Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Social and Educational Sciences
Department of Sociology and Political Science

Monika Lie Larsen

A Better Way of Working?

A Case Study of Innovation Culture in Agile Teams.

Master's thesis in Management
Supervisor: Petter Grytten Almklov
February 2022

NTNU
Norwegian University of
Science and Technology

Monika Lie Larsen

A Better Way of Working?

A Case Study of Innovation Culture in Agile Teams.

Master's thesis in Management
Supervisor: Petter Grytten Almklov
February 2022

Norwegian University of Science and Technology
Faculty of Social and Educational Sciences
Department of Sociology and Political Science

NTNU
Norwegian University of
Science and Technology

Abstract

Many organizations, both within and outside of the IT industry, are busy implementing "agile" ways of working. But are agile working methods the right answer for all types of organizations?

This single-case study explores an "agile" software organization's cultural characteristics. The culture appears close to what is associated with Employee-Driven Innovation (EDI).

By following an inductive approach, I have found some patterns in how the organization upholds its cultural values and how it applies the agile methodology. There are challenges in the project organization related to overall direction and cross-team coordination, as the chosen agile methodology itself does not provide much support for multi-team work. The organization "works around" these shortcomings by applying a set of informal and formal coordination strategies "above" the agile practices. These project-level mechanisms work together with the "core" agile practices to support mutual trust, openness, and emotional safety beyond the team level.

However, this is not sufficient to uphold team-level autonomy, engagement, and collaboration orientation, which implies that the "agile" culture in question does not fulfil its potential for innovation.

This study supports previous findings in studies of coordination in large-scale agile projects (Moe & Stray, 2020; Šāblis et al., 2020).

I propose that EDI cultural traits can be a useful measure of "innovation capacity" in agile organizations.

Sammendrag

Mange organisasjoner, både i og utenfor IT-bransjen, er opptatt av å iverksette «smidige» måter å jobbe på. Men er smidige arbeidsmetoder det riktige svaret for alle typer organisasjoner?

Denne casestudien utforsker de kulturelle egenskapene til en «smidig» programvareorganisasjon. Kulturen fremstår med sterke fellestrekk til det som forbindes med medarbeiderdrevet innovasjon (MDI).

Ved å følge en induktiv tilnærming har jeg funnet noen mønstre i hvordan organisasjonen opprettholder de kulturelle verdiene sine og hvordan den smidige metodikken blir anvendt.

Det er utfordringer i prosjektorganisasjonen knytta til overordna retning og koordinering på tvers av team, da den smidige metodikken som er valgt i seg sjøl ikke gir mye støtte til fler-teams-arbeid. Organisasjonen omgår disse manglene i metodikken ved å bruke et sett med uformelle og formelle koordineringsstrategier «over» den smidige praksisen. Disse mekanismene på prosjektnivå fungerer sammen med den smidige praksisen for å støtte gjensidig tillit, åpenhet og trygghet utover teamnivå.

Dette er imidlertid ikke tilstrekkelig til å opprettholde autonomi, engasjement og samarbeidsorientering utover teamnivå. Dermed oppfyller ikke den «smidige» kulturen potensialet sitt for innovasjon.

Denne studien støtter tidligere funn i studier av koordinering i storskala smidige prosjekter (Moe & Stray, 2020; Šāblis et al., 2020).

Jeg foreslår at kulturelle egenskaper knyttet til MDI kan være et nyttig mål på «innovasjonskapasitet» i smidige organisasjoner.

Preface

This thesis is written as part of a Master of Management at the Norwegian University of Science and Technology (NTNU), specializing in Innovation and Change Management.

Working on this thesis has been a joy.

Great thanks to software group manager Torstein Heggebø at Nordic Semiconductor for letting me get on with it; to my supervisor at NTNU, Petter Grytten Almklov, for cheering me on; and to Magne, Hauk and Halvor for accepting reduced service levels at home.

The largest thanks I owe to my informants, The Project team members. You know who you are. There wouldn't be much to write about without your kind help and support. As our CTO likes to say, "We only hire smart people." This saying has come to mind frequently, while reading transcripts and writing summaries, and I can only apologize if the results do not give justice to everything that I have learnt during this process.

Thank you all for sharing your thoughts!

Table of Contents

Figures

Tables

Abbreviations

CoP	Community-of-Practice
EDI	Employee-Driven Innovation
NSD	Norwegian centre for research data
PM	Project Manager
ROI	Return On Investment
SM	Scrum Master
SW	Software
TPM	Technical Product Manager

1 Introduction

1.1 About This Thesis

"Happy people do good work." I don't know who said it first, but it has been at the back of my mind through management studies at NTNU Videre. For all the people that I have worked with, as a project manager in the IT industry, I think it is mostly true.

What holds people back from being happy and doing good work? Often, it comes down to not being able to work in a way that feels right. Often, people will blame "culture" if things are not good at work (www.stateofagile.org, 2020). This thesis explores how popular ways of working on software development are related to this thing called "organizational culture" (Schein, 2010).

Software developers are often unhappy. There is always more to build than there is time for. Creative problem-solving and writing beautiful code makes developers happy. Work that steals focus away from the creative process is a constant source of unhappiness. How can we organize work so that people avoid being more unhappy than they have to be?

We live in a time of technological breakthroughs that have great impact on how we live and how we work. This is not just an opportunity; it forces companies to rethink how work is organized. Companies that have employees who are willing and able to drive change will have a competitive advantage in almost any industry. For employees, this means that being a domain expert today may have little relevance tomorrow, and so the ability to learn, adapt, and respond to changing environments is becoming more important than knowing how to do the one thing you learnt in school really well (McAfee & Brynjolfsson, 2018).

This situation is driving many organizations towards "agility" both from the top-down and bottom-up perspective (McKinsey&Company, 2017; Moe & Mikalsen, 2020; West et al., 2021; www.stateofagile.org, 2020). Top-down, because managers need their organizations to react quickly and respond to change without waiting for directions and approvals. Bottom-up because people want flexible ways of working to fulfil their potential and take ownership of their work. From the top, we have the concept of "agile leadership" (Dalton, 2018). From the bottom, we have several "agile methodologies" for doing software development work.

I believe in an agile methodology called "Scrum" (Schwaber & Sutherland, 2020). Scrum is a work methodology for software development for small, self-organizing workgroups. I like it because I think self-organization is a neat idea. Also, I think it is innovation-friendly, as it prescribes working in short iterations in a "muddling through" sort of fashion.

There are two problems with Scrum:

1) Scrum is a kind of belief system. You must subscribe to its values, and you have to understand how work practices relate to those values. If you do not believe in the values, then the methodology is not meant for you.

2) Many software development projects are too large to be taken on by some small, self-organizing workgroup. (There are several frameworks prescribing additional methods for coordinating work between self-organizing teams, but a fundamental problem remains: Being coordinated means giving up autonomy.)

In many organizations, there is a recurring discussion on "scaling Scrum": Does it make sense to use this methodology for large and complex projects? Isn't it rather so that people will become more unhappy and less productive if they try to follow practices that do not fit with their work situation? Could it even be that by insisting on working in this way, we restrict what kind of solution architectures we are able to implement successfully? And while we're at it: How smart is the idea of self-organizing workgroups anyway? It may be easy to self-organize in a carefully selected group of people with just the right variety of competencies, "soft" skills, problem-solving styles and ambitions for self-improvement – but what about the rest of us?

As it happens, I work in an R&D department where one of the projects have been scaling Scrum for quite a while. I decided to use this project as a case study. To be exact: I decided to design a study that could demonstrate how using the right methodology in the wrong way can have a negative impact on organizational culture.

Spoiler Alert: It did not work out that way. The project teams did not come across as particularly unhappy, and the work culture was not dysfunctional as far as I could tell. To the contrary, there was an unmistakably positive vibe all around. I seem to have stumbled upon the truth that "culture trumps strategy"[1] – but in a good way.

In the remainder of this chapter, I introduce my research questions (1.2) and explain the methodological approach and purpose of my case study (1.3).

1.2 Background and Problem Statement

Towards the end of the last century, the software industry went through a period of rapidly changing environment while contracts and working methods favoured stability. Technological breakthroughs, in everything from hardware size to programming paradigms, made possible ever larger and more complex technical solutions. To coordinate and control work of ever-increasing complexity, the industry spent vast resources on project management, waterfall planning[2], and requirement specifications (Cockburn & Highsmith, 2001).

This project management approach[3] did not work well for software delivery. Eventually, the developers said so: We need "A better way of working" (*Manifesto for Agile Software Development*, 2001). Such ways of working, meant to support flexible software development by teamwork, have since become commonly known as "Agile methods" (Conboy, 2009) (Cockburn & Highsmith, 2001). Their foundation is the "Agile Manifesto" containing these core elements:

- **Individuals and interactions** over processes and tools;
- **Working software** over comprehensive documentation;
- **Customer collaboration** over contract negotiations;

[1] Peter Drucker (1909-2005) might have mentioned something along these lines already.
[2] Sequential, phase-based planning. For instance: Specification, Design, Development, Test phases to be executed in sequence so that one project phase must be closed before the next can be started.
[3] See also: "scientific management" (Frederick Winslow Taylor (1856-1915))

- **Responding to change** over following a plan.

(*Manifesto for Agile Software Development*, 2001).

The Manifesto can be read as the expression (by its signatories) of basic assumptions about what is deemed good and valuable (Wendorff, 2002).

My topic of interest for this study is how agile methods work in a larger organizational context, where not everyone can be expected to share the same set of basic assumptions, values, and cultural preferences. People might not have the same understanding of purpose, or of what makes work meaningful. Software development is creative work, and we do know that organizational climate[4] can be a driver or a barrier to creativity in the workplace (Hunter et al., 2007). We also know that certain cultural characteristics such as autonomy and engagement are important for employee-driven innovation (Aasen & Amundsen, 2015). What are the relationships between work methods and cultural factors when several agile teams are coordinated in joint software development efforts?

This leads to a two-part research question:

- RQ1: How can cultural factors affect agile practices in multi-team projects?
- RQ2: How can agile practices affect cultural factors in multi-team projects?

Thus, the overall research question is the interrelationship between organizational culture and agile practices. The Theory chapter (2) will explain in some more detail what "agility" is; why agility can be difficult in large organizations; - and what we already know about the relationships between organizational culture and agility, based on empirical research as well as theoretical foundations.

1.3 Approach

This research has been exploratory and followed an inductive approach. The purpose of my case study was to understand in some detail the circumstances in one particular organization and, if possible, extract some transferable knowledge. I also wanted to involve the people in this organization in the process.

In the initial planning phase, my approach was deductive. I thought of doing a large organizational climate survey with lots of statistical analysis. I did a fair bit of reading up on agile culture and how to measure it before discarding this idea: For one thing, I didn't find any pre-validated survey that I really liked that seemed relevant to my type of organization. I thought that Aasen & Amundsen's list of cultural characteristics for employee-driven innovation (Aasen & Amundsen, 2011, pp. 173-176) was most in line with what I would like to measure, but couldn't find any surveys for that. Eventually I came to realize that the type of information that I wanted would be better communicated in conversation than by Likert scale.

[4] Organizational climate: "the general character of the total organizational environment as perceived by those who work within it. It is an expression of the organizational culture." (American Psychological Association dictionary) Association, A. P. Retrieved 29.12.2021, from https://dictionary.apa.org/

So, I changed the approach to a more open-ended design based on focus group interviews, turned back to Aasen & Amundsen, and made my own survey to have as a conversation starter. "Individuals and interactions over processes and tools," I thought.

This decision meant giving up control and accepting that people might want to talk about other things than what I had in mind. Rather than trying to fit their statements into a predefined coding scheme, it seemed reasonable to base the analysis on the code groups that emerged as I sorted the different conversation topics (Tjora, 2021). "Responding to change over following a plan," I said to myself.

The last part of my work was trying to fit the analysis into a larger context, asking questions such as - what kind of phenomenon is this? Are there any theories that can explain this? It may come as an unhappy surprise to the reader to find that there is more theory in the Discussion (Chapter 5). On the other hand, to pretend that I had planned from the outset to use all this theory would obfuscate the inductive premise of the study. Adding theory in the text where it appears to be relevant seemed to be a better way of working.

1.4 Structure

This chapter aimed to present the problem domain that is the subject for my thesis, and quickly explain my research approach.

The next chapter (2) introduces relevant theory and current research.

The Methodology chapter (3) describes research methodology and case, retrofitted with some discussion on the appropriateness of my design choices.

The analysed results are reproduced in the "Results and Analysis" chapter (4). This is deliberately kept short and close to actual conversations.

Based on the analysis, I discuss the research questions one by one (Chapter 5), pulling in new theory as needed to make my points.

Finally, based on the discussion, Chapter 6 is a summary of what we have learnt about the research questions; what the research contribution is; and what practical use the studied organization can make of this knowledge.

2 Theory

2.1 About This Chapter

Many organizations, both within and outside of the IT industry, are busy implementing "agile" ways of working (McKinsey&Company, 2017; Moe & Mikalsen, 2020; West et al., 2021). This can imply new work processes, re-organization, and new ways of thinking about leadership. Such change processes often meet with resistance: In recent surveys, many agile professionals assess their own organization as "not very agile," citing "existing culture" as the main barrier. (www.stateofagile.org, 2020). This has inspired a line of research into how existing organizational cultures can be influenced to become "more agile" - so-called "Agile Transformations" (Abrar et al., 2020; Goran et al., 2017; Hoda et al., 2017; Kalenda et al., 2018; Laanti & Kettunen, 2019; Spiegler et al., 2019; Struckman et al., 2020; Tan Trung et al., 2019). But what is "agility," and why do we want it?

This chapter first elaborates on what agility means and explains that there is no clear definition (2.2). I then discuss the concept of "agile culture" (2.3). I suggest that cultural values expressed as agile values are related to cultural properties associated with Employee-Driven Innovation (2.4). This is the theoretical foundation for my research design. I then introduce some more general concepts from organizational theory related to *trust*. These theories will be relevant for the discussion in Chapter 5.

2.2 What is "Agility"?

There is no unanimous definition of "agility" in software development (Conboy & Carroll, 2019). Some will argue that «being agile» means following the intention behind the Agile Manifesto, while others apply the term to using agile *methods* as part of a work practice. In this text, because intentions are hard to measure, I will refer to teams following agile practices as «agile teams», - fully acknowledging that this is an over-simplification.

Having "agility" as a guiding principle for an organization fits badly with the (tayloristic) image of the organization as a machine, with work being executed in an assembly line and controlled by a central managerial "brain". A better fit is the "organic" organization: A living organism, where clusters of cells react to changes in the environment (McKinsey&Company, 2017) (Figure 1).

Figure 1 - "The 5 trademarks of agile organizations" (McKinsey&Company, 2017)

This is not new ideas: The "organic" organization is described in the works by organizational theorists Burns and Stalker (Burns & Stalker, 1966) and similar to Henry Mintzberg's "adhocracy" (Mintzberg, 2009), - and Gareth Morgan elaborates on different "species" of organizational organisms as well as "organizations as brains" in *Images of Organization* (Morgan, 2006). What is new, then, is the eagerness of consultancy companies and industry players alike to discard the pyramid[5] and embrace the spheric organization – and to this end, their affinity for work methods developed by and for small teams of software developers.

The most commonly used framework for work management following agile methodology is Scrum (www.stateofagile.org, 2020). Scrum is based on the idea of self-organizing teams (Schwaber, 1995). The name as well as idea is taken from Nonaka and Takeuchi's description of how rugby teams operate, as a picture of "lean" product development (Takeuchi & Nonaka, 1986)).

[5] Possibly accelerated by Covid-19 making it difficult to exercise command-and-control type leadership from the home office…

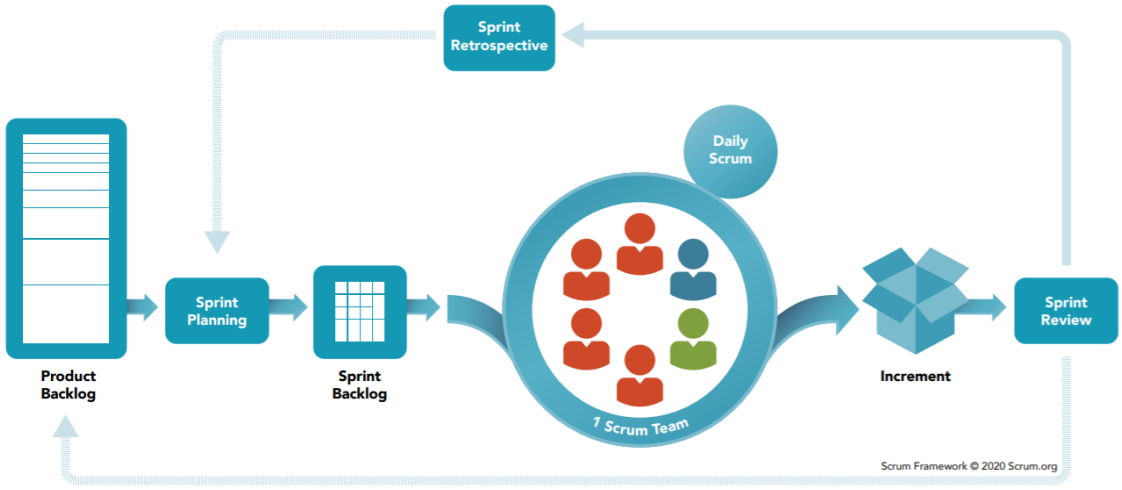


Figure 2 - The Scrum cycle. Taken from (What is Scrum?, 2020).

Scrum embeds the assumption that "talk is good" in organizational practices, through frequent, periodic "events" where the whole team meet, talk, and make joint decisions. Formal roles are of lesser importance: Someone must be "Scrum Master" and make sure that the team follows practice. Someone must take the role of "Product Owner" and set priorities. Apart from that, it is up to the team to decide if more roles are needed (Cohn, 2010, pp. 137-153).

The figure (Figure 2) illustrates the eternal Scrum sprint cycle. In "Sprint Planning" the team agrees on goals and tasks for the next work iteration (sprint), and during the sprint the team meets daily in the "Daily Scrum" to exchange status and agree on who will be doing what next. The sprint iteration is closed with an open presentation of work done in the "Sprint Review" followed by a retrospective meeting – and then the cycle repeats with a new Sprint Planning (*What is Scrum?*, 2020). A sprint cycle typically takes two to four weeks. By following the cycle repeatedly, teams learn and improve their work practices over time, in small iterations – just as they explore and develop working software solutions over time, in small iterations.

We see that the Scrum cycle is completely team centric. When several teams collaborate, one more event is recommended: The Scrum-of-Scrums. This is a standup[6] where representatives from every collaborating team meet to coordinate their work, "as often as necessary". Higher-level coordination is not part of the Scrum methodology, although there are several frameworks to choose from, with varying levels of rigidity, for "large-scale agile".

2.3 Is There Such a Thing as an "Agile Culture"?

In this context, "culture" should be read as "organizational culture" following the definition by Edgar Schein (Schein, 2010, p. 18):

> The culture of a group can now be defined as a pattern of shared basic assumptions learned by a group as it solved its problems of external adaptation and internal integration, which has worked well enough to be considered valid and, therefore, to be taught to new members as the correct way to perceive, think, and feel in relation to those problems.

[6] Brief meeting, preferably taken standing up if done in-person.

If we refer back to the Agile Manifesto (1.2) for a moment, we see that it seems to be built on a pattern of basic assumptions related to the problem of adapting traditional production techniques to software development.

The founders of Scrum, Ken Schwaber and David Sutherland, have made their basic assumptions even more explicit by defining five "Scrum Values":

Commitment, **Focus**, **Courage**, **Openness** and **Respect** (Schwaber & Sutherland, 2020).

Schwaber and Sutherland claim in the Scrum Guide ("Scrum Values" section) that teams following their methodology will over time **become** "agile": "The Scrum Team members learn and explore the values as they work with the Scrum events and artifacts" (Schwaber & Sutherland, 2020). If this is correct, it means there is a cause-effect relationship between practice and cultural values. This is indeed a popular belief among practitioners: "it is possible to influence and improve culture by implementing DevOps practices" (Forsgren et al., 2018, p. 29) – and "I believe that changing practices and methods sooner or later must lead to a cultural change" ((Küpper, 2016) – interview with agile coach).

Empirically, it is difficult to verify claims that Scrum "works" and makes the culture more agile, because we do not have a clear definition of agility (Conboy & Carroll, 2019; Iivari & Iivari, 2011). A growing number of researchers argue that perhaps is it more precise to say that Scrum is **made for** a certain type of culture rather than **creating** a certain type of culture (see for instance Bunyakiati et al for a discussion of agile "cultural fit" from a non-European perspective (Bunyakiati et al., 2016)) (Šmite et al., 2020; Tolfo et al., 2011).

An early comparative study by Strode et al. (Strode et al., 2009) linked six cultural factors to successful application of agile methods (Table 1).

Table 1 - Organizational culture factors linked to agility, according to Strode (Strode et al., 2009)

	Organizational culture factors
1	The organization values feedback and learning. Social interaction in the organization is trustful, collaborative, and competent. The project manager acts as a facilitator. The management style is that of leadership and collaboration.
2	The organization values teamwork is flexible and participative and encourages social interaction.
3	The organization enables empowerment of people.
4	The organization is results oriented.
5	The leadership in the organization is entrepreneurial, innovative, and risk taking.
6	The organization is based on loyalty and mutual trust and commitment.

Strode et al made it clear that this was a quantitative study, and as such should not be used to infer any causality.

Other research suggests that some culturally distinct behaviours will impede the application of agile methods. A recent study by Šmite et al summarizes previous work on

barriers to agility (Table 2) (Šmite et al., 2020), identifying a set of behaviours that are counterproductive to agile ways of working.

Table 2 - Culturally distinct behaviours impeding agile ways of working (see (Šmite et al., 2020) for a list of references))

Level	Impeding behaviour
Management behaviour	Command-and-control mindset, reinforced deference to superiors
	Leadership style discouraging team members from exposing problems
	Leadership style discouraging from proposing alternatives to perceived directives from superiors
Engineers' behaviour	Willingness to say yes to most requests in deference to superiors, reluctance to warn about non-feasible deadlines
	Reluctance to expose problems
	Lack of commitment to self-learning, reliance on top-down improvements
	Reluctance to engage in constructive disagreements and challenging discussions or voicing criticism
	Reluctance to propose alternatives to perceived directives from superiors

With these studies in mind, we do know quite a lot about cultural prerequisites for a successful "agile transformation". Still, there is limited knowledge about the cultural characteristics of organizations that have already been applying agile methods on a large scale over a long time.

2.3.1 How to Measure Culture

Organizational culture, as defined above, is not a straight-forward thing to measure. One approach is to use "organizational climate" surveys, which are general tools for probing culture indirectly. Such surveys are based on comprehensive questionnaires covering employees' perceptions of different aspects of the workplace: working environment, management approach, communication styles and behaviours – including dimensions such as autonomy, communication, and supervisory support (Patterson et al., 2005). There are also climate surveys specifically targeted to measure creativity and innovation (Amabile et al., 1996) (Hunter et al., 2007; Isaksen & Akkermans, 2011). Such climate measures have been found to be effective predictors of creativity and innovation at both the individual and group level, according to a systematic literature review by Hunter et al (Hunter et al., 2007). So for practical purposes, it may be possible to use a questionnaire to find out how well a culture supports specific capabilities - if the questionnaire was made for organizations similar to the one you are trying to measure. The main problem with questionnaires as I see it is that the results in themselves aren't very usable: There is no way of knowing why respondents answered in a certain way, so it is not very clear (to me, at least) how to understand such results or how to act on them.

2.4 Innovation as Part of Everyday Work

Two complementary definitions of innovation and creativity have been proposed by Toril Oddane (Oddane, 2017, p. 260) (my translation):

> "Innovation is a collective, open activity aiming to create and implement new, applicable products/processes which create values of an economical and/or other nature"

and

> "Creativity is an individual and collective ability to create something new and applicable as a response to an open problem".

From a software developer's perspective, it seems rather obvious that development practices must allow for and support creativity to be effective. Scrum does seem to integrate "creative" activities in the daily work, by how the iterative work process encourages disciplined improvisation, prototyping, and experiments (Cohn, 2010; Conboy et al., 2009). Following the methodology, these creative activities could be driven by any team member and over time lead to new and innovative software solutions and processes. This type of bottom-up innovation resembles the definition of Employee-Driven Innovation:

> "Employee-Driven Innovation refers to the generation and implementation of new ideas, products and processes – including the everyday remaking of jobs and organizational practices – originating from interaction of employees, who are not assigned to this task. The processes are unfolded in an organization and may be integrated in cooperative and managerial efforts of the organization. Employees are active and may initiate, support or even drive/lead the processes. (Høyrup, 2012: 8.)"

(Amble et al., 2020, p. 43).

How to encourage such innovation among company employees may depend on many things, related to work climate, organization, type of company, and type of industry. Drivers for innovation can be different in Eastern and Western cultures (Anderson et al., 2014), and according to a study of Norwegian companies which "each in their own way have succeeded in involving employees in innovation and continuous improvement" (my translation) – there is no one best practice, even within the national culture (Nærings- og handelsdepartementet et al., 2011). However, according to the same study, the Norwegian companies that succeeded show common cultural characteristics: Åpenhet (Openness), Tillit (Trust), Samarbeidsorientering (Collaboration Orientation), Sikkerhet (Safety), Toleranse (Tolerance), Engasjement (Engagement), Stolthet (Pride), Utviklingsorientering (Development Orientation), and Autonomi (Autonomy) (my translation in parenthesis).

The following table (Table 3) gives a summary of the above characteristics and how each is described by Aasen & Amundsen (Aasen & Amundsen, 2011, pp. 173-176). Although they are addressing different aspects of culture, the traits are interrelated (for instance, "Safety" and "Tolerance" – it is hard for anyone to feel safe in an intolerant work environment, and if you do not feel safe you are perhaps not so inclined to be tolerant towards your co-workers either). Some traits are more fundamental than others (for instance, to be open with someone you first need to trust them). This makes it a little bit difficult to discuss each trait in isolation. I have tried to clarify by adding from the descriptions which traits seem to act as prerequisites for others in the "Supported by" column. Then, going back to the Scrum Guide (Section 2.3), I have made an attempt to map each Scrum value to these EDI traits. For instance, I choose to assume that the values Courage and Respect map to a culture characterized by Safety and Tolerance.

Table 3 - EDI cultural traits and Scrum values

Cultural trait	Description	Supported by	Scrum Value
Openness	Making information freely available, being open to new ideas, allowing divergent thinking.	Trust, Safety, Tolerance	Openness
Trust	Mutual horizontal and vertical trust, feeling confident that people are competent and able to take responsibility.	Openness, Safety	Respect
Collaboration Orientation	Seeking opportunities for collaboration, joint decision-making.	Trust	
Safety *	Feeling free to share thoughts and ideas and ideas, even if they might seem "stupid" or "difficult". Organizational encouragement.	Tolerance, Trust	Courage
Tolerance	Accepting that people are different and accepting that people make mistakes.	Trust, Safety, Openness	Respect
Engagement	Willingness to put in an extra effort to reach goals, based on commitment and ownership.	Trust, Safety and Autonomy	Focus Commitment
Pride	Feeling good about work environment, product, and company reputation.	Engagement	
Development Orientation	Seeking opportunities for improvement and learning.	Safety, Trust	
Autonomy	Enabling people to take full ownership of their work by allowing them independence, empowerment, and self-organization.	Trust, Safety, Openness	

*) Amundsen and Rismark (Amble et al., 2020) associates "Trygghet" with the broader term "**organizational encouragement**" as used by McLean (McLean, 2005). McLean refers to Amabile in his definition: "'organizational encouragement'" encompasses several aspects, including encouragement of risk taking and idea generation, supportive evaluation of ideas, collaborative idea flow, and participative management and decision making (Amabile et al., 1996)" (McLean, 2005, p. 236).

It is worth noting that these cultural characteristics were identified through a large *qualitative* study and as such should not be mistaken for quantitatively validated constructs with similar names. Terms like "Autonomy" and "Engagement" could mean something different here than what would be measured in a climate survey (ref. Section 2.3.1). It should also be noted that no comparison was made between "EDI" and "non-EDI" organizations in Aasen & Amundsen's study, so we do not know if any or all these factors are specific to EDI. What we do know is that they appear to be a prerequisite for EDI (at least in a national context). Literature reviews such as Ahmed (Ahmed, 1998) and Hunter et al (Hunter et al., 2007) list similar traits as indicators of "innovation climate". Moreover, these traits seem to have a great deal of overlap with Strode's cultural factors for agility (Table 1), and to be related to the Scrum values. This leads to the question, which elements of an EDI culture are most important to an agile organization?

2.5 Trust

Based on the inductive analysis (coming up in Chapter 4), trust emerged as a key concept for explaining the Scrum teams' way of working. In my discussion, I will apply some different theoretical perspectives to shed light on trust in organizations.

The role of trust in organizational culture is well known: In a society with high level of trust between its members, relationships and behaviours are governed by shared norms and values rather than formal governing mechanisms (Jacobsen & Thorsvik, 2013, p. 129).

The presence of trust seems inversely proportional to the need for monitoring and control (Ouchi, 1980). Trust frees up organizational resources: Vertically, because managers can engage in coaching and servant leadership rather than command-and-control behaviours if they trust their subordinates to do the right thing. Horizontally, because people can mind their own business without worrying about what their peers are up to if there is an "assumption of good intent" between co-workers.

Loss of trust breaks down existing cultures and creates negative spirals of fear and defensiveness. Trusting relations are thus a prerequisite for building and improving culture. This creates a paradox, since collaboration requires some level of trust and trust is created through collaboration (Sørhaug, 1996). In recent years, the concept of trust has received much attention within the fields of transformational leadership and in change management, since it is notoriously hard to implement any kind of change successfully if the people asked to contribute to the change do not trust those initiating the change (or vice versa) (Amundsen & Kongsvik, 2016; Burns, 1979; Kotter, 1995).

Closely related to trust is the concept of "psychological safety" (Edmondson, 1999). It was popularized in the software industry after Google's Project Aristotle (Google, 2015). This company-internal research project asked "What makes a team effective at Google?" After two years, Google concluded that the key factor was "psychological safety": "Team members feel safe to take risks and be vulnerable in front of each other". Obviously, if team members do not feel that it is safe to take risks in front of each other, this will limit the flow of communication. People will keep their thoughts to themselves. But there are other, severe, consequences as well if psychological safety is lacking: According to neuropsychology, emotions of fear, stress and anxiety can be detrimental to our capacity to learn (Edmondson, 2019). In other words, investing in psychological safety is smart because it will unlock several other cultural capabilities.

2.5.1 Social Network Theory

Social network theory is a line of research targeted at explaining how social relations shape organizations and societies. Social network theory is based on four concepts (Kilduff & Brass, 2010, p. 310):

- the primacy of social connections,
- embeddedness in social fields,
- the social structuring of activity, and
- the social utility of connections.

Network researchers Kilduff and Brass have argued that the classic works on job design by Oldham and Hackman fail to consider these concepts. By treating individuals as

"atoms" without looking at the forces between them, they say, you do not have the full picture of what drives job satisfaction and job performance (Kilduff & Brass, 2010)[7].

From a larger perspective, every organization is a network of people. There is the formal network, defined by organizational charts and reporting lines, but in addition to that, there are informal network connections between people. These informal networks within the organization are key to understanding "invisible" connection brokering and dissemination of information (Schiefloe, 2015).

Networks with lots of connections between people are known as "dense" networks. Dense networks with few connections to the outside are (not surprisingly) called "closed" networks. Information and ideas tend to stay inside a "closed" network. Conversely, dense, "open" networks with many connections to other networks have a potential to circulate knowledge more widely. Organizations are often networks-of-networks, with network "clusters" of varying density (Schiefloe, 2015).

According to Schiefloe (Schiefloe, 2015), Cross and Parker have shown how different network topologies will have different types of impact on organizational and individual performance (Cross & Parker, 2004): Organizations with high-density networks outperform organizations with clustered or star networks. Individuals with large and varied networks outperform individuals with more limited networks. This research has probably inspired a fair bit of "team building" activities over the years and kept event management agencies in business. However, more recent studies show that there is most likely an inverse U-shaped relationship between network density and network performance: Increasing the number of network connections will increase the network's performance until it becomes too much and performance drops off (Wise, 2014).

The term "Social Capital" was first made popular by sociologist Pierre Bourdieu, referring to how different social classes have access to different social resources. Robert D. Putman associates it with "features of social organization, such as networks, norms, and trust, that facilitate coordination and cooperation for mutual benefit" (Putnam, 1994, p. 7). A simpler definition is "investment in social relations with expected returns" (Dubos, 2017).

Nahapiet and Ghosal (Nahapiet & Ghoshal, 1998) suggested that it is useful to think of social capital as having a structural, relational, and cognitive dimension. With this distinction, we can talk about trust and trustworthiness as key facets of the relational dimension of social capital (Nahapiet & Ghoshal, 1998).

Looking at social capital as an investment in social relations from the individual's point of view, the value of your investment is related to who and how many people you connect to (this is the "structural" dimension). But there is a difference between strong ("friends") and weak ("acquaintances") connections. Although strong connections are important, these connections are often confined to a limited and relatively closed group of people. Weak connections are more likely to be spread out over a larger variety of people, each with their own strong connections, and so are a better source of diverse information. Individuals who build social capital in both strong and weak connections become "brokers" who can connect the disconnected people (Schiefloe, 2015). Since

[7] In all fairness, job design theorists Oldham and Hackman themselves were also picking up on this in 2010: "But circumstances change, and the time is now right for research that focuses squarely on the social aspects of the work itself." Oldham, G. R., & Hackman, J. R. (2010). Not what it was and not what it will be: The future of job design research. *Journal of organizational behavior*, *31*(2-3), 463-479.

people tend to cluster – bond with those in the same work group, discipline, or with other similarities – brokers become bridges between clusters in the workplace. Putnam talks about "bonding" social capital as opposed to "bridging" social capital: "Bonding" capital is what you build within a cluster, while "bridging" capital is the social capital you get from connecting different clusters.

There are many beneficial outcomes for the organization if people connect across sub-units (build "bridging" social capital). According to Kilduff and Krackhardt, the greater the ratio of external to internal friendship ties across an organization's subunits, the more capable is the organization of engaging effectively in "new, untested, unlearned behaviors to obtain or maintain the organization's desired goal state" (Kilduff & Krackhardt, 2008, p. 211). (One can only assume this means that having friends outside of your own department is good, and not that it means having friends inside of your own department is bad.)

2.5.2 Communities-of-Practice

"Communities-of-Practice" (CoPs) researchers specifically address those types of networks where practice is shared (the practice is what the network is about and what gives it identity) and seek to explain how they turn information into knowledge. (Duguid, 2005)

"Communities-of-Practice" was introduced as a concept by learning theorists Lave and Wenger (Lave & Wenger, 1991) as a by-product of their studies in situated learning. When you enter a new work environment as an apprentice, you go through a period of adapting to the environment before you become an insider in that specific CoP. This kind of adapting to the environment, where you acquire the skills and ability to behave like a community member, is called "legitimate peripheral participation".

Brown & Duguid (Brown & Duguid, 1991) went on to suggest that what happens inside CoPs is fundamental to understanding how working, learning and innovation can fit together as complementary activities.

Duguid argues that learning situations should be as close to the work situation as possible. If we accept Polanyi's statement "we know more than we can say", it follows that knowledge has a tacit (implicit) as well as an explicit dimension (Duguid, 2005). Having a conversation is different from working together. *Knowing how the thing is done is something we learn by doing it*[8]*.* This shared knowledge is what sets a community apart as a community *of practice* as opposed to a community *of interest* or other social network.

Within a community, says Duguid, as part of learning the practice, people will develop a shared identity related to the practice; common ways of interpreting information; and shared ways of thinking about the practice. Having these shared ways of thinking are important because if we do not have "epistemic commitment", Duguid argues, then "no amount of bowling together will bring about shared, actionable knowledge."[9] (Duguid, 2005). Bad news for team event agencies.

[8] This is also the reasoning behind Scrum's emphasis on teamwork, ref. Takeuchi and Nonaka.Takeuchi, H., & Nonaka, I. (1986). The New Product Development Game. *Harvard Business Review*(January). https://hbr.org/1986/01/the-new-new-product-development-game

[9] OK, probably pun on Putnam's "Bowling alone".

Brown & Duguid see organizations with different independent communities as having a high potential for innovation because of the possible "friction of ideas" between CoPs (Brown & Duguid, 1991, p. 54).

However, since the CoP is not a formal structure and will change over time, it can diverge into sub-CoPs with their own identities and understandings (cultural differentiation), or the CoP can lose relevance to its members (cultural fragmentation) - which can make collaboration awkward, even within the CoP (Duguid, 2005).

According to Wenger, managers must find ways to cope with the boundaries between communities so that CoPs do not become barriers to shared understanding and collaboration in the workplace. Different ways of doing this can be to rely on people who are part of several CoPs to act as brokers between them; by having shared "artefacts" (stuff that is common to several CoPs, such as processes and tools); and by having interactions (such as task forces or communities of interest) between the CoPs as ways of "negotiating meaning" (Wenger, 1999, pp. 103-108). Tidd & Bessant further distinguish between "translators" as people who can explain one CoP's perspective to another CoP, and "knowledge brokers" who participate in several CoPs and thereby transfer knowledge between them (Tidd & Bessant, 2018, p. 511).

With regards to agile organizations, the need for boundary-spanning activities has been recognized and discussed for some time. A well-known attempt to create an organizational structure that balances close-knit teams with larger communities is what we now know as the "Spotify model" (Smite et al., 2019). This model was first described in 2012 in a blog post by two agile coaches in streaming company Spotify's R&D department (Kniberg & Ivarsson, 2012). The model introduces the "Guild" as a semi-formal community-of-interest where people from different teams and departments (called "Squads" and "Tribes") come together. Other companies have tried to copy this model with varying levels of understanding and success (West et al., 2021).

Joanne Roberts makes the observation that when it comes to deliberately trying to build CoPs into the organizational structure, again, trust is the key ingredient: a CoP will not be very effective in sharing ways of thinking about things if there isn't a high degree of trust between its members. If power relations and top-down management styles undermine trust and openness, there will be less informal knowledge transfer within the organization – meaning that trying to facilitate CoPs is not always a good management tool: "Indeed, communities of practice may be better suited to harmonious and trusting organizational environments in which workers are given a high degree of autonomy." (Roberts, 2006, pp. 628-629).

2.6 Summary

This chapter was focused on a few key concepts which are central to understanding the subject for this study: "Agility" (2.2), "Agile culture" (2.3), and "Innovation culture" (2.4). I used "Agility" as a bucket term for certain ways of working, with Scrum as the most predominant agile form; "Agile culture" for work culture among professional groups using Scrum or other agile work methods; and "Innovation culture" for work culture associated with Employee-Driven Innovation (EDI). I claimed that "Agility" and "EDI" are related concepts. Trust was introduced as a key concept to understanding organizational cultures and related to Social Capital and Communities-of-Practice.

3 Methodology

3.1 About This Chapter

This chapter describes my case (3.2) and methods used for data collection (3.4) and analysis (3.5). I give some rationale for using a mixed-methods approach (3.3) and evaluate my design choices with regards to reliability and generality (3.6).

3.2 Case

This study is based on a single case (one project organization) from my own workplace. Having only one unit of analysis provides an opportunity for mixed-methods data gathering and in-depth analysis. Since my data gathered by different methods are known to apply to the same case, there is no risk of misinterpreting the results due to cross-case differences.

The case is a software development project, using project participants as the primary data source. This project, hereafter referred to as "The Project", has been running for years. It has gone through several stages, from a one-team "startup" type customer-driven research effort, via technological changes and standardization work, to a streamlined development machinery of forty people across six teams (at the time of writing). It is an interesting case to study because it started out as one Scrum team and has stayed true to the agile methodology while growing in size and complexity. The software developed by The Project today is time-critical "low-level" components widely used in consumer electronics. The work itself favours specialization, as it requires a deep understanding of the underlying hardware, knowledge of different parts of the current code base, and competencies within a plethora of tools and frameworks.

The formal project organization and coordination mechanisms are illustrated by Figure 3 (this information is based on pre-study interviews and access to the project wiki).

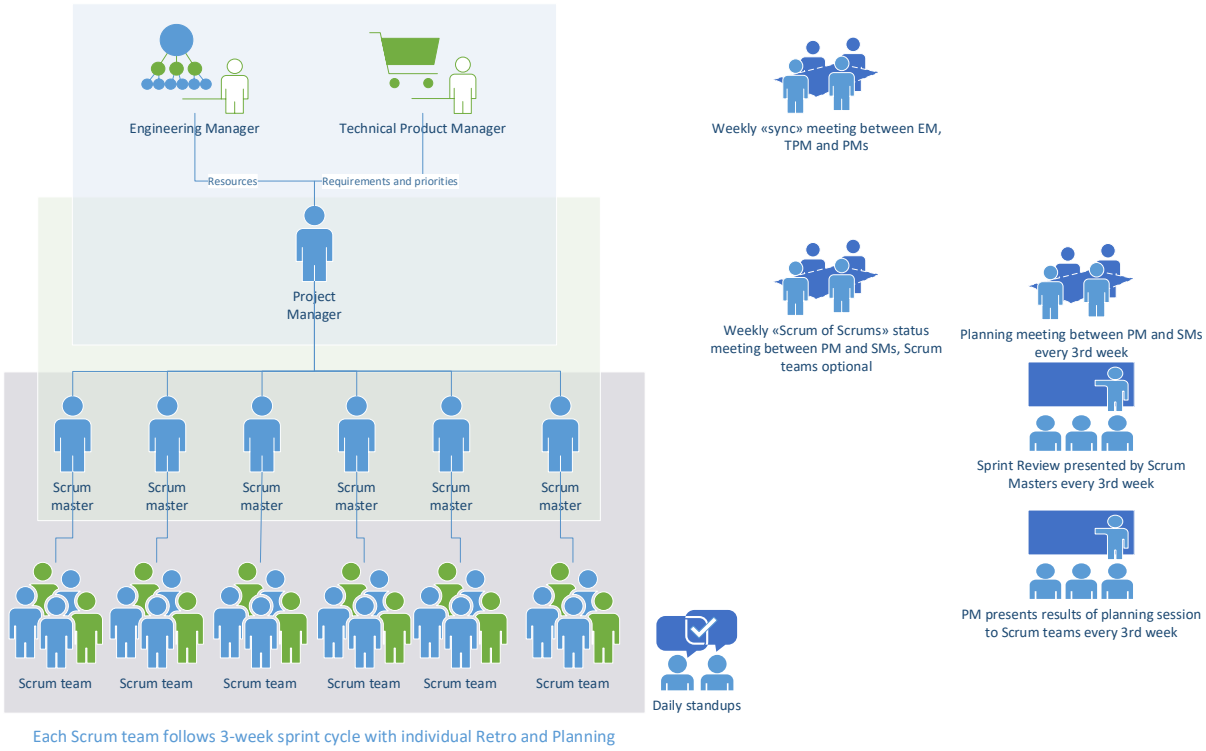


Figure 3 - Project organization and formal coordination mechanisms (orgchart and communication lines to the left, coordination meetings at each level to the right)

The project organization consists of around 40 people spread across six Scrum teams, and a Project Manager reporting to an Engineering Manager. The Project Manager acts as a bridge to the outside world. He receives customer requests from a Technical Product Manager (TPM) and communicates requests and priorities from the TPM to the teams. Communications between Project Manager and Scrum Teams mostly go through the Scrum Masters.

This gives three levels of formal coordination,

a) Coordination between the project (represented by Project Manager and Engineering Manager) and the environment. Most of the external coordination happens through the TPM.

b) Coordination between project management and teams within the project. Project coordination happens almost exclusively between project manager and the teams' Scrum Masters.

c) Coordination within Scrum teams. Team coordination often happens in a spokes-and-wheel manner with Scrum Master as the "wheel" coordinating work tasks between the others.

3.3 Research Design

My textbook in social science *Introduksjon til samfunnsvitenskapelig metode* says "It is argued that social reality is by definition soft and must be studied with soft data" (google's translation) - «Det hevdes at den sosiale virkeligheten per definisjon er myk og må studeres med myke data (Nyeng 2004)» (Johannessen et al., 2016, p. 34). I agree based on the assumption that the phenomena I try to study are in the domain of social reality and as such are best understood by communicating with the inhabitants of that

reality. As a pre-study, I have interviewed a few people to get familiar with The Project. Being an engineer, however, I could not resist using a survey for gathering *some* quantitative information as a starting point for defining my scope of research. This would give me an overview of how the project participants saw different aspects of their culture and could be combined with the data gathered from interviews to generate a richer data source for analysis. Using multiple methods and data sources like this is known as "data triangulation" and discussed in Section 3.6.2.

In Grounded Theory terms, the pre-study results became my "sensitizing concepts" (Bowen, 2006). The survey pointed to certain themes that became topics for my main data gathering activities, which were focus group conversations. My analysis, based on recorded data from these conversations and interviews, followed the Stepwise-Deductive Inductive method (Tjora, 2021).

The goal of the Stepwise-Deductive Inductive method is to generate new theoretical concepts through an iterative bottom-up search for emergent patterns in empirical data (the process is described in Section 3.5). My reason for choosing this method, rather than mapping data to a predefined coding scheme, was mainly because the stepwise coding process allowed me to create codes based on what people were saying ("open" coding). I did not hope to come up with a brand new theory, as would be the objective of a "real" Grounded Theory study, but rather to use the method to compare my results with current theoretical perspectives.

The research process is summarized in Figure 4.

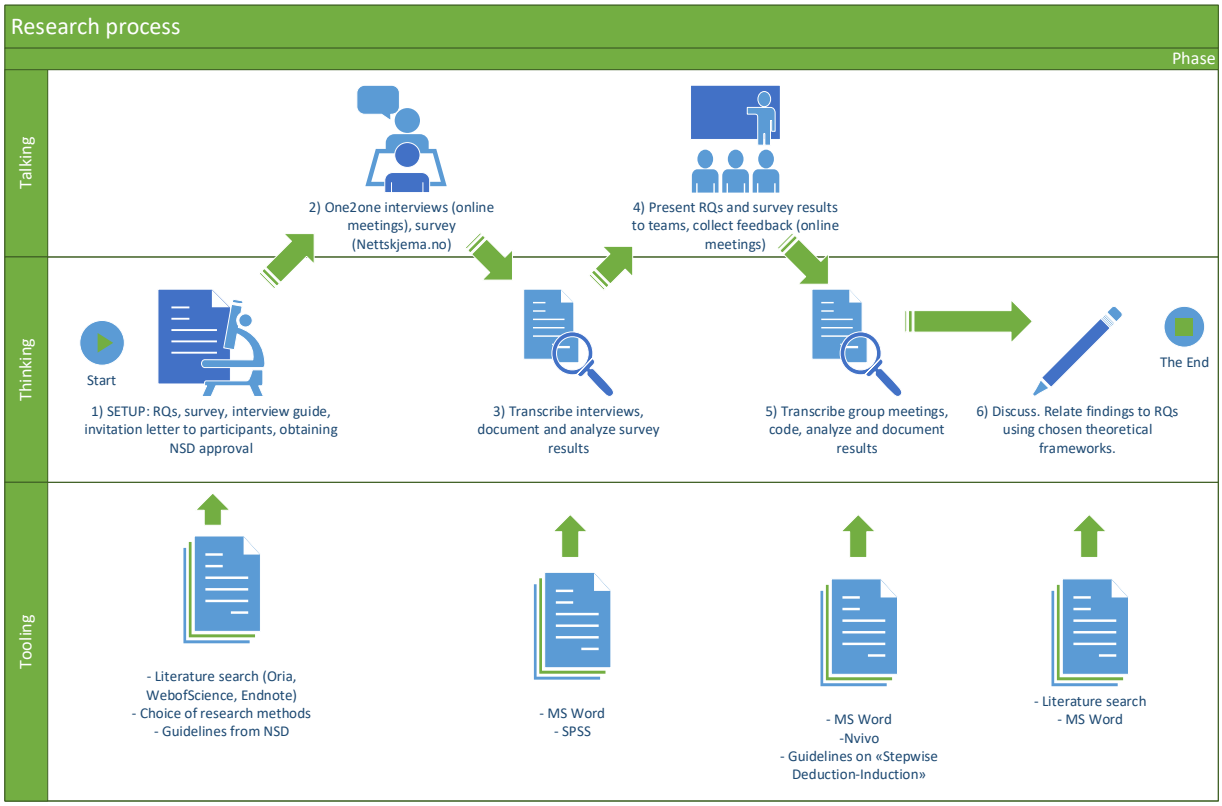


Figure 4 - Research process

Kongsvik and Almklov say that when studying social systems, ”combining methods has a synergistic effect on knowledge generation that goes beyond a simple additive effect” (Kongsvik & Almklov, 2021, p. 139). In other words, we can learn things by mixing methods that could not have been learned by applying either method alone. The downside is that the process may seem “fuzzy” when several methods and schools of thought are mixed and matched. The researcher must take care to document every step in sufficient detail, so that the reader is able to follow and to evaluate the quality of research (Johannessen et al., 2016). I have used an analysis tool (Nvivo) to add notes and link files along the way and kept a work log so that I could check the chronological context of my notes while writing up the final report (this document).

3.4 Data Collection

3.4.1 Prestudy: Survey

The purpose of my pre-study survey was to find out if the agile project participants did perceive the organizational culture as innovation-friendly, and to see if there were any cultural factors that stood out. To this end, I wanted to use a cultural measurement that was relevant to our Norwegian work environment as well as the R&D context. I opted for questions related to the elements known from the EDI study by Aasen et al (Aasen & Amundsen, 2011) (see Section 2.4). As I would not rely on the results for statistical analysis, I saw no point in limiting the survey to pre-validated schemas. I also wanted to limit the number of questions/statements in the survey, so that participation would be a low-effort task for busy team members (see Section 3.6.2 for the inevitable discussion on reliability and validity).

The survey was designed with one composite variable per EDI trait, each with two indicators. The two indicators were statements to be graded on a (Likert) scale from 1 to 5, either two positive indicators or one positive and one negative (with negative indicators weighted reversely).

To reduce the risk of linguistic misunderstandings, I borrowed what I deemed to be fitting statements from previously published surveys (Forsgren et al., 2018; Isaksen et al., 1999; Patterson et al., 2005). As an example, the variable “Engagement” was operationalized as “The work atmosphere is filled with energy” from Isaksen’s Situational Outlook Questionnaire (Isaksen et al., 1999) and “People are prepared to make a special effort to do a good job” from Patterson’s Organizational Climate Survey (Patterson et al., 2005).

The survey was set up in nettskjema.no and had a response rate of 49%.

The full survey is presented in Appendix A.

3.4.2 Prestudy: Interview (One-to-One)

In parallel with running the survey, I did a small number (4) of semi-structured interviews with the different project roles (developer, tester, scrum master, project manager). The purpose of these interviews was to understand formal coordination and how the agile methodology was implemented in The Project. Key questions that I wanted answered were

- What are the most common modes of communication in The Project?
- How is information shared?

- How are decisions made?

Introduksjon til samfunnsvitenskapelig metode (Johannessen et al., 2016) advice that one should stop conducting interviews when no new information is found. As the four people being interviewed gave similar descriptions of the work processes, I decided to move on to focus group conversations.

(The interview guide is included in Appendix B).

3.4.3 Focus Group Conversations

My main source of information has been group interviews, one per Scrum team. The purpose of these interviews has been to get the team members to elaborate on how they see their culture, and to look for examples of how culture and methodology might influence each other. There were six teams, and all team members were invited to interviews. The interviews were conducted online and with a participation of 70%-100%. Each interview took between 65-110 minutes, depending on the size of the group.

Each session followed the same pattern: Introduction, presentation of the research questions, presentation of the pre-study survey results, discussion. I would usually invite the group to make some guesses before revealing the survey results. Looking at the results, I would ask the group what they found most surprising about them, and the discussion would usually center around two or three factors. I would also ask what was the *least* surprising to them, and if they thought the displayed cultural profile was a fair representation of the project culture - or if something should be adjusted. When asked if there was something to learn from the project, some teams were reluctant to engage, while others were happy to summarize strengths and weaknesses. At the end of the interview, some asked to have the survey results so they could continue the discussion, while others seemed relieved that it was over.

This approach was a slight deviation from my initial design: I had expected the pre-study to show some significant differences between the project culture and EDI culture. The original plan was to steer the focus group sessions towards those differences and ask "why". However, the pre-study did not reveal large gaps, - and so I invited the focus groups to reflect more openly on the survey results (asking "what").

One limitation to the "focus group" approach is that the interviews were conducted in a period where most people worked from home, so participants did not meet in the same room. This made team discussions more awkward, as people in noisy environments would have to mute-and-unmute to participate, and it was not always clear who was talking to whom. I think the online setting made attendance easier (people could join from their desks), but active participation became less engaging.

See Appendix C for interview guide.

3.4.4 Summary of Data Collection Methods

Data collection methods are summarized in Table 4.

Table 4 - Methods for data collection

Collection method	Who	How	What	Where	When	Why

Survey 18 state-ments	All team members. 37 invited, 18 respondents (49% response)	Online survey	"EDI culture" traits	Nettskjema.no	March /April 2021	«Measure» innovation culture (sensitizing concepts & scoping)
Interview 4x45 minutes	Project members (selection): • Project manager (1) • Scrum master (1) • Developer (1) • Tester (1)	Semi-structured one-to-one interviews (online)	Processes related to project coord.	Microsoft Teams	March /April 2021	Gather information on how project implements agile methods (sensitizing concepts & scoping)
Focus groups 6x90 minutes	All teams (70%-100% attendance)	Team meetings (online)	Present survey results and invite feedback	Microsoft Teams	May /June 2021	Gather information on (how team members perceive) culture and relationship with methodology (feedback on sensitizing concepts, preliminary analysis)

3.5 Methods for Analysis and Interpretation

For the pre-study, I exported data and code book from the online survey to a statistical analysis tool (SPSS), and after processing imported the results to a simple spreadsheet for visualization.

For interviews and focus group conversations, I transcribed all meeting recordings as text documents and imported the files to a qualitative data analysis tool.

The quantitative pre-study was purely theory-driven (deductive) – that is, survey questions were chosen based on known relationships with theoretical constructs (within organizational and creativity research), and my interpretation of the results is based on my understanding of that theory.

For the qualitative data, the process was empirically driven – I gathered the results first and then looked for theory that could explain my findings.

Stepwise-Deductive Induction follows a pattern with similarities to the brainstorming technique most often used in team retrospectives. First, we

a) collect statements or ideas on sticky notes,
b) read each note out loud, asking what the author *really* meant, and
c) group together stickies that are identical in meaning (this is surprisingly difficult. Is it the same, or is it slightly different? Are the nuances important?). Then we

d) group stickies addressing the same topic, and finally
e) take a step back and look for groups that "belong" in the sense that they should be discussed together.

This is what I did as a one-person exercise during the reading of transcripts (substituting sticky with interesting text fragment, and only occasionally reading aloud). I assigned a "code" to each group of identical statements. For instance, I created the code **"Proud of the product"** based on two text fragments "I think people are proud of the product, because it's a good product" and "If the question is focused on what we deliver, we are proud of it", but made a different code **"Not proud of the code base"** for "That's one of those things for old and large code bases, that it's hard to be proud of things" and "You don't necessarily have ownership of certain parts that you can be proud of". As one can imagine, this left me with a huge set of codes (240 codes, to be exact).

Looking for common topics, my two examples from before were lumped together under **"Product ownership".** After this round of coding, I still had 127 codes. Some of them seemed to belong to larger themes, while others were more "random".

Looking for larger themes, I frequently had to look up the original text associated with each code and ended up shuffling and re-shuffling the groups. Eventually, I had a list of top-level topics for analysis and discussion. From this list, I decided to focus mainly on the project (organizational) level topics and interfaces between project and (Scrum) teams. The final grouping is described in Section 4.1.

Methods for data analysis are summarized in Table 5.

Table 5 - Methods for data analysis

Analysis method	Data source	Tool
Statistical analysis	Data file from online survey	IBM SPSS (processing of composite variables, T-test) Microsoft Excel (generation of radar diagrams for visualization)
Stepwise-Deduction Induction	Transcripts of meetings (focus group conversations and individual interviews)	Microsoft Word (transcripts) NVivo (coding and categorization)

3.6 Evaluation of Design Choices

3.6.1 Ethical Considerations

There is an element of "action research" (Greenwood & Levin, 1998; Kongsvik & Almklov, 2021) in telling subjects about the research questions and preliminary observations and inviting them to give the answers. Even though participation has been voluntary, my work on this thesis has influenced the project by putting work methodology and culture

on the agenda. One might think that this is harmless – or even beneficial to the project - but there are some considerations to keep in mind:

People feel strongly about their work. Quite a few of my focus group participants have even left their home country to be in the job position they hold today. The nature of work as well as the social climate in the organization is of great importance to its members and must be treated with respect.

People do not have equal power. There is a power distance between organizational levels. Even within the teams, the Scrum Masters have more power than their peers. Focus group discussions on topics such as autonomy and trust between people with unequal power can be uncomfortable for some (in particular, for those without power).

Circumstances might change. In the current situation, it seems unlikely that anyone would be held against anything (s)he might be quoted on in this report. But people do sometimes get into conflicts of interest in the workplace, and things might read differently if taken out of context.

Taken together, these considerations have made it important for me to

a) Inform people beforehand about the study with a written invitation letter and make it clear that participation was voluntary,
b) avoid pressuring people into talking about things that might be emotionally disturbing, especially in the focus group setting,
c) provide anonymity, by using a secure 3rd party service for the survey; by using fictious names in transcripts; and by leaving out any names in the final report,
d) not referring sentiments or examples that could be interpreted as personal criticism (unless directly relevant to my discussion).

With these precautions, I hope that the results can be read as both truthful and relevant without throwing anyone under the bus.

The Project as such can be recognized by colleagues. It is important to me that it is presented in a way that feels authentic to the participants. I did consider sending out transcripts to focus groups for review but decided against it. People might start modifying their statements if they feel that some sort of approval is required. As Tjora reminds us, data gathering is not journalism, and as a rule one doesn't ask informants to comment interviews unless there is a specific reason to do so (Tjora, 2021, p. 192).

In my transcripts I have used made-up team names and person names. These have been removed from the final report, as it would be possible to attribute different statements to one person, and that person could be identified if the team was recognized.

All data are stored securely (one must assume) on NTNU servers according to NSD (Norwegian centre for Research Data) guidelines and will be deleted once the report is submitted.

The study is approved by NSD (Norwegian centre for Research Data). The letter of approval is attached in Appendix D.

3.6.2 Reliability and Validity

Acting as a "researcher" within your own organization is not without issues. Both you and your research subjects need to understand when you are acting as researcher, and when you are acting as a colleague. Predisposition may occlude facts so that the researcher draws (faulty) inferences during interviews, making results biased and unreliable (Johannessen et al., 2016, p. 232). I have tried to avoid these effects, first by choosing as my case a project with which I have had no previous engagement, also by following a fixed structure in interviews, and by transcribing each conversation verbatim.

As an alternative, possibly more "objective" approach to data collection, I could have relied more heavily on a standardized survey, used observation and/or document reviews. Although I have considered these options, the latter goes out the window due to the dubious nature of written documentation in this project, and observation just seemed too awkward. There will be a difference between what people say and what people do, but ultimately, I made the choice to rely on what my informants chose to share. It is the team members' subjective perception of reality that is of most interest to my research questions (even if it might occasionally be at odds with objective truth).

Method triangulation – that is, using different methods and settings for data collection– is another way to strengthen internal validity (ensuring that what we measure is what we mean to measure) (Johannessen et al., 2016, p. 230). In my case, the pre-study has limited value on its own: The quantitative survey is not based on a pre-validated questionnaire, and it cannot be validated based on the results because the population sample and number of survey respondents is insufficient for statistical analysis. Moreover, it is aiming to measure complex phenomena which would probably take more indicators per variable to give valid constructs. Likewise, a small handful of brief interviews does not provide a reliable nor complete source of information. However, when these data are explored and elaborated further by the focus groups, they become useful and provide context to the analysis.

3.6.3 Generality

Generality is a different matter. Even if we assume that the study by design is valid and reliable, how do we know that the results are transferable to other cases? The short answer is that we do not. We are dealing with "soft" data where each case is unique, and working stepwise from the specific towards the general, as we do with SDI, does not guarantee that what we end up with is applicable outside of our own case. This can only be tested by following the SDI method through to conceptualization, and then applying the concepts to other cases and tying it in with existing theory (Tjora, 2021, p. 271) – which is a bit of a stretch goal for a master's thesis. I can claim ignorance and put the responsibility on the reader to decide if s/he wants to believe that my results are transferable – but I prefer to claim "moderatum generalization" (Payne & Williams, 2005, p. 297) by saying that present-day agile software organizations of moderate size (+-50 people) in Nordic countries might find my results useful.

3.7 Summary

This chapter described my research design and how the strategy evolved from a predominantly deductive to a predominantly inductive approach. My reasons for choosing mixed methods were explained and evaluated, and so were my ethical considerations. My final thoughts about the research contribution from this study will follow in Section 6.2.

4 Results and Analysis

4.1 About This Chapter

Apart from a few explanations and observations from my side, the analysis is a condensed version of the project participants' own analysis of practices and culture (this is one of the benefits of working with smart people).

The structure of this chapter is based on recurring themes that emerged in focus group interviews. The contents are kept close to actual conversations.

In the final analysis, I ended up with seven code groups:

1. Project-level culture and fit with EDI culture
2. How practices influence culture, and what to do when culture and practice does not fit
3. How project-level bureaucracy conflicts with personal engagement and ownership
4. Integration of new team members
5. Team-level work and the boundaries of self-organization
6. Product scope and direction
7. Working remotely

Each of these topics could be a thesis on its own. Since I am primarily interested in discussing project-level cultural phenomena and the relationships between the project and team level, I have (with some regret) omitted code groups 5, 6 and 7 from the analysis.

Group 1 "Project-level culture and fit with EDI culture" (4.24.1) compares participants' perception of the project culture with the EDI cultural traits measured by the survey. The culture is described in terms like "friendly", "informal", "supportive," "open" and "sharing". People consider these traits to fit well with the EDI traits.

Group 2 "How practices influence culture" (4.3) is about how people relate to the agile practices (quite pragmatically). There are two sub-topics:

a) People emphasize the importance of openness and explain that this is enforced by the Scrum methodology, and

b) They discuss why collaboration orientation is difficult. "Collaboration orientation" was the most controversial topic in group discussions. Some would argue that "There is so much talking going on" and hint that the urge to reach consensus on any subject indicates an extreme level of collaboration orientation – others would take the opposite position and say that "collaboration has degraded" for the very same reason. People will talk a lot about fixing problems but not work together on "real" development, it was said. There appears to be little interaction with groups outside of the project organization, and cross-team collaboration happens mostly ad hoc when individuals from different teams try to fix immediate problems.

Group 3 "Project-level bureaucracy" (4.5) is another group of sub-topics: One topic is about following rules and procedures (4.5.1), and the other topic is about project

hierarchy (4.5.2). It must be said that this is quite a large group, because, for all the positive things to be said about the project culture, there are still some aspects of the project structure that people see as problematic. Coping with complexity by creating formal reporting lines and work procedures are perceived as necessary evils that work against autonomy and engagement. Long communication lines between customer and developer makes it hard for teams to know if they are responding well to customer needs. Feedback loops from teams to management are also scarce: Teams are busy and want to move on, and usually do not bother trying to communicate their ideas back to project management.

Group 4 "Integration of new team members" (4.4) turned out to be a key to understanding how The Project *really* works. Again, this is a collection of subgroups:

a) the onboarding process (4.4.1),

b) the role of the informal communications network (4.4.2), and

c) differences in how "new" and "old" project members perceive culture (4.4.3).

New team members agree that they are more likely to feel personally engaged than people who have been around for a while. This is not just because they are eager to prove their worth:

- New members are freely given a lot of information
- New members go through a learning process
- New members are given opportunities for collaboration.

Over time, these benefits and opportunities shrink. The boundaries of information sharing become visible; less time is spent learning new things; and the seasoned team member masters his work tasks without so much interaction with other people. Overall, seniors are a little less enthusiastic about project culture.

Group 5 "Team-level work and the boundaries of self-organization" is naturally a large group, since the actual work is happening inside of the teams. It deals with team composition, team-internal work practices, and the relationships between individuals and team. Teams are designed to hold the competencies required for SW development without depending on external entities, so they are cross-functional with a mix of people with verification, development, and architectural skills. Relevant to my discussion is that there is a feeling of high individual autonomy for most team members. Also relevant is that the lack of involvement in project-level planning puts a limit to each team's capability to self-organize. Apart from these topics, I have chosen not to explore single-team issues further in this thesis. The cultural impacts of agility at the single-team level have been thoroughly investigated – and found to be beneficial for autonomy, collaboration and engagement - many times over (see for instance (Biddle et al., 2018; Kakar, 2017; Küpper, 2016; Pikkarainen & Wang, 2011)).

Group 6 "Product scope and direction" partly touches upon Group 3 "project-level bureaucracy" as it deals with how the product is developed as well as with what is on the product roadmap. As many will argue, what we are able to build is determined by how we are organized, so this is a topic well worth exploring within the context of innovation management. There is indeed a bit of a sociotechnical revival going on with the "DevOps" (Forsgren et al., 2018) movement in software development, so the interested reader can

pick up a recent book such as "Team Topologies" (Skelton & Pais, 2019) to explore this subject further.

Group 7 "Working remotely" is the last group that I have intentionally left out. Some of the teams in this study are mostly co-located, others are mostly virtual. Everyone uses the same tools for communications and workflow management, so the Covid-induced move to home office did not cause dramatic changes in work processes. However, as a remote worker commented, virtual-only teams level the field with regards to being in-the-know: "so before, I would feel that a lot of things happened and I didn't know about it, but now that everyone is on Teams, I don't see that as much." The social impact is there: People from different teams no longer meet at work, and they do not accidentally bump into each other on Microsoft Teams. The long-term effects on project culture and possible fragmentation into sub-cultures is something to look forward to knowing more about in the future.

The remainder of this chapter goes into each of the code groups 1-4 in more detail.

Table 6 gives an introductory overview of all the practices that were mentioned across different focus group conversations, together with the groups' explanations of how each practice impacts their culture. This summary is my interpretation, based on a read-through of all transcripts.

Table 6 – Agile practices impacting culture

Level	Practice	Impact	Ref. section
Project	Scrum-of-scrums	Promotes openness on all levels.	4.3.1
	Top-down planning with PM and Scrum Masters only	Hinders engagement at team and individual levels. Hinders autonomy at team level.	4.5.2
	Splitting work into team-sized Epics (tasks)	Promotes autonomy at team level. Hinders collaboration orientation at project level.	4.3.2
	"Generalist" teams	Promotes development orientation at team/individual level. Hinders engagement at team level. Hinders pride at individual level.	4.4.1 4.5.1
	Joint Sprint Review	Promotes openness on all levels.	4.3.1
Team	Daily standups	Promotes openness at team/individual level. Promotes collaboration orientation at team/individual level.	4.3.1
	Retrospectives	Promotes openness and development orientation at team level.	4.3.1
	"Cross-functional" teams	Promotes autonomy at team level.	4.3.2
Individual	Individual ad hoc interactions	Promotes trust, openness and safety at individual level.	4.4.2
	Code reviews	Promotes openness at individual level.	-
	Pair programming	Promotes openness at individual level.	-

4.2 The Project Culture is an "EDI-type" Culture

> And then, equally important, is that we do meet outside of work. This further enhances the bonds we have, and the ability to work more effectively and more efficiently together. It is very important, the fact that we are in contact outside work, so it is my colleague, and it is my friend. Even though it is not necessary, you *can* keep a clean barrier – but if you go beyond that barrier, that is the best. And that is the culture.

I did not know from the outset if there was anything like a "project identity" – the culture could be fragmented into multiple subgroups with widely different views and opinions, in which case it would be meaningless to talk about THE Project culture. However, informants at different organizational levels told the same story: "People are competent and responsible," "People support each other rather than compete," and "We have a good tone between colleagues". Was this wishful thinking or reality? The survey results (Figure 5) pointed in the same direction.

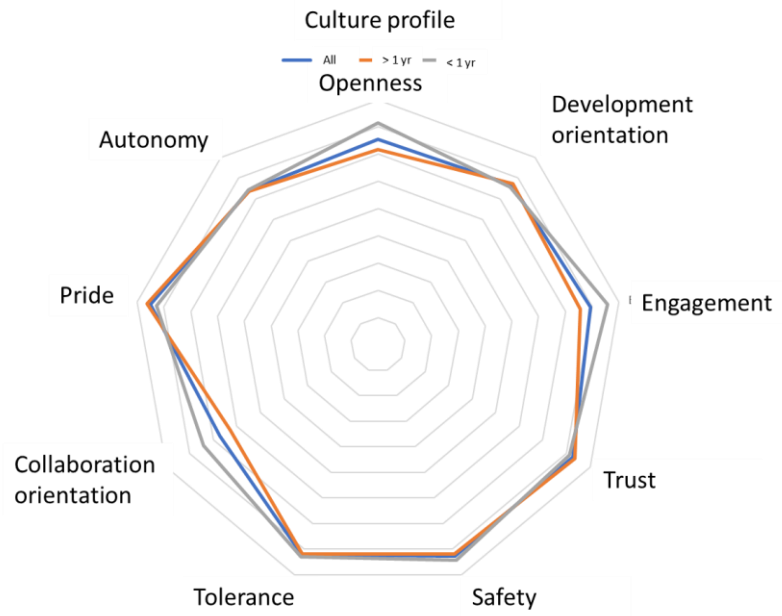


Figure 5 - Cultural profile (pre-study)

The radar diagram in Figure 5 was presented to the focus groups together with the numbers (we have an affinity for numbers) in Table 7. The diagram shows the average score for each factor (**blue** line) as well as the score for long-term (more than one year) team members (**red** line) and newer team members (**grey** line). The average score is close to 4 (out of 5) on all variables. To me, this indicates a kind of "fit" between project practices and an EDI-type culture. We do not know why certain variables stand out with a higher/lower average (it could have to do with how the questions were phrased and not with the culture per se), or why new and old team members seem to rate some things differently. (This latter point has been the subject of much interesting speculation in the focus groups, and will be returned to, notably in the analyses of onboarding practices (section 4.4) and project bureaucracy (4.5)).

Table 7 - Pre-study results, average score and standard deviation

	Open-ness	Dev. Orient.	Engage-ment	Trust	Safety	Tole-rance	Collab. Orient.	Pride	Auto-nomy

Avg/ StD	3.8/0.6	3.8/0.5	4.0/0.6	4.1/0.5	4.1/0.6	4.1/0.5	3.4/0.7	4.3/0.5	3.7/0.6
"Old"	3.6	3.9	3.8	4.2	4.1	4.1	3.1	4.3	3.7
"New"	4.1	3.8	4.3	4.1	4.2	4.1	3.7	4.1	3.7
T-test	p=0.05		P=0.06				p=0.09		

Before revealing these results to the focus groups, I would invite the group to guess how the project had scored. Some participants had not seen the survey questions and were naturally reluctant to make a guess, but after some debate all groups would settle for "we do well in most of those areas" or similar statements (the non-surprise when I did show the result was noticeable). There was no real disagreement about the overall shape of the profile, but there was also no consensus on which characteristic deserved the relative top and bottom score. Disagreement often centered around *Autonomy* (see Section 4.5.2), *Engagement* (4.5.1) and *Collaboration Orientation* (4.3.2).

The organization identifies itself as *open*, *supporting*, and *informal*, with a high level of horizontal and vertical trust. *Trust* and *safety* seem to be taken for granted, and several groups commented that the unanimous high scores on these variables were not surprising. Team members trust their peers to do their best. They trust the project manager to make the right decisions, and they feel that they are trusted to deliver when they take on work. Although people prefer not to bypass the hierarchy and talk to management directly, they do find managers to be open and listening if approached.

There is not much fear of consequences when admitting to mistakes: Wasting time and effort on something that doesn't work is recognized as a bad situation for those who are in it, and the willingness to help is high. At the same time, the project appears to be a meritocracy (the members who are perceived as most competent hold most influence), and as one member puts it, "I have a feeling that someone may feel that we have a good level of trust and mutual respect within the team, while others feel that we do not share that trust."

A new team member observed that "some of the people they just struggle with social skills." This was mentioned in passing, as something to be aware of in interactions, rather than viewed as a problem. It was acknowledged by the rest of the group that many team members crave stability and predictability in their working environment, and as such are not predisposed towards "agility". Only a few people confessed to thriving on changing social interactions as learning opportunities. There was an attitude of tolerance towards both dispositions: People's needs must be accommodated by the team setup, and "you have to be careful not to upset the balances" as the group explained it.

If many people prefer to be deeply embedded in their own teams, what keeps the project culture together? This was explained by *sharing* and the importance it holds in the organization. Sharing was referred to in multiple contexts. I made a list of all the different types of sharing I heard mentioned during interviews:

- Sharing information about the teamwork in reviews and Scrum-of-Scrums
- Sharing knowledge and how-tos in the project wiki
- Sharing work by dividing it up in parallelizable parts
- Sharing responsibility for the code base
- Sharing resources by allowing team members to "guest star" in other teams
- Sharing workload to reach deadlines

- Sharing interests outside of work.

The way I understand it, the sharing mechanisms give project members a chance to align across teams and informally exchange values and beliefs as well as competence and knowledge. These are voluntary behaviours: People choose to share because "this is how we do it around here".

I was told that cultural traits are reinforced by the recruitment process:

> The people who are doing the interviewing and hiring are hiring people similar to the people who are already here. So in that way it's sort of continuing the culture, I think. A big part of the hiring is whether someone fits in the team or not, not just their technical competence.

At the first reading of transcripts, I was surprised – and a little frustrated – by the fact that each focus group conversation would take off in a new direction, even though I was using the same slide set, following the same agenda, and basically saying the exact same thing in every meeting. I think this has to do with how the teams function: Each team has different things on their agenda. This does not necessarily imply that there are per-team subcultures. But it could be that, even if the cultural traits discussed here are common to The Project, there are other factors that are not so evenly shared. It may indicate that the discourse within each team is different and that there is little joint reflection on practice and culture across teams.

4.3 Practices Do Influence Culture!

The following sub-sections describe how the agile practices are perceived as affecting project culture.

4.3.1 Scrum Events Promote Openness

> We have practices that determine that at least to some degree we should have open communications and have collaboration and common priorities. After having worked with these practices for a while that's become part of our work culture.

The Project's attention to openness can be explained by the fact that openness is, in fact, a stated Scrum value. The teams attribute the high score on openness to the practices – in particular, to the daily Scrum and the weekly Scrum-of-Scrums (see Section 2.2).

The daily Scrum is *the* most important Scrum practice in The Project. This is what glues the team together: "Everyone will know what everyone else is doing". By following the same routine every day, the Dailies are thought to serve at least four functions (different teams bring up different things, but all agree that Dailies are important for openness):

- Disseminate knowledge within the team about the task that is being worked on
- Raise team awareness of potential problems and risks
- Provide the opportunity to ask for and offer help
- Agree on what to work on next.

Similarly, the Scrum-of-Scrums glue the different teams together: "Because my work is related to someone [in other teams], and that's why they want to help moving it forward." Most project members listen in to the Scrum-of-Scrums because, as they explain, having an overview of what other teams are working on makes it easier to offer help and to know when and whom to ask for assistance – it makes problem-solving more efficient.

The Sprint Review is another opportunity to check in on the others and hear what they are doing. The review usually takes the form of a short PowerPoint presentation of achievements from the last work iteration. The Scrum-Of-Scrums and Sprint Review are important for openness at the project level. Giving every team access to the same information enables "generalist" teams taking responsibility for the whole rather than "specialist" teams caring only for their own component.

Although the Scrum-of-Scrums and Sprint Review meetings enable ad hoc cooperation between teams and individuals by disseminating knowledge, there is not a lot of interaction between people in terms of idea generation or discussion in these forums. Close cooperation is contained within the team:

> You're kinda enforced by agile, you get quite tight in the teams, you know each other very well. But in some ways, that stops people from knowing – the other people within [the project], for instance, because you're working so closely with your own agile team.

4.3.2 Scrum Hinders Project-Level Collaboration

There is no tradition within the project to seek collaboration with other projects, and for most team members there is little interaction with the outside world. The project organization has been, and still is, a component provider for other teams within the company as well as for external customers. It is my impression that feedback and inputs from other parts of the company have mostly been pushed, from those teams to this project, and not actively sought.

With a new, common, software platform, more interactions are needed. The people who are working with the new platform find this situation difficult, especially since processes around the platform are evolving and "everything is moving all at once".

Within the project, if there is an opportunity to help another team in reaching a deadline or solving a problem, people like to swarm – that is, focus on the problem, offer solutions and "help out" across teams. Planned collaboration between two or more teams to solve larger problems is less common. The strategy is to avoid involving more teams if one team can do the job alone (within the time available), and so others get involved only if the responsible team calls for help.

> If I just ask you, - Hey, do you know about this, and the answer is No,- that's certainly not a collaboration. If people share some information, maybe - half-collaboration? But if we are really doing something together, like actually – pair programming, for example, that's probably collaboration.

Except for "listening in" on the Scrum-of-Scrums, joint planning and Sprint Review meetings, team members have no routine interactions with people from other teams (unless they are Scrum Masters): "I think for a lot of the other people, it's almost… restricted, in a way". Depending on what you are working on, you may happen to have a lot of ad hoc interactions with people outside of your own team, but at the team level there is rarely much collaboration.

Ideally, each team should have a sprint-sized work assignment (so-called Epic) that is parallelizable and just big enough for one Scrum Team to work on. Team members talk fondly about when they succeeded in "getting it right" so that everyone on the team worked together: "I really like it. For example, now that we have a common Epic and people are working on similar tasks, it's much easier to talk about it and share experiences, stuff like that." In practice, it is not so easy to make the Epic fit the team. A one-to-one interview centred on this: According to theory, the full team should work

together to analyse the Epic and break it down into tasks, but it is hard to involve everyone from the start. The informant observed that since the team does not know what to pick up next, team members not fully involved in the planning tend to look backwards and work on clean-up tasks, rather than looking ahead. Teams end up being spread out over a variety of tasks instead of working together towards one common goal.

The focus groups attribute the lack of cross-team collaboration to the methodology: Scrum makes a point of "shielding" the Scrum Team from the outside world and minimizing "interruptions". However, people also note that nothing in the methodology prevents teams from working together on larger tasks if they so choose: But this would require a level of planning that is missing today. "We do not plan for collaboration," one Scrum Master explains. There is also concern about the effectiveness of two teams working together since not all work is parallelizable – "there is a thin line between inefficiency and collaboration". Many enjoy the emergency-type swarming, but also argue that more planned cross-team collaboration could be more efficient than rushing-to-the-rescue mode of working together.

4.3.3 Agile Practices Can be Changed

> What I value is that if we see something that doesn't fit the way we work - if we spend enough effort in trying to change it, we can also change it. The processes can all be changed, so the mentality is not that "it was always like that, so we will keep it like that". We can change things.

Agile is seen as a "mindset" thing in The Project, with the practices as "common sense" ways of working to follow the mindset. Following the Scrum cycle is good because it provides a routine that "works efficiently". But if the practices are not efficient, they should be changed, people say: Inefficient practices will have a bad influence on the working environment, and they warn that over time the culture will deteriorate if teams are forced to work in ways that do not fit with the culture. But the teams are confident that "processes can all be changed," and that working on processes can create virtuous cycles improving the fit between process and culture.

4.4 How to Turn New (and Shiny) People Into Old (and Grumpy) People

These sub-sections explore different aspects of how The Project integrates new members, and how this affects those integrating as well as those being integrated.

4.4.1 Onboarding is Verbal

As the interviews took place in the time of pandemic, there was a concern about new recruitments: "We don't know these new people!" This seemed troubling because integration of new team members has been very much based on face-to-face interactions, sitting together, and working together. Traditionally, people say, when they were new, they were placed in a team and given a warm-up task together with some advice to listen and learn. Recently, as the project organization has been growing, the approach has become more structured: Each new member will have a mentor, and a training programme, with recommended reading materials and a fixed set of tasks to solve.

A frustrating (to the newcomer) aspect of the training tasks is that they are difficult to complete without help: "You must ask someone". People suspect that this is done deliberately to encourage collaboration and openness. For the newbie, actively seeking

domain knowledge and understanding the culture is only part of the challenge. It is equally important to get to know the "right" people – the experts - and to make himself known to the rest of the organization:

> I don't think it's easy for a newcomer just to figure out all by himself. You need to ask people. Also, I mean… documentation is not that – good, from my point of view. You need to ask people anyways, ask the experts. Yeah, we are not **forced**, but **almost** forced, I would say.

When conversations turned to onboarding, several groups started laughing at the thought of "fixing" the documentation.

> A: I think all these points can be linked to the new people having to be more collaborative in their efforts to find out how things work. Through that they are experiencing higher engagement, higher openness and more collaboration. Just through the fact that they have to contact other people to find out the answers they need. Whereas that's not something that people who've been here for a long time need to do, necessarily.
>
> B: which means that… if you fix your documentation… [laughter]
>
> C [laughing]: then you would have less collaboration!

It seems to me that because collaboration across teams happens ad hoc and based on informal communications, people need to know each other. And because people know each other, collaboration across teams happens ad hoc and based on informal communications. This aspect of onboarding takes precedence over other concerns, such as structuring project documentation or providing newbies with "easy" first tasks. Newbies themselves also argue that they need "real" first tasks to learn from.

4.4.2 The Informal Communication Backbone

> We are good at talking, but horrible at writing. Since our written documentation is not up to date, we are forced to talk to people, and because we are always talking to people, documentation is never updated. From a documentation point of view, it's a negative feedback loop. But from a communications point of view, it's a positive feedback loop, it increases our ability to communicate, but it decreases our ability to have static information readily available.

As mentioned above, reliable project documentation is limited. There are customer-facing product documents, and there are comments in the code and recorded video trainings – but the agile principles are deeply embedded in the ways of working. In effect, "Working software over comprehensive documentation" and "Individuals and interactions over processes and tools" work together so that newcomers are molded into the existing way of doing things. Asking around, and sometimes being sent in the wrong direction, is just the way it is – "when you are a new employee, you go around and poke people until you have a network".

Over time, team members build a mental map of the network and know the key people in each area of expertise. People explain that as they become more self-sustained, they do not have so much interaction with others in their daily work - *unless* there is a problem. The initial fumbling around has taken down the threshold for when to contact experts outside of the team, because now that you are a full member you know who to talk to. It makes it easy to alert the right person of a potential issue without going through formal channels. The steady trickle of information between the "know-it-all's" and more peripheral parts of the network enables the project to react quickly if a crisis occurs. It reduces the chance of one team trailing off the common path.

> A: I think the advantage of this is that you kind of break down the initial barrier of contacting people, you get familiar with a lot of different people and then you are less

conscious about contacting people when you need to. Like, if you never contacted anyone and you always found everything in the documentation, then you might feel quite a social barrier to reaching out to someone when you need to.

B: Yeah. The more people you talk to, I think the more efficient you become at figuring out who is the most helpful for certain types of questions and things like that.

4.4.3 Differences Between "New" and "Old" People

New guy: I think it's the same as we talked about previously, we need to be open, to learn, and probably the more senior people they already know.. yeah, they know what they need to do their jobs, so they might score a bit less on this. They already **know**, or at least they know where to go to find what they need. We need to be open in order to.. [laughs]

Monika: [also laughing] …in order to survive?

Old guy: In order to become like the old guys!

New guy: Yeah! That's the goal!

New team members agree that they are more likely to feel personally engaged than people who have been around for a while. There are two lines of explanation for this; one related to project history and one related to work situation.

The **"project history"** explanation goes like this:

Things were better before. In a successful project, there will always be some "regression to the mean" – people will remember a time when it was even better than it is today. The pioneers who have been with the project since it was just one team mentioned this.

People will also remember the things that weren't so great. Even the things that they don't want to talk about. When you know too much, "openness" becomes more challenging. This was brought up by team members coming in from another project, wondering what they possibly didn't know about.

Before the new training programme was in place, there were less opportunities for new people to collaborate. You might have been working in the organization for a long time without feeling fully integrated, as a medium-term member commented.

New people are recruited from other companies with other cultures, possibly "less open and engaging", and they think their new employer is just GREAT! "If you're joining a company and you feel worse about it than the people who are there, you are coming to the wrong place" was the laconic remark.

Then there is the **"work situation"** story:

During your first year of working on the project, you are in a continuous learning situation, everyone wants to get to know you, you are encouraged to collaborate, and you feel you can talk about anything. This came from new members explaining that they have objective reason to be enthusiastic.

When you get deeply embedded in your work, it is easy to lose track of the overall direction of the project. "It's like slowly boiling the frog alive, and he doesn't really notice what is going on."

As an experienced team member, you don't need that much interaction with others to get the job done. Unless people seek you out for help, you lose touch with the information network.

If you become one of the experts that are always asked for help, you lose out on teamwork. Interruptions make it hard to stay involved. "I feel like collaboration has degraded."

The project's environment is changing more frequently than it used to do. This generates uncertainty and frustration among old-timers. "Everything changes all the time! There is no stability!"

Taken together, we agreed in one group session that the "project history" and "work situation" stories would suggest that it might pay off to create more learning opportunities and get-togethers on a regular basis for everyone involved in the project. Some also suggested that encouraging people to "guest star" in other teams more often could help keeping up the project spirit (while others visibly shivered at the thought of leaving their team to "guest star" with someone else).

> If you try to mix people around, and maybe try to change the group more often, you could try to encourage this collaboration a lot longer. You know, keep changing groups, something like that.

During these discussions, one person brought up that there is more to being agile at work than just work processes and team culture, however great those aspects may be. The component of *personal motivation* plays perhaps the biggest role. People are in fact not dropped into a project as blank sheets ready to be imprinted with project values and practices; on the contrary, people bring with them unique personal experiences and expectations ("some might have experienced companies less open and engaging than Nordic"). I think that the question of how personal experience flavours people's perceptions of the working environment is an interesting (and within Scrum literature, under-communicated?) topic.

4.5 Project-Level Bureaucracy

In the next subsections, people talk about the downsides of being part of a larger organization with common tools and procedures and several layers of coordination.

4.5.1 Procedures for Everything

"They are the victims of their own success," one brand-new project member observed. As the project has grown and matured, much of the workload has shifted from new development to maintenance. To cope with all the different types of work tasks, processes have been streamlined. There are procedures and automation for every (?) kind of work that can be routinized. The teams are not sure whether this is a good or a bad thing: You want to have common "traffic rules" and standardized ways of doing things, and people are quite happy to be "shielded" from management and other interruptions, so this is good. And yet, there is a feeling that the project has "solidified," and that new team members are inheriting ways of working which they did not choose for themselves and do not always understand. New team members struggle with the volume of procedures:

> When the company is so big, you need to have procedures in place so that everyone doesn't do crazy things. But then, at the same time, it sort of collides with the fact that you want to be productive and you want to do things, and sometimes you feel like I just wanna do this, and I can't do it because it's like the System doesn't trust me.

These "procedures for everything" are silently accepted because they do enable people to do their job without knowing how "everything" works. Without having the full domain

knowledge, you are safe not to ruin the work of others if you follow procedure. One Scrum Master reminds the team that in their previous project

> Although we had rules, they were rules that we came up with ourselves, and we knew when we could break them and we could be quite flexible about it. Which is OK when you have a very small team, but it wouldn't work so well in [The Project] where there's just too many people and it's too risky.

4.5.2 Hierarchy Undermines Commitment

The Project is scaling by having a set of formal coordination mechanisms "above" the Scrum practices. Formal reporting lines between project management and Scrum Masters limit the communication interfaces for most team members (see Figure 3). Scrum Masters will usually speak on behalf of their teams, in Scrum-of-Scrums, planning meetings, and Sprint Reviews. "Ordinary" team members do not have a "formal" voice. Team members perceive these mechanisms to sometimes be at odds with the core Scrum principles, and "because of the Scrum Master having so many responsibilities, no one wanted to be a Scrum Master". This makes it easy for other team members to "disengage" and leave the talking to the Scrum Master.

Within a Scrum team, members can self-assign to the work tasks they want to take on. "It's not like I'm given the easiest task because I'm new, I can actually choose and try to find interesting tasks myself." In contrast, it is usually the Project Manager that assigns work to the teams in meetings with the Scrum Masters: "We are autonomous in the sense that we are free to organize our work as we please, but we are not autonomous to decide on what to do work on next. That is mandated."

Early in a group session, one person observed that "I am considering my own personal autonomy to be high, but I am surprised that so many people would score team autonomy high." This kicked off a discussion on ambiguity in level of autonomy. People seemed to agree that this explains why it is sometimes hard to make teams fully accountable for the work they take on. The same topic came up in individual interviews: If teams are not empowered to self-organize, it was argued, they will not feel obliged by deadlines agreed between Scrum Master and Project Manager, and they will not see it as their responsibility to align their work activities with other teams. Such alignment is up to the entity that assigned the work to them in the first place and presumably "knows best".

A side-effect of this planning process is that when the Scrum teams sit down to analyse their new work assignment, they might find that the work will be hard to fit into the timeframe. When this happens, they will usually go ahead and try to get it done anyway. One Scrum Master explained that

> "We don't actually have "committing" (sic) to the goal, because that creates fear of not delivering, and rush to deliver, and compromise of quality. We try to not commit. The team does not commit to anything. The team just say "we'll try to do it"."

A common observation in the focus groups was that *there are no feedback loops* in this organization. Information and priorities are communicated from the TPM via the Project Manager to the Scrum Masters, but there is no arena for team members to push information (such as unrealistic deadlines) the other way. Long communication pathways - in terms of the time it takes to get a message through, as well as in terms of number of hops – make people reluctant to engage in issues that require vertical communication.

> As a team, [The Project] is spinning a lot of plates. […] the team has taken on a little bit too much, and it's sort of struggling under the weight of the things it's got to do and achieve. Perhaps a little more focus would help. It's hard to know exactly what I think

> should be different, but I just feel that there's a lot of different things going on. It's not always clear.. why.

Uniting by a common goal can sometimes be difficult within a team. It is even more challenging between teams. In case of emergency, such as a release deadline or critical bug, priorities are clear. But when there is "business as usual" there will be plenty of things going on in parallel, and nothing to join forces on. Instead, each team will work to complete their own task. But they do not always know why they have been assigned that particular task, how important it is relatively to other teams' work, or what problem it will solve for the customer. This can create frustration:

> As for what we do, I think that is pretty clear, it sort of trickles down from project management to the agile teams, - but as for why we do them, that is not always so clear to me at least.
>
> So you are kind of told what you are supposed to do, but if you are not told why you are supposed to do it, then you don't really have a chance to find alternatives that might be better.

There is also no way for the teams to request feedback on their work. This can be very disconcerting for a "true" Scrum team that delivers software iteratively and relies on customer feedback for adjusting plans and gaining understanding on their way to final product. Instead, having to rely on their own best guess, it is my impression that teams may tend to make complete up-front plans based on whatever information they have, and not really knowing if their design solves the customer's problem. The distance between team and customer also makes it easy to hide behind the Scrum Master, one developer comments, and claim ignorance if the delivered feature for some reason should not fulfil expectations.

The lack of direct customer involvement is used by some teams as an explanation for why "Engagement" does not score higher: People do not know the customers well enough to share their successes or their failures. This leads to a feeling of detachment for some, especially for those who are used to working more closely with customer representatives.

4.6 Summary

The most interesting part of the analysis for me has been to see how The Project is applying agile principles ("Individuals and conversations" etc) to cross-team coordination. There is no practice for this described by "the book" (aka the Scrum Guide), so The Project has developed its own onboarding practices to teach new members how to keep up the conversations. This helps to preserve the common project culture as well as learning new members project-specific skills and ways of working.

It was surprising to me to see that team members identify so closely with all the cultural traits that I chose for the pre-study. It would seem like The Project is more than ready to engage in employee-driven innovation activities.

Less surprising, we saw that organizational hierarchy and project-wide routines were perceived to have negative effects such as reduced commitment, low team autonomy, loss of engagement and feelings of lack of focus and purpose.

The next chapter will discuss how these findings answer my research questions and how the answers can be related to theoretical concepts.

5 Discussion

5.1 About This Chapter

This chapter returns to the research questions, "How can cultural factors affect agile practices in multi-team projects?" and "How can agile practices affect cultural factors in multi-team projects?". By splitting the discussion into these two questions, I hope to clarify that, although there are mutual relationships, "culture" and "practice" work at different levels. Some cultural factors are more fundamental to agility than others; these are not necessarily the same factors as those which are most easily influenced by agile practices.

Starting with **how cultural factors can affect practices**, I discuss the significance of trust. For a high-trust organization such as The Project, it could be that other methodologies would do equally well, if common beliefs and behaviours could be preserved. The culture will make the methodology work.

With this disclaimer, we can go on to discuss what The Project is doing to make its methodology work. I use Social Network theory for this. My main point is that The Project culture is supporting the Scrum practices by following behavioural norms that uphold agile values.

Moving on to **how practices can affect culture,** I claim that openness is reinforced by the Scrum events *if* trust and safety are already present. I explain how the Scrum methodology's lack of focus on collaboration orientation beyond the team level impacts organizational learning and innovation. I turn to studies of Communities-of-Practice for this part of the discussion. Finally, I discuss why it is that coordination of multiple Scrum teams can have a bad effect on autonomy and angagement.

5.2 How Can Cultural Factors Affect Agile Practices in Multi-Team Projects?

From the analysis it looks like the culture is a deciding factor for how practices are followed. This is also as expected from the theoretical discussion on "Agile" culture (2.3). Referring to how The Project members talk about their culture as informal, friendly and consensus-seeking (4.2), it makes sense that the risk-taking and experimental aspects of the practices (2.4) do not receive so much attention.

5.2.1 The Role of Formal Coordination in High-Trust Cultures

There is close to unanimous agreement among team members that trust, safety, openness and tolerance is high (4.2).

A high-trust culture does not need much formal control because its members operate by shared values and common norms of behaviour. Being loyal to their peers, people will try as well as they can to avoid breaking the social rules. We can assume that The Project's formal methodology is adhered to insofar as people subscribe to its values, and that the practices are aligned with social norms. The Scrum values (2.3) fit well with the project culture (4.1), and so The Project **makes Scrum work** by tweaking and adding

procedures to make it fit the multi-team context – even in circumstances where it is not ideal (4.3.3). It is however tempting to suggest that in this case, *any* methodology would do if it had a reasonable fit with the cultural values of the organization: Because trust, safety, tolerance, and openness are so high, people will adapt, make the best of the situation, and assume that practices will improve over time.

We should also bear in mind that The Project has grown organically from a single team, and the Scrum methodology was chosen from the outset by the original team members. Mandated ways of working that could otherwise be interpreted as controlling may be tolerated, because they have a long tradition, and because people believe there are good intentions behind them. It could be that any kind of methodology, built on a reasonably similar set of values, would work equally well in this situation – simply because formal coordination does not hold that much importance.

5.2.2 Culturally Conditioned Behaviours That Make Scrum Work

There must be a two-way relationship between the agile values and cultural norms and behaviours: The values must be relevant to the culture, and the social norms should make it easy to behave in accordance with those values. Since "everyone" agrees that trust, safety, openness, and tolerance are high, there must be something that people do in The Project to build relations that can uphold these traits.

Trusting relationships are built through giving and taking responsibility (Aasen & Amundsen, 2011). In The Project, we see that the daily verbal interactions create opportunities for team members to display trustworthiness (4.3.1). This is embedded in the Daily Scrum event and helps consolidate the team over time as people learn to trust their peers. But this trust does not extend beyond the Scrum team, so something else is needed to build a project-wide culture that can support agility. We saw in Section 4.4.1 that The Project is solving this by deliberately making people – and in particular, new team members - look for help outside of their own team. By doing this, junior members learn that project members outside of their team can be trusted to support them. This trust feeds back into psychological safety (2.5) for newcomers when interacting with seniors. People learn that it is safe to reach out to someone in another team, and that they should not be afraid to speak openly (4.4.2).

In other words, there are cultural prerequisites for agility, and building relational social capital (2.5.1) is a way to support these prerequisites.

If we look closer at The Project from the social network perspective, what can be said about how the network affects practices? We can see that The Project is a network-of-networks, with Scrum teams as clusters and a few key people (notably Scrum Masters) acting as brokers. Architects and other experts are bridges with weak ties across the clusters. Within the company, The Project itself is a closed network, only loosely connected to the larger company network (as far as I can tell from how The Project explains relations with the outside world, section 4.3.2).

Schiefloe explains the benefits of high-density network organizations by how highly connected people draw on their network for advice and use the network to exert influence; how information travels; and how critical situations are resolved in dense networks (Schiefloe, 2015). This is very similar to how The Project members explain the role of informal communications between seniors and juniors, and how the network enables them to react quickly to emergencies (4.4.2). It pays off for The Project to

encourage individuals to reach out across teams, because this makes The Project more efficient.

If we look even more closely at the relational dimension social capital, we see that the actual ROI will depend on the cultural norms in the organization. In a network of little moral obligation, there can be lots of interactions without much return on investment. In a case like The Project, however, where trust is high and the network is "closed", people will feel committed by their relations. Keeping up the social capital is not just a necessity to keep the organization together, but it gives a high return on investment by ensuring that everyone behaves in accordance with expectations.

In addition to "forcing" people to talk to each other and build relations across teams, another "hack" used by The Project to promote networking is to recruit similar people to the people who are already there (4.2). According to Kilduff and Brass, this makes network-building easier:

> "Although we expect that strong, direct relationships will have more effect than distant, weak relationships, we also know people seek out others who have attitudes similar to their own. Similarity breeds interaction and interaction is the medium of influence and increased similarity as individuals with similar attitudes reinforce each other and become even more similar." (Kilduff & Brass, 2010)

Assuming that people really do prefer to interact with others similar to themselves, hiring similar people is a cheap way of building social capital. It could, however, backfire, if the "similar people" prefer to only interact with each other, and in effect create a "closed" network of likeminded peers. Krackhardt and Kilduff warn about potential consequences for individuals working within closed networks:

> "Individuals within closed networks are likely to experience lowered autonomy, less variety, and redundant feedback (cf. work on Simmelian ties: Krackhardt &Kilduff, 2002); but such effects may be outweighed by interpersonal trust and support." (Kilduff & Brass, 2010).

Kilduff and Brass explain that "less variety" is more likely in a closed network because you do not get new input on interesting work outside of your network, and easily fall into the trap of doing the same thing over and over. And if you always ask the same people, you tend to get the same answers ("redundant feedback"). "Lowered autonomy" refers to the works of Coleman, saying that individual autonomy goes down in a closely knit group because everyone within the group can monitor everyone else, and so people will feel forced to follow the common norms of behaviour whether they really want to or not (Coleman, 1994).

Scrum teams in general have been identified as dense networks in literature (Shafiq et al., 2019). There is some evidence that Scrum team members are more productive when the network is closed (Ehrlich & Cataldo, 2012); but there is also an increased risk of quality issues and rework when connections to the outside are few (Ehrlich & Cataldo, 2012). The analysis does not show whether Scrum teams in The Project suffer from productivity issues or quality issues. I will, however, argue that Coleman's assumption of lowered individual autonomy does not hold true in our case. To the contrary, The Project team members talk about high individual autonomy as a source of engagement (4.5.2) (levels of autonomy are discussed further in Section 5.3.3). This is also as one would expect from theories saying that trusting relations are based on choice, and that social capital goes together with autonomy (Allik & Realo, 2004).

We see that internal network building has itself become a cultural norm in The Project. It makes the Scrum methodology work in the multi-team setting by creating bridging social capital across teams.

5.3 How Can Agile Practices Affect Cultural Factors in Multi-Team Projects?

Although culture seems to determine which methodology is suitable and how to put it into practice, it is also appears that work practices will have an influence on culture. Team members point to several ways in which the agile practices affect their culture (Table 6 – Agile practices impacting culture). The most obvious example is how Scrum events promote openness (4.3.1). From the analysis and previous discussion, one might speculate that practices can be more effective in reinforcing existing culture than used as a tool to change culture.

5.3.1 Scrum Events and Openness

When The Project members talk about openness, they mainly talk about sharing information. Making a habit of sharing information in the Daily is clearly beneficial to the teams and to the project. Talking freely "in public" about what is happening in the project, both in Scrum-of-Scrums and Sprint Reviews, is maybe stretching the listeners' patience, but "all this talking" turns out to be very important for spreading odd bits of information. There are a number of reasons why this aspect of openness – making information freely available – is so important in multi-team work:

Firstly, there is the mutual relationship with trust and safety (section 2.5): I tell you everything because I trust you, and you trust me because I do not hide information from you. If I do not have to guard my words or second-guess what you are not telling me, that mental capacity can be used for something else (more productive) (Edmondson, 1999).

Secondly, at the organizational level, the effects of subgroups *not* sharing information with each other (not to mention management keeping information from employees) would quickly multiply as each group engage in guesswork to fill in the informational blanks ("sensemaking": put us in an uncertain situation, and we will try to make sense of it with the knowledge we have (Weick, 1995)).

Thirdly, there is the "efficiency vs effectiveness" aspect: From an efficiency point of view, it might better to give others only the information they need to perform their task. From an effectiveness point of view, since how people perform their task will be based on what they know, letting them know more reduces the risk of them doing the wrong thing (ref. risks of "star"-type networks and "closed" Scrum teams, section 5.2.2). This is a well-known issue in the software industry: Eager developers can be very productive and make products that no one needs or wants to use - in a highly efficient manner (Perri, 2018).

A fourth point is related to the third: Spreading information on more people ("redundancy of information") is a necessity for effective self-organization. People need overlap in what they know, to figure out how to organize. In a cross-functional team, this means that openness also entails functional specialists freely sharing their competence. If people guard their knowledge, teamwork efficiency will be limited by each specialist's work capacity. (Morgan, 2006) (One example that comes to mind is verification specialists and developers refusing to share tasks on a Scrum team).

In sum, the "information sharing" aspect of openness is a prerequisite for creativity and innovation. Coming up with new and useful ideas and products means using domain knowledge to put different pieces of information together in new ways. The employee who is driving innovation or creating new solutions really does need access to odd pieces of information (Oddane, 2017). Openness as a default behaviour is important for a free flow of communication in any kind of project work. For The Project, it is perhaps most important in the way that it supports trust and safety beyond team level. Speaking openly about problems and failures in Sprint Reviews and Scrum-of-Scrums demonstrates that the speaker lives by the shared values and that he can be trusted.

Openness as in, "being open to new ideas," is not so much visible at the project level. At the team level, new ideas are discussed in Sprint Retrospectives (2.2). Retrospectives are forums for self-evaluation, brainstorming and making suggestions for improvements within the teams. But since there are no project-wide retrospectives or formal feedback mechanism from the team retrospectives, new ideas often go unnoticed by management. This is unfortunate, because one thing we do know from change management research is that if people do not receive feedback on improvement ideas, they will eventually stop generating improvement ideas (Amundsen & Kongsvik, 2016; Kotter et al., 2006; Meyer & Stensaker, 2011).

5.3.2 Teamwork and Collaboration

It should be clear by now that Scrum is a methodology for teamwork. The practices encourage behaviours that build "bonding" social capital and tie the team together in a dense cluster. This ensures efficient teamwork. But as we have learnt from social network theory (2.5.1 and 5.2.2), dense clusters can have problems related to effectiveness. Network theory even told us that clustered organizations are outperformed by high-density organizations. This can be because the cluster has insufficient interaction with the outside world to know if it is doing the right thing, as The Project team members explain in Section 4.5.2. It can also be because it is difficult for a dense cluster to collaborate with other dense clusters.

As we have seen in the analysis (Section 4.3.2), project members were concerned about the potential inefficiency of collaboration. Why is efficient collaboration difficult? The cognitive dimension of social capital is important: Without some accumulated shared knowledge, it is hard to act on any piece of information. We must have a mutual agreement on what something *means* before we can act. This explains a lot of the "sharing" behaviours in The Project: These behaviours are a way of aligning how information is understood and related to work practices (4.3.1) so that collaboration is possible.

To further understand how information is turned into shared knowledge in The Project, we may think of The Project as a Community-of-Practice (2.5.2).

When The Project onboards new members, it is initiating them into the community of project-specific firmware development practice. The new people are learning it by doing it together with their mentors and teammates. As people come out of their training period, they continue to work and learn together, but now predominantly within their Scrum team. The team becomes their mini-community.

It is interesting to see that Project members have been echoing statements made by Wenger et al:

> "In their book, Cultivating Communities of Practice, Wenger et al. (2002, p. 141) devote a chapter to what they refer to as the 'downside' of communities of practice arguing that the 'very qualities that make a community an ideal structure for learning – a shared perspectives on a domain, trust, a communal identity, longstanding relationships, an established practice – are the same qualities that can hold it hostage to its history and its achievements'." (Roberts, 2006, p. 628)

In other words, having a well-established CoP can make change difficult because the practice will be so deeply embedded in the culture. It *is* "restricted, in a way" (4.3.2). Thinking back to what Wenger said about CoP boundaries and boundary objects (2.5.2), we see that reaching out across teams and interacting with brokers is important because it creates shared knowledge between the Scrum team mini-CoPs. The common processes and tools (4.5.1) that everyone must adhere to – whether they understand them or not – have a function as shared artefacts that give people a common language and a basis for interaction.

So far, we have discussed how The Project culture and The Project practices promote accumulation of social capital (sections 5.2.2 and 5.3.1). It is perhaps not so clear how to cash in on the investment. We have learnt that weak connections create a potential for learning and growth in an organization by giving people access to information outside of their own domain (Section 2.5.1). We have seen that The Project is trying to create weak connections at least for new employees (Section 4.4.1). But it also seems like having some vague connection to someone working across the organization does not make much of a difference when there is no work-related interaction[10] (4.3.2). Scrum does not make it easy to have these work-related interactions across the organization; to the contrary, as we have seen, the methodology is specifically designed to avoid such interactions and maximize efficiency within small, dense clusters by embedding people in self-contained Scrum teams (2.2 and 4.3.2). The cost of following Scrum, then, is that the effort put into making new people part of the larger social network does not pay off in organizational learning and innovation because the fully integrated team members do not use the network to this end (4.4.3). At the individual level, people will try to learn by doing things *within* the Scrum team and not involve others unless there is a problem they cannot solve on their own. At the project level, management will do their best to promote team autonomy by splitting the work so that each team can work independently (4.3.2). As we will see in the next section (5.3.3), external dependencies are not good for team autonomy, but ironically, removing dependencies and making every team an island also limits the opportunities for several teams to work together. With this way of working, it is hard to keep up the basic assumption of "collaboration orientation" – that is, the assumption that the best solutions are invented as collective efforts.

If we accept that The Project is a CoP, it seems like it would be a good idea to encourage more project-level CoP activities. One might question if trying to change a CoP from the management side is a smart thing to do, or if it requires a bottom-up initiative to be effective. Considering the concept of "legitimate participation," it is not really within management's power to decide who is in and who is out of a CoP. A manager in need of influencing the negotiation of meaning within a CoP will have a hard time doing so unless (s)he is a CoP member.

[10] In much the same way as creative ideas do not bounce between departments just because they are on neighboring floors. FYI for colocation enthusiasts.

However, in the case of The Project, given that it is a harmonious and trusting environment, it might be worth trying to encourage more boundary-spanning CoP activities as an experiment to neutralize the negative effects of Scrum team-centrism.

So what are the consequences for The Project of not collaborating? "Collaboration orientation" holds several aspects: Individuals seeking ways of working together; teams and subunits trying to organize their work so that everyone is contributing towards the same goal; and different organizational levels discussing strategy and process. Whichever way we define it; from an innovation perspective, the assumption that collaboration is good is rather fundamental. As a rule, successful innovations are not one-man shows: even the simplest idea usually takes a team of people to follow through from ideation to realization (Aasen & Amundsen, 2011, p. 141). More complex innovations require more diverse skill sets, problem-solving styles and personality types driving different parts of the process. Not every organization will have equal need for innovation, but if people do not believe that they can come up with qualitatively better solutions together than individually, the organization is likely to lose out on improvement opportunities. There is no reason to believe that The Project has reached the end of all the things it has to build, so it will have to pay attention to the downsides of not collaborating.

5.3.3 Coordination vs Autonomy

In the previous section, we discussed effectiveness in clustered networks and mentioned that interaction with the outside world – or lack thereof – can be a problem. Now we can return this and other consequences of trying to coordinate Scrum teams in a hierarchical fashion (4.5.2).

Working in an environment where you do not know when it is OK to break the rules is not ideal from an autonomy point of view. Following a procedure without knowing exactly why it is there makes it hard to feel ownership for the work you are doing, or (in the words of Karl Marx) to alienation. People talked about the need for coordination between teams as "something to be aware of" that can negatively impact team autonomy and engagement in project-level activities. It was alluded to in several settings:

- The lack of forums for exchanging feedback and ideas across the project limits the opportunities to get engaged even if one wants to.
- The project management practices regarding task assignments makes it easy to sit back and let others decide what to do.
- Pre-defined work procedures that every team member must follow makes it hard to take ownership of the work.
- Being "generalist" teams without a specific purpose is less committing than having a goal and a specific piece of code to take care of.

Autonomy in agile teams, and the paradox of autonomy vs coordination in multi-team software projects, have been thoroughly researched in recent years (Dahl et al., 2018; Gundelsby, 2018; Moe et al., 2019; Stray et al., 2018).

Moe et al. (Moe et al., 2008) discuss autonomy at the individual (personal), internal (team) and external (project) level:

Individual autonomy refers to how much discretion a person has in deciding how to do his/her job. At the individual level, autonomy is associated with intrinsic motivation[11] (interest in the task itself, rather than its rewards) (Deci & Ryan, 1985), which again triggers creativity (Amabile, 1997), learning (Deci & Ryan, 2000), willingness to take task ownership (Mayhew et al., 2007), and a feeling of doing meaningful work (Albrecht et al., 2021). Meaningful work is seen as an important mediator for Engagement: In a recent study by Albrecht et al, meaningful work was shown to be more strongly associated with employee engagement than either job variety, development opportunities or autonomy itself. Employee engagement has been associated with outcomes such as improved health, increased performance and company financial outcomes (Albrecht et al., 2021).

According to Moe et al, high individual autonomy can be problematic for teamwork efficiency, if team members simply choose to go their independent ways and fail to interact (Moe et al., 2008) (This follows Colemans perspective on individual autonomy, ref. section 5.3.2).

In The Project, we know from the analysis that people claim to have high individual autonomy. Individuals feel free to self-organize and pick their tasks (within the team and the sprint) and work relatively independently. At the same time, they are not empowered to choose tools and procedures. Looking to Deci & Ryan (Deci & Ryan, 1985), we can assume that choosing what to work on will be motivating, while following step-by-step instructions on how to do it could dampen engagement.

Internal autonomy refers to how the team operates internally with regards to joint decision-making and work coordination. A high internal autonomy implies that the team chooses its own path as a team and has the freedom to delegate decision-making authority within the team (Moe et al., 2008), - which means it could conflict with individual autonomy.

An example of a team with high internal autonomy, borrowed from Oddane (Oddane, 2017), is the jazz orchestra: As a group, the band can improvise a tune that no team member has thought out beforehand. But it can improvise successfully only if the team members a) know how to play their instruments, b) have some common musical

[11] Intrinsic vs extrinsic motivation, very long footnote but also very informative: As explained by Ahmed, P. K. (1998). Culture and climate for innovation. European journal of innovation management.

Rewarding individuals for their contribution to the organisation is widely used by corporations. However, while recognition can take many forms there is a common distinction: rewards can be either extrinsic or intrinsic. Extrinsic rewards are things such as pay increases, bonuses and shares and stock options. Intrinsic rewards are those that are based on internal feelings of accomplishment by the recipient. For example, being personally thanked by the CEO, or being recognised by the peer group, being awarded an award or trophy.
Innovative companies appear to rely heavily on personalised intrinsic awards, both for individuals as well as groups. Less innovative companies tend to place almost exclusive emphasis on extrinsic awards. It appears that when individuals are motivated more by intrinsic desires than extrinsic desires then there is greater creative thought and action. Nevertheless, it has to be stated that extrinsic rewards have to be present at a base level in order to ensure that individuals are at least comfortable with their salary. Beyond the base salary thresholds it appears that innovation is primarily driven by self-esteem level rather than external monetary rewards. It appears that extrinsic rewards often yield only temporary compliance. Extrinsic rewards promote competitive behaviours which disrupt workplace relationships, inhibit openness and learning, discourage risk-taking, and can effectively undermine interest in work itself. When extrinsic rewards are used, individuals tend to channel their energies in trying to get the extrinsic reward rather than unleash their creative potential.

"patterns" – and c) know each other well enough that they can take turns in leading and following.

It appears that many of the things The Project does to integrate new people and make them community members are also needed to enable internal autonomy: Teaching people the tools, demonstrating "how we do things", encouraging them to talk to others and getting to know their competencies. This is all well and good, but on the other hand, maybe a bit wasted since the teams in The Project do not choose their own paths? Likewise, from the outside it does not look like the improvisation-like aspects of agile methods (experiments, delivering work in increments and seeking feedback… Section 2.2) are practiced much in The Project. It seems like The Project is following the book on how to create internally autonomous teams, but without taking the last step and giving teams autonomy.

External autonomy refers to how the team is influenced by its surroundings – that is, management, customers, and such. In general, external interference is problematic because it lowers autonomy (Moe et al., 2008). In more recent works, Moe et al have examined the barriers to team autonomy in multi-team projects and identified two recurring themes: "overall direction" and "external dependencies" (Moe et al., 2019).

- Overall direction refers to deliverables and deadlines set by management without involving the teams.
- External dependencies are organizational constraints forcing team members to take on additional tasks outside of the planned work (firefighting, sales support, tools maintenance etc) – and also dependencies between teams that force team members to wait for other teams to finish their work.

To summarize, we can find at least three ways in which project-level coordination works against external autonomy in The Project:

- Self-organization is limited. Team roles and work processes are defined centrally by the project organization and not by the teams.
- Empowerment is limited. Teams do not decide what to work on next, but get work assigned from the project manager.
- Independence is limited. Teams do not have sufficient information to make up their own minds about priorities between tasks. They rely on the project manager to convey priorities from the Technical Product Manager.

When we talked about "Autonomy" in group conversations, this was often interpreted as "freedom from" external dependencies (as defined above). An alternative interpretation is "freedom **to**" – freedom to engage with customers, freedom to get involved in the product strategy, and freedom to collaborate across teams. We see that the project-level coordination mechanisms serve to protect the teams from external dependencies and provide a certain level of "freedom from". There are still "Overall direction"-type barriers which are limiting both individual and team opportunities for "freedom to".

5.4 Summary of the Discussion

This chapter made two main observations regarding how cultural factors can influence agile practices:

- Formal coordination is not so important in high-trust cultures.
- Social capital enables agility beyond team boundaries.

We then discussed how agile practices can affect cultural factors:

- Scrum events promote openness.
- Team-level practices counteract project-level collaboration orientation.
- Project-level coordination reduces autonomy and engagement.

These main points will be the basis for my conclusion (next chapter).

6 Summary

6.1 Conclusion

This study indicates an interrelationship between cultural traits and agile practices.

My first research question was **"How can cultural factors affect agile practices in multi-team projects?"**

The Project shows us at least two ways in which the culture is influencing the practice:

- A high-trust culture enables Scrum as a project management practice, because formal coordination is not so important in high-trust cultures.
- Organizational encouragement makes the practice work beyond the team level, because social capital enables agility beyond team boundaries.

I have not found any signs of "resistance" to the agile ways of working among project members. I think this is because The Project culture is built upon similar values as the Scrum methodology. In particular, because The Project values trust, safety and openness so much, it promotes practices and behaviours that can support these values. The Project acts as a Community-of-Practice, and since the original community members decided to use Scrum, the methodology has been passed on to each new member as a right way of working. If another methodology had been chosen from the outset, that might have been the gospel today.

The Scrum practices promote the basic cultural traits of trust, safety and openness *within* Scrum teams, but between the teams, the core methodology provides less opportunities to live by the values. The Project overcomes this hurdle by having individual, informal interactions that strengthen these values as part of the cultural norms of behaviour. By having network building as part of the onboarding process, and by promoting openness in interactions between old and new project members, The Project clears the information flow and enables Scrum to work beyond the team level.

The Project culture does not seem very geared towards external collaboration, flexibility, or change, and so those aspects of the methodology are not emphasized by The Project practices either. However, contrary to popular industry reports, I cannot say that the culture is holding The Project back from becoming "more agile". If anything, it is held back by its chosen way of working with Scrum in a project hierarchy. I can see two opposite ways of dealing with this: Either splitting The Project into smaller, flatter, autonomous entities – which can be difficult because of the nature of the product – or, introducing new practices that encourage more project-wide or company-wide collaboration. This could be done for instance by establishing a few communities of interest with voluntary participation.

This brings us to the second research question**: "How can agile practices affect cultural factors in multi-team projects?"**

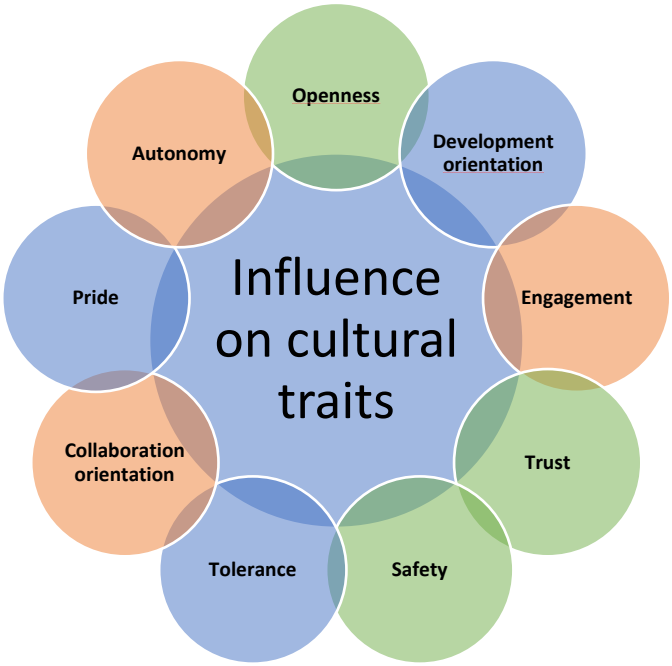


Figure 6 - Cultural traits influenced by agile practices in The Project (red = predominantly negative, green = predominantly positive, blue = neutral)

My answer is that agile practices affect The Project culture mainly by reinforcing cultural traits and behaviours that are already there. Over time, they have effectively turned The Project into a Community-of-Practice.

- The agile practices reinforce the high-trust culture because Scrum events promote openness, and thereby trust and safety.
- The agile practices do not support project-level learning and innovation because team-level practices counteract project-level collaboration orientation.
- The need for project-level coordination of Scrum teams reduces autonomy and engagement.

The Daily, Scrum-of-Scrums and Sprint Review are examples of practices that promote Openness in an already trusting and safe environment. If one of these traits were missing, we do not know if the practices would be effective. It is however hard to imagine that people in The Project would be willing to share failures and mistakes if they did not feel that it was safe to speak, or that they would ask for help if they did not trust each other.

The other noticeable effect that agile practices have on The Project culture is that they have a mixed effect on some of the more complex cultural traits related to how people work together. The negative effects all seem related to trying to use the one-team methodology in a multi-team setting: The Scrum practices promote autonomy and collaboration orientation *within* the teams. When these traits are not well supported at the team level by project-level practices, teams lose engagement and clarity of purpose. Instead of setting their own goals, they will wait for higher levels in the project hierarchy to tell them what to do. They will work on their separate tasks rather than sharing knowledge with other teams through collaboration.

Being a long-time project manager, it has been my preconception that if people do not perceive the way their work is organized as suitable for the task at hand, they will lose engagement, ownership, and motivation. Over time, I have speculated, this might

become a barrier to organizational learning, so that those teams who need it most become least capable of change. If so, it follows that applying agile methods that do not «fit» will have negative consequences for innovation and capacity for change – quite contrary to what agile methods are supposed to achieve according to popular belief.

I have not found much support for my preconceptions in this study. I have found a culture that is inward-looking and somewhat self-preserving, but with a fair amount of optimism and willingness to evolve. The agile practices are treated pragmatically, as a means to upholding shared values.

Maybe I did not get the expected results because my research approach was not strict enough, but I think that my findings from this study are correct. If people want to use Scrum because they believe in Scrum, they will make Scrum work. No harm will fall on their culture because of it. They will have to struggle with the balance between autonomy and coordination, and they must be careful not to become too inwards-focused. It may be difficult to keep everyone involved and engaged. This may be a reasonable price to pay for having a methodology that is aligned with established norms and values.

6.2 Research Contribution

With the limitations from section 3.6 regarding validity and generality in mind, the research contribution from this thesis is a small step towards better understanding of how culture and work methods influence each other in software projects. As far as I know, this is the first time EDI has been applied as a model for culture in agile organizations. Considering that the model seems to be useful as an approximation for "agile" culture, it could be interesting to refine the pre-study survey further as a measure of "innovation capacity" for technology companies and businesses where innovation is a part of daily work. Applying this type of questionnaire to R&D projects regularly could provide insights into cultural trends and give early warning on organizational issues.

6.3 Final Words

This work presents a snapshot of how team members perceived The Project around Easter 2021. That snapshot is, of course, no longer valid. Since the time of my interviews, a few people have left The Project and more have joined. Project management has changed. Roles have been clarified: The Product Owner is more involved, and most team members have attended Scrum Master training now and know what to expect from these roles. Practices are also changing. Feedback loops are being established with project-wide retrospectives. Planning meetings have been opened for all project members to attend.

Can The Project do more to improve? Within the areas of planning, collaboration, and customer involvement, I would like to see The Project turning more of its potential into action.

I would recommend first looking into how to keep senior project members involved and engaged.

- "Job rotation" is an ugly phrase, but making it easy to change teams as well as "guest star" from time to time is one way of diffusing knowledge and maintaining the social network.

- Instead of treating the teams as interchangeable units, giving each team a unique purpose and allowing people to flow between teams would enable longer-term planning and more forward-looking teams.
- Longer-term perspectives on teamwork might give people an incentive to think about career development and competence building, thus promoting development orientation.
- Training activities should not be restricted to new team members. Competence building could be encouraged by setting aside time and resources for project-wide learning activities and allowing people to self-subscribe to communities of interest.
- Cross-team collaboration activities should not be restricted to new team members. Larger development activities could be kicked off with two teams collaborating from the start (rather than rushing-to-the rescue when the deadlines approach).

There are also some "simple" fixes that could be done at the team level and be beneficial at project level:

- Stop overloading the Scrum Master role. If this role is not mixed with HR or delivery responsibilities it can be shared between (motivated) team members so that everyone gets a chance to be heard and responsibility is really shared.
- Encourage demos! Making a habit of demonstrating the team's work builds pride, and this should be part of the agile practice.

And finally, The Project should have the difficult talk about "customer collaboration" and what it means. "Customer" does not have to be someone paying for the software, but could be anyone using the software. Is it possible to develop new features more iteratively and involve customer representatives in providing frequent and constructive feedback? Should The Project take on a larger scope and work on samples and applications instead of being a component provider? Would it be feasible to have "user forums" for developers to meet and collect feedback and improvement ideas from other organizational units?

As usual, it is not just what you do but also how you do it that matters. For any of these ideas to be successful, they should be discussed, refined, *and improved* together with The Project members. Because the best outcomes really come from collective efforts.

References

Aasen, & Amundsen. (2011). *Innovasjon som kollektiv prestasjon*. Gyldendal akademisk.

Aasen, T. M. B., & Amundsen, O. (2015). *Innovasjonsarbeid : organisasjon, kultur og ledelse*. Gyldendal akademisk.

Abrar, M. F., Sohail, M., Ali, S., Majeed, M. F., Shah, I. A., Rashid, N., & Ullah, N. (2020). De-motivators for the adoption of agile methodologies for large-scale software development teams: An SLR from management perspective. *Journal of Software-Evolution and Process*, *32*(12), Article e2268. https://doi.org/10.1002/smr.2268

Ahmed, P. K. (1998). Culture and climate for innovation. *European journal of innovation management*.

Albrecht, S. L., Green, C. R., & Marty, A. (2021). Meaningful Work, Job Resources, and Employee Engagement. *Sustainability (Basel, Switzerland)*, *13*(4045), 4045. https://doi.org/10.3390/su13074045

Allik, J., & Realo, A. (2004). Individualism-Collectivism and Social Capital. *Journal of Cross-Cultural Psychology*, *35*(1), 29-49. https://doi.org/10.1177/0022022103260381

Amabile, T. M. (1997). Motivating creativity in organizations: On doing what you love and loving what you do [Article]. *California Management Review*(1), 39-58. https://doi.org/10.2307/41165921

Amabile, T. M., Conti, R., Coon, H., Lazenby, J., & Herron, M. (1996). Assessing the work environment for creativity [Article]. *Academy of Management Journal*, *39*(5), 1154-1184. https://doi.org/10.2307/256995

Amble, N., Amundsen, O., Rismark, M., Gabrielsen, I. M., Johansen, E. M., Stiklestad, S. S., & Waaler, S. (2020). *Medarbeiderdrevet innovasjon*. Gyldendal.

Amundsen, O., & Kongsvik, T. Ø. (2016). *Endringskynisme : og kunsten å skape god endringspraksis* (2. utg. ed.). Gyldendal akademisk.

Anderson, N., Potocnik, K., & Zhou, J. (2014). Innovation and Creativity in Organizations: A State-of-the-Science Review, Prospective Commentary, and Guiding Framework. *Journal of Management*, *40*(5), 1297-1333. https://doi.org/10.1177/0149206314527128

Association, A. P. Retrieved 29.12.2021, from https://dictionary.apa.org/

Biddle, R., Meier, A., Kropp, M., Anslow, C., & Ieee. (2018). *MyAgile: Sociological and Cultural Effects of Agile on Teams and their Members*. https://doi.org/10.1145/3195836.3195845

Bowen, G. A. (2006). Grounded Theory and Sensitizing Concepts. *International Journal of Qualitative Methods*, *5*(3), 12-23. https://doi.org/10.1177/160940690600500304

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.

Bunyakiati, P., Surachaikulwattana, P., & Ieee. (2016). Fit Between Agile Practices and Organizational Cultures. In *2016 13th International Joint Conference on Computer Science and Software Engineering* (pp. 438-443). <Go to ISI>://WOS:000389036200079

Burns, J. M. (1979). *Leadership*. Harper Row.

Burns, T., & Stalker, G. M. (1966). *The management of innovation* (Vol. 6). Tavistock.

Cockburn, A., & Highsmith, J. (2001). Agile software development: The people factor. *Computer*, *34*(11), 131-133. https://doi.org/10.1109/2.963450

Cohn, M. (2010). *Succeeding with Agile: Software Development Using Scrum.* Addison-Wesley.

Coleman, J. S. (1994). *Foundations of social theory*. Cambridge, Mass: Belknap Press of Harvard University Press.

Conboy, K. (2009). Agility from First Principles: Reconstructing the Concept of Agility in Information Systems Development. *Information systems research*, *20*(3), 329-354. https://doi.org/10.1287/isre.1090.0236
Conboy, K., & Carroll, N. (2019). Implementing Large-Scale Agile Frameworks: Challenges and Recommendations. *Ieee Software*, *36*(2), 44-50. https://doi.org/10.1109/MS.2018.2884865
Conboy, K., Wang, X., & Fitzgerald, B. (2009). Creativity in Agile Systems Development: A Literature Review. In G. Dhillon, B. C. Stahl, & R. Baskerville (Eds.), *Information Systems - Creativity and Innovation in Small and Medium-Sized Enterprises* (Vol. 301, pp. 122-+). https://doi.org/10.1007/978-3-642-02388-0_9
Cross, R., & Parker, A. (2004). *The hidden power of social networks : understanding how work really gets done in organizations*. Harvard Business School Press.
Dahl, B., Schjødt-Osmo, S., Karlsen, L. S., Finnestrand, H. O., & Moe, N. B. (2018). *Challenges for Autonomous Development Teams - A Multiple Case Study* NTNU].
Dalton, J. (2018). *Great Big Agile*. APress.
Deci, E. L., & Ryan, R. M. (1985). *Intrinsic Motivation and Self-Determination in Human Behavior* (1st ed. 1985. ed.). Springer US : Imprint: Springer.
Deci, E. L., & Ryan, R. M. (2000). The "What" and "Why" of Goal Pursuits: Human Needs and the Self-Determination of Behavior. *Psychological inquiry*, *11*(4), 227-268. https://doi.org/10.1207/S15327965PLI1104_01
Dubos, R. (2017). *Social capital: Theory and research*. Routledge.
Duguid, P. (2005). 'The Art of Knowing': Social and Tacit Dimensions of Knowledge and the Limits of the Community of Practice. *The Information Society*, *21*(2), 109-118. https://brill.com/view/book/edcoll/9789460919152/BP000008.xml
Edmondson, A. (1999). Psychological Safety and Learning Behavior in Work Teams. *Administrative science quarterly*, *44*(2), 350-383. https://doi.org/10.2307/2666999
Edmondson, A. C. (2019). *The fearless organization : creating psychological safety in the workplace for learning, innovation, and growth* (1st edition. ed.). Wiley.
Ehrlich, K., & Cataldo, M. (2012). *All-for-one and one-for-all? a multi-level analysis of communication patterns and individual performance in geographically distributed software development* Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work, Seattle, Washington, USA. https://doi.org/10.1145/2145204.2145345
Forsgren, N., Humble, J., & Kim, G. (2018). *Accelerate. Building and Scaling High Performing Technology Organizations.* IT Revolution.
Google. (2015, Feb, 15). The five keys to a successful Google team. *re:Work*. https://rework.withgoogle.com/blog/five-keys-to-a-successful-google-team/
Goran, J., LaBerge, L., & Srinivashan, R. (2017, July). Culture for a Digital Age. *McKinsey Quarterly*.
Greenwood, D. J., & Levin, M. (1998). Action research, science, and the co-optation of social research. *Studies in cultures, organizations and societies*, *4*(2), 237-261.
Gundelsby, J. (2018). Enabling autonomous teams in large-scale agile through architectural principles. In *ACM International Conference Proceeding Series* (pp. 1-4): ACM.
Hoda, R., Noble, J., & Ieee. (2017). Becoming Agile: A Grounded Theory of Agile Transitions in Practice. In *2017 Ieee/Acm 39th International Conference on Software Engineering* (pp. 141-151). https://doi.org/10.1109/icse.2017.21
Hunter, S. T., Bedell, K. E., & Mumford, M. D. (2007). Climate for creativity: A quantitative review. *Creativity Research Journal*, *19*(1), 69-90. https://doi.org/Doi 10.1080/10400410709336883
Iivari, J., & Iivari, N. (2011). The relationship between organizational culture and the deployment of agile methods. *Information and Software Technology*, *53*(5), 509-520. https://doi.org/10.1016/j.infsof.2010.10.008
Isaksen, S. G., & Akkermans, H. J. (2011). Creative Climate: A Leadership Lever For Innovation. *Journal of Creative Behavior*, *45*(3), 161-187. https://doi.org/10.1002/j.2162-6057.2011.tb01425.x

Isaksen, S. G., Lauer, K. J., & Ekvall, G. (1999). Situational Outlook Questionnaire: A Measure of the Climate for Creativity and Change. *Psychological reports*, *85*(2), 665-674. https://doi.org/10.2466/pr0.1999.85.2.665
Jacobsen, D. I., & Thorsvik, J. (2013). *Hvordan organisasjoner fungerer, 4. utgave*. Fagbokforlaget.
Johannessen, A., Christoffersen, L., & Tufte, P. A. (2016). *Introduksjon til samfunnsvitenskapelig metode* (5. utg. ed.). Abstrakt.
Kakar, A. K. (2017). Assessing Self-Organization in Agile Software Development Teams. *Journal of Computer Information Systems*, *57*(3), 208-217. https://doi.org/10.1080/07362994.2016.1184002
Kalenda, M., Hyna, P., & Rossi, B. (2018). Scaling agile in large organizations: Practices, challenges, and success factors. *Journal of Software-Evolution and Process*, *30*(10), Article e1954. https://doi.org/10.1002/smr.1954
Kilduff, M., & Brass, D. J. (2010). Job design: A social network perspective. *J. Organiz. Behav*, *31*(2-3), 309-318. https://doi.org/10.1002/job.609
Kilduff, M., & Krackhardt, D. (2008). *Interpersonal networks in organizations : cognition, personality, dynamics, and culture* (Vol. 30). Cambridge University Press.
Kniberg, H., & Ivarsson, A. (2012). Scaling Agile @Spotify. https://blog.crisp.se/wp-content/uploads/2012/11/SpotifyScaling.pdf
Kongsvik, T., & Almklov, P. (2021). Combining lenses: Pragmatics and action research in safety science. In *Inside Hazardous Technological Systems* (pp. 133-150). CRC Press.
Kotter, J. P. (1995). Leading change: Why transformation efforts fail. *Harvard Business Review*, *73(2)*, 59-67.
Kotter, J. P., Mueller, P., & Rathgeber, H. (2006). *Our iceberg is melting : changing and succeeding under any conditions*. Macmillan.
Küpper, S. (2016). *The impact of agile methods on the development of an agile culture: research proposal: [the agile evolution]* Proceedings of the 20th International Conference on Evaluation and Assessment in Software Engineering, Limerick, Ireland. https://doi.org/10.1145/2915970.2915977
Laanti, M., & Kettunen, P. (2019). SAFe Adoptions in Finland: A Survey Research. In R. Hoda (Ed.), *Agile Processes in Software Engineering and Extreme Programming - Workshops* (Vol. 364, pp. 81-87). https://doi.org/10.1007/978-3-030-30126-2_10
Lave, J., & Wenger, E. (1991). *Situated learning: Legitimate peripheral participation*. Cambridge university press.
Manifesto for Agile Software Development. (2001). https://agilemanifesto.org/
Mayhew, M. G., Ashkanasy, N. M., Bramble, T., & Gardner, J. (2007). A Study of the Antecedents and Consequences of Psychological Ownership in Organizational Settings. *J Soc Psychol*, *147*(5), 477-500. https://doi.org/10.3200/SOCP.147.5.477-500
McAfee, A., & Brynjolfsson, E. (2018). *Machine, Platform, Crowd*. Ww Norton & Co.
McKinsey&Company. (2017). The 5 Trademarks of Agile Organizations. *McKinsey&Company*. Retrieved December, from https://www.mckinsey.com/business-functions/organization/our-insights/the-five-trademarks-of-agile-organizations
McLean, L. D. (2005). Organizational Culture's Influence on Creativity and Innovation: A Review of the Literature and Implications for Human Resource Development. *Advances in developing human resources*, *7*(2), 226-246. https://doi.org/10.1177/1523422305274528
Meyer, C. B., & Stensaker, I. G. (2011). *Endringskapasitet*. Fagbokforl.
Mintzberg, H. (2009). *Managing*. Prentice Hall Financial Times.
Moe, N. B., Dahl, B., Stray, V., Karlsen, L. S., & Schjødt-Osmo, S. (2019). Team Autonomy in Large-Scale Agile. 52nd Hawaii International Conference on System Sciences, Hawaii.
Moe, N. B., Dingsoyr, T., & Dyba, T. (2008). Understanding Self-Organizing Teams in Agile Software Development. In (pp. 76-85): IEEE.

Moe, N. B., & Mikalsen, M. (2020). Large-Scale Agile Transformation: A Case Study of Transforming Business, Development and Operations. In V. Stray, R. Hoda, M. Paasivaara, & P. Kruchten, *Agile Processes in Software Engineering and Extreme Programming* Cham.
Moe, N. B., & Stray, V. (2020). *A Decade of Research on Autonomous Agile Teams: A Summary of the Third International Workshop*
Morgan, G. (2006). *Images of organization* (Updated ed. ed.). Sage.
Nærings- og handelsdepartementet, International Research Institute of Stavanger, & Ntnu samfunnsforskning. (2011). *Håndbok i medarbeiderdrevet innovasjon*. Nærings-og handelsdepartementet.
Nahapiet, J., & Ghoshal, S. (1998). Social Capital, Intellectual Capital, and the Organizational Advantage. *The Academy of Management review*, *23*(2), 242-266. https://doi.org/10.2307/259373
Oddane, T. (2017). *Kreativitet og innovasjon : fem sider av nesten samme sak*. Fagbokforl.
Oldham, G. R., & Hackman, J. R. (2010). Not what it was and not what it will be: The future of job design research. *Journal of organizational behavior*, *31*(2-3), 463-479.
Ouchi, W. G. (1980). Markets, Bureaucracies, and Clans. *Administrative science quarterly*, *25*(1), 129-141. https://doi.org/10.2307/2392231
Patterson, M. G., West, M. A., Shackleton, V. J., Dawson, J. F., Lawthom, R., Maitlis, S., Robinson, D. L., & Wallace, A. M. (2005). Validating the organizational climate measure: links to managerial practices, productivity and innovation. *Journal of organizational behavior*, *26*(4), 379-379+. https://doi.org/http://dx.doi.org/10.1002/job.312
Payne, G., & Williams, M. (2005). Generalization in Qualitative Research. *Sociology*, *39*, 295-314. https://doi.org/10.1177/0038038505050540
Perri, M. (2018). *Escaping the build trap: How effective product management creates real value*. O'Reilly Media.
Pikkarainen, M., & Wang, X. (2011). *An Investigation of Agility Issues in Scrum Teams Using Agility Indicators*. https://doi.org/10.1007/978-1-4419-7355-9_38
Putnam, R. D. (1994). Social Capital and Public Affairs. *Bulletin of the American Academy of Arts and Sciences*, *47*(8), 5-19. https://doi.org/10.2307/3824796
Roberts, J. (2006). Limits to communities of practice. *Journal of Management Studies*, 623-639. https://doi.org/info:doi/
Šāblis, A., Šmite, D., & Moe, N. B. (2020). Team-external coordination in large-scale software development projects. In.
Schein, E. H. (2010). *Organizational Culture and Leadership* (Fourth Edition ed.). Jossey-Bass.
Schiefloe, P. M. (2015). *Sosiale landskap og sosial kapital : nettverk og nettverksforskning* (2. utg. ed.). Universitetsforl.
Schwaber, K. (1995). *SCRUM Development Process* OOPSLA '95, Austin, Texas.
Schwaber, K., & Sutherland, J. (2020). *The 2020 Scrum Guide*. Retrieved 03.30.2021 from https://scrumguides.org/scrum-guide.html
Shafiq, S., Inayat, I., & Abbas, M. (2019, 28-30 Aug. 2019). Communication Patterns of Kanban Teams and Their Impact on Iteration Performance and Quality. 2019 45th Euromicro Conference on Software Engineering and Advanced Applications (SEAA),
Skelton, M., & Pais, M. (2019). *Team Topologies: Organizing Business and Technology Teams for Fast Flow*. IT Revolution.
Šmite, D., Gonzalez-Huerta, J., & Moe, N. B. (2020). "When in Rome, Do as the Romans Do": Cultural Barriers to Being Agile in Distributed Teams. In (Vol. 383, pp. 145-161).
Smite, D., Moe, N. B., Levinta, G., & Floryan, M. (2019). Spotify Guilds: How to Succeed With Knowledge Sharing in Large-Scale Agile Organizations. *Ieee Software*, *36*(2), 51-57. https://doi.org/10.1109/MS.2018.2886178
Sørhaug, T. (1996). *Om ledelse : makt og tillit i moderne organisering*. Universitetsforl.

Spiegler, S. V., Heinecke, C., & Wagner, S. (2019). The Influence of Culture and Structure on Autonomous Teams in Established Companies. In R. Hoda (Ed.), *Agile Processes in Software Engineering and Extreme Programming - Workshops* (Vol. 364, pp. 46-54). https://doi.org/10.1007/978-3-030-30126-2_6

Stray, V., Moe, N., & Hoda, R. (2018). Autonomous agile teams: challenges and future directions for research. In *ACM International Conference Proceeding Series* (pp. 1-5): ACM.

Strode, D. E., Huff, S. L., & Tretiakov, A. (2009). The Impact of Organizational Culture on Agile Method Use. In (pp. 1-9): IEEE.

Struckman, C., Sanchez, Daniel, Reina, G., Ed, & Ramirez, J. (2020). *The Culture PRISM: 5 Dimensions That Shape Your Culture*.

Takeuchi, H., & Nonaka, I. (1986). The New Product Development Game. *Harvard Business Review*(January). https://hbr.org/1986/01/the-new-new-product-development-game

Tan Trung, L., Sivarajah, U., & Weerakkody, V. (2019). Do Agile Managed Information Systems Projects Fail Due to a Lack of Emotional Intelligence? *Information Systems Frontiers*. https://doi.org/10.1007/s10796-019-09962-6

Tidd, J., & Bessant, J. R. (2018). *Managing Innovation: Integrating Technological, Market and Organizational Change* (Sixth Edition ed.). John Wiley & Sons.

Tjora, A. H. (2021). *Kvalitative forskningsmetoder i praksis* (4. utgave. ed.). Gyldendal.

Tolfo, C., Wazlawick, R. S., Ferreira, M. G. G., & Forcellini, F. A. (2011). Agile methods and organizational culture: reflections about cultural levels. *Journal of Software Maintenance and Evolution-Research and Practice*, *23*(6), 423-441. https://doi.org/10.1002/smr.483

Weick, K. E. (1995). *Sensemaking in organizations* (Vol. 3). Sage.

Wendorff, P. (2002). Organisational culture in agile software development. In M. Oivo & S. KomiSirvio (Eds.), *Product Focused Software Process Improvement, Proceedings* (Vol. 2559, pp. 145-157). <Go to ISI>://WOS:000181620800011

Wenger, E. (1999). *Communities of practice: Learning, meaning, and identity*. Cambridge university press.

West, M., Wan, D. W., Blosen, B., Sklavounakis, A., & Chan, W. F. (2021). Market Guide for Enterprise Agile Frameworks.

What is Scrum? (2020). Retrieved 03.30.2021 from https://www.scrum.org/resources/what-is-scrum

Wise, S. (2014). Can a team have too much cohesion? The dark side to network density. *European management journal*, *32*(5), 703-711. https://doi.org/10.1016/j.emj.2013.12.005

www.stateofagile.org. (2020). 14th Annual State of Agile report. Retrieved February, 2021, from www.stateofagile.org

Appendices

Appendix A - Questionnaire

Appendix B – one-to-one interview guide

Appendix C – Focus group interview guide

Appendix D – NSD approval

Appendix A - Questionnaire

Anonymous questionnaire, cultural factors. Distributed electronically to all project team members.

EDI-factor	Statement: In this project…
Åpenhet/ Openness	a) Information is widely shared.** *b) Changes are made without talking to the people involved in them***
Utviklingsorientering/ Development orientation	a) Information is actively sought.* b) Failures are treated primarily as opportunities to improve the system.*
Engasjement/ Engagement	a) The work atmosphere is filled with energy*** b) People are prepared to make a special effort to do a good job **
Tillit/ Trust	a) Responsibilities are shared. * b) Supervisors show that they have confidence in those they manage. **
Trygghet/ Safety	a) We encourage and support each other. b) People usually feel welcome when presenting new ideas.*/***
Toleranse/ Tolerance	*a) Messengers are punished when they deliver news of failure or other bad news.* * b) A wide variety of viewpoints are expressed here.***
Samarbeidsorientering/ Collaboration orientation	a) Cross-functional collaboration is encouraged and rewarded. * b) Collaboration between teams and departments is very effective. ***
Stolthet/ Pride	*a) We do not have much of a reputation for top-quality products* ** b) People are happy to say they work on the project.
Autonomi/ Autonomy	a) People make choices about their own work *** b) Management trust people to take work-related decisions without getting permission first. **

General	
How long have you worked on this project?	1. Less than one year 2. One year or more
This survey covers some aspects of project culture, but not all. Are there other aspects you'd rather be asked about?	Free text
Other comments	Free text

*(Forsgren et al., 2018)

**(Patterson et al., 2005)

***(Isaksen et al., 1999)

Appendix B – one-to-one interview guide

One-to-one interviews with small number of project stakeholders.

- What is your role in/related to the project?
- Modes of communication - How do you communicate with the project?
    - Formal communication channels
    - Informal communications
- Information sharing - How is information shared in the project?
    - How do you find information relevant to the project?
    - How do you share information with others?
- Decisions – How are decisions made in the project?
    - Do you take part in project-level decisions?
    - Do you take part in team-level decisions?
    - Do you know who makes decisions that affect your work?
- Related to these topics, what do you see as strengths and weaknesses (that we haven't covered so far) in the way work is done in the project?

Appendix C – Focus group interview guide

Group conversations with project teams, 1-2 hours depending on group size.

- How well do you think the survey results reflect the situation in your team? In the project as a whole?
- How do you think other teams will react to these findings? Do you think the results are more/less representative of other teams?
- Which findings do you think are more surprising?
- Is there anything in these results you think should be investigated further?

Appendix D – NSD approval (next page)

NSD NORSK SENTER FOR FORSKNINGSDATA

Vurdering

Referansenummer

997794

Prosjekttittel

Masteroppgave, innovasjonskultur i smidige multi-team-prosjekter

Behandlingsansvarlig institusjon

Norges teknisk-naturvitenskapelige universitet / Fakultet for samfunns- og utdanningsvitenskap (SU) / Institutt for sosiologi og statsvitenskap

Prosjektansvarlig (vitenskapelig ansatt/veileder eller stipendiat)

Petter Grytten Almklov, Petter.Almklov@ntnu.no, tlf: 73559998

Type prosjekt

Studentprosjekt, masterstudium

Kontaktinformasjon, student

Monika Lie Larsen, monikall@stud.ntnu.no, tlf: 92443422

Prosjektperiode

22.03.2021 - 01.02.2022

Vurdering (1)

18.03.2021 - Vurdert

Det er vår vurdering at behandlingen av personopplysninger i prosjektet vil være i samsvar med personvernlovgivningen så fremt den gjennomføres i tråd med det som er dokumentert i meldeskjemaet med vedlegg den 18.03.2021 samt i meldingsdialogen mellom innmelder og NSD. Behandlingen kan starte.

DEL PROSJEKTET MED PROSJEKTANSVARLIG
For studenter er det obligatorisk å dele prosjektet med prosjektansvarlig (veileder). Del ved å trykke på knappen «Del prosjekt» øverst til venstre i meldeskjemaet. Prosjektansvarlig bes akseptere invitasjonen innen en uke. Dersom invitasjonen utløper, må han/hun inviteres på nytt.

MELD VESENTLIGE ENDRINGER
Dersom det skjer vesentlige endringer i behandlingen av personopplysninger, kan det være nødvendig å melde dette til NSD ved å oppdatere meldeskjemaet. Før du melder inn en endring, oppfordrer vi deg til å lese om hvilke type endringer det er nødvendig å melde:

https://www.nsd.no/personverntjenester/fylle-ut-meldeskjema-for-personopplysninger/melde-endringer-i-meldeskjema
Du må vente på svar fra NSD før endringen gjennomføres.

TYPE OPPLYSNINGER OG VARIGHET
Prosjektet vil behandle alminnelige kategorier av personopplysninger frem til 01.02.2022.

LOVLIG GRUNNLAG
Prosjektet vil innhente samtykke fra de registrerte til behandlingen av personopplysninger. Vår vurdering er at prosjektet legger opp til et samtykke i samsvar med kravene i art. 4 og 7, ved at det er en frivillig, spesifikk, informert og utvetydig bekreftelse som kan dokumenteres, og som den registrerte kan trekke tilbake.

Lovlig grunnlag for behandlingen vil dermed være den registrertes samtykke, jf. personvernforordningen art. 6 nr. 1 bokstav a.

PERSONVERNPRINSIPPER
NSD vurderer at den planlagte behandlingen av personopplysninger vil følge prinsippene i personvernforordningen om:

· lovlighet, rettferdighet og åpenhet (art. 5.1 a), ved at de registrerte får tilfredsstillende informasjon om og samtykker til behandlingen
· formålsbegrensning (art. 5.1 b), ved at personopplysninger samles inn for spesifikke, uttrykkelig angitte og berettigede formål, og ikke behandles til nye, uforenlige formål
· dataminimering (art. 5.1 c), ved at det kun behandles opplysninger som er adekvate, relevante og nødvendige for formålet med prosjektet
· lagringsbegrensning (art. 5.1 e), ved at personopplysningene ikke lagres lengre enn nødvendig for å oppfylle formålet

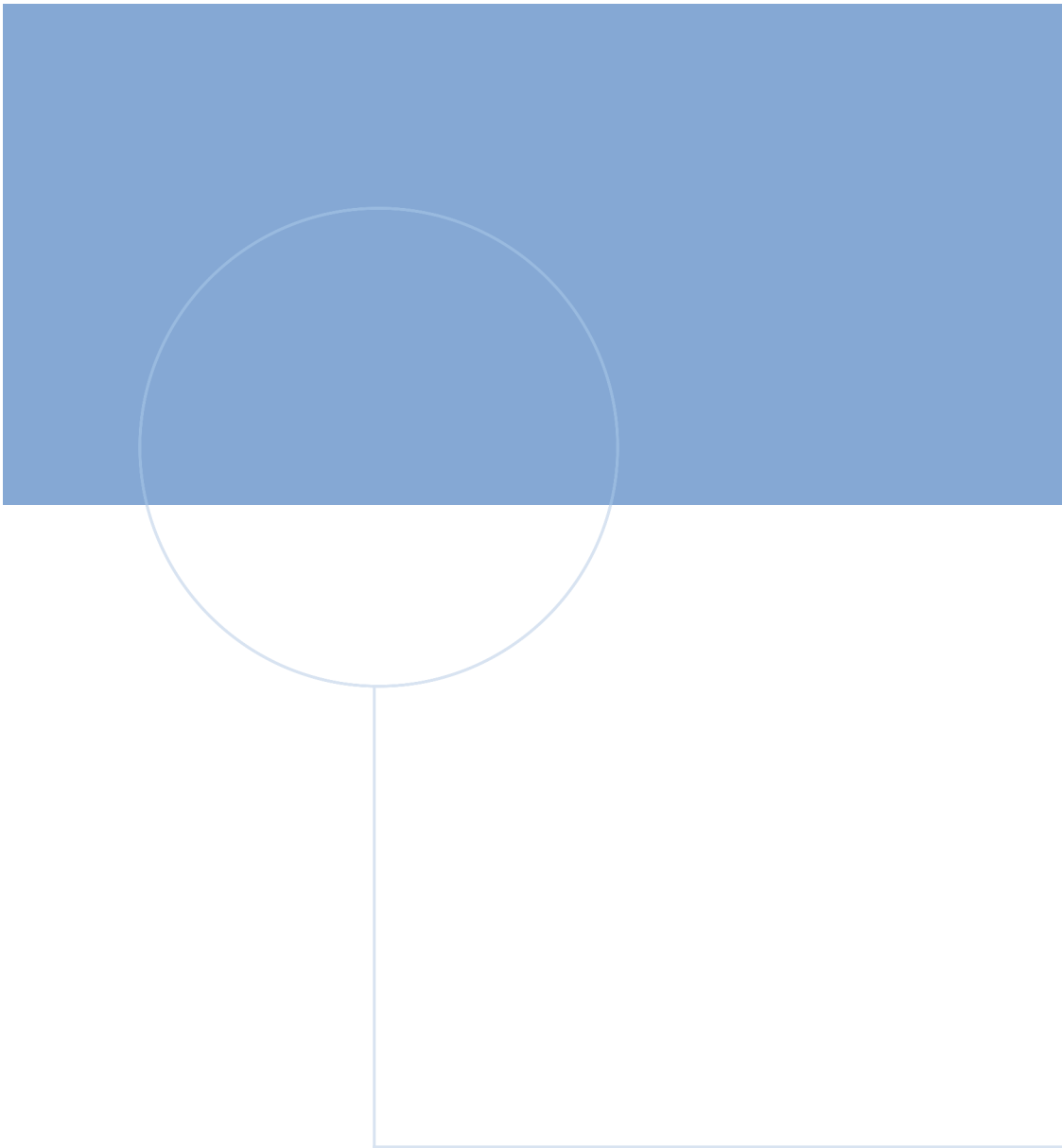
DE REGISTRERTES RETTIGHETER
Så lenge de registrerte kan identifiseres i datamaterialet vil de ha følgende rettigheter: innsyn (art. 15), retting (art. 16), sletting (art. 17), begrensning (art. 18) og dataportabilitet (art. 20).
NSD vurderer at informasjonen om behandlingen som de registrerte vil motta oppfyller lovens krav til form og innhold, jf. art. 12.1 og art. 13.

Vi minner om at hvis en registrert tar kontakt om sine rettigheter, har behandlingsansvarlig institusjon plikt til å svare innen en måned.

FØLG DIN INSTITUSJONS RETNINGSLINJER
NSD legger til grunn at behandlingen oppfyller kravene i personvernforordningen om riktighet (art. 5.1 d), integritet og konfidensialitet (art. 5.1. f) og sikkerhet (art. 32).
For å forsikre dere om at kravene oppfylles, må dere følge interne retningslinjer og/eller rådføre dere med behandlingsansvarlig institusjon.

OPPFØLGING AV PROSJEKTET
NSD vil følge opp ved planlagt avslutning for å avklare om behandlingen av personopplysningene er avsluttet.

Lykke til med prosjektet!
Tlf. Personverntjenester: 55 58 21 17 (tast 1)



NTNU

Norwegian University of
Science and Technology