections from Follett and Foucault. European Management Journal, 38(6), 829-835.
Chadwick, I. C., & Raver, J. L. (2015). Motivating organizations to learn: Goal orientation and its influence on organizational learning. Journal of Management, 41(3), 957-986.
Chen, G., Zhou, Q., & Liu, W. (2017). Organizational learning from experience: Current status in multilevel perspective, integration model and future direction. Nankai Business Review International. Vol. 8 No. 2, 2017 pp. 122-157. 10.1108/NBRI-01-2017-0006
Cherns, A. (1976). The principles of sociotechnical design. Human Relations, 29(8), 783- 792.
Cherns, A. (1987). Principles of sociotechnical design revisited. Human Relations, 40 (3), 153-161.
Chmiel, N. E. (2008). An introduction to work and organizational psychology: a European perspective. Blackwell Publishing.
Clark, T. R. (2020). The 4 Stages of Psychological Safety: Defining the Path to Inclusion and Innovation. Kindle Edition.
Clarke, V., & Braun, V. (2014). Thematic analysis. In T. Teo (Ed.), Encyclopaedia of critical psychology, (pp. 1947– 1952). Springer. https://doi.org/10. 1007/978-1-4614-5583-7
Clegg, C. W., Waterson, P. E., & Axtell, C. M. (1996). Software development: knowledge-intensive work organizations. Behaviour & Information Technology, 15(4), 237-249.
Clegg, S. R. (2003). Managing organization futures in a changing world of power/knowledge. In H. Tsoukas & C. Knutsen (Eds.), The Oxford handbook of organization theory: Meta-theoretical perspectives. Oxford University Press, pp. 536-567.
Cockburn, A. (2007). Agile software development: The cooperative game (2nd edition). Addison Wesley.
Conchúir, E. Ó., Holmstrom, H., Agerfalk, J., & Fitzgerald, B. (2006). Exploring the assumed benefits of global software development. In 2006 IEEE International Conference on Global Software Engineering (ICGSE'06), pp. 159-168.
Contu, A., & Willmott, H. (2003). Re-embedding situatedness: The importance of power relations in learning theory. Organization Science, 14(3), 283-296.
Cusumano, M.A. (2004). The Business of Software: What Every Manager, Programmer, and Entrepreneur Must Know in Good Times and Bad. Free Press.
Davies, A. & Hobday, M. (2005). The business of projects: Managing innovation in complex products and systems. Cambridge University Press.
Denzin, N. K., & Lincoln, Y. S. (Eds.). (2011). The Sage handbook of qualitative research. Sage.
Dingsøyr, T., Dybå, T., & Moe, N. B. (2010). Agile software development: an introduction and overview. In Agile Software Development (pp. 1-13). Springer.
Dougherty, D., & Hardy, C. (1996). Sustained product innovation in large, mature organizations: Overcoming innovation-to-organization problems. Academy of Management Journal, 39(5): 1120-1153.
Dybå, T. (2000). Improvisation in small software organizations. IEEE Software 17(5), 82-87.
Edmondson, A. C. (2002). The local and variegated nature of learning in organizations: A group-level perspective. Organization Science, 13(2), 128-146.
Edmondson, A. C. (2011). Strategies for learning from failure. Harvard Business Review 89(4), 48-55.
Enberg, A., Lindkvist, L., & Tell, F. (2006). Exploring the dynamics of knowledge integration - Acting and interacting in project teams. Management Learning, 37(2), 143-165.
Fachrunnisa, O., Adhiatma, A., Lukman, N., & Ab. Majid, M. N. (2020). Towards SMEs' digital transformation: The role of agile leadership and strategic flexibility. Journal of Small Business Strategy, 30(3), 65-85.
Faraj, S., & Xiao, Y. (2006). Coordination in fast-response organizations. Management Science, 52(8), 1155-1169.
Fenwick, T. (2008). Understanding relations of individual-collective learning in work: A review of research. Management Learning, 39(3), 227-243.
Foucault, M. (1980). Power/knowledge: Selected interviews and other writings, 1972-1977. Pantheon Books.
Fægri, T. E. (2011). Collaborative learning in software development: An investigation of characteristics, prerequisites (Doctoral dissertation, Norwegian University of Science and Technology).
Fægri, T. E. (2010, June). Adoption of team estimation in a specialist organizational environment. In International Conference on Agile Software Development (pp. 28-42). Springer, Berlin, Heidelberg.

Giddens, A. (1984). The constitution of society: Outline of the theory of structure. University of California Press.
Graves-Brown, P. (2013). Inside is Out: An Epistemology of Surfaces and Substances. In: A. González-Ruibal (Ed.), Reclaiming Archaeology: Beyond the Tropes of Modernity. Routledge, 298-310.
Gross, G.P. (2016). The Myth and Reality of German Warfare: Operational Thinking from Moltke the Elder to Heusinger. University Press of KY.
Hagen, Ø., & Steiro, T. (2001). Kunnskapsøkonomien-gamal vin på ny flaske. Magma, 4 (5), 83-95.
Hansen, M. T., Nohria, N., & Tierney, T. (1999). What's your strategy for managing knowledge? Harvard Business Review.
Hatch, M. J. (1997). Jazzing up the theory of organizational improvisation. In: J. P. Walsh, & Huff (Eds.), Advances in Strategic Management. Greenwich, CT: Jain.
Hernes, T., & Irgens, E. J. (2013). Keeping things mindfully on track: Organizational learning under continuity. Management Learning, 44(3), 253-266.
Hoegl, H. & Gemuenden, H. G. (2001). Teamwork quality and the success of innovative projects: A theoretical concept and empirical evidence. Organization Science 12(4), 435-449.
Janz, B. D., & Prasarnphanich, P. (2009). Freedom to cooperate: Gaining clarity into knowledge integration in information systems development teams. IEEE Transactions on Engineering Management, 56(4), 621-635.
Katz, R. (Ed.). (1988). Managing professionals in innovative organizations: A collection of readings. Ballinger Publishing Company.
Kezar, A. (2012). Bottom-up/top-down leadership: Contradiction or hidden phenomenon. The Journal of Higher Education, 83(5), 725-760.
Kolb, D. A. (1984). Experiential learning: Experience as the source of learning and development. Prentice-Hall.
Kvale, S., & Brinkmann, S. (2009). Interviews: Learning the craft of qualitative research interviewing. Sage.
Law, J. (2002). Aircraft Stories: Decentring the Object in Technoscience. Duke University Press.
March, J. G., & Simon, H. A. (1993). Organizations. John Wiley & Sons.
March, J. G. (1988). Bounded rationality, ambiguity, and the engineering of choice. In J. G. March (Ed.), Decisions and Organizations. Basil Blackwell, 266-293.
March, J. G., & Olsen, J. P. (1976). Organizational choice and ambiguity. In: J. G. March, & J. P. Olsen (Eds.). Ambiguity and choice in organizations. Universitetsforlaget
Mark, G. (2002). Extreme collaboration. Communications of the ACM, 45(6), 89-93.
Marshall, N., & Brady, T. (2001). Knowledge management and the politics of knowledge: illustrations from complex products and systems. European Journal of Information Systems, 10(2), 99-112.
Brady, T. (2001). Knowledge management and the politics of knowledge: Illustrations from complex products and systems. European Journal of Information Systems 10(2): 99-112.
McGregor, D. (1960). Theory X and theory Y. Organization theory, 358(374), 5.
Mills, C. W. (1940). Situated actions and vocabularies of motive. American Sociological Review 5(6): 904-913.
Mintzberg, H., Jorgensen, J., Dougherty, D., & Westley, F. (1996). Some surprising things about collaboration-knowing how people connect makes it work better. Organizational Dynamics, 25(1), 60-72.
Morgan. G. (2006). Images of organization (Updated edition). Thousand Oaks, California: Sage Publications.
Moshenska, G. (2016). Reverse engineering and the archaeology of the modern world. Forum Kritische Archäologie, 5, 16-28.
Mumford, A. (2006). The story of sociotechnical design: reflections on its successes, failures, and potential. Information Systems Journal, 16(4), 317-342.
Mumford, A. (2000). A sociotechnical approach to systems design. Requirements Engineering, 5(2), 125-133.
Mumford, E. (1983). Designing human systems for new technology: The ETHICS method. Manchester Business School.
Nerur, S., & Balijepally, V. (2007). Theoretical reflections on agile development methodologies: The traditional goal of optimization and control is making way for learning and innovation. Communications of the ACM, 50(3), 79-83.
Noe, R. A., Clarke, A. D., & Klein, H. J. (2014). Learning in the twenty-first-century workplace. Annu. Rev. Organ. Psychol. Organ. Behav., 1(1), 245-275.
Okhuysen, G. A., & Bechky, B. A. (2009). Coordination in organizations: An integrative perspective. The Academy of Management Annals, 3(1), 463-502.
Perker, D. W., Holesgrove, M., & Pathak, R. (2015). Improving productivity with self-organised teams and agile leadership. International Journal of Productivity and Performance Management, 64(1), 112–128. https://doi.org/http://dx.doi.org/10.1108/MRR-09-2015-0216
Ploumis, M. (2020). Mission command and philosophy for the 21st century. Comparative Strategy, 39(2), 209-218.
Poppendick, M., & Poppendick, T. (2003). Lean Software Development: An Agile Toolkit. Addison Wesley Professional.
Powell, W. W., & Snellman, K. (2004). The knowledge economy. Annu. Rev. Sociol., 30, 199-220.

Raelin, J. A. (2015). Rethinking Leadership. MIT Sloan Management Review, 56(4), 95.
Rawlins, B. (2009). Give the emperor a mirror: Toward developing a stakeholder measurement of organizational transparency. Journal of Public Relations Research, 21, 71-99.
Roberts, J. (2006). Limits to communities of practice. Journal of Management Studies 43(3): 623-639.
Roberts, R. (2018). Twelve Principles of Modern Military Leadership. Non Commissioned Officer Journal. United States Army website. https://www.armyupress.army.mil/Journals/NCO-Journal/
Rogers, C. R. (1957). The necessary and sufficient conditions of therapeutic personality change. Journal of consulting psychology, 21(2), 95.
Rolfsen, M., & Wulff, K. (2014). Lean programvareutvikling. In; Rolfsen, M.(red.)., Lean blir norsk. Lean i den norske samarbeidsmodellen] [Lean becomes Norwegian. Lean in the Norwegian Co-operation Model Fagbokforlaget.
Rulke, D. L., & Galaskiewicz, J. (2000). Distribution of knowledge, group network structure, and group performance. Management Science 46(5), 612-625.
Ryan, R. M., & Deci, E. L. (2017). Self-determination theory: Basic psychological needs in motivation, development, and wellness. Guilford Publications.
Salas, E., D.E. Sims, D. E., & Burke, C. S. (2005). Is there a "big five" in teamwork? Small Group Research 36(5), 555-599.
Senge, P. (1995). Rethinking Leadership in the learning organization. In Leading Learning Organizations. Cambridge, MA. MIT Center for Organizational Learning Research Monograph. Edited version by Colleen P Lannon. https://thesystemsthinker.com/rethinking-leadership-in-the-learning-organization/
Schwaber, K. (2004). Agile project management with Scrum. Microsoft press.
Simon, H. A. (1991). Bounded rationality and organizational learning. Organization Science 2(1): 125-134.
Sinha, R., Shameem, M., & Kumar, C. (2020). SWOT: strength, weaknesses, opportunities, and threats for scaling agile methods in global software development. In Proceedings of the 13th innovations in
software engineering conference on formerly known as India software engineering conference (pp. 1-10).
Sinek, S. (2014). Leaders eat last: Why some teams pull together, and others don't. Penguin.
Steiro, T. J., & Torgersen, G.-E. (2018 a). Samhandling Under Risk: Applying Concurrent Learning to Prepare for and Meet the Unforeseen. In G.-E. Torgersen (Ed.), Interaction: 'Samhandling' Under Risk. A Step Ahead of the Unforeseen. Cappelen Damm Akademisk, 251-266. https://doi.org/10.23865/noasp.36.ch14 License: CC BY-NC 4.0
Steiro, T. J., & Torgersen, G.-E. (2018 b). Weltbürger Perspectives and Samhandling. In G.-E. Torgersen (Ed.), Interaction: 'Samhandling' Under Risk. A Step Ahead of the Unforeseen (pp. 189–198). Cappelen Damm Akademisk. https://doi. org/10.23865/noasp.36.ch10 License: CC BY-NC 4.0.
Subramanian, K. R. (2017). Psychological contract and transparent leadership in organisations. International Research Journal of Advanced Engineering and Science, 2(1), 60-65.
Suchman, L. (2007). Human-Machine Reconfigurations: Plans and Situated Actions. Cambridge University Press, 2nd ed.
Swaminathan, N. (2011). Digging into Technology's Past. Archaeology 64(4)
http://archive.archaeology.org/1107/features/mos_technology_6502_computer_chip_cpu.html
Tabaka, J. (2006). Collaboration explained. Addison-Wesley.
Takeuchi, H., & Nonaka, I. (1986). The new product development game. Harvard Business Review, 64(1), 137-146.
Tell, A. (2004). What do organizations know? Dynamics of justification contexts in R&D activities. Organization 11(4), 443-471.
Thorsrud, E., & Emery, F. E. (1970). Industrial democracy in Norway. Industrial Relations: A Journal of Economy and Society, 9(2), 187-196.
Trist, E. L., & Bamforth, K. W. (1951). Some social and psychological consequences of the Longwall method of coal-getting. Human Relations 4(1), 3-33.
Trist, E. L. (1981). The evolution of sociotechnical systems. Occasional Paper No. 2. Toronto, Quality of Working Life Centre.
Verner, J. M., Brereton, O. P., Kitchenham, B. A., Turner, M., & Niazi, M. (2014). Risks and risk mitigation in global software development: A tertiary study. Information and Software Technology, 56(1), 54-78.
Vennatrø, R., & Høgseth, H. B. (in prep.) "Retro-Engineering Archaeology within the Haptic Multiple". In Melko, N. & von Rüden, C. (Eds.). Touching Things – An Archaeology of Perception. Springer.
Volini, E., Schwartz, J. Indranil, R., Hauptman, M., Van Durme, Y., Denny, B., & Berson, J. (2019). Leadership for the 21st century: The intersection of the traditional and the new. 2019 Global Human Capital Trends. Deloitte Insights.
Williams, L., & Cockburn, A. (2003). Guest Editors' Introduction: Agile Software Development: It´s about Feedback and Change. Computer, 36(06), 39-43.
White, R. (2007). Vulcan 607: The Epic Story of the Most Remarkable British Air Attack Since WWII. Corgi.

Zhou, W., Hu, H., & Shi, X. (2015). Does organizational learning lead to higher firm performance? An investigation of Chinese listing companies. Vol. 22 No. 5, 2015 pp. 271-288.

Zuboff, S. (1988). In the age of the smart machine: The future of work and power. Heinemann Professional Publishing.

Yulk, G. (2008, April). The importance of flexible leadership: The importance, assessment, and development of flexible leadership. Practitioner forum presented at the 23rd Annual Conference of the Society for Industrial Organizational Psychology, San Francisco.

Younas, M., Jawawi, D. N., Ghani, I., Fries, T., & Kazmi, R. (2018). Agile development in the cloud computing environment: A systematic review. *Information and Software Technology*, *103*, 142-158.