What Makes Effective Leadership in Agile Software Development Teams?

Lucas Gren
Volvo Cars and Chalmers | University of Gothenburg
Gothenburg, Sweden
Blekinge Institute of Technology
Karlskrona, Sweden
lucas.gren@lucasgren.com

Paul Ralph
Dalhousie University
Halifax, Canada
paulralph@dal.ca

ABSTRACT

Effective leadership is one of the key drivers of business and project success, and one of the most active areas of management research. But how does leadership work in agile software development, which emphasizes self-management and self-organization and marginalizes traditional leadership roles? To find out, this study examines agile leadership from the perspective of thirteen professionals who identify as agile leaders, in different roles, at ten different software development companies of varying sizes. Data from semi-structured interviews reveals that leadership: (1) is dynamically shared among team members; (2) engenders a sense of belonging to the team; and (3) involves balancing competing organizational cultures (e.g. balancing the new agile culture with the old milestone-driven culture). In other words, agile leadership is a property of a team, not a role, and effectiveness depends on agile team members' identifying with the team, accepting responsibility, and being sensitive to cultural conflict.

CCS CONCEPTS

• **Software and its engineering** → **Agile software development**; **Programming teams**.

KEYWORDS

agile methods, leadership, management, culture

1 INTRODUCTION

Leadership is one of the key drivers of organizational and project success [8]. However, agile software development involves a cultural change [cf. 40, 73] that flattens the organizational hierarchy, de-emphasizes managers and empowers teams [51]. One might therefore wonder whether there is leadership in agile teams at all.

However, management is different from leadership. "Leadership [involves] the articulation of an organizational vision, introduction of major organizational change, providing inspiration, and dealing with highly stressful and troublesome aspects of the external environments of organizations" [37, p. 444]. In contrast, "Management consists of implementing the vision and strategy provided by leaders, coordinating and staffing the components of organizations, administering the infrastructures of organizations, and handling the day-to-day problems that inevitably emerge in the process of strategy and policy implementation and ongoing organizational functioning" [37, p. 445].

Put more simply, "leadership is getting group members to achieve the group's goals" [35, p. 315]. *Leadership is a kind of work.* Even in self-organized and self-managed agile teams, someone is doing leadership. Indeed, many team members can dynamically share the leadership work of driving the team toward its goals.

Research on leadership in software engineering is scarce. One notable exception is Kalliamvakou et al.'s study of Microsoft managers, which concluded that "we still don't know what to look for in a great software engineering manager, and how to further develop their skills to support the teams they manage" [43]. A recent systematic review concluded that "agile leadership research needs further attention and that more empirical studies are needed to better understand agile leadership in general" [50]. This motivates the following research question.

> **Research question:** *What is the nature of leadership in agile software development, from the perspective of professionals who identify as agile leaders?*

Here, agile software development refers to both the ideology laid out in the agile manifesto [21] and the specific methodologies and practices that embody that ideology (e.g. Extreme Programming [9], Scrum [63], Dual-Track Development [65]).

To address this research question, we conducted a qualitative survey (a series of interviews) of thirteen self-described

Lucas Gren and Paul Ralph

Table 1: Participant Information

PID	Org.	Industry	Employees	Role	Prior Method	Reason for agile
A1	A	Consumer packaged goods	26,000	Agile coach	Waterfall	Engagement, job satisfaction, and quality
B1	B	Retail business	16,000	Project manager lead	Waterfall	Improve business value
C1	C	Automatic identification/data capture	1,800	Project portfolio management responsible	Ad hoc process	Project priorities not personal interest
D1	D	Industrial supply distribution	18,400	Project Manager (initiator of agile).	Waterfall	Innovation in value delivery
E1	E	Personal care	56,000	Team leader sales and distr.	Waterfall	Improve the company
E2	E	Personal care	56,000	Lead of 25 project managers	Waterfall	Improve the company
F1	F	Oil and gas	82,000	Project manager (project execution)	Waterfall	Read about agile methods
G1	G	Software	74,400	Scrum Master/Project Manager	Started agile	Speed
G2	G	Software	74,400	Scrum Master of two teams	Started agile	Speed
G3	G	Software	74,400	Scrum Master	Started agile	Speed
H1	H	Online media and social networking	5,000	Scrum Master/manager in first agile team	A culture of guessing what users liked	Better and faster feedback/ solve org. problem
I1	I	Training and education	100	Multi-type supporting role	Similar to agile	Started as agile
J1	J	Software consultancy	35	Founder	Agile principles with flat org. structure	Started as agile

agile leaders from ten different organizations (see Section 2). Section 3 presents our findings. Next, we describe existing research related to our findings (Section 4), and then integrate our findings with this work (Section 5). Section 6 describes the study's limitations and Section 7 concludes the paper with some thoughts on future work.

2 METHOD

This section explains our approaches to data collection and analysis. Since our research question concerns the perspectives of agile leaders, we employ a qualitative survey [59] targeting professionals who identify as agile leaders.

2.1 Participants

The first author approached industry contacts—with whom he had no direct relationship or conflict of interest—to recruit participants who identify as agile leaders (e.g., Scrum Masters, founders, agile coaches, and various project managers). We focused on agile leaders rather than other professionals because we wanted to understand the leaders' perspective. Of course, comparing the leaders' perspective with software professionals who explicitly do not identify as leaders is a potential extension. We also wanted participants of different ages, from both small and large companies, and from different countries, to maximize sample diversity—a key aspect of representativeness [7]. Moreover, we want to investigate components of agile leadership that are similar across these contextual characteristics.

More formally, the recruitment criteria were: (1) Employees who identify as agile leaders, (2) Companies of different size, (3) Mix of age and gender, and (4) Mix of countries and national cultures.

Nine men and four women between 25- and 50-years-old participated. Six participants were located in the USA, three in Brazil, two in Germany, one in the UK, and one in Tunisia. All participants indicated that their teams use an agile approach. Participants roles, industries, organization sizes and methodological contexts varied (see Table 1). SAP America Inc. mediated some of the contacts to the companies; four of the participants were working on Agile SAP implementations, meaning that they use agile methods in customizing an SAP product for an organization [61]. All four also worked on other, non-SAP projects.

2.2 Data Collection

The first author completed 13, 45–90 minute, semi-structured interviews: ten via teleconference; three face-to-face. He began by introducing himself and the project, getting permission to record the interview and emphasizing how participants' identities would be protected. The complete interview guide is available in Supplementary Material (see Data Availability). All interviews were transcribed word-for-word.

2.3 Data Analysis

The data was analyzed using ATLAS.ti[1] following the steps recommended by Braun et al. [11]:

(1) read each transcript
(2) highlight all statements broadly related to concept under investigation (i.e. *leadership* in this case)
(3) sort the highlighted statements into categories
(4) name each category
(5) for each category, re-read all of its statements together; reassess category cohesion and name

_____

[1]https://atlasti.com/

The first author began by reading all the transcripts and tagging all quotations related to leadership in Atlas.ti—176 in all. Next, quotations were grouped into categories and the categories were iteratively refined.

We applied the two concepts of saturation as described by Aldiabat and Navenec [1], i.e., *code* and *meaning* saturation. We recognized *code* saturation when no new codes emerged from the last few interviews. After this step, the second author audited the categories for coherence. The categories were then revisited and rephrased again. We recognized *meaning* saturation when we understood our themes well enough to see their relationship to existing theories in social psychology (described in Section 4). We then integrated our categories back into existing theories (see Section 5). The three themes described next emerged from this synthesis of our categories with existing research. Tables showing examples for all emerging themes and categories are included in Supplementary Material (see Data Availability). Two categories—"communication" and "staffing problems"—were dropped as out of scope for this paper. The reason they were out of scope was that the participants who mentioned communication and staffing problems discussed them as general issues, not directly related to effective agile leadership.

3 FINDINGS

Figure 1 and Table 2 summarize our findings. Briefly, agile leadership comprises three dimensions: *dynamic team leadership*, *social identity* and *organizational culture*.

In other words, we found that agile teams:

(1) **share leadership work rather than having an individual leader;**
(2) **build a sense of belonging and common purpose across and within organizational functions;**
(3) **adapt their processes to the different cultures which exist simultaneously within a single organization.**

3.1 Dynamic Team Leadership

Ten out of thirteen interviewees highlighted that agile team leadership is only effective when it is shared by the team. All the interviewees saw themselves as an active part of obtaining a shared-leadership dynamic.

3.1.1 Have Team Members that Take Initiative and Responsibility. Interviewees emphasized having a team where everybody could take initiative. One interviewee stated that high-performing individuals do not make good agile team members because teamwork skills are more important than individual performance. Interviewees also described how team members who were "initiators" could enroll others in leadership efforts and help make shared leadership a team norm.

Another interviewee stressed how a single team member could ruin the team's effectiveness due to passiveness and not being able to create an active work-mode. This finding mirrors prior research on dysfunctional teams [19].

Interviewees felt that complaining was not necessarily detrimental to the team's effectiveness as long as the team members felt like they were in charge of making a change for the better. Complaining seems more infectious and destructive if teams do not have autonomy because they tend to complain about factors beyond their control.

One interviewee explained that teams need some sort of leadership, whether or not they are agile, to solve problems effectively. This interviewee saw the positive effects of having all team members be leaders somehow but also stressed that many people in industry are used to having managers that assume responsibility for project failures. Getting people to think differently about leadership as shared was therefore challenging. Interviewees' perspectives on shared leadership are complicated by the inability to divide projects neatly into agile vs. plan-driven—some officially plan-driven projects were described as being quite agile and vice versa.

The interviewees also stressed that there was often a difference in that agile methods have more onerous expectations for all team members doing leadership work than plan-driven methods. One interviewee was skeptical that agile is good for all projects because agile methods assume that team members are willing to accept responsibility, take initiative, and make decisions.

People with a mindset of doing leadership themselves appear to focus more on agile principles and less on implementing user stories as quickly as possible. Teams with such members focused more on being collaborative, self-organized, and seemed more passionate about their work.

Another interviewee attributed the team members' active participation in driving the teamwork to the agile approach to projects. The motivation to lead was therefore not only an a priori property of individual team members but also an emerging team norm. The interviewee emphasized this distinction because it implies that it is possible to activate leadership behavior by setting other group norms [72].

3.1.2 Let Team Members Lead. Having members who want to take initiative is necessary but insufficient. Eight interviewees indicated that more traditional forms of leadership (i.e. managers) surrounding a team could undermine self-leadership. Teams were described as needing to feel heard and autonomy not only for technical or engineering practices but also for team improvements and other organizational issues.

Another key skill that surfaced was a leader's ability to take a step back when leadership can be shared. This was described as being tricky because guidance was often needed early on and it could be difficult to spot when an individual or a team is ready for increased responsibility. One interview explained that, especially when leadership is a formal role, the people in that role need to actively avoid taking over the leadership of the team. Some interviewees believe that strong, formal leadership roles prevent self-organization and shared leadership.

It was sometimes unclear to interviewees whether individual leadership remained because it was needed, or for other
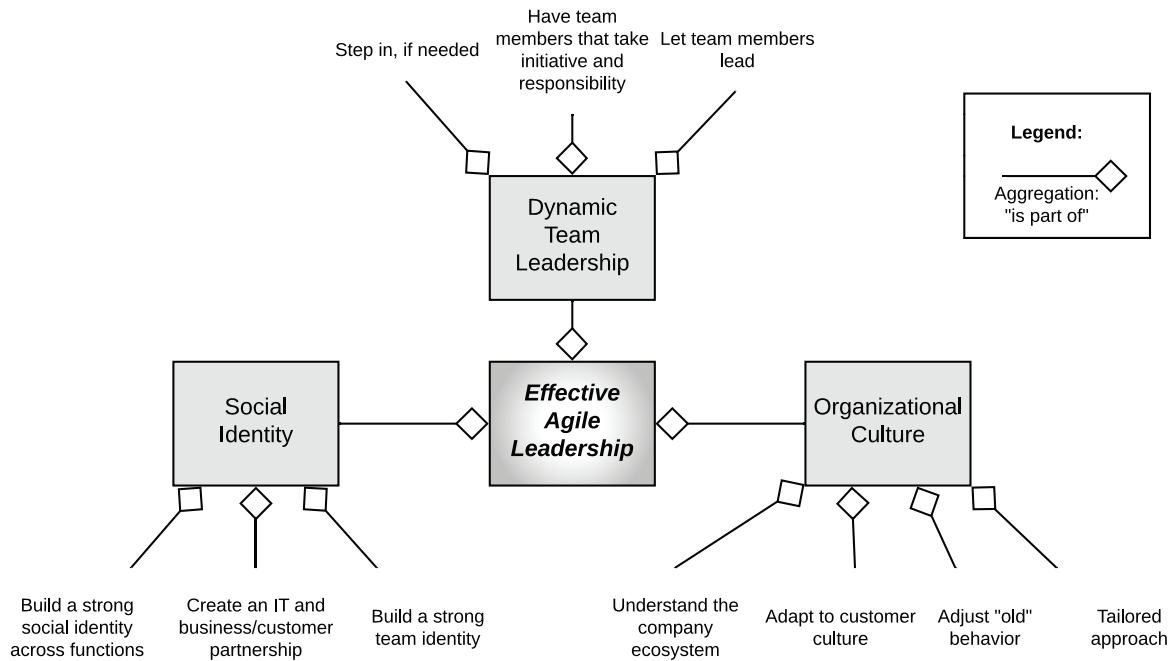
Figure 1: Components of effective leadership in agile software development teams.

reasons. One interviewee described a situation where their team's effectiveness was reduced by the manager's unwillingness to share leadership. Another stressed that leadership, when not shared, can become destructive.

3.1.3 *Step In, If Needed.* Two of the interviewees explicitly stated that, even if they strove for self-organization, an experienced supervisor sometimes needed to intervene. One person described this as being valued by the teams because the ways in which the supervisor could intervene were clear. The other interviewee expressed frustration in having to help teams make better decisions. They felt that the team should be ready to accept responsibility faster, but were not, and therefore tried to help. Having the support of a senior employee was also described as helpful since team members could validate issues with that person before feeding the issues back to the team itself.

3.2 Social Identity

Of the ten organizations where our interviewees work, organizations G, I, and J, embraced agile methods from their founding. In contrast, organizations A–F and H are large companies with a long history of milestone-driven software projects. They were organized around strong functional silos (e.g. marketing, production, software development, accounting & finance). Many participants were therefore leading transitions from a functional organizational hierarchy to a project-based organization, or leading cross-functional agile

teams within companies that were organized around organizational functions.[2]

All of the participants viewed cross-functional teams with high levels of team spirit as advantageous because it helps more employees take responsibility. The participants explained that they obtained higher levels of team spirit, not only for their own work context and field of expertise but also for other important organizational functions (e.g. finance). None of the participants used the term "social identity"; however, they all discussed consolidating, stabilizing, and anchoring different social categories (e.g. business people, developers, testers). A person's social identity is the collection of social categories with which that person identifies [36].

3.2.1 *Build a Strong Social Identity Across Functions.* One of the most highlighted benefits of agile software development is increased team spirit; that is a greater sense of team-belonging and social cohesion. One interviewee explained how, as members come together from different functional backgrounds, a strong sense of team spirit (i.e. feelings of belonging) helps them on-board, integrate with the team and begin contributing. As team membership becomes part of the new team member's social identity, they are more willing to accept responsibility and do leadership work.

Another participant highlighted that a separation of roles within the team decreased team cohesion. They therefore focused on getting team members to realize the value of sharing all aspects of software delivery instead of focusing on

---

[2]Throughout this paper, "functions" refers to organizational functions (e.g. marketing, accounting).

Table 2: Overview of Themes and Categories with Example Quotations

Theme	Category	Examples of supporting quotations
Dynamic team leadership	Have team members that take initiative and responsibility	"Another guy said, 'well you won't have the time, so I'm the Scrum Master now' and that was it . . . they don't see the Scrum Master as a person, they see it as a role that they all can take." (Participant I1) "Teams that are doing it right . . . are really thinking about being collaborative and self-organized and all those things. They become passionate about it and really want to do more of it." (Participant C1)
	Let team members lead	"I guess we were just really lucky with the team members we had. This was basically also my objective; careful! Don't become a project manager, don't start planning stuff, don't start assigning or defining tasks. Stay out of that!" (Participant G3) "There are teams that have a strong leader, and there the problem is another one which is this whole idea of having someone telling everybody what to do. For me, it's the leadership will make it the best or the worst." (Participant J)
	Step in, if needed	"Most of the time we have to interfere as leaders in order to help the team come together and make a better decision." (Participant J1) "I could [force the teams to try a practice]. And I did. . . . And if they don't like it, they're free to go back to the old one." (Participant I1)
Social identity	Build a strong social identity across functions	"A new resource from finance for example didn't used to have much to do with the other functions. Now with these meetings and planning, they are all one team working together and cheer each other's aspects." (Participant F1) "It's very easy to see that, well, 'my part in the project is just coding,' but that's not quite true. They [the team] get less cohesive by having a clear separation of roles." (Participant H)
	Create an IT and business/customer partnership	"The business partnership and engagement you create is absolutely tremendous compared to the traditional approach!" (Participant B1) "So they can again, work to be a part of and share the accountability [in agile]. I mean, for a whole variety of reasons that involvement of the business partner as a part of the team, not them versus us." (Participant D)
	Build a strong team identity	"It's interesting because when we deploy software in the middle of the day, the [people] here actually call the other units and say 'ok, we're stopping the software for a couple of minutes for deploying.' So then everyone stops. We could do a night deploy, but we prefer to do that." (Participant H1) "The Scrum part takes between seven to ten minutes. Then it takes another five to ten minutes to fill out the time, and it's in the spreadsheet, so then I got to the tab that shows the burn down. People wait for that! Like: 'Oooh let's see what the burn-down is today!' And then the third part is just an open forum, an open meeting, and people sort of take care of things that will only take a short amount of time to discuss." (Participant G1)
Organizational culture	Tailored approach	"I have a kind of waterfall-like representation of our sprints and the main milestones and I show that to them. But then I just take care of all the requirements from the organization, you know, at a higher level. The team doesn't have to worry about that stuff." (Participant G1) "What one individual pilot project starts to practice, you always have difficulty scaling that, because that's what they were doing and got really good at doing exactly that specific to that project. But that's not necessarily well translated into practice format across the enterprise. That needs to be more of the fundamentals and the process behind it."(Participant A1)
	Understanding the company ecosystem	"If you haven't brought those people along in your agile journey, then subsequently you will have that hurdle to overcome because of the different culture and behaviors." (Participant D1) "To some people it's a subtle difference and they sometimes have a hard time grasping why we don't just have one list for finance, one list for sales, ops, etc. But it has been a key drive for us since we have shared resources, to have one cross-functional list." (Participant F1)
	Adapt to customer culture	"Often, they don't want changes as often as we could deliver them." (Participant C1) "We used to work mostly with start-up companies, and with start-up companies, that's a big discussion because, with a tight budget, why would you have 17 layers of tests?" (Participant J1)
	Adjust "old" behavior	"Allowing the team to become a high performance team without the problem of having a lot of politics involved in the execution of the project." (Participant E2) "Waterfall approach provides some very familiar and comfortable handrails, called milestones, which everybody grasps and clings to for life." (Participant G1)

'

their individual expertise. By sharing in different tasks, team members expanded their social identity to include belonging to a team that was defined by its *goal*, rather than its *roles*.

3.2.2 Create an IT and Business/Customer Partnership. Moreso than previous approaches, agile software development focuses on social identities. Transitioning to agile necessitates defining one's work identity based on social categories other than roles. In other words, agilists are supposed to identify with their cross-functional team instead of, or at least in addition to, their role. For example, employees should say 'I'm part of the *Mass Effect* team,' or 'I'm on project Urithiru' not

'I'm a front-end developer' or 'I'm a requirements analyst'. Interviewees consistently described how their teams obtained a common social identity by sitting and working together.

Participants emphasized how leaders expanded their team members' responsibilities by communicating who was responsible for what. One participant described attempting to enroll customers into the team's shared identity. They felt that projects in which customers and developers felt like they were on the same team were more successful.

Two participants described cases when they did not succeed in creating a common social identity with business people or other external customers. Such situations were described as

leading to team members not feeling prioritized or sharing accountability for the work being done.

3.2.3 Build a Strong Team Identity.
Social identity building within the teams was also described as important. All practitioners highlighted the importance of the social ceremonies (often taken from agile ceremonies like a retrospective meeting or daily stand-ups [67]) in creating a shared social category that became a part of the individuals' social identity. One participant described a ceremony that was possible to conduct more efficiently, but the team preferred deploying code in the middle of the day so that everyone could get together and get an important sense of belonging in the team.

Building a strong and common social identity was also described as being easier for co-located teams, which is consistent with prior research [44]. Two of the participants described putting extra effort into online ceremonies to engender a sense of belonging in their distributed teams.

3.3 Organizational Culture

Many large companies do not really want an agile transition. They just want to increase speed and productivity by adopting some elements of agile while retaining their functional silos and tyrannical power hierarchies. Agile leaders, not only within these large companies but also within smaller organizations beholden to them, therefore have to balance conflicting organizational cultures—the new agile culture and the old, milestone-driven culture. This balancing involves tailoring the agile approach, understanding the company ecosystem, adapting to customer culture, and adjusting "old" behaviors. All of the participants repeatedly contrasted effective leadership in an agile context to other leadership behavior they had also observed or experienced.

3.3.1 Tailored Approach.
One common topic was the difficulty of scaling up small agile pilot projects to other contexts, even within the same company. The larger companies tended to want a standard set of agile practices, but participants reported that the diversity of contexts hinders a standardized approach. Some participants manage this conflict between diversity and standardization by focusing on the agile principles while tailoring the practices for each, or a few, teams.

One strategy described by three participants was to translate the agile principles into the company's context rather than rigidly implementing agile practices and rituals. One participant described tweaking the waterfall model to make it more incremental as a first step toward agile. Another interviewee claimed that agile was not appropriate for some phases in their development—it simply did not fit where the company was at the time of the interview. Returning to the idea of balancing opposing cultures, another participant suggested that agile leaders should shield the team from the control needs of their organizations so that the team can focus on delivering value.

3.3.2 Understand the Company Ecosystem.
With large-scale agile, one participant highlighted the issues with having only one product owner for each team since that person struggled to set priorities for the teams since that person alone could not balance many different functions (e.g. stakeholders, finance, line management). They instead focused on the function and purpose of product ownership, and created a business council where up to seven people from different organizational functions set the priorities together. One representative from each department (e.g. one from finance, one from marketing, one from operations, etc.) met every three weeks to review all projects and their previous priorities. The council then created one single list of company priorities across the organizational functions and, through the creation of that list, removed silos and forced collaboration across the company. This appeared to improve consensus that IT was working on the most important projects for the company.

Another participant emphasized that some companies they worked with as customers did not understand that transitioning to agile development entails a cultural change. Balancing two conflicting cultures was described as very important within the participant's own organization, but also to customize teams for customer culture. This internal adaptation required much effort because customers that did not incorporate and focused on agile principles did not obtain as much added values as the ones that did, which was described as frustrating but necessary.

Participants not only had to see to the teams' need to contextualize but also to balance other functions within the same company, all of them with a different culture. One example being that a development team could have an agile financial department integrated into their project so that they could make rapid changes to what was being built, while others had a more milestone driven external financial department hindering their agility.

3.3.3 Adapt to Customer Culture.
All participants discussed balancing their internal organizational cultures with their customers' organizational cultures. One participant claimed that their team could deliver updates much more often than the customer could handle them. Therefore, the participant felt that they had to be less agile to adapt to their customers. (The implication that agility is determined by release schedule is questionable [81] but many participants seem to think agile means delivering fast and often, rather than being responsive to change).

One participant said that agile teams and their stakeholders meet and synchronize their expectations instead of specifying requirements and then controlling delivery. They felt that this synchronization nurtured more discipline and commitment in stakeholders.

Another cultural tension concerned contracting. One participant complained that traditional fixed-price contracts hampered agile ways of working, which is consistent with previous research [e.g. 41]. The contracts instead defined what type of collaboration was possible with the customers, and partly determined how the two different cultures could be balanced. What this participant described was contracting based on the agile principles, i.e. fixed time, fixed cost, but

flexible scope. They also had fully dedicated teams for each project and avoided developers in multiple teams.

3.3.4 Adjust "Old" Behavior. The dialectic tension between the old, milestone-driven culture and the new agile culture, including power, values, and principles of working, exists at both the organizations individual levels. All participants described old habits and old ways of working resurfacing continuously in the new agile environments. Agile leaders therefore felt that they needed to guard new work principles and continuously adjust deviating behaviors.

One participant explained that they were so used to the old ways of working that many employees struggled to follow the reasoning behind the new ways of working. They described partially losing control of their project and having to let go of aspects of control that used to be key for them to know how to do their job.

All participants described working hard to change behavior. One participant claimed that changing the behavior of people in their organization was the most challenging part of creating a more customer value driven company.

All interviewees mentioned how effective leadership meant letting go of control and reporting, while trusting teams and focusing on incremental delivery of customer value. Participants reported varying results with about half of their companies having reasonably successful translations while the other half were still struggling.

3.4 Summary

In summary, participants identified three broad capabilities necessary for effective leadership in modern software development:

(1) The ability to share responsibility and leadership work with other team members.
(2) The ability to get team members from different functional silos to identify as members of a cross-functional team.
(3) The ability to manage ongoing tension between the competing logics of the old, milestone-driven culture and the new agile culture.

4 RELATED WORK

This section summarizes existing literature on leadership in software engineering and social psychology. Then, in Section 5, we will integrate the findings from Section 3 with the existing theory and research reviewed in this section.

4.1 Research on Leadership in Software Engineering

In software development, *agility* has at least three meanings:

(1) the ability to act and react quickly and easily;
(2) adherence to the principles laid out in the agile manifesto [15];
(3) the extent to which a team implements the practices associated with one or more ostensibly agile methods.

Some agilists believe that agile principles must be implemented using specific agile practices tailored to their own context [80]. However, instead of providing empirical evidence that these practices increase a team's ability to act quickly and easily, proponents of agile simply redefined *agility* as adhering to their methods [81].

Even if we accept that adopting ostensibly agile practices increases a team's agility, neither the agile manifesto nor popular books espousing agile practices [e.g 10, 63] address leadership in-depth or explain how what type of leadership behavior is needed over time. Scrum, in particular, promotes a facilitating role—the "Scrum Master"—and does not acknowledge the necessity of adapting one's leadership style to one's social context [64]. In reality, many companies use a mixture of methods and adopt agile methods incrementally [44].

Most leadership research in software engineering views leadership as an individual role [e.g. 22, 43]). Studies also show how some agile roles include leadership behavior, e.g., Scrum Masters [70] and agile coaches [6].

Some studies investigate specific leadership styles. For example, Veiseh et al. [76] showed that agile leaders are more transformational. Garcia et al. [22] found that the effectiveness of different leadership styles are independent of the software development process (e.g. agile, waterfall). Other studies have investigated how leadership styles change as teams mature [25, 27]. What leadership behavior is needed depends on individual readiness [29] and the group's development stage [79]. Leadership roles typically become more distributed as a team matures [68, 69].

Andrias et al. [3] found three leadership styles—Tigers, Cranes, and Elephants—in Information Systems Development projects. Przybilla et al. [56, 57] suggests a research plan and a theoretical model where they want to investigate individual traits (represented by personality traits and cognitive ability) and the perceived leadership distribution to explain self-organization. They hypothesize that many individuals would be perceived as leaders in agile teams. Meanwhile, Moe et al. [54] argue that agile teams should be trained in leadership. The same authors later studied challenges of shared decision-making in agile teams, including "alignment of strategic product plans with iteration plans, allocation of development resources, and performing development and maintenance tasks in teams" [52].

We did not find any software engineering studies that investigate social identity (see Section 4.2) and leadership together. However, developers' social identities play a key role in the effectiveness of software projects [5].

In contrast, much research investigates agile teams' self-organization and self-management [e.g. 31, 32, 53, 60]. Furthermore, Kakar [42] showed that agile teams have higher degrees of self-organization than teams in a more traditional team setup. We extends these works by differentiating leadership from management and organization. Only one study has investigated explicit leadership aspects of getting teams to self-organize [26], but from the perspective of an appointed agile leader.

As early as 2003, scholars argued for including cultural aspects in software process improvement efforts [55]. Later studies confirmed that considering culture improves outcomes in software process improvement [66]. Others argued that social norms, values and beliefs are all important when deploying systems development methodologies in general [39] and agile software development in particular [71]. Our theme of participants balancing conflicting cultures resonates with these prior studies.

4.2 Research on Leadership in Social Psychology

Early (19$^{\text{th}}$ century) conceptions of leadership were dominated by the *Great Man Theory* [34].[3] It posits that the success of nations, organizations, and endeavors is largely determined by leaders—amazing individuals who changed the world using inherent talent for leading. This theory is no longer taken seriously because both of its premises are false: leadership is a learnable set of skills rather than an innate talent [14], and that the fate of nations is determined by multitudinous factors rather than individuals. Criticism of the *Great Man Theory* is not new; in 1869 Tolstoy wrote: "To elicit the laws of history we must leave aside kings, ministers and generals, and select for study the homogeneous, infinitesimal elements which influence the masses" [74, p. 977].

Hogg et al. [35] categorize 20$^{\text{th}}$-century leadership research into seven different focus areas:

(1) *Personality traits and individual differences* (including the Great Man Theory): a body of research that views leadership as "innate or acquired individual characteristics" [35, p. 316].

(2) *Situational perspectives*: a body of research that views leadership as a product of a context, which is considered oversimplified because the characteristics of individual leaders can also matter [35].

(3) *What leaders do*: a body of research that investigates the actual behaviors of leaders and typically divides leaders into task-focused or relationship-focused [35]. This is problematic because it ignores the interplay between leader behavior and context.

(4) *Contingency theories*: a body of research that connects situational and behavioral factors. The main problem with these theories is that they are rather static [35] and only focus on one specific leadership behavior in one specific context at a time.

(5) *Transactional leadership*: a body of research that focuses on "the transaction of resources between the leader and the followers" [16]. These ideas were extended to the leader-member exchange (LMX) theory [23], which posits that leader-follower groups proceed through a series of phases similar to Tuckman's stages of group development [75]. Transactional leadership theories are criticized for failing to appreciate that leadership is a group process [33].

(6) *Transformational leadership*: a body of research that focuses on how leaders can transform goals into action using charisma. This work posits that leaders should offer employees an identity within the organization such that employees take personal responsibility for the success of the company and see company failures as their own [8]. Transformational leadership improves follower performance at the individual, team and organizational levels [78]. However, immoral charismatic leaders may use power only for personal gain or to promote their own personal vision, censure critical or opposing views, demand unquestioning acceptance of their decisions, use one-way communicatio, and ignore followers' needs [38].

(7) *Leader perceptions*: a body of research built on *social cognition*, a branch of cognitive psychology that focuses on how people process, store, and apply information about other people and social situations [35]. For instance, Leader Categorization Theory posits that our perception of leadership influences our selection and endorsement of leaders [48].

In summary, the first three areas do not have much explanatory power [14]. Transactional leadership has merit, while transformational leadership is quite effective but open to abuse. Understanding perceptions of leadership also helps to explain who people choose to endorse as leaders.

Most of the leadership studies in software engineering fit into one or more of Hogg et al.'s categories; for example, leadership as an individual role [22, 43], or the style of being a transformational leader [76]. However, leadership research in the 21$^{\text{st}}$ century has taken new directions. As we describe next, leadership is now seen as dynamic and integrated into the social process of group formation.

4.3 Dynamic Team Leadership

Recent studies of team effectiveness suggest that appointed leaders of complex projects create, maintain, and refine collective systems of meaning and expectation, which, in turn, guide the behavior of individuals [45]. This different perspective of leadership entails less control and direction and more of the leader's "developmental and instructional capabilities applied over time" [45].

Members of effective teams share leadership dynamically [46]; that is, the distribution of leadership work among team members is constantly in flux. The team constantly adjusts (based on their circumstances) what leadership work is being done, who is doing it and how they are doing it. Ford [20] suggests replacing the term *leadership* with *leadingship* to capture the fact that it is a shared property of the team rather than a property of designated "leaders".

These ideas have existed in parallel with some of the previous leadership theories described above—for example, Cranach [77] suggested that leadership is a function of group action in 1986.

In summary, leadership is a kind of work—a shared function of driving the work forward toward the team's goal—which teams share. Teams constantly adjust their shared leadership work to their changing circumstances.

---

[3]We preserve the original, gender-specific name here because it highlights how out-dated this theory is.

4.4 Social Identity Theory

Above, we described Hogg et al.'s seven *historical* areas of leadership research. Hogg et al.'s point, in categorizing leadership research thusly, is to contrast these historical views with a contemporary, *Social Identity Theory* of leadership [34]. Whereas all seven historical areas assume that leadership is an individual *role*; Social Identity Theory models leadership as a *relationship* where group members can influence each other to adopt (and perceive as their own) new values, attitudes and goals.

A person's *social identity* is the part of their sense of self that derives from group membership; that is their self-categorization into different social groups [36]. Effective leadership transforms individual action into group action that defines group membership. Leadership is an *identity function*: it consolidates, stabilizes, and anchors the group's identity [34]. For example, the congruence between the social identities of leaders and followers leads to higher ratings of transformational leadership [4].

Social identities also play an important role in how people collaborate. To collaborate effectively, teams need to create a common social category. Team members do *not* have to abandon their prior social categories to identify with the team; rather, they create a new social category for the cross-functional team [18]. For example, if some members of a software development team are also part of group that promotes DevOps across the organization; they need not suppress that part of their social identity to feel like part of the software team as well. Furthermore, the new social category does not require commonalities beyond team membership to take hold. For example, the team members can all have different hobbies, nationalities, belief systems or interests.

However, the more a person identifies with one social category—be it a role or a team—the more they exhibit inter-group bias; that is, favoring the groups we are part of (the "in-group") over others ("the out-group") [30]. Inter-group bias has both positive and negative effects; for example, inter-group bias can magnify team member's motivation, but it can also lead employees to put their team's best interests ahead of the organization's [30].

4.5 Organizational Culture

Organizational culture refers to "the pattern of beliefs, values and learned ways of coping with experience that have developed during the course of an organization's history, and which tend to be manifested in its material arrangements and in the behaviours of its members" [12]. Culture is widely-studied in many fields including anthropology, management, political science, and sociology. For the purposes of this paper, however, the key point is that leadership (and social identity) are inseparable from culture [62].

Organizational culture is "a particularistic system of symbols shaped by ambient society and the organization's history, leadership and contingencies, differentially shared, used and modified by actors in the course of acting and making sense out of organizational events" [2, p. 216]. Rather than a single, monolithic culture, organizations can have a mishmash of conflicting cultural views [24]. Any nuanced understanding of leadership in agile teams therefore must be situated in the organization's ongoing cultural conflict.

4.6 Summary

In summary, leadership research has evolved considerably since the 19th century. Three recent trends in leadership research help explain leadership in agile software development:

- social identity helps explain how leaders get individuals to feel like part of a team and accept responsibility;
- dynamic team leadership helps explain how leadership work is shared among team members;
- the idea that leadership is inextricable from culture helps explain how leaders balance conflicting cultures associated with agile and milestone-driven processes.

More broadly, increased focus on people and their interactions is a core dimension of transitioning to, and maintaining, agile software development. Leadership is part of this increased focus on human factors. Next, we will further integrate our findings back into recent leadership research.

5 DISCUSSION

The main contributions of this paper—the concepts illustrated by Figure 1—resonate well with recent studies on leadership in social psychology, most of which have not previously been applied to the software engineering context.

5.1 Dynamic Team Leadership

Our findings suggest that leadership in cross-functional agile teams is dynamically shared: many team members accept responsibility, take initiative, do leadership work; and furthermore, who does what leadership work changes over time. This resonates with research on dynamic team leadership, which has found that effective leaders develop and guide team members instead of controlling or directing them [45]. Agile teams appear to need leadership that consistently adapts to their evolving situation; agile team members need to actively drive teamwork.

Our results suggest that self-appointed agile leaders let other team members take on leadership work when they (the others) are ready to do so. Not allowing team members to take on leadership work and drive the teamwork forward appears to inhibit the development of the desired agile team dynamic [46]. Having traditional managers in or around teams can therefore prevent them from becoming agile.

Agile methods frequently prescribe self-managed teams run without interference from external leaders. However, both our findings and existing research on dynamic team leadership suggest that external or appointed leaders creating, maintaining, and refining expectations for the team can be beneficial [45].

5.2 Social Identity

Our findings suggest that building a strong social identity across functions, such that employees identify with their new team in addition to their old role, is important for effective agile leadership. Initially, participants added to their colleagues' identities by simply designing cross-functional teams. Having team members from different functions collaborating was described as one key to breaking the silos and focusing on increased customer value.

Our results are consistent with prior findings in psychology that designing teams is a kind of leadership behavior [34]. The positive effects of cross-functional teams are also well known and include increased organizational learning, job satisfaction, and overall effectiveness [17]. The agile leaders we interviewed described creating a sense of belonging and shared responsibility not only within teams but also with customers and other organizational functions.

Our interviewees' emphasis on building a strong team identity (i.e. clearly defining a social category for the team) is consistent with the main premise of social identity theory: our membership in groups is a major part of who we are and how we fit into our social world. Our interviewees' descriptions of how employees identify with different roles and teams are consistent with the idea that an individual's social identity comprises many social categories [30, 36].

Furthermore, synthesizing prior literature with our findings suggests the following recommendation. New team members do not have to abandon their prior social categories to identify with their new team. Instead of trying to remove a person's prior social category (e.g. a software tester), we simply add the cross-functional team identity (e.g. "I'm a member of team Eta"). The different categories do not interfere with each other, and trying to suppress aspects of a person's social identity is problematic. Moreover, developing employees' social identities can decrease inter-group bias and make collaboration more effective across the organization [e.g. 17].

5.3 Organizational Culture

Like prior software engineering research on culture [40, 73], our findings suggest that transitioning to agile software development is inextricable from sifting organizational cultures. Realizing the benefits of agile software development requires more than just implementing agile practices and process improvements—the organizational culture has to change from one of individual leaders commanding and controlling to one of decentralized, dynamic team leadership. That is why our participants focused on implementing agile principles, tailored to their specific contexts, rather than inflexibly rolling out a standard set of agile practices.

Furthermore, the agile leaders saw themselves as shields between the new agile cultures of their teams and the old, milestone and function-oriented cultures in their surrounding company ecosystems. This shielding resonates with prior psychological literature on how understanding organizational culture is essential to making sense of the organization's history, its leadership, and events [2]. Even the participants

from the smaller companies contrasted effective leadership in agile teams to other leadership behavior.

Consistent with prior research on agile transitions [40, 73], our results show that agile leaders perform a cultural balancing act across the organizational functions to facilitate a transition toward agile processes. More general research on organizational culture similarly shows that a multitude of organizational cultures existing simultaneously within large organizations. Multicultural models are needed because large organizations have problems of "cross-cultural" conflict [24]. When a company adopts agile, the new agile culture must therefore be balanced with existing, and sometimes conflicting, cultures. For example, some of our participants described adapted to their customers' culture by decreasing their release frequency.

5.4 Answering our Research Question

This study sought to address the question, what is the nature of leadership in agile software development, from the perspective of professionals who identify as agile leaders? Put simply, our answer is as follows.

> Agile leadership is dynamically shared among team members, engenders a sense of belonging to the cross-functional team, is inextricable from organizational culture, and often requires balancing competing organizational cultures.

5.5 Implications

The above findings have several implications for researchers, professionals and educators.

Future research on leadership in software engineering should be more connected to recent research in social psychology and organizational behavior. "What makes a good agile leader?" is the wrong question because effective agile teams *share* leadership work. Counter-intuitively, to understand "what to look for in a great software engineering manager, and how to further develop their skills to support the teams they manage" [43], we need to conceptualize leadership as a property of a team, rather than an individual. Once we understand that leadership is a kind of work, rather than a kind of person, we can recognize that the effectiveness of software engineering managers depends on their ability to (1) get employees to identify as part of team, (2) get team members to accept greater responsibility, and (3) manage cultural conflict. Indeed, transitioning to agile development may increase the importance of these aspects of leadership.

The proposed model explains that, as organizations gradually shift to agile, they must balance competing cultures, including conflicts between teams, within teams, and even within individuals. Practitioners may benefit from collaborating with academics on this because theories of leadership are challenging to digest and apply.

Meanwhile, educators may find the proposed model useful for teaching agile project management—a problematic subject for many software engineering educators [cf. 58]—especially at the graduate level. The model can be used both to explain to students the relationship between management

and leadership in self-managed, self-organized software teams and to frame class discussion of taking on leadership work and responsibilities.

6 LIMITATIONS

Being purely qualitative, we examine this study's limitations through common qualitative categories including Credibility, Resonance, Originality, Usefulness and Transferability [13, 47].

Credibility is the degree to which findings are demonstrably grounded in observation. We provide detailed analysis tables (see Data Availability showing how each theme and category is grounded in specific interviewee statements.

Resonance is the degree to which findings make sense to participants. We assessed resonance using *dialogical interviewing*, a technique in which, during interviews, the interviewer reformulates the interviewee's statements in the interviewer's words to synchronize their interpretations and co-construct meaning [28]. This technique was used throughout the interviews as interviewee expressed something that seemed important to them. It lead to many small improvements and revisions to our understanding of the interviewees' perspectives, which filter into the emerging themes in many small ways.

Originality is the degree to which the study produces novel concepts. While the themes that emerge from this study map into existing theories in social psychology, this is the first study to show that leadership is present in, and critical for, self-managed agile teams. Although dynamic team leadership and social identity are widely researched in social psychology, this is the first study that demonstrates that these theories apply to, and are useful for, understanding software teams.

Usefulness is the degree to which a study's results can be applied by someone (e.g. practitioners, researchers). Section 5.5 describes the potential uses of our findings by researchers, professionals and educators. The conceptual transition from leadership as a role to leadership as a shareable form of work has the potential to spawn a new direction in research, education, and practice.

Transferability is the degree to which other researchers could apply a study's proposed concepts in different contexts. Transferability is an alternative to generalizability for qualitative studies that do not statistically generalize from a sample to a population. The mapping of our findings into prior theories, which apply across diverse contexts, suggests high transferability. However, agile leaders are scarcer than developers and much scarcer than consumers, so we were not able to find as many participants as we would have liked. We attempted to mitigate bias among participants by recruiting a diverse group in terms of gender, race, nationality, job description, company size, and industry. Future research could improve confidence in the transferability of results by investigating agile leadership from the perspective of professionals who do *not* identify as agile leaders, and professionals from a wider variety of organizational contexts.

Moreover, the analysis may be colored by the first author's experiences with agile leadership at Volvo Cars; specifically, the first author's practical experience with agile culture clashes may have affected the coding of the data. We attempted to mitigate this threat by having the second author audit the coding. Similarly, both of the authors have backgrounds in social psychology, which could affect the degree to which our findings converge with existing psychological theory in ways we can neither perceive nor mitigate.

7 CONCLUSIONS AND FUTURE WORK

This study was motivated by the three realizations: (1) leadership is crucial for projects and organizations, (2) little research has investigated how leadership works in self-managed agile teams, and (3) software engineering research has not embraced recent advances in psychological research on leadership. We therefore conducted a qualitative survey of thirteen software professionals who identify as agile leaders. Our main contribution is the model of agile leadership shown in Figure 1. Briefly, the model suggests that: (1) leadership is a kind of work rather than a job role; (2) self-organized/self-managed teams dynamically share leadership work among team members; (3) employees are more willing to accept leadership work and responsibility when they identify as belonging to their teams; and (4) effective leadership often involves balancing conflicting cultures (e.g. agile vs. milestone-driven).

This qualitative study aims to explain rather than predict. Future work could quantitatively assess the antecedents and consequences of effective leadership using validated scales for concepts including social identity [49], or explore under-researched aspects of leadership such as *prototypicality* (the idea that more typical team members are better at leadership work [34]) and inter-group bias (favoritism toward one's teammates [30]) in software teams.

Software engineering researchers need to get better at integrating theories and findings from reference disciplines and collaborating directly with scientists from other disciplines. While cross-disciplinary research is intrinsically challenging, software engineering is intrinsically interdisciplinary. Therefore, only by appreciating research from other disciplines can we effectively understand, explain, predict and prescribe software engineering phenomena.

REFERENCES

[1] Khaldoun M Aldiabat and Carole-Lynne Le Navenec. 2018. Data saturation: The mysterious step in grounded theory methodology. *The Qualitative Report* 23, 1 (2018), 245–261.

[2] Yvan Allaire and Mihaela E Firsirotu. 1984. Theories of organizational culture. *Organization studies* 5, 3 (1984), 193–226.

[3] Mone Stepanus Andrias, Sabine Matook, and Richard Vidgen. 2018. Towards a typology of agile ISD leadership. In *Research-in-Progress Papers, Proceedings of the European Conference on Information Systems*, Vol. 28. AIS.

[4] Bruce J Avolio, Fong T Keng-Highberger, Robert G Lord, Sean T Hannah, John M Schaubroeck, and Steve WJ Kozlowski. 2020. How leader and follower prototypical and antitypical attributes influence ratings of transformational leadership in an extreme context. *Human Relations* (2020). DOI:http://dx.doi.org/10.1177/0018726720958040

[5] Andreas Bäckevik, Erik Tholén, and Lucas Gren. 2019. Social identity in software development. In *Proceedings of the 12th International Workshop on Cooperative and Human Aspects of Software Engineering*. IEEE Press, 107–114.

[6] Gisela Bäcklander. 2019. Doing complexity leadership theory: How agile coaches at Spotify practise enabling leadership. *Creativity and Innovation Management* 28, 1 (2019), 42–60.

[7] Sebastian Baltes and Paul Ralph. in press. Sampling in software engineering research: A critical review and guidelines. *Empirical Software Engineering* (in press).

[8] Bernard M Bass and Ronald E Riggio. 2006. *Transformational leadership*. Psychology press.

[9] Kent Beck. 1999. Embracing change with extreme programming. *IEEE Computer* 32, 10 (1999), 70–77.

[10] Kent Beck. 2000. *Extreme programming explained: embrace change*. Addison-Wesley Professional.

[11] Virginia Braun and Victoria Clarke. 2006. Using thematic analysis in psychology. *Qualitative research in psychology* 3, 2 (2006), 77–101.

[12] A.D. Brown. 1998. *Organisational Culture* (2nd ed.). Financial Times.

[13] Kathy Charmaz. 2014. *Constructing grounded theory*. sage.

[14] Martin M Chemers. 2000. Leadership research and theory: A functional integration. *Group Dynamics: Theory, research, and practice* 4, 1 (2000), 27.

[15] David Cohen, Mikael Lindvall, and Patricia Costa. 2004. An introduction to agile methods. *Advances in Computers* 62 (2004), 1–66.

[16] David De Cremer and Eric Van Dijk. 2005. When and why leaders put themselves first: Leader behaviour in resource allocations as a function of feeling entitled. *European Journal of Social Psychology* 35, 4 (2005), 553–563.

[17] Daniel R Denison, Stuart L Hart, and Joel A Kahn. 1996. From chimneys to cross-functional teams: Developing and validating a diagnostic model. *Academy of management journal* 39, 4 (1996), 1005–1023.

[18] John F Dovidio, Samuel L Gaertner, and Ana Validzic. 1998. Intergroup bias: status, differentiation, and a common in-group identity. *Journal of personality and social psychology* 75, 1 (1998), 109–120.

[19] Will Felps, Terence R Mitchell, and Eliza Byington. 2006. How, when, and why bad apples spoil the barrel: Negative group members and dysfunctional groups. *Research in organizational behavior* 27 (2006), 175–222.

[20] Randal Ford. 2010. Complex adaptive leading-ship and open-processional change processes. *Leadership & Organization Development Journal* 31, 5 (2010), 420–435.

[21] M. Fowler and J. Highsmith. 2001. The Agile Manifesto. In Software Development, Issue on Agile Methodologies, last accessed on December 29th, 2006. (Aug. 2001).

[22] Fernando Andre Zemuner Garcia and Rosária de Fatima Segger Macri Russoi. 2019. Leadership and Performance of the Software Development Team: Influence of the Type of Project Management. *Revista Brasileira de Gestão de Negócios* 21, 5 (2019), 970–1005.

[23] George B Graen and Mary Uhl-Bien. 1995. Relationship-based approach to leadership: Development of leader-member exchange (LMX) theory of leadership over 25 years: Applying a multi-level multi-domain perspective. *The leadership quarterly* 6, 2 (1995), 219–247.

[24] Kathleen L Gregory. 1983. Native-view paradigms: Multiple cultures and culture conflicts in organizations. *Administrative science quarterly* 28, 3 (1983), 359–376.

[25] Lucas Gren, Alfredo Goldman, and Christian Jacobsson. 2020. Agile ways of working: A team maturity perspective. *Journal of Software: Evolution and Process* 32, 6 (2020), e2244.

[26] Lucas Gren and Magdalena Lindman. 2020. What an Agile Leader Does: The Group Dynamics Perspective. In *International Conference on Agile Software Development*. Springer, 178–194.

[27] L Gren, R Torkar, and R Feldt. 2017. Group development and group maturity when building agile teams: A qualitative and quantitative investigation at eight large companies. *The Journal of Systems and Software* 124 (2017), 104—–119. DOI:http://dx.doi.org/10.1016/j.jss.2016.11.024

[28] Lou Harvey. 2015. Beyond member-checking: A dialogic approach to the research interview. *International Journal of Research & Method in Education* 38, 1 (2015), 23–38.

[29] Paul Hersey, Kenneth H Blanchard, and Walter E Natemeyer. 1979. Situational leadership, perception, and the impact of power. *Group & Organization Management* 4, 4 (1979), 418–428.

[30] Miles Hewstone, Mark Rubin, and Hazel Willis. 2002. Intergroup bias. *Annual review of psychology* 53, 1 (2002), 575–604.

[31] Rashina Hoda, James Noble, and Stuart Marshall. 2012. Self-organizing roles on agile software development teams. *IEEE Transactions on Software Engineering* 39, 3 (2012), 422–444.

[32] Rashina Hoda, James Noble, and Stuart Marshall. 2013. Self-organizing roles on agile software development teams. *IEEE Transactions on Software Engineering* 39, 3 (2013), 422–444.

[33] Michael A Hogg, Robin Martin, Olga Epitropaki, Aditi Mankad, Alicia Svensson, and Karen Weeden. 2005. Effective leadership in salient groups: Revisiting leader-member exchange theory from the perspective of the social identity theory of leadership. *Personality and Social Psychology Bulletin* 31, 7 (2005), 991–1004.

[34] Michael A Hogg and Daan van Knippenberg. 2003. Social identity and leadership processes in groups. *Advances in Experimental Social Psychology* 35 (2003), 1–52.

[35] Michael A. Hogg and Graham M. Vaughan. 2014. *Social Psychology* (7 ed.). Pearson, Harlow, England.

[36] Matthew J Hornsey. 2008. Social identity theory and self-categorization theory: A historical review. *Social and Personality Psychology Compass* 2, 1 (2008), 204–222.

[37] Robert J House and Ram N Aditya. 1997. The social scientific study of leadership: Quo vadis? *Journal of management* 23, 3 (1997), 409–473.

[38] Jane M Howell and Bruce J Avolio. 1992. The ethics of charismatic leadership: submission or liberation? *Academy of Management Perspectives* 6, 2 (1992), 43–54.

[39] Juhani Iivari and Magda Huisman. 2007. The relationship between organizational culture and the deployment of systems development methodologies. *Mis Quarterly* 31, 1 (2007), 35–58.

[40] J Iivari and N Iivari. 2011. The relationship between organizational culture and the deployment of agile methods. *Information and Software Technology* 53, 5 (2011), 509–520.

[41] Magne Jørgensen, Parastoo Mohagheghi, and Stein Grimstad. 2017. Direct and indirect connections between type of contract and software project outcome. *International Journal of Project Management* 35, 8 (2017), 1573–1586.

[42] Adarsh Kumar Kakar. 2017. Assessing self-organization in Agile software development teams. *Journal of computer information systems* 57, 3 (2017), 208–217.

[43] Eirini Kalliamvakou, Christian Bird, Thomas Zimmermann, Andrew Begel, Robert DeLine, and Daniel M German. 2019. What Makes a Great Manager of Software Engineers? *IEEE Transactions on Software Engineering* 45, 1 (2019), 87–106. DOI:http://dx.doi.org/10.1109/TSE.2017.2768368

[44] Kirsi Korhonen. 2013. Evaluating the impact of an agile transformation: a longitudinal case study in a distributed context. *Software Quality Journal* 21, 4 (2013), 599–624.

[45] Steve WJ Kozlowski and Daniel R Ilgen. 2006. Enhancing the effectiveness of work groups and teams. *Psychological science in the public interest* 7, 3 (2006), 77–124.

[46] Steve WJ Kozlowski, Daniel J Watola, Jaclyn M Jensen, Brian H Kim, and Isabel C Botero. 2009. Developing adaptive teams: A theory of dynamic team leadership. In *Team effectiveness in complex organizations: Cross-disciplinary perspectives and approaches*, Eduardo. Salas, Gerald F. Goodwin, and C. Shawn. Burke (Eds.). Routledge, New York, US, 113–155.

[47] Yvonna S. Lincoln and Egon G. Guba. 1985. *Naturalistic Inquiry.* Sage.
[48] Robert G Lord, Douglas J Brown, Jennifer L Harvey, and Rosalie J Hall. 2001. Contextual constraints on prototype generation and their multilevel consequences for leadership perceptions. *The Leadership Quarterly* 12, 3 (2001), 311–338.
[49] Riia Luhtanen and Jennifer Crocker. 1992. A collective self-esteem scale: Self-evaluation of one's social identity. *Personality and social psychology bulletin* 18, 3 (1992), 302–318.
[50] Sunila Modi and Diane Strode. 2020. Leadership in Agile Software Development: A Systematic Literature Review. In *ACIS 2020 Proceedings.* Springer, 55.
[51] N Moe, T Dingsøyr, and T Dybå. 2009. Overcoming barriers to self-management in software teams. *IEEE Software* 26, 6 (2009), 20–26.
[52] Nils Brede Moe, Aybüke Aurum, and Tore Dybå. 2012. Challenges of shared decision-making: A multiple case study of agile software development. *Information and Software Technology* 54, 8 (2012), 853–865.
[53] Nils Brede Moe, Torgeir Dingsøyr, and Tore Dybå. 2008. Understanding self-organizing teams in agile software development. In *19th Australian Conference on Software Engineering (aswec 2008).* IEEE, 76–85.
[54] Nils Brede Moe, T Dingsyr, and O Kvangardsnes. 2009. Understanding shared leadership in agile development: A case study. In *2009 42nd Hawaii International Conference on System Sciences.* IEEE, 1–10.
[55] Ojelanki Ngwenyama and Peter Axel Nielsen. 2003. Competing values in software process improvement: an assumption analysis of CMM from an organizational culture perspective. *IEEE Transactions on Engineering Management* 50, 1 (2003), 100–112.
[56] Leonard Przybilla, Alexander Präg, Manuel Wiesche, and Helmut Krcmar. 2020. A Conceptual Model of Antecedents of Emergent Leadership in Agile Teams. In *Proceedings of the 2020 Computers and People Research Conference.* 164–165.
[57] Leonard Przybilla, Manuel Wiesche, and Helmut Krcmar. 2019. Emergent Leadership in Agile Teams–an Initial Exploration. In *Proceedings of the 2019 Computers and People Research Conference.* 176–179.
[58] Paul Ralph. 2018. Re-imagining a course in software project management. In *Proceedings of the 40th International Conference on Software Engineering: Software Engineering Education and Training.* 116–125.
[59] Paul Ralph, Sebastian Baltes, Domenico Bianculli, Yvonne Dittrich, Michael Felderer, Robert Feldt, Antonio Filieri, Carlo Alberto Furia, Daniel Graziotin, Pinjia He, and others. Empirical Standards for Software Engineering Research. (????).
[60] Mats Angermo Ringstad, Torgeir Dingsøyr, and Nils Brede Moe. 2011. Agile process improvement: diagnosis and planning to improve teamwork. In *European Conference on Software Process Improvement.* Springer, 167–178.
[61] Sean Robson. 2013. *Agile SAP: Introducing Flexibility, Transparency and Speed to SAP Implementations.* IT Governance Publishing, Cambridgeshire.
[62] Edgar Schein. 2010. *Organizational culture and leadership* (4 ed.). Jossey-Bass, San Francisco.
[63] Ken Schwaber. 2004. *Agile project management with Scrum.* Microsoft Press, Redmond, Wash.
[64] Ken Schwaber and Jeff Sutherland. 2020. The 2020 Scrum Guide. (2020). https://www.scrumguides.org/scrum-guide.html
[65] Todd Sedano, Paul Ralph, and Cécile Péraire. 2020. Dual-Track Development. *IEEE Software* 37, 6 (2020), 58–65.
[66] Chiao-Ching Shih and Sun-Jen Huang. 2010. Exploring the relationship between organizational culture and software process improvement deployment. *Information & Management* 47, 5-6 (2010), 271–281.
[67] Chaehan So and Wolfgang Scholl. 2009. Perceptive agile measurement: New instruments for quantitative studies in the pursuit of the social-psychological effect of agile practices. In *Agile Processes in Software Engineering and Extreme Programming.* Springer, 83–93.
[68] Simone V Spiegler, Daniel Graziotin, Christoph Heinecke, and Stefan Wagner. 2020. A Quantitative Exploration of the 9-Factor Theory: Distribution of Leadership Roles Between Scrum Master and Agile Team. In *International Conference on Agile Software Development.* Springer, 162–177.
[69] Simone V Spiegler, Christoph Heinecke, and Stefan Wagner. 2021. An empirical study on changing leadership in agile teams. *Empirical Software Engineering* 26, 3 (2021), 1–35.
[70] Pallavi Srivastava and Shilpi Jain. 2017. A leadership framework for distributed self-organized scrum teams. *Team Performance Management* 23, 5/6 (2017), 293–314.
[71] Diane E Strode, Sid L Huff, and Alexei Tretiakov. 2009. The impact of organizational culture on agile method use. In *2009 42nd Hawaii International Conference on System Sciences.* IEEE, 1–9.
[72] Alvin Teh, Elisa Baniassad, Dirk Van Rooy, and Clive Boughton. 2012. Social Psychology and Software Teams: Establishing Task-Effective Group Norms. *IEEE Software* 29, 4 (2012), 53–58.
[73] Cristiano Tolfo and Raul Wazlawick. 2008. The influence of organizational culture on the adoption of extreme programming. *Journal of systems and software* 81, 11 (2008), 1955–1967.
[74] L. Tolstoy. 1869. *War and Peace.* Penguin, Harmondsworth, UK.
[75] Bruce Tuckman and Mary Jensen. 1977. Stages of small-group development revisited. *Group & Organization Management* 2, 4 (1977), 419–427.
[76] Seidmehdi Veiseh, Neeman Eghbali, and others. 2014. A study on ranking the effects of transformational leadership style on organizational agility and mediating role of organizational creativity. *Management Science Letters* 4, 9 (2014), 2121–2128.
[77] Mario Von Cranach. 1986. Leadership as a function of group action. In *Changing conceptions of leadership*, Carl F. Graumann and Serge Moscovici (Eds.). Springer, New York, US, 115–134.
[78] Gang Wang, In-Sue Oh, Stephen H Courtright, and Amy E Colbert. 2011. Transformational leadership and performance across criteria and levels: A meta-analytic review of 25 years of research. *Group & organization management* 36, 2 (2011), 223–270.
[79] S. Wheelan and J. Hochberger. 1996. Validation studies of the group development questionnaire. *Small Group Research* 27, 1 (1996), 143–170.
[80] Laurie Williams. 2012. What agile teams think of agile principles. *Commun. ACM* 55, 4 (2012), 71–76.
[81] Michael Wufka and Paul Ralph. 2015. Explaining Agility with a Process Theory of Change. In *Agile Conference (AGILE).* IEEE, 60–64.