# A Theory of Scrum Team Effectiveness

CHRISTIAAN VERWIJS, The Liberators, The Netherlands
DANIEL RUSSO*, Department of Computer Science, Aalborg University, Denmark

Scrum teams are at the heart of the Scrum framework. Nevertheless, an integrated and systemic theory that can explain what makes some Scrum teams more effective than others is still missing. To address this gap, we performed a seven-year-long mixed-method investigation composed of two main phases. First, we induced a theoretical model from thirteen exploratory field studies. Our model proposes that the effectiveness of Scrum teams depends on five high-level factors - responsiveness, stakeholder concern, continuous improvement, team autonomy, and management support - and thirteen lower-level factors. In the second phase of our study, we validated our model with a Covariance-Based Structural Equation Modeling (SEM) analysis using data from about 5,000 developers and 2,000 Scrum teams that we gathered with a custom-built survey. Results suggest a very good fit of the empirical data in our theoretical model ($CFI = 0.959$, $RMSEA = 0.038$, $SRMR = 0.035$). Accordingly, this research allowed us to (1) propose and validate a generalizable theory for effective Scrum teams and (2) formulate clear recommendations for how organizations can better support Scrum teams.

CCS Concepts: • **Social and professional topics** → *Computing and business*; *Computing profession*; • **Software and its engineering** → **Software development techniques**.

Additional Key Words and Phrases: Agile, Scrum, Teams, Structural Equation Modeling, Case Studies.

**ACM Reference Format:**
Christiaan Verwijs and Daniel Russo. 2022. A Theory of Scrum Team Effectiveness. *ACM Trans. Softw. Eng. Methodol.* 37, 4, Article 111 (November 2022), 52 pages. https://doi.org/10.1145/1122445.1122456

## 1 INTRODUCTION

Software development teams are at the core of the Scrum framework [133]. A recent investigation by Russo [127] identified Scrum teams as the most important driver of project success [124]. Like other Agile development methods, Scrum follows a collaborative, iterative, and human-oriented approach to software development by reflecting the principles of the Agile Manifesto [12]. This is reflected by an increasing recognition that the ability to work effectively in teams is a foundational skill for citizens in the future of work [39].

Among Agile methods, Scrum is the most popular [155]. According to market research led by VersionOne, in 2021, 81% of organizations reported that they use Scrum or a hybrid version of it. Software houses routinely use it for their internal development process [159]. Scrum teams focus on delivering incremental business value through the continuous interaction with stakeholders within an established framework [133]. With a minimal set of prescribed roles and events, it is up to teams to build on this foundation by adding practices for stakeholder collaboration, estimation, task breakdown, portfolio management, and so on that make sense in their context. Thus, to be

*Corresponding author.

Authors' addresses: Christiaan Verwijs, christiaan.verwijs@theliberators.com, The Liberators, The Netherlands; Daniel Russo, daniel.russo@cs.aau.dk, Department of Computer Science, Aalborg University, A.C. Meyers Vaenge, 15, 2450, Copenhagen, Denmark.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
© 2022 Association for Computing Machinery.
1049-331X/2022/11-ART111 $15.00
https://doi.org/10.1145/1122445.1122456

effective, Scrum assumes a high degree of autonomy for teams to perform their work [68]. The 2020 Scrum Guide highlights the importance of Scrum teams over the individual roles that make it up [135]. By shifting control from middle management to autonomous Scrum teams, the role of management has become less relevant. Therefore, the contemporary understanding of Scrum is to support teams to deliver and deploy software as frequently as is helpful [68], which is summarized in the 2020 Scrum Guide as "Sprints are the heartbeat by which ideas are turned into value." [135].

In the last twenty years, researchers performed several empirical investigations into Scrum, mostly field studies [45]. So far, research typically focused on process improvement factors (such as security [119]), transformation processes [38], methodological and tailoring practices [105], or Scrum artifacts [26]. Some scholars also explored teaming aspects, like autonomy [110] and proactive behavior [76]. However, few studies have specifically explored which team-level factors are most important to Scrum team effectiveness, with few exceptions [108, 110]. Additionally, most works performed so far are not generalizable because they are field studies or use a research strategy not aimed at generalizing results [45, 142]. Thus, we do not know if the accumulated knowledge about Scrum teams is specific to a few contexts or if it is generalizable to larger populations. A sample study, using adequate statistical techniques to test research hypotheses [126], would bridge this gap [142]. Accordingly, we formulate our Research Questions (RQ):

- $RQ_1$: *Which are the key factors that determine Scrum team effectiveness, and how do they relate to each other?*
- $RQ_2$: *How generalizable is the theoretical model for Scrum teams?*

This study focuses on Scrum teams, and not more broadly on Agile teams of other kinds, for two reasons. The first is that Scrum provides a well-defined framework with a small set of roles and events for teams to implement. This brings focus and a shared vocabulary to the investigation. The second reason is that the prevalence of Scrum makes it easier to reach a sufficiently large sample.

The characteristic of sample studies is that they rely on pre-existing theoretical evidence [127]. Accordingly, we had to induce a working theory first to address $RQ_1$. To do so, we performed 13 case studies from 2014 to 2019, involving informants from over 45 Scrum teams. Using the multiple cases study approach by Eisenhardt [49], we inferred a theoretical model from our qualitative findings and extant literature to explain how team-level factors contribute to Scrum team effectiveness.

In this model, we define *Scrum team effectiveness* as the satisfaction of team members with their work process and stakeholders' satisfaction with the outcomes of that process. We found that team effectiveness is determined by five high-level factors with thirteen sub-factors. The first high-level factor is the ability of teams to be responsive to the changing needs of stakeholders, as expressed through frequent releases and effective refinement. This is amplified through the second high-level factor we identified; a strong concern for the needs of stakeholders. This involves a focus on value, clear Sprint Goals, collaboration with stakeholders, and high-quality Sprint Reviews. Third, teams require a climate of continuous improvement to incrementally improve their ability to be responsive and focus on stakeholder needs. This is expressed through a learning environment, a strong concern for quality, opportunities for shared learning, psychological safety, and high-quality Sprint Retrospectives. In turn, teams need to have the autonomy to make necessary changes, through self-management and cross-functionality. Finally, we show that strong support from management is vital to make all this possible.

To answer $RQ_2$, we ran a large-scale cross-sectional study by collecting data from 4,940 software professionals of 1,978 Scrum teams, resulting in one of the most extensive sample studies performed in software engineering. We aimed to test whether the inferred theoretical model also fit empirical data and matched the day-to-day reality of Scrum teams. Using Covariance Based Structural Equation Modeling (CB-SEM or SEM in short), we found that the model fit the data well and that

its proposed factors explained a substantial amount of variance in Scrum team effectiveness. A replication package is also openly available on Zenodo to encourage secondary studies.

In the remainder of this paper, we describe the related work in Section 2. Afterward, we introduce our Mixed-Methods research design in Section 3 and induce our theoretical model from 13 multiple exploratory case studies in Section 4.1. Subsequently, in Section 6, we validate our model through a large-scale cross-sectional study using Covariance Based Structural Equation Modeling. Finally, we discuss the implications for research and practice along with the study limitations in Section 7 and draw our conclusion by outlying future research directions in Section 8.

## 2 RELATED WORK

Software development teams have been at the core of Scrum since the first guide was issued in 2010 by Jeff Sutherland and Ken Schwaber[1]. Consequently, software engineering literature investigated several aspects of Scrum teams over the last decade. Since this investigation seeks relevance instead of completeness, we primarily address studies that focus on Scrum teams to identify our research gap.

Agile software development is widely studied in software engineering research [45]. Much of this work has focused on the transition from plan-driven to Agile approaches at the organizational level. Dikert, Paasivaara & Lassenius [38] and López-Martínez et al. [92] identified common challenges in Agile transformations, such as lack of management support, limited understanding of Agile methodologies, and a cultural mismatch with Agile principles. However, plan-driven development may co-exist with Agile practices if management takes proper actions to mitigate challenges and confusion that might arise in Scrum teams [154]. Another challenge is that Agile practices do not guide HR processes such as onboarding and knowledge transfer, so organizations have to figure out how to deal with it [40]. Other studies have investigated requirements engineering in Agile environments [132] and how Agile methodologies impact contracting [128].

Scholars have also investigated the transition from plan-driven to Agile approaches at the team level. Hoda & Noble [68] used a Grounded Theory approach to understand how Agile teams transition from traditional to Agile practices. Gren et al. [59] suggest that group developmental aspects of Scrum teams are critical success factors during the transition from a plan-driven to an Agile development fashion.

Several studies have attempted to identify critical success factors for Agile teams. Melo et. al. [104] performed an interpretative case study to develop an *Input-Process-Output* (IPO) model for team effectiveness. The IPO framework has been used in teamwork research [61, 102] to understand team effectiveness as resulting from the interplay between input states (e.g. characteristics about the team, task, organization or individuals) and group processes (e.g. team dynamics, communication quality, learning, conflict) [53]. Melo et. al. [104] observed a negative impact of member turnover as a result of conflict, which hampered team productivity. Team size, diversity, team development stage and collocation were identified as relevant input factors, with shared learning, communication, conflict management and the use of Agile practices as group processes.

Recently, the Agile Success Model was proposed by Russo [124] to provide a general understanding of how the dynamics within an Agile project lead to its success. This study identified (Scrum) teams and the skills of their members as the most critical enabler.

Other studies have applied quantitative methods to identify success factors across organizations. Chow & Cao [25] surveyed 109 Agile projects to identify factors that contributed to their perceived success. They identified a frequent delivery strategy, Agile software techniques and capable teams as the most important factors. Surprisingly, no such effects were found for factors outside the team, like

[1]www.scrumguides.org/revisions.html

executive support, organizational culture and project attributes like size and type. This is partially supported by a recent survey study of 40 Scrum projects by Van Kelle et. al. [153]. They also found the adoption of Agile development practices to be a critical success factor, and also that project size did not matter. In addition, the researchers identified the leadership of Scrum Masters, and a shared understanding of what is valuable (value congruence) as important contributors. Serrador & Pinto [136] investigated survey data from 1.386 Agile projects in an extensive survey study. They found that high-quality goals and a clear product vision contribute to successful projects, but not project complexity or team experience. Furthermore, Lee & Xia [89] used a mixed-method approach with case studies and a survey study to investigate how team autonomy and team diversity influence Agile project success. While both were found to influence project success, autonomy also adversely impacted results when performance goals were not prioritized.

Taken together, these studies suggest that much of what makes Agile teams and projects successful lies at the team level. Frequent delivery, team autonomy, the use of Agile development practices, and skilled teams seem to be widely recognized. It is less clear how these factors interact and which additional factors are relevant. There is also no consensus on the influence of the characteristics of teams, projects, or organizations.

### 2.1 Research on Scrum

The majority of the studies discussed to this point investigated Agile projects and Agile teams in a broad sense, and not Scrum teams specifically. While the Scrum framework remains the most popular implementation of the Agile manifesto [155], there are others like Extreme Programming (XP), DevOps, Kanban, and custom implementations. It is reasonable to assume that the findings up to this point generalize to Scrum teams. However, the specificity of the Scrum framework along with its prevalence in the workplace provides an opportunity to investigate more specific factors. For example, not all Agile methodologies have roles similar to Product Owners or Scrum Masters. Nor do all Agile methodologies prescribe iterations to drive frequent delivery. So in this section, we will explore related work to Scrum teams. But we begin with an overview of the Scrum framework to provide consistent terminology throughout this paper.

The Scrum framework is designed around self-managing Scrum teams [135]. A *"Scrum team"* consists of a Scrum Master, a Product Owner, and developers. *"Stakeholders"* are external parties with a stake in the outcomes of the team, like customers and users. The Scrum framework prescribes five recurring events to foster collaboration and coordination within the team and its stakeholders. The first recurring event is the *"Sprint"* itself. It has a consistent duration of a month or less. Each Sprint begins with *"Sprint Planning"*, where a team sets a goal on what it aims to deliver that Sprint (the *"Sprint Goal"*) and selects the work to achieve it. This work is selected from a prioritized list called the *"Product Backlog"*. It contains all requirements, features, and ideas deemed valuable for the product. The selection of work for the current Sprint is called the *"Sprint Backlog"*. It changes to reflect discoveries that teams make as they collaborate to achieve the goal. One daily opportunity for such discoveries is the *"Daily Scrum"* where members coordinate their work intending to deliver an increment that achieves their Sprint Goal. These increments need to be done, meaning that they conform to the quality standards set out by the team as captured in their *"Definition of Done"*. At least before the end of the Sprint, the team shares its increment with stakeholders and gathers feedback during the *"Sprint Review"*. The Product Backlog is updated accordingly to reflect discoveries. Finally, a *"Sprint Retrospective"* is held within the team to learn and improve its process. Within this framework, the *"Scrum Master"* is accountable for enacting Scrum while the *"Product Owner* is accountable that what a team works on is indeed valuable. All other members of the team are considered *"Developers"* of the product and include all the skills and specializations needed to deliver done increments.

The role of Scrum Master and Product Owner have been of particular interest to researchers. Shastri, Hoda & Amor [137] found that Scrum Masters typically spend time on six core activities; facilitating, mentoring, negotiating, adapting processes, coordinating and protecting. But tensions occur when they are also responsible for project management, which negatively impacts team performance [112]. The Product Owner is more suited to these activities. This is also clearly reflected in the eight roles that Bass et. al. [10] identified as typical for Product Owners, which center around requirement analysis, prioritization, release management and stakeholder management. This focus on the discovery and communication of business needs is also noted by other studies [16, 149]. However, Sverrisdottir et. al. [145] note from interviews with Product Owners that their success is largely dependent on organizational factors. Thus, how the role is practiced varies widely by organization, and often deviates from theory. Several studies suggest that Product Owners are more successful when they involve teams in strategy formulation, refinement and stakeholder management [10, 75].

Another area of investigation has been how Scrum can be scaled across multiple teams. Paasivara & Lassenius [113] investigated challenges related to inter-team coordination when scaling Scrum across many teams. High-quality communication and coordination by Product Owners has been shown to impact project success [16].

Effective self-managing teams are at the center of the Scrum framework. Several studies have investigated team-level factors. For example, Baumgart, Hummel & Holten [11] explored which personality types are most desirable in Scrum teams. Furthermore, developers need both soft and hard skills to be effective as a team [67, 124]. Surprisingly few studies have attempted to conceptualize Scrum team effectiveness and its determinants, even though such insights are invaluable to practitioners. Although productivity-based measures such as the SPACE framework [52] have been proposed, they are difficult to aggregate and compare across organizations and are often constrained by factors beyond a team's control [102]. Melo et al. [104] proposed a more comprehensive definition of productivity of Agile teams by including outcomes such as customer satisfaction, creativity, and product quality.

Moe et al. [108, 110] were among the first to investigate Scrum team effectiveness from the perspective of *teamwork*. Salas et al. [129] define this as "the set of interrelated thoughts, actions, and feelings of each team member that are needed to function as a team and that combine to facilitate coordinated, adaptive performance and task objectives resulting in value-added outcomes". They propose a model for effective teamwork based on five core components (shared leadership, peer feedback, redundancy, adaptability, and team orientation) and three coordinating mechanisms (mutual trust, shared mental models, and closed-loop communication). This model has been successfully applied to Scrum teams by Moe & Dingsøyr [108] and Agile teams by Strode, Dingsøyr & Lindsorn [143].

A strength of this perspective on Scrum team effectiveness is that it grounds it in a tradition of teamwork research. However, the generalized model for effective teamwork by Salas et al. [129] may not consider idiosyncrasies of Scrum teams that are critical to their effectiveness. We consider this particularly relevant to three areas. First, it does not consider how teams interact with their stakeholders. Several studies have emphasized the role of the Product Owner and shared Product Ownership as a contributor to team effectiveness [10, 86, 122, 153]. This is particularly relevant to Scrum teams as their interactions with stakeholders are a critical part of Agile methodologies [12]. Second, the capability of Scrum teams to be responsive to changing needs in their environments is not considered, even though this is an important reason to use Agile methodologies over plan-driven approaches [40]. Third, Scrum teams are always embedded in larger systems that constrain or enable their effectiveness in dynamic ways [36]. So teams have to learn how to continuously evaluate and improve their work methods, processes, and environment to remain or become effective. Scholars

such as Argyris [6] argue that such continuous improvement is difficult without a culture that encourages learning [54]. Similarly, the self-managing nature of Scrum teams demands a different approach from management for teams to make their own decisions. Manz et al. [97] describe this shift as "leading others to lead".

These areas are illustrative of a systems perspective on teams [93]. Although effective teamwork within a team is undoubtedly a solid contributor to team effectiveness, much of it will be negotiated on the boundaries with the larger systems it is part of [36]. For example, what support do teams receive from the organization? How do they collaborate with stakeholders, customers, and users? Furthermore, how is their effectiveness enabled or impeded by technology and organizational culture?

Thus, this study aims to provide a systemic understanding of the factors that contribute to the effectiveness of Scrum teams. This includes factors internal to teams and those on the boundaries of teams.

## 3 RESEARCH DESIGN

We deployed a Mixed Method approach to answer our research questions [29]. Such approaches are increasingly used in the social sciences, humanities [73], and also in software engineering [125, 141] to increase the fidelity of theories and findings over purely quantitative or qualitative inquiries [106]. The different research phases of our research design are summarized in Figure 1.

In the **first phase**, we used a multiple case study to identify potentially relevant factors by observing Scrum teams in 13 different cases. We combined our observations with extant literature to induce a preliminary, testable theory (the objective of $RQ_1$). While the natural research setting offered by case studies is excellent for developing such a theoretical understanding of realistic phenomena [142], it is not aimed at the generalization of findings across settings (the objective of $RQ_2$). Furthermore, the exploratory nature of our multiple case studies would not be sufficient to claim a robust theory.

In the **second phase**, we tested the preliminary theoretical model in a neutral research setting [142] using Structural Equation Modeling[2] (SEM) and a substantial sample of Scrum teams from different settings. Structural Equation Modeling is commonly used to test hypothesized relationships between unobserved (latent) constructs through observed indicator variables [20, 83]. The hypothesized relationships in the model are specified into a series of structural (regression) equations that predict the variance-covariance matrix that should be observed between variables in the data if the model fits perfectly. A second variance-covariance matrix is calculated from sample data and compared with the first matrix. The smaller the discrepancy between the matrices, as indicated by the statistical goodness of fit indices [20], the better the theoretical model "fits" the observed data. A strength of Structural Equation Modeling is that it is an inherently confirmatory, hypothesis-driven approach to model testing [20]. Because the entire model is tested simultaneously, researchers can perform factor analysis and regression analysis in a single model. Finally, Structural Equation Modeling allows researchers to assess and adjust for measurement error by explicitly estimating it as part of the model, which is impossible in traditional multivariate approaches. This allows researchers to arrive at more accurate estimates, especially when there is a substantial error in the measurements, such as in sample studies [20].

[2]Structural Equation Modeling divide itself into Covariance-Based Structural Equation Modeling and Partial Least Squares Structural Equation Modeling. This paper focuses solely on Covariance-Based Structural Equation Modeling. For an overview of these different computation methods and their terminology, cf. Russo & Stol, 2021.

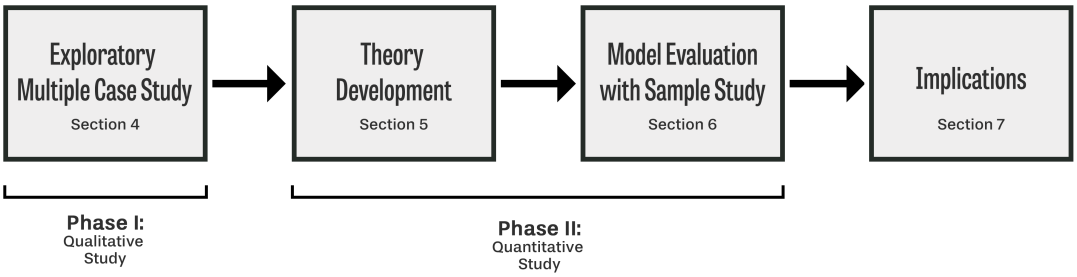


Fig. 1. The design of this two-phased study, and the corresponding sections in this paper.

## 4 PHASE I: MULTIPLE CASE STUDY

### 4.1 Case selection

This research investigates Scrum team effectiveness. Consequently, our unit of analysis is the Scrum team. Due to the exploratory nature of this first phase, we did not use *a priori* criteria to guide our sample selection. Instead, we opted for a flexible sampling design and selected cases to maximize case variability and improve our theoretical understanding [160]. In particular, we looked for a variation in size of organizations, commercial or non-commercial nature of the organizations, industry, and age of establishment. From a theory-building perspective, we relied on Eisenhardt's replication logic [48] i.e., each experience serves as a different experiment that stands on its own as an analytic unit, inducing theory through observations or data [49].

The sample for our multiple case study consisted of 13 separate cases that took place between 2014 and 2019 (see table 1). Each case study observation took place for several months to a year. Each case involved one or more Scrum teams in organizations in improving their effectiveness. Some of these organizations already worked with Scrum (Table 1: #10, #11, #12), while others implemented it during the case studies.

### 4.2 Collection of Qualitative Data

We used different techniques and multiple data sources for each case. The use of multiple data sources allows the triangulation of findings and increases internal validity [139].

Each case began with semi-structured **interviews** of the informants that led internal change initiatives to increase team effectiveness, usually Scrum Masters, Product Owners, and management. Most interviews lasted one hour. Informants were asked to define effectiveness and identify which factors they thought were relevant. We opted for semi-structured interviews to allow the broadest possible scope for data collection. We periodically interviewed these informants again during our engagements and at the end. After each interview, we summarized key findings and validated them with the informants. In total, we collected 155 pages of field notes and interviews.

We then hosted **workshops** that were attended by the Scrum teams and relevant stakeholders. The workshops lasted between 1 and 2 days. During the workshops, we moderated a conversation in the group to identify and build consensus on what made them effective and where and how they wanted to improve their process to become more effective. The groups generated **documentation** in the form of flip charts to capture insights in their own words. Sixty-two documents were created in this manner. We were also given access to other sources of documentation in most cases, such as a Definition of Done, Product Backlogs, developed Increments, and team charters.

Following the initial workshops, we were on-premise for one or more days per week, ranging between two months and a year. The duration depended on the length of the engagement and

Table 1. Description of the cases used in this study for observations and interviews

| # | Year | Domain | Org size | Nr. of teams | Roles involved in observations |
|---|---|---|---|---|---|
| 1 | 2014-2016 | Transportation | 4,000-5,000 | 5 | 2x Scrum Master, 1x Product Owner, 30+ Developers, 1x Program Manager |
| 2 | 2014-2017 | Software development | 10-20 | 2 | 2x Scrum Masters, 1x Product Owner, 8x Developers, 1x CxO |
| 3 | 2014-2018 | Software development | 10-20 | 1 | 2x Scrum Masters, 1x Product Owner, 1x CEO |
| 4 | 2014-2016 | Software development | 10-20 | 1 | 1x Scrum Master, 1x Product Owner, 12x Developers, 1x CxO |
| 5 | 2015-2017 | Retail | 50-100 | 2-5 | 5x Scrum Masters, 2x Product Owners, 25x Developers, 2x CxO |
| 6 | 2016-2017 | Information Technology | 100-200 | 5 | 5x Scrum Masters, 4x Product Owners, 25x Developers, 2x CxO |
| 7 | 2016-2017 | Non-Profit | 10-50 | 1 | 1x Scrum Master, 1x Product Owner, 6x Developers |
| 8 | 2016-2017 | Healthcare | 50-100 | 2 | 1x Scrum Master, 1x Product Owner, 8x Developers |
| 9 | 2017-2018 | Utilities | 500 | 1 | 1x Scrum Master, 1x Product Owner, 6x Developers |
| 10 | 2018-2019 | Tourism industry | 4,000-5,000 | 5 | 12x Scrum Masters |
| 11 | 2018-2019 | Recruitment | 50-100 | 1 | 1x Scrum Master, 1x Product Owner, 5x Developers, 1x CxO |
| 12 | 2019 | Retailer | $> 10,000$ | 12 | 15x Scrum Masters, 4x Product Owners, 3x Agile Coaches |
| 13 | 2019 | Research & education | 1,000-2,000 | 5 | 4x Scrum Masters, 2x Product Owners, 38x Developers, 1x Department Manager |

the organization's size. In most cases, a desk in the team room was provided to us, or a desk in a room close to the teams. This allowed us to capture **observations** during more than 150 meetings and Scrum Events, as well as social interactions that happened naturally throughout the day (e.g., lunches, breaks, conversations). These observations allowed us to capture relevant group dynamics and intimate project knowledge from the participants and the organization. We captured the observations in field notes directly after they occurred. For case #12, we could not remain on-premise as this company was geographically distant. Instead, we visited the organization on two occasions and performed follow-up interviews remotely. We included this case due to the unique characteristics of its organizational culture and because the workshops were attended by a substantial cross-section of many Scrum teams and organizational layers.

### 4.3 Analysis of Qualitative Data

We used the analytical framework known as *Gioia methodology* [57] to analyze our qualitative data. In particular, Gioia et al. proposed a structured approach to foster scientific rigor in qualitative data analysis. This approach, which is more and more common within the management community, is now also being used in software engineering [124].

Following the Gioia methodology, we analyzed our data in three different stages. In the first stage, we used open coding to identify 61 distinct low-level concepts in the data [28]. We then used axial coding in the second stage to group related concepts into 16 second-order themes relevant to answering $RQ_1$. Finally, we grouped second-order themes into six high-level aggregate dimensions until theoretical saturation was achieved. In all stages, we aimed to retain the terminology used by practitioners to keep our theoretical model grounded. Researchers ideally perform the first and second stages with as little theoretical knowledge as possible to prevent confirmation bias [57]. In addition, we involved a domain expert unfamiliar with the study to categorize first-order concepts into themes and aggregate dimensions independently. Several iterations were performed until consensus was achieved. This three-stage approach is a data structure where grounded observations about success factors are induced into six high-level theoretical aggregate dimensions connected to existing literature.

### 4.4 Results from the case studies

The following sections report the six aggregate dimensions that we induced from the multiple case studies. Guided by our first research question ($RQ_1$), we begin our examination with team effectiveness and then explore relevant team-level dimensions that may contribute to it. We use the reporting structure recommended by Gioia [57] to provide clarity on how we aggregated our observations into higher-order concepts. We provide observations and exemplary quotes from informants for each concept and connect them to extant literature. The complete data structure is available in Table 13 in the Appendix.

*4.4.1 Team Effectiveness.* The Scrum teams broadly identified two ultimate outcomes of effective teamwork with Scrum. We summarize the observations and second-order themes in Table 2.

**Stakeholder satisfaction:** Most Scrum teams expected that effective teamwork with Scrum results in more satisfied stakeholders, particularly customers and users (see Tab 1: #1, #2, #3, #4, #5, #7, #8, #12). Many teams reasoned that this was because Scrum allowed them to release increments sooner and more frequently (#2, #3, #4) and deliver value to stakeholders more frequently (Tab 1: #2, #3, #4, #6, #7). The increased collaboration between teams and stakeholders was another reason. (Tab 1: #2, #5, #6).

> "A big opportunity [of Scrum] for us is to get more satisfied customers" (Tab 1: #3, Product Owner)

This expected outcome is congruent with the goal of the Scrum Framework, which is to deliver value to stakeholders more frequently [135]. Other studies also identified stakeholder satisfaction as a key indicator for Agile Teams of post-release quality [87] and team performance [94].

**Team Morale:** Another outcome that Scrum teams expected was increased morale and motivation (Tab 1: #2, #3, #4, #5, #6, #7, #8, #9, #13). For many Scrum teams, this resulted from increased collaboration (Tab 1: #1, #3, #6, #8). Other reasons that were given ranged from "do work that matters" (Tab 1: #6), "more personal autonomy" (Tab 1: #2, #6) to "getting things done" (Tab 1: #4, #6).

> "I expect that when we work well with Scrum, our team spirit will be higher" (Tab 1: #2, Scrum Master)

In extant literature on work motivation, Kahn [77] found that the psychological experience that one's work is meaningful contributes to the degree to which people engage in that work and push

on in the face of challenges. While work engagement is an individual experience, the team-level equivalent has been called team morale by organizational psychologists [71, 151] or *esprit de corps* in military teams [96]. The motivational quality of frequent releases can be understood with goal-setting theory [91] when we consider each release as a tangible and ambitious short-term goal to be achieved by the team. Specifically for Scrum teams, researchers have argued that team members are effectively internal stakeholders and that their satisfaction should be an important goal in addition to customer satisfaction [94].

Table 2. Data structure of observations and themes for the aggregate dimension 'Team Effectiveness'

| **1st Order Observations & concepts** | **2nd Order Theme** |
|---|---|
| Stakeholders are (more) satisfied with the work done by the team | Stakeholder Satisfaction |
| Stakeholders frequently complain about the quality of the product | |
| Stakeholders give compliments to the team for a job well done | |
| Team members feel motivated in their work | Team Morale |
| Team members frequently complain, but there is no conversation about how to solve things | |
| Team members feel like they're on a hamster wheel (different day, same problems) | |

*4.4.2 Responsiveness.* The Scrum teams we observed varied in their ability to respond quickly to emerging needs or issues. This manifested in two related areas. We summarize the observations and second-order themes in Table 3.

**Release Frequency:** The Scrum framework requires Scrum teams to deliver at least one increment every Sprint that can be released to stakeholders [135]. In three cases, teams generally released to stakeholders at least once during the Sprint (Tab 1: #9, #11, #12). However, most teams released intermediate versions to internal staging environments (Tab 1: #1, #2, #3, #4, #5, #8, #13) or used several Sprints before they were able to do so (Tab 1: #6, #10). These teams released batches to their stakeholders relatively infrequently - once per quarter or less - except for critical bug fixes. At the same time, teams recognized the benefits of releasing more frequently. One team stated that:

> "It allows us to deliver something to our customers sooner and allows them to suggest changes, or even stop" (Tab 1: #3, Team Member)

While other teams observed that frequent releases contribute to:

> "rhythm, clarity and getting things done" (Tab 1: #3, Team Member)
> "increased software quality" (Tab 1: #6, Team Member)
> "happier customers" (Tab 1: #3, Scrum Master)

We observed that the nature of Sprint Reviews changed as release frequency increased. Teams with high release frequencies (Tab 1: #1, #2, #3, #9, #11, #12) inspected the increments in staging or even production environments and used the Sprint Review to collect critical feedback from stakeholders. More stakeholders also attended these sessions. In contrast, teams with lower release frequencies often used static presentations or release logs instead of a working increment to share with stakeholders what was done (Tab 1: #5, #6, #10). These findings are congruent with other studies. For example, Chow & Cao [25] surveyed 109 Agile projects and identified a strategy

optimized for frequent delivery as the most important success factor. This has also been found in other studies [74].

**Refinement:** The Scrum teams we observed varied in approaching the clarification and decomposition of work on the Product Backlog. This is generally called "refinement" in professional literature [135]. Teams engage in this activity to break down large chunks of work into smaller chunks that can be delivered within the scope of a Sprint or less. Most teams performed refinement during weekly workshops attended by the entire team (Tab 1: #1, #3, #4, #5, #9, see also Figure 2), which is consistent with other case studies [81, 113]. Other teams performed it mainly during Sprint Planning (#2, #7, #8). One consistent pattern was that while most teams performed refinement, they also felt it should be more. For example, one team concluded that:

> "We lack the discipline to refine well" (Tab 1: #1, Team Member)

While another Scrum Master observed that:

> "When we do refine work, the Sprint seems to be much smoother" (Tab 1: #9, Scrum Master)

Some teams struggled to see the purpose of refinement. They perceived Sprints as artificial time-boxes where any uncompleted work could be carried over into the next Sprint (Tab 1: #1, #5, #7). In particular, we observed that Sprint Planning tended to go faster for teams that refined during the Sprint. In addition, most Scrum teams frequently raised (the lack of) refinement as a topic during Sprint Retrospectives (Tab 1: #1, #2, #3, #13). This was usually driven by a desire to have more clarity as to what is needed, identify external dependencies earlier, or make Sprint Planning easier. One Scrum Master described the purpose of refinement as:

> "a way to identify dependencies on other teams upfront, so that we are not surprised by it during a Sprint" (Tab 1: #1, Scrum Master)

Table 3. Data structure of observations and themes for the aggregate dimension 'Responsiveness'

| 1st Order Observations & concepts | 2nd Order Themes |
|---|---|
| The team feels that iterations are artificial (i.e., "There's always a next Sprint" or "Let's just carry it over to the next Sprint") | Refinement |
| Team members spend time during a Sprint to prepare work for the coming Sprints | |
| Sprint Planning tends to drag on as teams try to discover what they need to create that Sprint | |
| The Sprint Backlog contains only a handful of large Product Backlog Items | |
| Sprints generally result in at least one releasable increment for this team | Release Frequency |
| Acceptance testing is done after the Sprint and not during | |
| Sprint Reviews are used to review an Increment that is running on an environment that is as close to production as possible (e.g., not a developers pc) | |
| Sprint Reviews do not demonstrate working software (e.g., instead, PowerPoint presentations are used, or the team simply talks about it) | |

Fig. 2. A Scrum team engaged in refinement (see Table 3). Picture by Christiaan Verwijs.

*4.4.3 Stakeholder Concern.* We found that Scrum teams varied in the degree to which the needs of stakeholders were considered in their decision-making. We define *"stakeholders"* as external parties that have a stake in the outcomes of a team, like customers, key users, and investors. The differences manifested primarily in four areas.

**Value focus:** Not all Scrum teams focused equally on the value their work delivered to stakeholders. The Product Owners played an important role here. Some Product Owners frequently discussed with their team how the work on the Product Backlog fit in the overarching strategy (Tab 1: #1, #3, #5, #9, #12). These teams also periodically removed items from the Product Backlog that no longer fit with that strategy. Three Product Owners also shared how the work returned on investments and how it impacted stakeholders (Tab 1: #1, #3, #12). Other Product Owners primarily passed requirements on without elaboration (Tab 1: #6, #7). We also observed that teams differed in how work was described on the Product Backlog. Some teams (Tab 1: #3, #6) captured work in technical terms without connecting it to the needs of stakeholders (e.g., "Update connector API to expose module composition."). Other teams explicitly clarified how items on their Product Backlog connected to the needs of stakeholders (e.g., "Administrators can sort submissions by entry-date so they can address the most recent ones first.").

The role that Product Owners play in creating a sense of what is valuable is also recognized in extant literature [9, 150]. Bass et al. [10] concluded from a qualitative study with 46 Product Owners that the role "is critical for translating business needs into software implementation by gathering and prioritizing requirements.". Product Owners can use different strategies to promote collective product ownership and develop a shared understanding of value [75, 86, 149]. A shared understanding of value among team members (or "Value congruence") was identified as a critical success factor to Agile project success by Van Kelle et. al. [153].

**Stakeholder collaboration:** There was a substantial difference between Scrum teams in how they involved stakeholders. Some teams frequently invited stakeholders to their Sprint Reviews (Tab 1: #1, #2, #3, #9). The Product Owner often took the initiative here. Nevertheless, we also observed many teams where developers interacted directly with stakeholders (Tab 1: #1, #3, #4, #7, #9, #14). Developers visited stakeholders on-site or contacted them by e-mail or telephone for clarification. In some cases, developers trained stakeholders to use their product (Tab 1: #1, #2, #3). In some teams, only the Product Owner interacted with stakeholders (Tab 1: #2, #5, #6, #8, #11, #12). The purpose of stakeholder collaboration was mostly described in terms of gathering feedback:

> "It allows us to involve users in development to get early feedback" (Tab 1: #7, Product Owner)
> "Prioritize work together with all internal stakeholders" (Tab 1: #1, Product Owner)

Stakeholder collaboration is recognized as a success factor for Scrum teams in extant literature [25, 68, 69, 92, 117]. Hoda, Stuart & Marshall [69] found that limited collaboration between Agile teams and their stakeholders can lead to unclear requirements, lack of feedback, and loss of productivity and business. Furthermore, less-experienced teams may be more inclined to delegate stakeholder collaboration to their Product Owners [68].

**Sprint Review Quality:** The Sprint Review is a formal event in Scrum where feedback is gathered from stakeholders and adjustments are identified [135]. Not all teams used the Sprint Review for this purpose. Some teams never invited stakeholders (Tab 1: #4, #6, #7). In the teams that did, the way feedback was gathered differed. Some teams offered a formal presentation (Tab 1: #1, #6, #8, #11). Other teams offered stakeholders opportunities to "take the wheel" and try new features on the spot (Tab 1: #3, #5, #8, #9, #13). This interactive approach tended to draw out more constructive feedback than formal presentations. Only in a few cases did teams use Sprint Reviews to reflect on the larger objective and inform the following steps (Tab 1: #1, #9). Product Owners were often instrumental in shaping the quality of Sprint Reviews by actively inviting stakeholders (Tab 1: #1, #9 and #13) by initiating preparation with the team (Tab 1: #7 and #9) or by suggesting interactive formats (Tab 1: #9). In one example of an interactive format, the Product Owner set up four stations where stakeholders could try new features with help from team members. Feedback was collected at the stations on postcards as stakeholders rotated across the stations during the Sprint Review (see Figure 3).

We could not find studies that specifically identified Sprint Reviews as a success factor. However, professional literature emphasizes the Sprint Review as a critical opportunity to align the work of a team with its stakeholders [116, 121, 156]. We speculate that the reason for this gap is that the quality of Sprint Reviews is assumed to result from other factors in this dimension, such as lack of value focus, low collaboration with stakeholders, and unclear goals. However, we observed that some of the teams we studied improved in these areas when they started inviting more stakeholders to their Sprint Reviews.

**Shared goals:** Shared goals are used in the Scrum framework to encourage collaboration and clarify how individual tasks connect to larger business goals [134]. In practice, the use of shared goals varied greatly. In some cases, Product Owners began Sprint Planning by stating a business objective that then guided the ordering and selection of work for that Sprint (Tab 1: #1, #2, #7, #8, #9, #12, #13). Other teams selected the top items, even when they were unrelated (Tab 1: #3, #4, #5, #6, #10). Teams that did not use shared goals often acknowledged that it would bolster team spirit, customer orientation, and collaboration if they would (Tab 1: #1, #4, #7).

Fig. 3. Example of an interactive Sprint Review (see Table 5). Picture by Christiaan Verwijs.

The importance of shared goals is also recognized in the extant literature on work teams. The presence of salient shared goals positively influences team effectiveness [7, 31, 72], cohesion [95], motivation [82] and cooperation [111]. For Scrum teams specifically, shared goals allow Scrum Teams to align on important business objectives, both within teams and the broader organization [86, 158]. Such a congruence on goals has been identified as an important enabler of Agile project success in several studies [136, 153].

*4.4.4 Team Autonomy.* The observed teams varied in their autonomy in making decisions about how to perform their work. This broadly concerned two types of constraints to their autonomy. The first type was imposed from outside the team (self-management). The second type emerged from skill-based constraints within the team (cross-functionality). We summarize the observations and second-order themes for this aggregate dimension in table 5.

**Self-Management:** The ability of the observed Scrum teams to self-manage their work varied by case. For example, the use of the work method was often suggested by management and co-opted by teams after exploratory workshops (Tab 1: #1, #2, #5, #6, #8, #9, #10, #11, #12, #15). During these workshops, the participants distributed roles, scheduled the formal Scrum events, and clarified collective ambitions. Scrum teams controlled their composition in some cases (Tab 1: #1, #3, #6, #14). One department manager initially proposed a composition for five teams and then decided against this during a collective kickoff workshop with everyone present:

> "I discovered today that I should not make decisions for others. So I will leave the composition of the teams to you instead" (Tab 1: #13, Department Manager)

Table 4. Data structure of observations and themes for the aggregate dimension 'Stakeholder Concern'

| 1st Order Observations & concepts | 2nd Order Theme |
|---|---|
| The team frequently spends time cleaning up the Product Backlog by removing items that do not make sense anymore (e.g., in light of changed product vision, strategy) | Value Focus |
| Team members other than the Product Owner can reiterate the strategy behind the upcoming sprints | |
| The team talks about how or why items on the Product Backlog are valuable to their stakeholders during various Scrum Events | |
| The Backlog contains mostly technical tasks and is difficult to connect to specific business needs by stakeholders | |
| The team is familiar with the vision and strategy for the product they are working on | |
| The work that a team does is aligned with other aspects of the business (e.g., support, sales, marketing) | Stakeholder Collaboration |
| Except for the Product Owners, the members of a team do not generally speak with users, customers, or other stakeholders | |
| Product Owners involve customers/users/stakeholders in the prioritization of the Backlog | |
| Product Owners frequently meet with the customers or users using their software | |
| The Backlog is not easily available to stakeholders, end-users, and other teams | |
| There is no interaction during Sprint Reviews between members of the team; people sit back and watch | |
| Stakeholders have opportunities during Sprint Reviews to interact and review a working product or working features | Sprint Review Quality |
| The team prepares for the Sprint Review by determining what they would like feedback on and in what order | |
| The Sprint Review is attended by users, customers, and other stakeholders | |
| The Product Owner uses the Sprint Review to get feedback from stakeholders on the work that was done | |
| Most Sprints have a Sprint Goal that explains how the work in a Sprint tie together | Shared Goals |
| There is a roadmap of Sprint Goals for the upcoming sprints | |

In another case, a department manager proposed an initial composition for several teams and encouraged them to switch members when they discovered better compositions (Tab 1: #6). While all teams in our study could distribute work internally as they saw fit, their control over incoming work and when to perform it varied greatly. In one case, a Scrum Master observed:

> "Our Product Owners have no autonomy. They are told what to do, when to do it, and often how to do it, and they pass that on to the team" (Tab 1: #10, Scrum Master)

Similarly, contractual agreements regarding the scope and delivery dates were often controlled by external departments or steering committees (e.g., Tab 1: #1, #2, #4, #7). Only a few teams were directly responsible for their contractual agreements with stakeholders (Tab 1: #3, #14).

The notion of self-management is central to Scrum. The Scrum framework describes Scrum teams as *"self-managing, meaning they internally decide who does what, when, and how"* [134]. This follows the principle of *responsible autonomy* from sociotechnical systems (STS) [148]. The

team as a whole is made responsible for its composition, task distribution, and leadership to make them more resilient in unpredictable environments where managers cannot constantly regulate work. Other studies have also identified high autonomy as an important success factor for Agile teams [40, 89, 104, 110, 146]. Self-managing teams have been associated with a range of positive effects by organizational psychologists [82], such as increased productivity, job satisfaction, and higher commitment.

**Cross-functionality:** Another constraint to autonomy emerged from the skills present in teams. Some of the Scrum teams covered a diverse range of skills from testing, business analysis, and design to coding (Tab 1: #1, #3, #5, #6, #9). The range of skills was much narrower in other teams, often only coding (Tab 1: #2 and #7). We observed two consequences of low versus high functional diversity. First, when a team lacked a specific skill and was not available in the environment, they would forego tasks altogether (e.g., testing, user interface improvements, security hardening). This impacted the quality and usability of their product (Tab 1: #6 and #7). Second, when a team lacked the skill, and it was available in their environment, this effectively introduced a dependency on people outside the team who possessed that skill. These dependencies increased development time as teams had to wait for dependencies to complete the task (Tab 1: #1, #2, #5, #6, #10). For example, teams in case #1 (Table 1) frequently waited for a vendor to deploy new versions to one of a dozen environments. This delayed internal testing and caused confusion as to which version was running on which environments at a given moment in time. Another striking example from this case was how the initial composition of Scrum teams included skills for coding, design, and testing but not requirements analysis. This was initially done by a separate team and created a recurring dependency as Scrum teams ran into questions about the requirements on the one hand and delivered work that did not meet the requirements on the other. This resulted in frequent rework and bugs. When the requirement analysts later joined the Scrum teams, the department manager concluded:

> "this greatly improved quality now that testing is part of the development cycle" (Tab 1: #6, Department Manager)

In extant literature on work teams, Keller [80] describes cross-functional teams as consisting of members with different functional backgrounds, which contributes to *organizational ambidexterity* [40]. Functional diversity has been linked to shorter development time [50], increased quality and schedule performance [80] and higher performance [34, 79] in general work teams. For Scrum Teams specifically, Moe et al. [110] found that team effectiveness is inhibited when teams lack cross-functionality, as expressed by a high degree of skill specialization and division of work. Similarly, Melo et. al. [104] identified high inter-dependencies between teams or team members as an impediment to team productivity.

*4.4.5 Continuous Improvement.* The teams we studied also varied in the degree to which they engaged in continuous improvement. This manifested primarily in five areas. We summarize the observations and second-order themes for this aggregate dimension in Table 6.

**Sprint-Retrospective Quality:** The Sprint Retrospective is a recurring opportunity for teams to "plan ways to increase quality and effectiveness" [134]. The majority of the Scrum teams we observed organized a Sprint Retrospective every Sprint (Tab 1: #1, #3, #5, #6, #7, #8, #9, #12, #13). However, other teams did so less frequently (Tab 1: #2, #4, #10, #11). In terms of what was discussed during Sprint Retrospectives, we observed that some teams used different formats to address different themes (e.g., the Definition of Done, customer satisfaction, or code quality) (Tab 1: #1, #2, #3, #4, #5, #6, #9, #12). Though a small number of Scrum teams generally repeated the same format,

Table 5. Data structure of observations and themes for the aggregate dimension 'Team Autonomy'

| 1st Order Observations & concepts | 2nd Order Theme |
|---|---|
| The Scrum Master is distributing work and chairing the Planning and the Sprint Review | Self Management |
| The Product Owner has a mandate to make decisions about the work that the team spends time on | |
| The team is responsible for planning its work capacity | |
| Teams are involved in or have complete control over decisions that affect their composition or work method | |
| Dependencies slow teams down on external teams or individuals who need to perform tasks that they cannot do themselves | |
| Teams are given responsibility only for specific functional areas (e.g., development, testing, design) | |
| Individual team members are made responsible (and rewarded) for specific specializations | Cross-Functionality |
| Team members collaborate on tasks from the Sprint Backlog, regardless of their specializations | |
| Team members believe they are solely responsible for their specialization (e.g., "I'm only here to code") | |
| Team members actively keep each other up to date about what they are working on throughout the day | |
| Teams are organized in terms of functional specializations (e.g, a 'continuous delivery team' or 'a design team') | |
| The testing for a particular item on the Sprint Backlog is performed by someone other than the developer (in or outside the team) | |

usual variations of a "plus delta"[3] to discuss general improvements (Tab 1: #7, #10). Compared with the repeated use of the same format, we observed that a more diverse range of formats and themes tended to generate more improvements in a broader range of areas (see Figure 4).

The value of continuous improvement and organizational learning, particularly for complex tasks, is well established in extant literature. With its iterative nature and recurring opportunities for process improvements, the Scrum framework is a continuation of Deming's PDCA-cycle [37]. Every Sprint allows teams to detect and correct mismatches in their expectations, tools, processes, quality standards, and collaboration within their team and the broader organization. This is essentially a form of group-level *kaizen*, where groups of employees work together to "find and solve problems faced during their day-to-day work without any interference from management" [17]. In research on organizational learning, Argyris [6] describes this process of detecting errors and correcting them as "learning" and argues that it involves changing actions (single-loop) or challenging existing systems and governing variables (double-loop). Our findings are also congruent with Hoda & Noble [68], who found that the use of Sprint Retrospectives and other reflective practices became more focused and embedded as teams became more experienced with Agile practices.

**Concern for Quality** The Scrum teams we observed also varied in their concern for quality standards. This was most evident in their Definition of Done, which is used in the Scrum framework to clarify the "quality standards required for the product" [134]. While some teams frequently challenged their Definition of Done to reformulate or add criteria (Tab 1: #1, #3, #5, #8, #9, #12, #12), others never or rarely did (Tab 1: #2, #6, #10). One Scrum team did not have an explicit Definition of Done (#11). The concern for quality was also apparent in how new practices and technologies

[3]In this format, teams generally create two columns on a shared work-space. One column lists the things that went well (the "plus"), while another shows improvements (the "delta")

were explored to improve quality. While some teams frequently did so (Tab 1: #1, #3, #5), others rarely did (Tab 1: #2, #6, #7, #10).

We grouped this second-order theme with "continuous improvement" because we observed that teams that were highly concerned with quality also tended to engage more in continuous improvement overall. Clear quality standards, as captured in the Definition of Done, provide a clear baseline for improvements and gaps between current and desired quality.

**Psychological Safety** We also observed that Scrum teams diverged in how they addressed internal conflicts. For example, in one team (Tab 1: #6), an argument built up between two members over several weeks eventually erupted when one member suddenly left the team, and another ended up at home with a burnout. The Scrum Master observed that:

> "[...] people interrupt each other constantly and do not listen" (Tab 1: #6, Scrum Master)

We observed a similar pattern of interrupting and talking over each other in several other Scrum teams (Tab 1: #1, #4, #6, #7). However, other teams purposefully made time for their members to express personal concerns and frustrations (Tab 1: #3, #5, #8, #9). In one instance, the teams actively sought help from an outside coach to address growing frustrations (Tab 1: #1). In another instance, two team members (#3) expressed deep concern about whether or not their customers were satisfied with their ability to meet deadlines. The team scheduled a 1-hour session to talk things through. One member of this team explained the benefits of this as:

> "It is important for me to talk about what is buzzing around in my head so that I can move on" (Tab 1: #3, Team Member)

These observations are indicative of psychological safety. The organizational psychologist Schein [131] conceptualized this as a climate where people can focus on shared goals over self-protection. More recently, Edmondson [46] defined it as *"the shared belief that it is safe to take interpersonal risk"*. Psychological safety has been demonstrated to influence learning behavior, decision quality, and performance in work teams [47], also specifically in Agile teams [42, 44]. Moe, Dingsøyr & Dybå [110] conclude that without sufficient trust at the group level, *"team members will expend time and energy protecting, checking, and inspecting each other as opposed to collaborating to provide value-added ideas"*. The ability to give and receive peer feedback is one of the core components of effective teamwork in Agile teams [143].

**Shared Learning** While Scrum teams can often make substantial improvements to their processes, many improvements are bound to transcend individual teams. The degree to which Scrum teams actively involved others varied greatly. Some teams rarely involved others (Tab 1: #2, #4, #7, #10). However, other teams shared their learning with other teams in a "marketplace," or "Scrum of Scrums" (#1, #6), invited management (Tab 1: #3, #8, #9), organized cross-team workshops on shared challenges (Tab 1: #6, #12, #13) or started Communities of Practice (Tab 1: #1, #10, #12, #13).

The need to extend learning beyond team boundaries is also well-established in the extant literature on organizational learning. This process is most commonly referred to as "boundary crossing" [36, 78]. This literature treats teams as open systems that are embedded in larger systems where their effectiveness is negotiated on the team boundaries [36]. Decuyper et al. [36] argue that "Teams can neither learn nor work effectively if they hold to share knowledge, competency, opinions or creative ideas across their boundaries with the different stakeholders in the learning process". Frequent-boundary crossing increases efficiency and innovation [144] and increases knowledge

Fig. 4. Scrum Masters from one organization engaged in a shared learning session (Table 6)

transfer [5]. The need for inter-team learning is also recognized as an important coordination mechanism in large-scale Agile by Berntzen et al. [15].

**Learning Environment** Finally, we observed that the degree to which the environment of Scrum teams fostered learning directly significantly varied. In some cases, team members were actively encouraged by management to learn new technologies during work hours (Tab 1: #3, #4, #5, #12). In other cases, learning was considered more of a personal responsibility to perform outside of work hours (Tab 1: #8, #6, #10). For example, some organizations did not allow team members to participate in professional meetups during work hours (Tab 1: #6, #10), whereas others actively hosted public, professional meetups during work hours (Tab 1: #2, #5).

The relevance of a learning environment to continuous improvement is also central in the extant literature on organizational learning. The presence of a culture that nurtures and encourages learning and treats it as an integral part of work is essential for the learning organizations [6, 99]. If learning is not adequately incentivized, it will be hard to develop productive learning behaviors in organizations [54].

*4.4.6 Management Support.* The role of managers varied greatly between cases. In some cases, managers primarily took a supporting role by asking the teams what they needed from them (Tab 1: #1, #2, #3, #4, #5, #7, #10, #13, #14). Management remained distant from the teams in other cases (Tab 1: #9, #10, #11) or retained a directive style (Tab 1: #6, #7). In one extreme example, a top manager would occasionally leave a memo in the team room with sharp criticism about development speed (Tab 1: #6). A Scrum Master in this organization initially described this as:

Table 6. Data structure of observations and themes for the aggregate dimension 'Continuous Improvement'

| 1st Order Observations & concepts | 2nd Order Themes |
|---|---|
| Sprint Retrospectives always yield the same issues over extended periods | Sprint Retrospective Quality |
| Sprint Retrospectives are used to dive into deeper problems, not just the superficial symptoms | |
| The action items from Sprint Retrospectives are usually not implemented | |
| A lot of bugs and issues are discovered during the Sprint review or quickly after | Concern for Quality |
| Team members often talk about how to improve the quality of their work (e.g., during Sprint Retrospectives) | |
| The Definition of Done does not define what high-quality looks like for a team | |
| Team members are quick to shoot down ideas (e.g., "We've tried that before...", "That doesn't work here") | Psychological Safety |
| Team members generally offer vague and unclear contributions to the Daily Scrum (e.g., "I've was busy yesterday, I will be busy today") | |
| Team members make an effort to listen to each other before sharing their views | |
| Team members openly ask for help when they are stuck | |
| When a mistake is made, people are not singled out or punished for it | |
| Teams share their learnings and challenges with other teams in the organization | Shared Learning |
| Teams involve people from outside the team to work together to overcome challenges they face | |
| Team members express a desire to try new things or experiment with new technologies | Learning Environment |
| Team members are motivated or encouraged to learn new things, read professional books, visit conferences or meetups | |

> "[...] management only performs lip service to Scrum, because they are rarely present to help us" (Tab 1: #6, Scrum Master)

In several cases, we observed how managers supported teams in expanding their skills. This involved training and personal development in some cases (Tab 1: #3 and #5), adding members with missing skills (Tab 1: #1, #2, #3, #4), or giving teams the freedom to change their composition as needed (Tab 1: #5, #6, #14). We also observed instances where managers actively managed the boundaries of the Scrum teams. For example, in one case (Tab 1: #3), the company owner openly renegotiated the distribution of responsibilities for particular activities (sales, salary setting, company vision) with the team. In another (Tab 1: #1), the program manager moved deployment engineers from an external vendor into each of the teams to remove a dependency that made it difficult for teams to deploy quickly.

Because Scrum teams are self-managing, managers have to shift their approach to what Manz et al. [97] describe as "leading others to lead". The locus of control for work-related decisions shifts from external managers to the teams themselves [33, 90, 115, 154]. Russo [124] investigated success factors for Agile transformations and found that commitment from top management is an important predictor. This requires management to understand why teams work with Scrum, to invest in skills and technologies, and observe the autonomy of teams [124]. These findings are congruent with research about Agile teams specifically [33, 154]. Young & Jordan [161] provide evidence based on case studies that support from top management may be the most critical factor

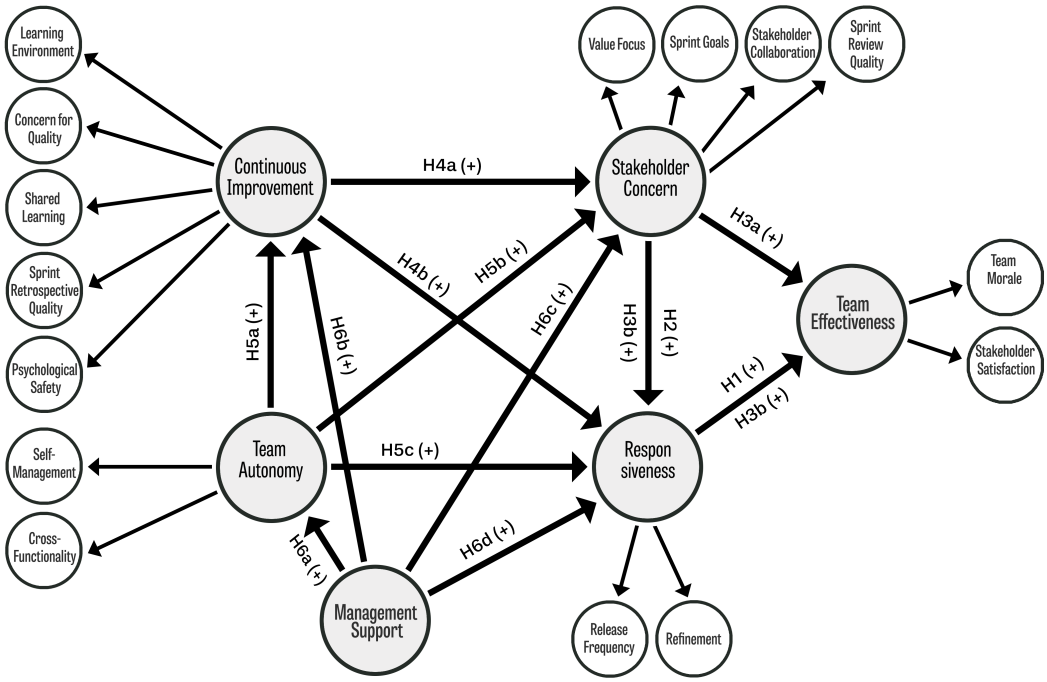


Fig. 5. Theoretical model and hypotheses.

to project success. However, a survey study among 109 Agile projects by Chow & Cao [25] did not show a significant effect of management support on project success.

Table 7. Data structure of observations and themes for the aggregate dimension 'Management Support'

| 1st Order Observations & concepts | 2nd Order Themes |
|---|---|
| Management predominantly tells team members how to perform their work, rather than asking where they can support them | Management Support |
| Management understands why this team works with Scrum | |
| Management is available to help teams remove impediments | |

## 5 THEORY DEVELOPMENT AND HYPOTHESES

The qualitative phase of our study aimed to induce a theoretical model. To answer $RQ_1$, this model should explain how the induced team-level factors influence team effectiveness. Regarding $RQ_2$, we aim to develop a model that can be generalized across settings. However, the exploratory nature of our case studies and their natural research setting [142] are not suited to statistical generalization to teams beyond those studied. This is why the second phase of our study aims to test our theoretical model with quantitative methods and a substantial sample of Scrum teams. We will now present this model and develop testable hypotheses. Our six hypotheses, and their sub-hypotheses, are visualized in figure 5.

As described in section 4.1, we induced "Stakeholder Satisfaction" and "Team Morale" as indicators of team effectiveness. Other outcomes of effective Scrum have been identified in the literature, like increased quality [24] and project success [124]. However, organizational psychologists often

define team effectiveness as *"the degree to which a team meets the expectations of the quality of the outcome"* [62]. In this sense, stakeholder satisfaction is the evaluation of team outcomes from the external perspective of stakeholders, whereas team morale is the evaluation of team outcomes from the internal perspective of team members. This is conceptually similar to how team effectiveness is defined in the "Team Diagnostic Survey (TDS)" by Wageman, Hackman & Lehman [157], and adds increased interdependence between members as a third criterion. In software development, team effectiveness is often understood more broadly to include developer productivity (e.g., lines of code, hand-offs, merge times, velocity), such as the SPACE framework by Forsgren et al. [52]. However, productivity and team performance are highly contextualized. What is considered highly productive in one organization may be low in another. Mathieu et al. [102] note that such objective measures are difficult to compare and aggregate across organizations, as we aim to do in this study. So similar to other studies that have investigated team effectiveness [41, 84, 120], we use the satisfaction of team members and stakeholders as a composite measure of team effectiveness.

We propose that the effectiveness of Scrum teams depends on five team-level factors: (a) the ability of Scrum teams to respond quickly to emerging needs, (b) the awareness that teams have of the needs of their stakeholders, (c) the autonomy that teams have to perform their work as they see fit, (d) a climate of continuous improvement and (e) support from management. Based on the findings from the case studies and supported by extant literature, we propose a causal process by which these factors interact to influence team effectiveness.

One factor that is central to our model is responsiveness. This is the capability of Scrum teams to respond rapidly to changing needs and emerging insights through the frequent release of increments. In the first phase of our research, we identified two relevant indicators for responsiveness: the release frequency of a team and their ability to refine large chunks of work into smaller chunks of work (see section 4.1). Higher responsiveness is effectively what distinguishes Agile methodologies from more plan-based approaches [12, 133]. Indeed, studies have shown that teams with a frequent delivery strategy are more likely to deliver successful project outcomes than teams that do not [25, 74]. We expect that responsiveness positively influences our indicators of team effectiveness. Regarding stakeholder satisfaction, we expect that more responsive Scrum teams have more satisfied stakeholders. Responsive teams can more quickly respond to emerging needs and deliver valuable outcomes to their stakeholders. Regarding team morale, we expect that more responsive Scrum teams will experience higher team morale. As we also observed in the case studies, frequent releases to stakeholders make the work more fulfilling, purposeful, and motivating for team members.

Hypothesis 1 (H1). *The responsiveness of a team is positively associated with team effectiveness.*

While responsiveness is a predictor of team effectiveness in our model, we expect responsiveness alone is not enough. It is also essential for Scrum teams to ensure that each release addresses required stakeholder needs. Otherwise, a high release frequency only results in a stream of increments of questionable value to stakeholders. We expect that Scrum teams are most effective when they combine responsiveness with a shared understanding of what their stakeholders need, and not just one or the other. In the qualitative phase of our research, we defined this as "Stakeholder Concern" and identified Sprint Reviews, stakeholder collaboration, value focus, and shared goals as relevant indicators. "Stakeholder Concern" can be thought of as a collection of team mental models about what is valuable to stakeholders from the perspective of team cognition [22, 123]. Cannon-Bowers & Salas [21] argue that team mental models contribute to behavioral coordination on interdependent tasks, the motivational state and cohesion of its members, and their ability to meet objectives. Team cognition has been shown to explain a significant amount of variance in

team performance [34]. Specifically for Scrum teams, Moe, Dingsøyr & Dybå [110] report that a lack of shared mental models about desirable team outcomes impedes cross-functionality and communication. We observed that Product Owners play a critical role in developing this shared understanding, which is supported by literature [9, 10, 150]. Product Owners can use different strategies to promote collective product ownership [75, 86, 149] The use of shared goals to clarify business objectives is one way [86]. There is extensive evidence for the positive influence of clear, shared goals on team effectiveness [7, 31, 72], cohesion [95], motivation [82] and cooperation [111].

We expect that stakeholder concern has two complementary effects on team effectiveness. The first is that Scrum teams that have a deep concern for stakeholder needs are more likely to see the benefit of releasing more frequently to those stakeholders and are thus more inclined to be responsive:

Hypothesis 2 (H2). *Stakeholder concern is positively associated with responsiveness.*

Second, we hypothesize that stakeholder concern is positively associated with our two indicators of team effectiveness. We observed in the qualitative phase of our research that teams found it easier to satisfy the needs of stakeholders when they understood those needs clearly. This also mirrors findings by Hoda, Stuart & Marshall [69] and Van Kelle et. al. [153]. Regarding team morale, we expect that a genuine concern for stakeholder needs bolsters the sense of purpose of a team. This finding is well-grounded in positive psychology [2, 101]. The psychological experience that one's work is meaningful encourages people to engage in that work and push through even in the face of challenges [71, 77, 130, 151].

In addition to this direct effect on Scrum team effectiveness, we also expect to see a mediating effect on responsiveness. This reflects a core principle of Agile methodologies that a strong focus on stakeholder needs has to be coupled with the capability to respond to those needs. Otherwise, it essentially remains a waterfall approach where stakeholders have to trust in a good outcome at the end of the development cycle [12]. We expect that Scrum teams are most effective when they release frequently *and* focus on the needs of stakeholders.

Hypothesis 3 (H3). *Stakeholder concern is positively associated with team effectiveness (H3a), and this relationship is further mediated by responsiveness (H3b)*

Our first three hypotheses capture how responsiveness and stakeholder concern interact at the team level to influence team effectiveness. Informed by the case studies and extant literature, we identified three additional team-level antecedents: a climate of continuous improvement, team autonomy, and management support.

We begin with the first antecedent, continuous improvement. Every Sprint offers opportunities to detect mismatches in tools, processes, and quality standards and adjust accordingly. This process of detecting errors and correcting them is what Argyris [6] defines as "learning". We observed that Scrum teams varied in the degree to which they engaged in continuous improvement, as indicated by the quality of Sprint Retrospectives, psychological safety, quality standards to guide improvement, and shared learning with others in the organization. These factors are reminiscent of work by Marsick & Watkins [99] on learning organizations, which they characterize as having "total employee involvement in the process of collaboratively conducted, collectively accountable change directed towards shared values or principles". Out of the seven dimensions they identify in the "Dimensions for Learning Organizations Questionnaire" (DLQP), they include a climate of openness and safety, recurring opportunities for learning, shared learning, and a collective vision to guide improvements. Hoda & Noble [68] also emphasize the role of effective learning as a means

for Scrum teams to overcome obstacles. We expect that Scrum teams that engage in continuous improvement are more likely to have overcome barriers to responsiveness and stakeholder concern and are thus more effective as a result.

Hypothesis 4 (H4). *Continuous improvement is positively associated with stakeholder concern (H4a) and responsiveness (H4b).*

The second antecedent concerns team autonomy. In the case studies, we observed that team autonomy was expressed in autonomy from external constraints (self-management) and autonomy due to fewer internal skill constraints (cross-functionality). Consistent with self-determination theory by Deci & Ryan [35], we expect that the shift from an external to an internal locus of control encourages more proactive behavior in teams towards achieving shared goals while also making them feel more responsible for the outcomes. Specifically, we hypothesize that high-autonomy teams are more inclined to engage in continuous improvement than low-autonomy teams. Many studies have shown that job autonomy increases proactive behavior to improve the work environment [3, 114, 147]. Specifically for Agile teams, Junker et al. [76] found that teams are more likely to initiate changes and improvements when autonomy is high. Similarly, we hypothesize that the increased sense of responsibility for their outcomes will lead high-autonomy teams to show greater concern for the needs of stakeholders and more proactive behaviors aimed at understanding those needs. Finally, we expect that high-autonomy teams are more responsive than low-autonomy teams due to decreased external dependencies.

Hypothesis 5 (H5). *Team autonomy is positively associated with continuous improvement (H5a), stakeholder concern (H5b), and responsiveness (H5c).*

Finally, we learned from the case studies how support from management varied between Scrum teams. Observations from the case studies also suggest that management support is a critical success factor. This is consistent with other studies that have investigated how management supports contributes to the success of Agile adoption [18, 124, 154] and change initiatives in general [161]. However, Chow & Cao [25] did not find a significant effect on Agile project success. Hoda & Noble [68] conceptualize the required change in management approach during Agile transitions as shifting from "driving" to "empowering". Rather than making decisions for teams, management ensures that teams have what they need to make decisions for themselves. Specifically for Scrum teams, we expect that support from management manifests through the other success factors in our model. So management contributes to team effectiveness by removing barriers to autonomy, continuous improvement, stakeholder concern, and responsiveness.

Hypothesis 6 (H6). *Management Support is positively associated with team autonomy (H6a), continuous improvement (H6b), stakeholder concern (H6c), and responsiveness (H6d).*

## 6 PHASE II: LARGE-SCALE CROSS-SECTIONAL STUDY

After the induction of the theoretical model described in Section 4.1, we will now evaluate it to assess its generalizability.

### 6.1 Collection of Quantitative Data

We performed our data collection process through a customized online survey between August 2020 and October 2021 [4]. The survey has been designed so that Scrum teams can self-assess their development process. The team member visiting the survey website would first complete the survey for their team and then invite other members to participate anonymously through a unique link for their team. This allowed us to aggregate the results from individual participants to the team level. Upon completion of the survey, we provided teams with an automatically generated and anonymized report of their results, allowing them to take concrete actions for improvements. We operationalized the sixteen second-order themes identified in the multiple case studies with Likert scales, ranging from 1 (Completely disagree) to 7 (Completely agree). We also added scales to control for common method bias and to control for the impact of Covid-19. The instrument is available in Table 15 in the Appendix. We adapted established scales to Scrum teams from existing literature where possible. Alternatively, we created new scales from first-order concepts we collected in the multiple case study. For example, we operationalized the independent variable *Sprint Review Quality* with two questions: "The Product Owner of this team uses the Sprint Review to collect feedback from stakeholders" and "During Sprint Reviews, stakeholders frequently try out what this team has been working on during the Sprint". As another example, the dependent variable *Stakeholder Satisfaction* was measured with the questions "Stakeholders frequently compliment this team with their work.", "Stakeholders are generally happy with the software this team delivers.", "Stakeholders are generally happy with how fast this team responds to their needs." and "Our stakeholders compliment us with the value that we deliver to them.". An overview of the scales is shown in Table 8. The complete questionnaire is available in Table 15 in the Appendix. To improve the usability and measurement reliability of the sample study, we performed three survey pilots between June 2019 and June 2020.

After the pilots, we employed three strategies to reduce response bias. First, we incentivized participation by offering participants a personalized report to support their teams. The report included team scores and a benchmark derived from the overall and aggregated team responses. In addition, teams received actionable improvement suggestions based on their scores drawn from a professional book [156]. Second, we sent reminders to participants who started the survey but did not complete it within two days. The survey was advertised on channels that are frequented by people interested in Scrum, ranging from industry platforms, blog posts, podcasts, and videos by industry influencers[5]. Third, we strongly emphasized the anonymous nature of the survey. Although participants could leave their e-mail addresses to receive research updates, this was not required. The personalized report we provided to teams only showed team-level averages when a sufficient number of members participated but never individual results.

In the pilot studies, we observed that many participants first tested the survey with a quick test run with fake answers and then returned for an actual attempt. To prevent such responses from influencing our results, we added an item at the start of the survey to ask respondents if their answers reflected a real team or were meant to test the survey.

The use of a single method - like a survey - introduces the potential for a systematic response bias where the method itself influences answers [118]. To control for such common method bias, the most objective approach in current literature is the use of a marker variable that is theoretically unrelated to other factors in the model [138]. We included three items from the social responsibility scale (SDRS5) [65] and found a small but significant unevenly distributed response bias. Following

[4]The survey is available at the following URL: www.scrumteamsurvey.org.

[5]Due to the opt-in nature of the survey, we were not able to compute a meaningful response rate. However, we were able to establish that 42% of the participants who started the survey completed it.

recommendations in the literature, we retained the marker variable "social desirability" in our causal model to control for common method bias [138].

### 6.2 Analysis of Quantitative Data

For our sample, we began with all individual responses from June 3, 2020, and onward ($N = 5,324$). Because our survey was public, we applied several strategies to remove careless responses from the sample and prevent them from biasing the results [103]. We first discarded 145 responses where the team's name indicated a test, e.g., "Fake" or "Test." To further reduce the impact of careless responses, we removed 239 participants that completed the survey below the 5% percentile (6.87 minutes) or entered only a few questions. We identified a handful of multivariate outliers based on their Mahalanobis distance but did not remove them. These cases may represent unusual cases, and their removal should be done with caution [32]. Our final sample contained 4,940 individuals from 1,978 teams. The composition of the sample is detailed in Table 9. Because we wanted to analyze our measures at the team level, we calculated a team-level mean for each item in the survey by aggregating it to a unique key that was assigned to each team. Such aggregation is justified when at least 10% of the variance exists at the team-level [63]. We tested this by calculating the team level's Intraclass Correlation (ICC). This measure describes the resemblance of individuals within a higher-level grouping [63]. We found that 51% of the variance existed at the team level, which exceeded the required threshold of 10% suggested by Hair et al. Accordingly, we ran a post-hoc power analysis using G*Power [51] version 3.1. Such a test allows researchers to determine the probability of correctly rejecting the null hypotheses, given the sample size and the effect size ($f$). We determined that the sample size allows us to correctly capture small effect sizes ($f = .02$) with a statistical power of almost 100% ($1 - \beta = .999$). In other words, we are very confident that our sample is big enough to provide a reliable outcome of our SEM analysis.

Next, we tested our data for necessary statistical assumptions to run our SEM analysis. We assessed normality by inspecting the skewness and kurtosis of individual items. The skew for all items remained well below the recommended thresholds for kurtosis ($< 3$) and skew ($< 2$) [32]. We tested linearity by entering all pairs of independent and dependent variables into a curve estimation [55] to see if the relationship was appropriately linear, which was significant in all cases ($p < 0.01$). We assessed homoscedasticity by inspecting the scatter plots for all pairs of independent and dependent variables for inconsistent patterns but found none. Finally, multicollinearity was assessed by entering all independent variables one by one into a linear regression [55]. The Variance Inflation Factor (VIF) remained below the critical threshold of 10 [63] for all measures but indicated some multicollinearity of Psychological Safety with other independent variables (between 7.38 and 9.81).

Finally, we identified the most appropriate strategy for dealing with missing data. Unless data is missing completely at random (MCAR), any patterns in missing data may bias the results of multivariate analysis [63]. For this, we calculated Little's MCAR test. This is a Chi-Square test that compares the observed patterns of missing data with the patterns that would be expected from a process that results in random missing data [63]. Our test showed that data wasn't completely missing at random ($Chi^2 = 13,200.799$, $df = 9,853$, $p < 0.001$) [63]. A closer inspection of patterns in the missing data revealed that the percentage of missing data was below 2% for most items but slightly higher (up to 3.8%) for four items that measured aspects relating to stakeholders. Overall, missing data remained below the recommended threshold of 10% [63]. Because the percentage of missing data was modest, and to prevent list-wise deletion of cases and lost information, we performed EM maximum likelihood imputation in SPSS [20].

We analyzed the data through Structural Equation Modeling (SEM) with the AMOS software package [4] to effectively test the multiple relationships between our independent and dependent

variables at the same time [20, 63, 127]. Our theoretical model is composed of a structural (or inner) component to describe as many relationships between independent and dependent variables as the researcher is looking for and a measurement model that allows the inclusion of multiple indicators to measure latent factors and account for measurement error [83]. It is important to note that this makes SEM an inherently confirmatory approach that combines multiple linear regressions and confirmatory factor analysis (CFA) with Maximum Likelihood estimation (ML) to produce more consistent and less biased estimates than those derived through Ordinary Least Squares (OLS) that is typically used in multiple regression and ANOVA [63]. Researchers evaluate their model through several "Goodness of Fit" indices and individual paths' statistical significance and effect sizes. The aim is to arrive at a model as parsimonious as possible while also providing a good fit (and thus explanatory power). We discuss the fit indices used in this paper in section 6.3.

Based on our theoretical model, we created a full latent variable model that contained both the measurement model and the structural model. The measurement model defines relationships between indicator variables (survey items) and underlying first-order latent factors and effectively acts as a CFA-model [83]. The structural model defines the hypothesized relations between latent variables and is effectively a regression model. This approach makes the results less prone to convergence issues because of low indicator reliability and offers more degrees of freedom to the analysis compared to a non-latent model [64]. Team effectiveness was modeled as a latent factor and measured through team morale and stakeholder satisfaction. Similarly, we modeled the five team-level factors from the case studies as latent factors, each measured through the associated themes that we identified in the case studies. In the resulting path model, team effectiveness is effectively a dependent variable with the other factors as independent variables that directly or indirectly influence it.

Following the approach outlined in literature [20, 63, 127], we entered all items from our survey as indicator variables into the model, with each indicator variable loading on a single first-order latent factor that represented the associated (scale) construct and first-order latent factors loading on the five proposed second-order factors. Then, we used AMOS to calculate factor scores and errors for each latent factor based on their indicators.

As recommended by the literature, we first evaluated and fitted the measurement model and then introduced the structural part of the model [20, 83].

### 6.3 Model fit evaluations

Before testing for the model fit, we assessed reliability, convergent, and discriminant validity for the resulting measurement model. The individual steps involved in the model-fitting process are included in Table 14 in the Appendix. Discriminant validity was assessed by analyzing the heterotrait-monotrait ratio of correlations (HTMT) with a third-party plugin in AMOS [56] and following the approach outlined in literature [63, 66]. This ratio between trait correlations and within trait correlations should remain below $R = .90$ to indicate good discriminant validity from other constructs in different settings. This was the case for all measures. We assessed convergent validity by inspecting composite reliability (CR) and average extracted variance (AVE). The AVE remained above the rule of thumb of > .50% [63] for all pairs of factors, ranging between .583 and .875. The CR was equal to or above the threshold of .7 [63] for all scales, except Sprint Review Quality ($CR = .689$). We decided to retain this scale due to the modest violation and its theoretical relevance.

We then proceeded with the fitting procedure. AMOS suggested three covariances between items from the same scales (PS2 and PS3, PS1 and PS4, VF2 and VF3) based on high modification indices. A closer inspection showed that the pairs used wording more similar than other items of the same scales, so we allowed the covariances. Finally, we investigated local fit by inspecting the

Table 8. Scales used in the survey study, along with attribution, number of items and reliability (Cronbach's Alpha) based on respondent-level response data ($N = 4,919$)

| Construct variable | Items adapted from | # Items | $\alpha$ |
|---|---|---|---|
| Concern for Quality | Created by authors from our case studies | 2 | .786 |
| Shared Learning | Created by authors from our case studies | 3 | .790 |
| Learning Environment | Adapted from 'Continuous Learning' scale in DLOQ [100] | 2 | .725 |
| Sprint Retrospective Quality | Created by authors from our case studies | 3 | .834 |
| Psychological Safety | Adapted from 'Inquiry & Dialogue' scale in DLOQ [100] | 5 | .862 |
| Shared Goals | Adapted from Van der Hoek, Groeneveld & Kuipers [152] | 2 | .924 |
| Stakeholder Collaboration | Created by authors from our case studies | 3 | .852 |
| Sprint Review Quality | Created by authors from our case studies | 2 | .691 |
| Value Focus | Created by authors from our case studies | 3 | .799 |
| Cross-Functionality | Adapted from Edmondson [46] | 2 | .735 |
| Self-Management | Adapted from Langfred [88] | 3 | .803 |
| Refinement | Created by authors from our case studies | 3 | .820 |
| Release Frequency | Created by authors from our case studies | 2 | .860 |
| Management Support | Created by authors from our case studies | 2 | .852 |
| Team Morale | Adapted from Van Boxmeer et. al. [151] and Schaufeli [130] | 3 | .894 |
| Stakeholder Satisfaction | Created by authors from our case studies | 4 | .842 |
| Social Desirability | Highest-loading items from SDRS-5 scale [65] | 3 | .688 |
| Covid-19 Impact ($N = 928$) | Created by authors | 2 | .922 |

residual covariance matrix. A standardized residual covariance is considered large when it exceeds 2.58 [20]. This indicates that an item does not sufficiently measure (only) its intended factor. We found two items, SH1 and PS5, with multiple violations and removed them from the model. No further modifications were made.

We then evaluated the goodness of fit using indices recommended by recent literature [20, 63, 83]; the Comparative Fit Index (CFI), the Root Mean Error of Approximation (RMSEA), the Standardized Root Mean Residual (SRMR) and the Tucker Lewis Index (TLI). A commonly used index that we reported but did not test for was $\chi^2$ (CMIN) and its corollary CMIN/df. These indices are highly susceptible to type I errors in larger samples ($N > 400$, [63]). So instead, we used the Comparative Fit Index (CFI) [14] which offers a similar test but with consideration of the sample size and its reliable properties have made it the most commonly used index today [63]. A cut-off value of .95 or higher is generally considered to indicate good fit [13, 20, 63]. The Root Mean Error of Approximation (RMSEA) by Steiger & Lind [140] also provides an index that considers sample size but adds to this a parsimony adjustment that leads it to favor the simplest model out of potential models with the same explanatory power [83]. A value below .05 is generally considered to indicate a good fit [20, 63]; additionally, we follow the advice to report the confidence interval in addition to only the absolute value [23]. The Standardized Root Mean Residual (SRMR) calculates a standardized mean of all the differences (residuals) between each observed covariance and the hypothesized covariance between variables [63]. A value below .05 is indicative of a good fit. We also inspected local fit by looking at the standardized residuals between pairs of variables, with values beyond 2.58 as a cut-off value for poor local fit [20]. Finally, we report and test the Tucker Lewis Index (TLI). This is another incremental fit index, like the CFI, that compares the relative improvement of the hypothesized model from a model where all variables are uncorrelated. Hair et al. [63] considers a value of .92 or above sufficient to conclude a good model fit. In addition to overall model fit, we

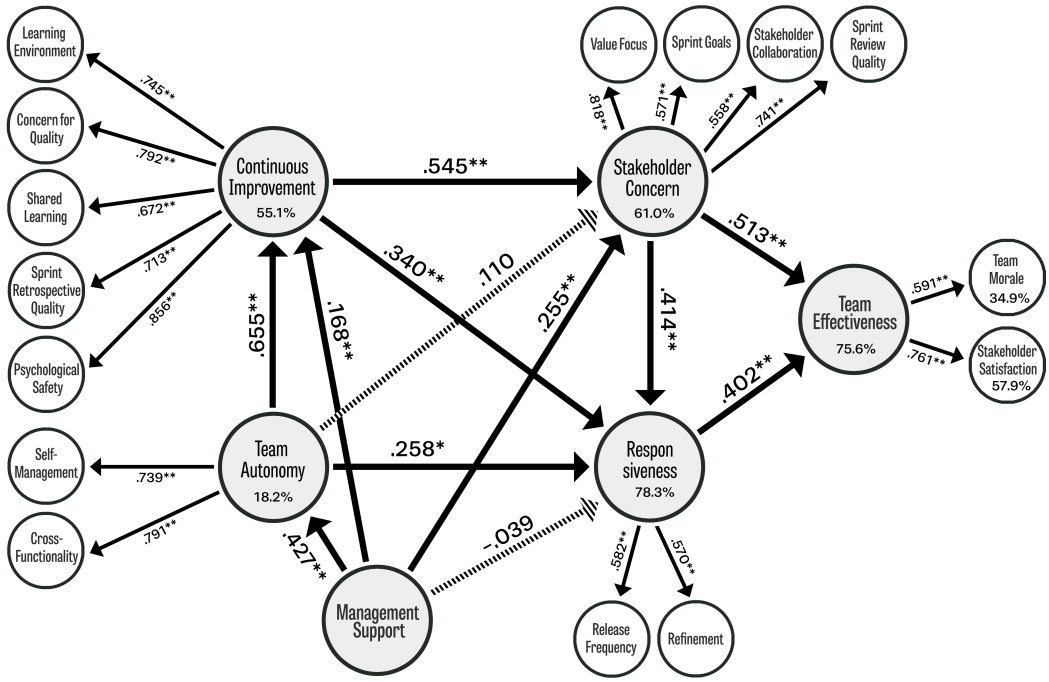


Fig. 6. Standardized factor loadings and standardized path coefficients for the research model ($** : p < .01$, $* : p < 0.05$). Squared Multiple Correlations are reported at the top right of each endogenous variable. The model shows first- and second-order latent factors. The indicator items are excluded from the model for the sake of clarity.

also evaluated our model on the percentage of variance that is explained in team effectiveness by all other variables in the model.

We also report a very good model fit of our data ($Chi^2(763) = 2.324,465$; $TLI = .963$; $CFI = .970$; $RMSEA = .032$; $SRMR = .028$). A Confirmatory Factor Analysis (CFA) is reported in the Appendix (Table 16) that shows that all items load primarily on their intended second-order factors. The cumulative Eigenvalues of 17 factors explain 79% of the total observed variance, which is well beyond the recommended threshold of 60% [63].

We then proceeded to test the path model for the effects we predicted from our theory. Our hypothesized theoretical model fits the data well on each of the fit indices, as described in Table 10: $Chi^2(710) = 2,757.805$; $TLI = .953$; $CFI = .959$; $RMSEA = .038$; $SRMR = .035$. The predictors in our model explain respectively 57.9% of the variance in stakeholder satisfaction and 34.9% of the variance in team morale. For studies in the social sciences, values above 26% are considered large [27].

All steps in the fitting process are summarized in Table 14 in the Appendix.

### 6.4 Hypothesis Testing and Interpretation

We reported the means, standard deviations, and (Pearson) correlations of all variables in Table 17 in the Appendix. Following recommendations in statistical literature [20, 83], we used a bootstrapping procedure with 2,000 samples and 95% bias-corrected confidence intervals to more accurately estimate parameters and their $p$-values for direct effects, factor loadings, and the hypothesized indirect effects. This resulted in a standardized, bias-corrected estimate ($\beta$) for each path, along

Table 9. Composition of the sample

| Variable | Category | N (%) |
|---|---|---|
| Respondents | | 4,940 |
| Teams | | 1,978 |
| Respondent role | Scrum Master | 1,216 (24.6%) |
| | Product Owner | 465 (9.4%) |
| | Developer (inc tester, analyst, design) | 2,631 (53.6%) |
| | Other | 601 (12.2%) |
| | Unknown | 27 (0.2%) |
| Respondents per team | 1 respondent | 1,366 (69.1%) |
| | 2-4 respondents | 255 (11,4%) |
| | 5-8 respondents | 294 (14.9%) |
| | 9+ respondents | 23 (4.7%) |
| Product Type | Product for internal users | 1,057 (53.4%) |
| | Product for external users / customers | 894 (45.2%) |
| Scrum Team Size | 1-4 members | 93 (4.7%) |
| | 5-10 members | 1,404 (71.0%) |
| | 11-16 members | 364 (4.9%) |
| | >16 members | 60 (5.1%) |
| | Unknown | 27 (1.4%) |
| Team Experience | No experience with Scrum | 79 (4.0%) |
| | Low experience with Scrum | 463 (23.4%) |
| | Moderate experience with Scrum | 898 (45.4%) |
| | High experience with Scrum | 525 (26.5%) |
| | Unknown | 13 (0.7%) |
| Organization Sector | Technology And Telecommunications | 745 (37.7%) |
| | Financial | 380 (19.2%) |
| | Government | 106 (5.4%) |
| | Other | 747 (37.8%) |
| Organization Size | 1-50 employees | 184 (9.3%) |
| | 51-500 employees | 537 (27.1%) |
| | 501-5,000 employees | 679 (34.3%) |
| | >5,000 employees | 499 (25.2%) |
| Region | Europe | 1,309 (66.2%) |
| | North America | 253 (12.8%) |
| | Asia & Oceania | 157 (7.9%) |
| | South America | 98 (5%) |
| | Africa & Middle East | 36 (1.8%) |
| | Other & Global | 125 (6.3%) |

with a $p$-value to test the null hypothesis of there being no effect at all. The parameter estimates for our hypotheses are reported in Table 11, with additional estimates and factor loadings available in Table 18 in the Appendix.

The results allowed us to confirm ten out of twelve hypotheses. We found a significant and positive association between teams' responsiveness and team effectiveness (H1, $\beta = .402, p < .01$). So more responsive teams are also more effective in terms of their team morale and stakeholder satisfaction. Regarding stakeholder concern, we found a direct positive effect on responsiveness (H2, $\beta = .414, p < .01$). This supports the notion that a high concern for stakeholder needs drives teams to be more responsive. We also hypothesized that stakeholder concern influences team

Table 10. Model Fit Indices

| Model fit index | Value | Interpretation |
|---|---|---|
| Chi-Square ( $\chi^2$) | 2,757.805 | n/a |
| Degrees of freedom (df) | 710 | n/a |
| CMIN/df | 3.884 | A value below 5 indicates an acceptable model fit [98], below 3 a good fit [70] |
| Root Mean Square Error of Approximation | .038 | Values $\leq$ .05 indicates good model fit [20] |
| RMSEA 90% CI | .037-.040 | |
| $p$ of Close Fit (PCLOSE) | 1.000 | Probability that RMSEA $\leq$ 0.05, where higher is better |
| Comparative Fit Index (CFI) | .959 | Values $\geq$ .95 indicates good model fit [63] |
| Tucker Lewis Index (TLI) | .953 | Values $\geq$ .92 indicates good model fit [63] |
| Standardized Root Mean Square Residual (SRMR) | .035 | Values $\leq$ .05 indicates good model fit [63] |
| Team effectiveness: Variance explained by predictors ($R^2$) | 75.6% | Values $\geq$ 26% indicates large effect [27] |
| Stakeholder satisfaction: Variance explained by predictors ($R^2$) | 57.9% | Values $\geq$ 26% indicates large effect [27] |
| Team morale: variance explained by predictors ($R^2$) | 34.9% | Values $\geq$ 26% indicates large effect [27] |

effectiveness, both directly (H3a) and mediated through responsiveness (H3b). This means that a high concern for stakeholder needs is not in itself beneficial when a team is not also responsive. The data supported partial mediation but not full mediation. Full mediation exists when there is only an indirect significant effect between two variables but not a direct one [8]. We did find a significant direct effect between stakeholder concern and team effectiveness (H3a, $\beta = .513, p = .02$) and also a significant indirect effect from stakeholder concern to team effectiveness (H3b, $\beta = .166, p < .01$). So while stakeholder concern in itself is a solid contributor to Scrum team effectiveness, the most effective Scrum teams are those that are pairing it with high responsiveness.

With responsiveness and stakeholder concern as the central contributors to team effectiveness, we hypothesized three team-level antecedents that create the right conditions. First, we hypothesized that teams are more likely to become responsive and develop a concern for stakeholders when they operate in a climate that encourages continuous improvement. We found support for this through a significant effect from continuous improvement to stakeholder concern (H4a, $\beta = .545, p < .01$) and (H4b, $\beta = .340, p < .01$). In a similar vein, we found support for the positive influence of team autonomy. A positive effect was found on continuous improvement (H5a, $\beta = .655, p < .01$) and on responsiveness (H5c, $\beta = .258, p = .02$), but not on stakeholder concern (H5b, $\beta = .110, p = .074$). So the more autonomous teams are, the more likely they are to improve continuously and the more responsive they can be.

For our final hypothesis, we expected that support from management is an important enabler for the other factors in our model. And while we found a positive effect from management support on team autonomy (H5a, $\beta = .427, p < .01$), continuous improvement (H5b, $\beta = .168, p < .01$) and stakeholder concern (H5c, $\beta = .255, p < .01$), management support was not significantly associated with responsiveness (H5d, $\beta = -.039, p = .466$).

Table 11. Parameter Estimates, Confidence Intervals, Standard Errors, and Standardized Coefficients for Direct Effects and Indirect effects for hypotheses (statistically significant hypotheses at $p < 0.05$ are set in boldface)

| Parameter | Unstandardized | 95% CI | SE | $p$ | Standardized |
|---|---|---|---|---|---|
| | | *Direct Effects* | | | |
| **H1: Responsiveness → Team Effectiveness** | **.235** | **(.111, .418)** | **.050** | **.007** | **.402** |
| **H2: Stakeholder Concern → Responsiveness** | **.400** | **(.265, .544)** | **.067** | **.002** | **.414** |
| **H3a: Stakeholder Concern → Team Effectiveness** | **.291** | **(.138, .417)** | **.047** | **.015** | **.513** |
| **H4a: Continuous Improvement → Stakeholder Concern** | **.616** | **(.491, .749)** | **.061** | **.001** | **.545** |
| **H4b: Continuous Improvement → Responsiveness** | **.372** | **(.166, .582)** | **.082** | **.008** | **.340** |
| **H5a: Team Autonomy → Continuous Improvement** | **.710** | **(.61, .825)** | **.054** | **.001** | **.655** |
| H5b: Team Autonomy → Stakeholder Concern | .135 | (.011, .273) | .061 | .074 | .110 |
| **H5c: Team Autonomy → Responsiveness** | **.306** | **(.110, .513)** | **.081** | **.015** | **.258** |
| **H6a: Management Support → Team Autonomy** | **.251** | **(.208, .292)** | **.020** | **.001** | **.427** |
| **H6b: Management Support → Continuous Improvement** | **.107** | **(.067, .147)** | **.020** | **.001** | **.168** |
| **H6c: Management Support → Stakeholder Concern** | **.184** | **(.143, .226)** | **.021** | **.001** | **.255** |
| H6d: Management Support → Responsiveness | -.027 | (-.078, .029) | .028 | .466 | -.039 |
| | | *Indirect Effects* | | | |
| **H3b: Stakeholder Concern → Responsiveness → Team Effectiveness** | **.094** | **(.044, .225)** | **.138** | **.004** | **.166** |
| **Continuous Improvement → Responsiveness → Team Effectiveness** | **.087** | **(.047, .152)** | **.092** | **.004** | **.137** |
| **Team Autonomy → Responsiveness → Team Effectiveness** | **.072** | **(.015, .175)** | **.051** | **.016** | **.104** |
| **Team Autonomy → Continuous Improvement → Responsiveness → Team Effectiveness** | **.062** | **(.034, .108)** | **.023** | **.003** | **.223** |

## 7 DISCUSSION

Teams are at the heart of the Scrum framework. To date, few studies have investigated how team-level factors contribute to Scrum team effectiveness. Moe et al. [108, 110] were among the first to do so. They applied a teamwork model by Salas et al. [129] to understand how cognitive and behavioral processes allow team members to collaborate effectively. Nevertheless, teamwork is one piece of the puzzle. Because teams are open systems, their effectiveness is constrained by the larger systems they are part of [36]. This is particularly relevant to boundary-crossing challenges typical

to Scrum teams, such as their interactions with stakeholders, the capability to release frequently, management support, team autonomy, and a climate of continuous improvement. Thus, this study contributes to existing research by offering a systematic perspective on the team-level factors that contribute to Scrum team effectiveness and manifest within the team and on the boundaries with larger systems.

We defined team effectiveness both from the perspective of external stakeholders (e.g., customers, users, and others outside the team) and internal stakeholders (team members). We used self-reported measures from team members to operationalize both perspectives and the success factors we expect to influence team effectiveness. To understand which team-level factors contribute to Scrum team effectiveness and answer $RQ_1$, we induced a five-factor model for team effectiveness from 13 case studies. To answer $RQ_2$, we tested the generalization of our model with Covariance-Based Structural Equation Modeling and a large sample of 1,978 Scrum teams and 4,940 respondents. Our sample included teams of varying levels of experience with Scrum from different regions in the world and different types and sectors of organizations (Table 9).

Our model demonstrated a good fit based on recommended fit indices with this sample. The five factors explained a substantial amount of variance in our two measures for team effectiveness: stakeholder satisfaction ($R^2 = 58\%$) and team morale ($R^2 = 35\%$). Although Scrum teams are embedded in larger organizational systems, this suggests that much of what makes Scrum teams effective can be designed and assessed at the team level. We discuss the findings and their implications below and summarize them in Table 12.

First, **Scrum teams that release to stakeholders frequently are indeed more effective than teams that do not**, as evidenced by higher stakeholder satisfaction and higher team morale (H1). We defined this more broadly as "responsiveness" and included release frequency and refinement. This finding reflects the premise of Agile methodologies that it is easier to manage complexity and discover emergent needs through many small releases [12] than through incidental large releases. This is also consistent with prior studies [25, 74]. It also stresses the motivational effect of frequent releases on teams that we observed in the case studies and is consistent with the evidence on motivation by Kahn [77].

Second, we found that **responsiveness to stakeholders alone is not enough**. Scrum teams negotiate much of their effectiveness in collaborating with stakeholders and understanding what is important to them. This "stakeholder concern" is indicated through frequent collaborations with stakeholders, the use of Sprint Reviews to gather feedback, shared goals, and a focus on what is valuable. Teams that score high in these areas develop a clearer sense of value and purpose than teams that do not. The need for such a shared understanding of value is also consistent with Van Kelle et. al. [153]. Our results suggest that such teams are more likely to develop the capabilities to be responsive to stakeholder needs (H2). We also found that a high concern for stakeholder needs contributes to team effectiveness directly and is mediated through responsiveness (H3a & H3b). This means that **while Scrum teams are indeed more effective when they understand the needs of their stakeholders, their actual ability to respond quickly to those needs further strengthens that effect**. In support of earlier studies by Bass et al. [10] and Judy & Krummins-Beens [75], we found that Product Owners are instrumental in shaping stakeholder concern. Product Owners can support Scrum teams by providing a vision and strategy to address stakeholder needs, using formal gatherings to gather feedback from stakeholders, and encouraging collaboration between the team and stakeholders.

Third, **we identified a climate of continuous improvement as a crucial team-level antecedent of responsiveness and stakeholder concern, and thus Scrum team effectiveness**. Continuous improvement is indicated by high psychological safety, a focus on quality, shared learning with other teams, Sprint Retrospectives that result in actionable improvements, and an

environment that encourages learning. Teams that score high in this area are more likely to be responsive (H4a) and show higher concern for stakeholder needs (H4b), which in turn increases their effectiveness. We theorized that this is because such teams are better able to continuously learn how to overcome emerging barriers to responsiveness and stakeholder collaboration and work with the larger system to affect change. Although the cross-sectional nature of our data does not allow us to support such a causal mechanism conclusively, the results are certainly consistent with research on organizational learning [99, 100] and learning as part of Agile transformations [68]. Another way to think of our conceptualization of continuous improvement is a team's ability to learn. This is also consistent with Strode, Dingsøyr & Lindsorn [143], who identify learning as an important influence on effective teamwork in Agile teams. Learning in particular is a process that is most effective when it crosses system boundaries [36, 78].

Fourth, **we identified team autonomy as an essential team-level antecedent of continuous improvement and stakeholder concern, and thus of Scrum team effectiveness**. Team autonomy is indicated by a high level of self-management and cross-functionality. Teams that score high in this area tend to operate in climates that are more conducive to continuous improvement (H5a) and are more responsive (H5b). If team autonomy is seen as a reduction in external and internal constraints a team has to navigate to become more effective, the results illustrate different types of constraints. For responsiveness, we found a modest positive effect on team autonomy. The more autonomy teams experience, the more likely they are to take ownership over improvements to their work as a team [76]. As teams expand their autonomy over their composition, work method, and work distribution, they also tend to be more responsive to emerging needs. We found a much stronger positive effect of team autonomy on continuous improvement, which is corroborated by other research [36]. However, contrary to our expectations, we did not find an effect of team autonomy on stakeholder concern (H5c). We expected that the mandate of a team to make decisions about their product is one type of constraint associated with team autonomy. We expected this mandate to influence the ability of teams to focus on stakeholder concerns. We found in the case studies that some Product Owners controlled scope and delivery dates and were able to adapt to emerging stakeholder needs. In contrast, other Product Owners had to follow established requirements and lacked such flexibility. However, no such effect was found. Team autonomy did not predict stakeholder concern in our analyses. This suggests that the autonomy of Scrum teams to make decisions about how to do their work (self-management) is mainly orthogonal to their autonomy to decide what is most valuable to their stakeholders (product mandate). We are not aware of studies that specifically explored this distinction, so this may be interesting to explore in future studies. However, this finding may also reflect a limitation of how we operationalized team autonomy in the survey (see Section 7.1).

A practical implication of these findings is that **organizations that seek to improve Scrum team effectiveness do well to invest in their autonomy**, which is supported by other studies of Agile teams [40, 104, 110]. Moe et. al. [107, 109] see this as particularly challenging in large-scale environments where the desire to increase team autonomy often competes with a desire to align teams and coordinate decision-making. A more effective strategy in such environments is to encourage teams to self-align by involving them in shared goal-setting and the development of a shared vision [107]. Team autonomy is also difficult in "specialist cultures" where redundancy of skills in teams is considered an inefficiency and thus discouraged, resulting in more dependencies and lower individual autonomy [109]. So while our study emphasizes the need for increased team autonomy, the preferred interventions are dependent on the context of a team.

Fifth, **this study stresses the importance of clear support from management**. We found that teams that experience such support also have higher team autonomy, continuous improvement, and stakeholder concern (H6a, H5b & H6c). The results did not support the hypothesized positive

effect on responsiveness (H6d). This also seems to be a result of different types of constraints. Where stakeholder concern, continuous improvement, and team autonomy are more susceptible to team-level constraints, responsiveness may be more constrained by individual skills, technologies, or legacy systems. A practical implication of these results is that support from management is most relevant where it involves the removal of constraints in the immediate environment of Scrum teams, particularly where it concerns the autonomy of Scrum teams and the quality and mandate of Product Owners. This is an excellent example of what Cummings [30] calls "boundary control"; one of the strategies that managers can use to support autonomous workgroups.

Thus, our resulting model identifies five factors and thirteen sub-factors contributing to Scrum team effectiveness that can be observed, measured, and influenced at the team level. The factors in our model also reflect the dynamic reality of Scrum teams as part of larger systems and how they continuously negotiate their effectiveness on their boundaries with regards to their autonomy, learning, stakeholder collaboration, and responsiveness. With this study, we provide an additional perspective on Scrum team effectiveness as compared to Moe et al. [108, 110]. Although teamwork did not naturally emerge from our case studies as a separate factor, the components of teamwork and coordinating mechanisms can certainly be recognized in variables such as psychological safety, shared goals, cross-functionality, and self-management.

So how do our results compare to prior investigations into the success factors of Agile and Scrum teams? Most importantly, we also show that team-level factors explain much of what makes teams effective. More specifically, the frequent delivery strategy that is emphasized by Chow & Cao [25] returns in our model as "responsiveness". The need for high-quality goals and a shared understanding of value that was identified by Serrador & Pinto [136] and Van Kelle et. al. [153] is represented through "stakeholder concern" in our model. We also see an important role for team autonomy, like Lee & Xia [89] and Melo et. al. [104]. However, there are also differences. First, and contrary to Chow & Cao [25], we did find a positive effect of management support. This may be the result of slightly different operationalizations and a different outcome variable. However, it is more likely that this effect is indeed indirect as our model shows. This makes sense as management support in itself won't make teams more effective, but rather creates better conditions for other team-level factors. A second difference is that our model does not include Agile development practices (like test automation, coding standards, and refactoring) as a separate factor, whereas Chow & Cao [25] and Van Kelle et. al. [153] do. We'd argue that such practices and skills effectively enable teams to release more frequently, and are thus measured indirectly through "responsiveness". We agree there is value in investigating how different Agile development practices impact responsiveness directly if only to offer more guidance to practitioners as to which practices are most beneficial. In summary, our study brings together many of the factors identified in earlier works into a single model. Our results also provide a deeper understanding of how those factors might interact in practice.

We did not specifically investigate the role of characteristics like team size, project size, skill level, organization size, and legacy software on these processes since these moderators did not emerge from our qualitative phase. Empirical studies have also cast doubt as to whether project size and complexity [25, 153] and team experience are significant moderators [136] at all. As with every model, the predictive power may be larger when additional moderators are introduced. For example, it may be more difficult to release frequently for legacy code or codebases with many external dependencies [19]. Teams may experience less autonomy as project size increases due to increased dependencies. It may also be harder for them to interact directly with stakeholders [154]. However, our resulting statistical model explains a large amount of variance in team effectiveness without such moderators. Since we aim to provide a parsimonious theoretical contribution, we have not included them in our model.

As a result of this, our model is primarily concerned with dynamic, team-level processes. This is particularly interesting from the perspective of the *Input-Process-Output* (IPO) framework that has seen frequent use in studies of team effectiveness [102, 104]. Models that use this framework organize variables into more or less static input states (e.g. team characteristics), group processes (e.g. communication), and outcomes (e.g. productivity). The core idea is that "input states affect group outcomes via the interaction that takes place among members" [61]. The aforementioned moderators are good examples of such input states. Of the factors in our model, "management support" is an input state to some extent in that affects team effectiveness through other factors. However, management support is clearly a dynamic process that changes over time. This reflects an emerging view of the IPO framework among scholars of team effectiveness that the distinction between inputs and processes is often artificial [43]. Even seemingly static inputs like team composition or skill level are in reality dynamic and probably influenced by team-level processes. Kozlowski & Ilgen [85] summarize this as "while the IPO model is a useful organizing heuristic, treating it as a causal model encourages taking a limited and static perspective on team effectiveness and the dynamic processes that underlie it". A strength of our model is that it highlights the dynamic nature and interplay of five team-level factors - or processes. However, we recognize that our model is still fairly linear and does not consider reciprocal effects. For example, we treat team autonomy as an important contributor to continuous improvement. It is entirely likely that teams expand their autonomy through the improvements they implement. Similarly, while stakeholder concern positively influences responsiveness, it is entirely likely that teams become more concerned with their stakeholders as a result of increased responsiveness. Such reciprocal, dynamic effects are interesting areas for future studies to explore.

### 7.1 Limitations

This is a Mixed Methods study. Following the recommendations of Russo et al. [125], we discuss both qualitative and quantitative limitations. In particular, we combined the threats to credibility, transferability, dependability, and confirmability [60] of our multiple field studies, with internal, construct, external, and conclusion validity from our sample study [58].

**Credibility & Internal**. The factors that we identified in the inductive research phase were derived from thirteen cases over five years. Triangulation was achieved through the use of multiple independent cases. For the deductive research part of our study, we employed several strategies to safeguard internal validity. One threat to public surveys is that samples are based on voluntary (non-probabilistic) responses and may be biased due to self-selection or non-completion. Therefore, we advertised the survey on industry platforms that are frequented by the target population. We also incentivized participation by offering teams a detailed score profile upon completion. Although we can not calculate the non-response rate without knowing how many people visited the survey, we did establish that 42% of the participants that started the survey also completed it. The cross-sectional nature of the data also does not allow conclusions about the causality of effects.

**Transferability & External**. A key strength of the Eisenhardt approach is that it grounds theories in the substantive area of use [49], and thus increases the likelihood of valid and relevant theories [1]. More importantly, the second phase of our study provided support for our theory with one of the most extensive sample studies ever performed in this area, surveying over 5,000 professionals of 2,000 Scrum teams from a range of industries, organizations, and countries (cf. Table 9), leading to generalizable results. Although we used a non-random sample for the case studies, it included Scrum teams that varied in size, sector, scale, and product. However, we can not make conclusions for teams that apply Scrum to other domains than software engineering or teams in general.

Table 12. Summary of Findings and Implications

| | Findings | Implications |
|---|---|---|
| **Five Factor Theory for Scrum Team Effectiveness** | From 13 case studies, we developed a theoretical model for Scrum teams from thirteen lower-order indicators grouped into five latent factors that predict Scrum team effectiveness. This model fit the data from a large and representative sample of Scrum teams well ($Chi^2(710) = 2,757.805$; $TLI = .953$; $CFI = .959$; $RMSEA = .038$; $SRMR = .035$). Together, the five factors explain a substantial amount of variance in the indicators we used to operationalize team effectiveness; stakeholder satisfaction (57.9%) and team morale (34.9%). | Design and assess Scrum teams with five team-level factors in mind: responsiveness, stakeholder concern, continuous improvement, team autonomy, and management support. Design the environment of Scrum teams to minimize constraints to these factors on the one hand and train and support them in the skills they need for each factor. |
| **Responsiveness** | Responsiveness is positively associated with team effectiveness ($\beta = .402, p < .01$). | Support Scrum teams in their ability to be responsive. Implement technical tooling, increase automation, and train necessary skills (particularly *refinement*). Invest in team autonomy, stakeholder concern, and management support to make the need for responsiveness more relevant to teams. |
| **Stakeholder Concern** | The concern that teams show for their stakeholders is positively associated with responsiveness ($\beta = .414, p < .01$). We also found that stakeholder concern contributes to team effectiveness directly ($\beta = .513, p < .05$), and also partially through responsiveness ($\beta = .166, p < .01$). So the positive effect of stakeholder concern is strongest when paired with high responsiveness. | Product Owners can increase Stakeholder Concern by co-opting teams in product strategy formulation, goal setting, and collaboration with stakeholders. If Scrum teams are unable to release frequently in the first place, efforts must be undertaken to remove organizational constraints, increase automation and build technical skills. |
| **Continuous Improvement** | The degree to which teams engage in continuous improvement is positively associated with stakeholder concern ($\beta = .545, p < .001$) and responsiveness ($\beta = .340, p < .01$). | Scrum teams are advised to direct their continuous improvement process towards the five key factors identified in this study: responsiveness, stakeholder concern, team autonomy, management support, and continuous improvement. These factors are most likely to highlight constraints to team effectiveness stemming from internal or external factors to the team. In turn, organizations should broaden the autonomy of teams to encourage teams to take control over improvements. |
| **Team Autonomy** | Team Autonomy was positively associated with continuous improvement ($\beta = .655, p < .001$) and responsiveness ($\beta = .258, p < .05$). Contrary to our expectations, we did not find an effect on stakeholder concerns. | Expand the autonomy of Scrum teams primarily in two areas. The first is internal to teams and concerns the degree to which its members are cross-functional. The second concerns constraints imposed by the organizational environment that limit control over tooling, team composition, choice of process, and Product Owners' mandate over their product. |
| **Management support** | Management Support was found to be positively associated with team autonomy ($\beta = .427, p < .001$), continuous improvement ($\beta = .168, p < .001$) and stakeholder concern ($\beta = .255, p < .001$), but no significant effect was found on responsiveness. | Management can most effectively contribute to Scrum teams by increasing their autonomy in terms of self-management and product mandate. Train management in the skills needed to support rather than direct. |

The data collection was partially performed during the COVID-19 pandemic. Some of the scales, like "Team Morale" and "Psychological Safety", may have been scored differently because teams worked from home. We added a 2-item 7-point Likert scale for a subset of teams ($N = 351$) to assess how they expected their results to have been different outside of COVID-19. The team mean score ($M = 2.81, SD = 1.63$) did not suggest a substantial impact. We also added the COVID-19 indicators as controls in our Structural Equation Model to further assess any confounding influence of COVID-19 on our results but found no significant effect on any of the other indicators in the survey.

**Dependability & Construct**. Triangulation was achieved through multiple cases for five years. Nevertheless, we can not exhaustively rule out personal bias in the case studies. Thus, we used well-established methodological guidelines to explain the research process and report our findings. Also, the case studies were performed to inform theory, so the emphasis lies on the second phase of our study. To maximize content validity for the second phase, we used existing validated scales where available. Where this was not the case, we derived scales from observations that recurred across cases. As a result, the proposed measurement model fits the data well. A heterotrait-monotrait

(HTMT) analysis confirmed discriminant validity for all measures. The reliability for all but one measure exceeded the cutoff recommended in the literature ($CR >= .70$ [63]).

Our survey consisted of self-reported items. We ran three pilots with the survey to identify problematic items, improve reliability and increase the response rate. A confirmatory factor analysis showed that all items loaded primarily on their intended scales (see Table 16). The scales also demonstrated good overall reliability. Although we statistically controlled for social desirability and common method bias, other confounding variables may have influenced how participants interpreted the items. This is particularly relevant to the operationalization of team effectiveness, which is based on self-reported scores on team morale and the perceived satisfaction of stakeholders. Mathieu et al. [102] recognize that such affect-based measures may suffer from a "halo effect". Future studies could ask stakeholders to rate their satisfaction with team outcomes directly. This does not rule out a halo effect entirely but is conceptually closer to what matters to organizations. Future studies could also attempt to find more objective measures for team effectiveness.

Furthermore, our operationalization for team autonomy in the survey did not include a measure for product mandate. This may explain why we did not find the hypothesized effect of team autonomy on stakeholder concern. Instead, we may have indirectly measured such mandate through the indicators for stakeholder concern, more specifically value focus ("The Product Backlog of this team is ordered with a strategy in mind", and "The Product Owner of this team has a clear vision for the product.").

Our sample composition (Table 9) shows that a wide range of teams participated in the questionnaire, with different levels of experience from different parts of the world and different types of organizations. We also observed a broad range of scores on the various measures. This provides confidence that a wide range of Scrum teams participated. Furthermore, our sample size and the aggregation of individual-level responses to team-level aggregates reduce variability due to non-systematic individual bias.

Finally, we performed a multi-group comparison to rule out that teams without experience with Scrum interpreted the questions differently from teams with experience. We created one group with teams without experience or an unknown level of experience ($N = 92$) and another for teams with at least some experience ($N = 1.886$). We found no significant differences in the factor loadings and the path effects between both groups ($Chi^2(86) = 91.450$; $p = .324$). If both groups would've interpreted the questions differently, we would have expected a different model for both groups, as indicated by different factor loadings and path effects.

**Confirmability & Statistical conclusion**. In this study, we used observations from thirteen exploratory case studies to inform a theoretical model about factors that contribute to the effectiveness of Scrum teams. In the second phase, we used Structural Equation Modeling to test the entire model simultaneously [20, 83]. The resulting model fits the data well on all fit indices recommended by statistical literature and explains a substantial amount of variance in the dependent variables. Finally, we published the team-level data and syntax files on Zenodo for reproducibility.

## 8 CONCLUSION

This paper proposes a theoretical model to understand which team-level factors determine Scrum team effectiveness. To date, only a few studies have investigated this question (e.g, [104, 108, 143]). This is surprising due to the prevalence of Scrum teams [155]. Furthermore, Scrum teams have been identified as the most important success factor for Agile projects in the Agile Success Model [124]. This study contributes to existing research by offering a systematic perspective on the team-level factors that determine Scrum team effectiveness. A generalizable theory for Scrum teams offers practical guidance to organizations and practitioners on designing, assessing, and supporting teams.

We applied a Mixed Methods approach by first collecting qualitative data from thirteen case studies to develop a preliminary theoretical model, which we then subjected to a quantitative test with a large sample of Scrum teams ($N = 1,978$). The data identified five high-order factors (responsiveness, stakeholder concern, continuous improvement, team autonomy, and management support) and thirteen lower-order factors (e.g., value focus, Sprint Review quality, cross-functionality, and refinement). Based on patterns in the observational data and supported by extant literature, we proposed six hypotheses to explain how these factors interact to make Scrum teams more or less effective, as indicated by their ability to satisfy stakeholders on the one hand and experience high team morale on the other.

Our findings highlight the interplay between stakeholder concern and responsiveness as drivers of Scrum team effectiveness. **The most effective teams are those that pair the ability to frequently release *with* a strong focus on the needs of their stakeholders. In turn, this demands a high degree of team autonomy, continuous improvement, and support from management**.

The model in this paper acts as a grounding framework to inform future research on Scrum team effectiveness. We recognize several areas that are useful to explore in future studies. First, we proposed our model as a systemic alternative to the teamwork model by Moe et al. [108]. It would be interesting to explore how both conceptualizations strengthen each other. Second, future studies can investigate how moderators such as project size, team experience, legacy systems, and skill-level of team members improve the predictive power of the Scrum Team Effectiveness model. Third, it is worthwhile to test the proposed causality with longitudinal or quasi-experimental designs. First, this may shed more light on which types of interventions are most relevant to practitioners and in which order. Fourth, it would be helpful to replicate our findings by directly measuring stakeholder satisfaction rather than indirectly through team members. Finally, we expect that much of our model can be generalized to other types of Agile teams. A few factors and a few items in our questionnaire turned out to be specific to the Scrum framework. It would be helpful to test the model with a version of the questionnaire that is agnostic to the Agile methodology in use by a team.

## 9 ACKNOWLEDGMENTS

The authors would like to thank all the organizations and informants for their efforts and time. We would also like to thank Johannes Schartau for acting as an additional rater in the qualitative case studies.

## 10 SUPPLEMENTARY MATERIALS

A replication package for the sample study is available at the following DOI to support Open Science: https://www.doi.org/10.5281/zenodo.6905812 under a CC-BY-NC-SA 4.0 license. The package includes the model definitions (AMOS), syntaxes for SPSS, and a fully anonymized, cleaned, and aggregated dataset of the analyzed Scrum teams.

## 11 RESPONSIBLE DISCLOSURE

Christiaan Verwijs has a financial interest in The Liberators BV.

## REFERENCES

[1] Lars-Johan Age. 2011. Grounded theory methodology: Positivism, hermeneutics, and pragmatism. *Qualitative Report* 16, 6 (2011), 1599–1615.

[2] Clayton P Alderfer. 1972. Human needs in organizational settings. *New York* (1972).

[3] Gopesh Anand, Dilip Chhajed, and Luis Delfin. 2012. Job autonomy, trust in leadership, and continuous improvement: An empirical study in health care. *Operations Management Research* 5, 3 (2012), 70–80.

[4] James L Arbuckle. 2011. IBM SPSS Amos 20 user's guide. *Amos Development Corporation, SPSS Inc* (2011).

[5] Linda Argote. 1993. Group and organizational learning curves: Individual, system and environmental components. *British Journal of Social Psychology* 32, 1 (1993), 31–51.

[6] Chris Argyris. 1999. *On Organizational Learning*. Wiley.

[7] Caroline Aube and Vincent Rousseau. 2005. Team goal commitment and team effectiveness: the role of task interdependence and supportive behaviors. *Group Dynamics: Theory, Research, and Practice* 9, 3 (2005), 189.

[8] Reuben M Baron and David A Kenny. 1986. The moderator–mediator variable distinction in social psychological research: Conceptual, strategic, and statistical considerations. *Journal of Personality and Social Psychology* 51, 6 (1986), 1173.

[9] Julian M Bass, Sarah Beecham, John Noll, and Mohammad Abdur Razzak. 2016. All Hands to the Pumps: The Product Owner Role in Small Companies Lero Technical Report: 2017_1. (2016).

[10] Julian M Bass, Sarah Beecham, Mohammad Abdur Razzak, Clodagh Nie Canna, and John Noll. 2018. An empirical study of the product owner role in scrum. In *International Conference on Software Engineering*. 123–124.

[11] Ruth Baumgart, Markus Hummel, and Roland Holten. 2015. Personality traits of Scrum roles in agile software development teams-a qualitative analysis. In *European Conference on Information Systems*. AIS.

[12] Kent Beck et al. 2001. Manifesto for Agile Software Development. www.agilemanifesto.org.

[13] Peter M Bentler. 1980. Multivariate analysis with latent variables: Causal modeling. *Annual Review of Psychology* 31, 1 (1980), 419–456.

[14] Peter M Bentler. 1990. Comparative fit indexes in structural models. *Psychological Bulletin* 107, 2 (1990), 238.

[15] Marthe Berntzen, Rashina Hoda, Nils B Moe, and Viktoria Stray. 2022. A Taxonomy of Inter-Team Coordination Mechanisms in Large-Scale Agile. *IEEE Transactions on Software Engineering* (2022).

[16] Marthe Berntzen, Nils Brede Moe, and Viktoria Stray. 2019. The product owner in large-scale agile: an empirical study through the lens of relational coordination theory. In *International Conference on Agile Software Development*. Springer, 121–136.

[17] Nadia Bhuiyan and Amit Baghel. 2005. An overview of continuous improvement: from the past to the present. *Management Decision* 43, 5/6 (2005), 761.

[18] Barry Boehm and Richard Turner. 2005. Management challenges to implementing agile processes in traditional development organizations. *IEEE Software* 22, 5 (2005), 30–39.

[19] Jan Bosch and Petra M Bosch-Sijtsema. 2011. Introducing agile customer-centered development in a legacy software product line. *Software: Practice and Experience* 41, 8 (2011), 871–882.

[20] Barbara M Byrne. 2016. *Structural Equation Modeling With AMOS Basic Concepts, Applications, and Programming*. Routledge.

[21] Janis A Cannon-Bowers and Eduardo Salas. 2001. Reflections on shared cognition. *Journal of Organizational Behavior: The International Journal of Industrial, Occupational and Organizational Psychology and Behavior* 22, 2 (2001), 195–202.

[22] Janis A Cannon-Bowers, Eduardo Salas, and Sharolyn Converse. 1993. Shared mental models in expert team decision making. *Individual and Group Decision Making: Current Issues* 221 (1993), 221–46.

[23] Feinian Chen, Patrick J Curran, Kenneth A Bollen, James Kirby, and Pamela Paxton. 2008. An empirical evaluation of the use of fixed cutoff points in RMSEA test statistic in structural equation models. *Sociological Methods & Research* 36, 4 (2008), 462–494.

[24] Tjan-Hien Cheng, Slinger Jansen, and Marc Remmers. 2009. Controlling and monitoring agile software development in three dutch product software companies. In *Workshop on Software Development Governance*. IEEE, 29–35.

[25] Tsun Chow and Dac-Buu Cao. 2008. A survey study of critical success factors in agile software projects. *Journal of systems and software* 81, 6 (2008), 961–971.

[26] Paolo Ciancarini, Angelo Messina, Francesco Poggi, and Daniel Russo. 2018. Agile knowledge engineering for mission critical software requirements. In *Synergies Between Knowledge Engineering and Software Engineering*. Springer, 151–171.

[27] Jacob Cohen. 1992. A power primer. *Psychological Bulletin* 112, 1 (1992), 155.

[28] Juliet M Corbin and Anselm Strauss. 1990. Grounded theory research: Procedures, canons, and evaluative criteria. *Qualitative Sociology* 13, 1 (1990), 3–21.

[29] John W Creswell and Vicki L Plano Clark. 2007. *Designing and conducting Mixed Methods research*. Wiley.

[30] Thomas G Cummings and Christopher G Worley. 2005. *Organization development*. Thomson South-Western.

[31] Josh Daspit, C Justice Tillman, Nancy G Boyd, and Victoria Mckee. 2013. Cross-functional team effectiveness: An examination of internal team environment, shared leadership, and cohesion influences. *Team Performance Management: An International Journal* (2013).

[32] Jackson De Carvalho and Felix O Chima. 2014. Applications of structural equation modeling in social sciences research. *American International Journal of Contemporary Research* 4, 1 (2014), 6–11.

[33] Paulo Henrique de Souza Bermejo, André Luiz Zambalde, Adriano Olímpio Tonelli, Samara Alyne Souza, Larissa Avelino Zuppo, and Priscila Luiz Rosa. 2014. Agile principles and achievement of success in software development: A quantitative study in Brazilian organizations. *Procedia Technology* 16 (2014), 718–727.

[34] Leslie A DeChurch and Jessica R Mesmer-Magnus. 2010. The cognitive underpinnings of effective teamwork: a meta-analysis. *Journal of Applied Psychology* 95, 1 (2010), 32.

[35] Edward L Deci and Richard M Ryan. 2000. The" what" and" why" of goal pursuits: Human needs and the self-determination of behavior. *Psychological Inquiry* 11, 4 (2000), 227–268.

[36] Stefan Decuyper, Filip Dochy, and Piet Van den Bossche. 2010. Grasping the dynamic complexity of team learning: An integrative model for effective team learning in organisations. *Educational Research Review* 5, 2 (2010), 111–133.

[37] Edwards Deming. 1986. *Out of the crisis.* MIT Center for Advanced Engineering Study.

[38] Kim Dikert, Maria Paasivaara, and Casper Lassenius. 2016. Challenges and success factors for large-scale agile transformations: A systematic literature review. *Journal of Systems and Software* 119 (2016), 87–108.

[39] Marco Dondi, Julia Klier, Frederic Panier, and Jörg Schubert. 2021. Defining the skills citizens will need in the future world of work. *McKinsey Global Institute. Available at: defining-the-skills-citizens-will-need-in-the-future-world-of-work. pdf (mckinsey. com)(Accessed: 30 June 2021)* (2021).

[40] Denniz Dönmez and Gudela Grote. 2013. The practice of not knowing for sure: How agile teams manage uncertainties. In *International Conference on Agile Software Development*. Springer, 61–75.

[41] Toni L Doolen, Marla E Hacker, and Eileen M Van Aken. 2003. The impact of organizational context on work team effectiveness: A study of production team. *IEEE Transactions on Engineering Management* 50, 3 (2003), 285–296.

[42] Tim Dreesen, Phil Hennel, Christoph Rosenkranz, and Thomas Kude. 2021. "The Second Vice is Lying, the First is Running into Debt." Antecedents and Mitigating Practices of Social Debt: an Exploratory Study in Distributed Software Development Teams. In *Hawaii International Conference on System Sciences*.

[43] James E Driskell, Eduardo Salas, and Tripp Driskell. 2018. Foundations of teamwork and collaboration. *American Psychologist* 73, 4 (2018), 334.

[44] Charles Duhigg. 2016. What Google learned from its quest to build the perfect team. *The New York Times Magazine* 26, 2016 (2016), 2016.

[45] Henry Edison, Xiaofeng Wang, and Kieran Conboy. 2021. Comparing Methods for Large-Scale Agile Software Development: A Systematic Literature Review. *IEEE Transactions on Software Engineering* (2021).

[46] Amy Edmondson. 1999. Psychological safety and learning behavior in work teams. *Administrative Science Quarterly* 44, 2 (1999), 350–383.

[47] Amy C Edmondson and Zhike Lei. 2014. Psychological safety: The history, renaissance, and future of an interpersonal construct. *Annual Review of Organizational Psychology and Organizational Behavior* 1, 1 (2014), 23–43.

[48] Kathleen M Eisenhardt. 1989. Building theories from case study research. *Academy of Management Review* 14, 4 (1989), 532–550.

[49] Kathleen M Eisenhardt and Melissa E Graebner. 2007. Theory building from cases: Opportunities and challenges. *Academy of Management Journal* 50, 1 (2007), 25–32.

[50] Kathleen M Eisenhardt and Behnam N Tabrizi. 1995. Accelerating adaptive processes: Product innovation in the global computer industry. *Administrative Science Quarterly* (1995), 84–110.

[51] Franz Faul, Edgar Erdfelder, Axel Buchner, and Albert-Georg Lang. 2009. Statistical power analyses using G* Power 3.1: Tests for correlation and regression analyses. *Behavior Research Methods* 41, 4 (2009), 1149–1160.

[52] Nicole Forsgren, Margaret-Anne Storey, Chandra Maddila, Thomas Zimmermann, Brian Houck, and Jenna Butler. 2021. The SPACE of Developer Productivity: There's more to it than you think. *Queue* 19, 1 (2021), 20–48.

[53] Donelson R Forsyth. 2018. *Group dynamics.* Cengage Learning.

[54] David A Garvin, Amy C Edmondson, and Francesca Gino. 2008. Is yours a learning organization? *Harvard Business Review* 86, 3 (2008), 109.

[55] James Gaskin. 2012. Data screening: Gaskination's StatWiki.

[56] James Gaskin and James Lim. 2016. AMOS Plugin In: Gaskination's StatWiki.

[57] Dennis A Gioia, Kevin G Corley, and Aimee L Hamilton. 2013. Seeking qualitative rigor in inductive research: Notes on the Gioia methodology. *Organizational Research Methods* 16, 1 (2013), 15–31.

[58] Kerri A Goodwin and C James Goodwin. 2016. *Research in psychology: Methods and design.* John Wiley & Sons.

[59] Lucas Gren, Richard Torkar, and Robert Feldt. 2017. Group development and group maturity when building agile teams: A qualitative and quantitative investigation at eight large companies. *Journal of Systems and Software* 124 (2017), 104–119.

[60] Egon G Guba. 1981. Criteria for assessing the trustworthiness of naturalistic inquiries. *Educational Technology Research and Development* 29, 2 (1981), 75–91.

[61] J Richard Hackman. 2012. From causes to conditions in group research. *Journal of Organizational Behavior* 33, 3 (2012), 428–444.

[62] Richard Hackman. 1976. *The design of self-managing work groups.* Technical Report. Yale University Press.

[63] Joseph F Hair Jr, Wiliam C Black, Barry J Babin, and Rolph E Anderson. 2019. *Multivariate data analysis.*

[64] Jeffrey R Harring, Brandi A Weiss, and Jui-Chen Hsu. 2012. A comparison of methods for estimating quadratic effects in nonlinear structural equation models. *Psychological Methods* 17, 2 (2012), 193.

[65] Ron D Hays, Toshi Hayashi, and Anita L Stewart. 1989. A five-item measure of socially desirable response set. *Educational and Psychological Measurement* 49, 3 (1989), 629–636.

[66] Jörg Henseler, Christian M Ringle, and Marko Sarstedt. 2015. A new criterion for assessing discriminant validity in variance-based structural equation modeling. *Journal of the Academy of Marketing Science* 43, 1 (2015), 115–135.

[67] Anita Hidayati, Eko K Budiardjo, and Betty Purwandari. 2020. Hard and Soft Skills for Scrum Global Software Development Teams. In *International Conference on Software Engineering and Information Management.* 110–114.

[68] Rashina Hoda and James Noble. 2017. Becoming agile: a grounded theory of agile transitions in practice. In *International Conference on Software Engineering.* IEEE, 141–151.

[69] Rashina Hoda, James Noble, and Stuart Marshall. 2011. The impact of inadequate customer collaboration on self-organizing Agile teams. *Information and Software Technology* 53, 5 (2011), 521–534.

[70] Li-tze Hu and Peter M Bentler. 1999. Cutoff criteria for fit indexes in covariance structure analysis: Conventional criteria versus new alternatives. *Structural Equation Modeling: a Multidisciplinary Journal* 6, 1 (1999), 1–55.

[71] Gary W Ivey, J-R Sébastien Blanc, and Janet Mantler. 2015. An assessment of the overlap between morale and work engagement in a nonoperational military sample. *Journal of Occupational Health Psychology* 20, 3 (2015), 338.

[72] Brian D Janz, Jason A Colquitt, and Raymond A Noe. 1997. Knowledge worker team effectiveness: The role of autonomy, interdependence, team development, and contextual support variables. *Personnel Psychology* 50, 4 (1997), 877–904.

[73] Burke Johnson, Anthony J Onwuegbuzie, and Lisa A Turner. 2007. Toward a definition of mixed methods research. *Journal of Mixed Methods Research* 1, 2 (2007), 112–133.

[74] Magne Jørgensen. 2016. A survey on the characteristics of projects with success in delivering client benefits. *Information and Software Technology* 78 (2016), 83–94.

[75] Ken H Judy and Ilio Krumins-Beens. 2008. Great Scrums need great product owners: Unbounded collaboration and collective product ownership. In *Annual Hawaii International Conference on System Sciences.* IEEE, 462–462.

[76] Tom L Junker, Arnold B Bakker, Marjan J Gorgievski, and Daantje Derks. 2021. Agile work practices and employee proactivity: A multilevel study. *Human Relations* (2021).

[77] William A Kahn. 1990. Psychological conditions of personal engagement and disengagement at work. *Academy of Management Journal* 33, 4 (1990), 692–724.

[78] Elizabeth Kasl, Victoria J Marsick, and Kathleen Dechant. 1997. Teams as learners: A research-based model of team learning. *The Journal of Applied Behavioral Science* 33, 2 (1997), 227–246.

[79] Eric Kearney, Diether Gebert, and Sven C Voelpel. 2009. When and how diversity benefits teams: The importance of team members' need for cognition. *Academy of Management Journal* 52, 3 (2009), 581–598.

[80] Robert T Keller. 2001. Cross-functional project groups in research and new product development: Diversity, communications, job stress, and outcomes. *Academy of Management Journal* 44, 3 (2001), 547–555.

[81] Youry Khmelevsky, Xitong Li, and Stuart Madnick. 2017. Software development using agile and scrum in distributed teams. In *International Systems Conference.* IEEE, 1–4.

[82] Bradley L Kirkman and Benson Rosen. 1999. Beyond self-management: Antecedents and consequences of team empowerment. *Academy of Management Journal* 42, 1 (1999), 58–74.

[83] Rex B Kline. 2015. *Principles and practice of structural equation modeling.* Guilford publications.

[84] Theresa JB Kline and Michelle MacLeod. 1997. Predicting organizational team performance. *Organization Development Journal* (1997).

[85] 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.

[86] Sigurhanna Kristinsdottir, Marta Larusdottir, and Åsa Cajander. 2016. Responsibilities and challenges of product owners at spotify-an exploratory case study. In *Human-Centered and Error-Resilient Systems Development.* Springer, 3–16.

[87] Eetu Kupiainen, Mika V Mäntylä, and Juha Itkonen. 2014. Why are industrial agile teams using metrics and how do they use them?. In *International Workshop on Emerging Trends in Software Metrics.* 23–29.

[88] Claus W Langfred. 2005. Autonomy and performance in teams: The multilevel moderating effect of task interdependence. *Journal of Management* 31, 4 (2005), 513–529.

[89] Gwanhoo Lee and Weidong Xia. 2010. Toward agile: an integrated analysis of quantitative and qualitative field data on software development agility. *MIS Quarterly* 34, 1 (2010), 87–114.

[90] Michael Y Lee and Amy C Edmondson. 2017. Self-managing organizations: Exploring the limits of less-hierarchical organizing. *Research in Organizational Behavior* 37 (2017), 35–58.

[91] Edwin A Locke, Karyll N Shaw, Lise M Saari, and Gary P Latham. 1981. Goal setting and task performance: 1969–1980. *Psychological Bulletin* 90, 1 (1981), 125.

[92] Janeth López-Martínez, Reyes Juárez-Ramírez, Carlos Huertas, Samantha Jiménez, and Cesar Guerra-García. 2016. Problems in the adoption of agile-scrum methodologies: A systematic literature review. In *International Conference in Software Engineering Research and Innovation*. IEEE, 141–148.

[93] Niklas Luhmann. 1995. *Social systems*. Stanford University Press.

[94] Viljan Mahnic and Ivan Vrana. 2007. Using stakeholder-driven process performance measurement for monitoring the performance of a Scrum-based software development process. *Electrotechnical Review* 74, 5 (2007), 241–247.

[95] Derek C Man and Simon SK Lam. 2003. The effects of job complexity and autonomy on cohesiveness in collectivistic and individualistic work groups: a cross-cultural analysis. *Journal of Organizational Behavior: The International Journal of Industrial, Occupational and Organizational Psychology and Behavior* 24, 8 (2003), 979–1001.

[96] Frederick J Manning. 1991. Morale, cohesion, and esprit de corps. (1991).

[97] Charles C Manz and Henry P Sims Jr. 1987. Leading workers to lead themselves: The external leadership of self-managing work teams. *Administrative Science Quarterly* (1987), 106–129.

[98] Herbert W Marsh and Dennis Hocevar. 1985. Application of confirmatory factor analysis to the study of self-concept: First-and higher order factor models and their invariance across groups. *Psychological Bulletin* 97, 3 (1985), 562.

[99] Victoria J Marsick and Karen E Watkins. 1999. *Facilitating learning organizations: Making learning count*. Gower Publishing, Ltd.

[100] Victoria J Marsick and Karen E Watkins. 2003. Demonstrating the value of an organization's learning culture: the dimensions of the learning organization questionnaire. *Advances in Developing Human Resources* 5, 2 (2003), 132–151.

[101] Abraham Harold Maslow. 1981. *Motivation and personality*. Prabhat Prakashan.

[102] John Mathieu, M Travis Maynard, Tammy Rapp, and Lucy Gilson. 2008. Team effectiveness 1997-2007: A review of recent advancements and a glimpse into the future. *Journal of Management* 34, 3 (2008), 410–476.

[103] Adam W Meade and S Bartholomew Craig. 2012. Identifying careless responses in survey data. *Psychological Methods* 17, 3 (2012), 437.

[104] Claudia de O Melo, Daniela S Cruzes, Fabio Kon, and Reidar Conradi. 2013. Interpretative case studies on agile team productivity and management. *Information and Software Technology* 55, 2 (2013), 412–427.

[105] Angelo Messina, Franco Fiore, Mauro Ruggiero, Paolo Ciancarini, and Daniel Russo. 2016. A new agile paradigm for mission-critical software development. *CrossTalk* 29, 6 (2016), 25–30.

[106] John Mingers. 2001. Combining IS research methods: towards a pluralist methodology. *Information Systems Research* 12, 3 (2001), 240–259.

[107] Nils Brede Moe, Bjørn Haug Dahl, Viktoria Stray, Lina Sund Karlsen, and Stine Schjødt-Osmo. 2019. Team autonomy in large-scale agile. In *Annual Hawaii International Conference on System Sciences*. AIS, 6997–7006.

[108] Nils Brede Moe, Torgeir Dingsøyr, and Tore Dybå. 2008. Understanding self-organizing teams in agile software development. In *Australian Conference on Software Engineering*. IEEE, 76–85.

[109] Nils Brede Moe, Torgeir Dingsøyr, and Tore Dybå. 2009. Overcoming barriers to self-management in software teams. *IEEE Software* 26, 6 (2009), 20–26.

[110] Nils Brede Moe, Torgeir Dingsøyr, and Tore Dybå. 2010. A teamwork model for understanding an agile team: A case study of a Scrum project. *Information and Software Technology* 52, 5 (2010), 480–491.

[111] Linda D Molm. 1994. Dependence and risk: Transforming the structure of social exchange. *Social Psychology Quarterly* (1994), 163–176.

[112] John Noll, Mohammad Abdur Razzak, Julian M Bass, and Sarah Beecham. 2017. A study of the scrum master's role. In *International Conference on Product-Focused Software Process Improvement*. Springer, 307–323.

[113] Maria Paasivaara and Casper Lassenius. 2016. Scaling scrum in a large globally distributed organization: A case study. In *International Conference on Global Software Engineering*. IEEE, 74–83.

[114] Sharon K Parker, Uta K Bindl, and Karoline Strauss. 2010. Making things happen: A model of proactive motivation. *Journal of Management* 36, 4 (2010), 827–856.

[115] John A Pearce III and Elizabeth C Ravlin. 1987. The design and activation of self-regulating work groups. *Human Relations* 40, 11 (1987), 751–782.

[116] Roman Pichler. 2016. *Strategize: product strategy and product roadmap practices for the digital age*. Pichler Consulting.

[117] Mariângela Pinton and Alvair Silveira Torres Junior. 2020. Human Aspects of Agile Transition in Traditional Organizations. *Journal of Technology Management & Innovation* 15, 3 (2020), 62–73.

[118] Philip M Podsakoff, Scott B MacKenzie, Jeong-Yeon Lee, and Nathan P Podsakoff. 2003. Common method biases in behavioral research: a critical review of the literature and recommended remedies. *Journal of Applied Psychology* 88, 5 (2003), 879.

[119] Christoph Pohl, Alf Zugenmaier, Michael Meier, and Hans-Joachim Hof. 2015. B. Hive: A zero configuration forms honeypot for productive web applications. In *International Information Security and Privacy Conference*. Springer, 267–280.
[120] Radostina K Purvanova and Renata Kenda. 2021. The impact of virtuality on team effectiveness in organizational and non-organizational teams: A meta-analysis. *Applied Psychology* (2021).
[121] Ryan Ripley and Todd Miller. 2020. *Fixing your scrum: practical solutions to common scrum problems*. Pragmatic Bookshelf.
[122] Knut H Rolland, Vidar Mikkelsen, and Alexander Næss. 2016. Tailoring agile in the large: Experience and reflections from a large-scale agile software development project. In *International Conference on Agile Software Development*. Springer, 244–251.
[123] William B Rouse and Nancy M Morris. 1986. On looking into the black box: Prospects and limits in the search for mental models. *Psychological Bulletin* 100, 3 (1986), 349.
[124] Daniel Russo. 2021. The Agile Success Model: A Mixed Methods Study of a Large-Scale Agile Transformation. *ACM Transactions on Software Engineering and Methodology* 30, 4 (2021), 1–46.
[125] Daniel Russo, Paolo Ciancarini, Tommaso Falasconi, and Massimo Tomasi. 2018. A Meta Model for Information Systems Quality: a Mixed-Study of the Financial Sector. *ACM Transactions on Management Information Systems* 9, 3 (2018), 1–38.
[126] Daniel Russo and Klaas-Jan Stol. 2019. Soft theory: a pragmatic alternative to conduct quantitative empirical studies. In *International Workshop on Conducting Empirical Studies in Industry and 6th International Workshop on Software Engineering Research and Industrial Practice*. IEEE, 30–33.
[127] Daniel Russo and Klaas-Jan Stol. 2021. PLS-SEM for Software Engineering Research: An Introduction and Survey. *Comput. Surveys* 54, 4 (2021).
[128] Daniel Russo, Gerolamo Taccogna, Paolo Ciancarini, Angelo Messina, and Giancarlo Succi. 2018. Contracting Agile Developments for Mission Critical Systems in the Public Sector. In *International Conference on Software Engineering*. IEEE, 47–56.
[129] Eduardo Salas, Dana E Sims, and C Shawn Burke. 2005. Is there a "big five" in teamwork? *Small Group Research* 36, 5 (2005), 555–599.
[130] Wilmar B Schaufeli, Marisa Salanova, Vicente González-Romá, and Arnold B Bakker. 2002. The measurement of engagement and burnout: A two sample confirmatory factor analytic approach. *Journal of Happiness Studies* 3, 1 (2002), 71–92.
[131] Edgar H Schein et al. 1992. How can organizations learn faster?: the problem of entering the Green Room. (1992).
[132] Eva-Maria Schön, Jörg Thomaschewski, and María José Escalona. 2017. Agile Requirements Engineering: A systematic literature review. *Computer Standards & Interfaces* 49 (2017), 79–91.
[133] Ken Schwaber. 1995. SCRUM Development Process. In *OOPSLA Conference*. 1–23.
[134] Ken Schwaber and Jeff Sutherland. 2020. Definiton of Scrum.
[135] Kent Schwaber and Jeff Sutherland. 2020. The Scrum Guide. *Scrum Alliance* (2020), 17.
[136] Pedro Serrador and Jeffrey K Pinto. 2015. Does Agile work?—A quantitative analysis of agile project success. *International journal of project management* 33, 5 (2015), 1040–1051.
[137] Yogeshwar Shastri, Rashina Hoda, and Robert Amor. 2021. Spearheading agile: the role of the scrum master in agile projects. *Empirical Software Engineering* 26, 1 (2021), 1–31.
[138] Marcia J Simmering, Christie M Fuller, Hettie A Richardson, Yasemin Ocal, and Guclu M Atinc. 2015. Marker variable choice, reporting, and interpretation in the detection of common method variance: A review and demonstration. *Organizational Research Methods* 18, 3 (2015), 473–511.
[139] Robert E Stake. 1995. *The art of case study research*. Sage.
[140] James H Steiger. 1980. Statistically based tests for the number of common factors. In *The Annual Meeting of the Psychometric Society*.
[141] Klaas-Jan Stol, Bora Caglayan, and Brian Fitzgerald. 2017. Competition-based crowdsourcing software development: a multi-method study from a customer perspective. *IEEE Transactions on Software Engineering* 45, 3 (2017), 237–260.
[142] Klaas-Jan Stol and Brian Fitzgerald. 2018. The ABC of software engineering research. *ACM Transactions on Software Engineering and Methodology* 27, 3 (2018), 1–51.
[143] Diane Strode, Torgeir Dingsøyr, and Yngve Lindsjorn. 2022. A teamwork effectiveness model for agile software development. *Empirical Software Engineering* 27, 2 (2022), 1–50.
[144] Eric Sundstrom, Michael McIntyre, Terry Halfhill, and Heather Richards. 2000. Work groups: From the Hawthorne studies to work teams of the 1990s and beyond. *Group Dynamics: Theory, Research, and Practice* 4, 1 (2000), 44.
[145] Hrafnhildur Sif Sverrisdottir, Helgi Thor Ingason, and Haukur Ingi Jonasson. 2014. The role of the product owner in scrum-comparison between theory and practices. *Procedia-Social and Behavioral Sciences* 119 (2014), 257–267.

[146] John Tripp and Deborah J Armstrong. 2018. Agile methodologies: organizational adoption motives, tailoring, and performance. *Journal of Computer Information Systems* 58, 2 (2018), 170–179.
[147] John Tripp, Cindy Riemenschneider, and Jason B Thatcher. 2016. Job satisfaction in agile development teams: Agile development as work redesign. *Journal of the Association for Information Systems* 17, 4 (2016), 1.
[148] Eric Lansdown Trist and Kenneth W Bamforth. 1951. Some social and psychological consequences of the longwall method of coal-getting: An examination of the psychological situation and defences of a work group in relation to the social structure and technological content of the work system. *Human Relations* 4, 1 (1951), 3–38.
[149] Carolin Unger-Windeler, Jil Ann-Christin Klünder, Timothy Reuscher, and Kurt Schneider. 2020. Are Product Owners communicators? A multi-method research approach to provide a more comprehensive picture of Product Owners in practice. *Journal of Software: Evolution and Process* (2020), e2311.
[150] Carolin Unger-Windeler, Jil Ann-Christin Klünder, Timothy Reuscher, and Kurt Schneider. 2021. Are Product Owners communicators? A multi-method research approach to provide a more comprehensive picture of Product Owners in practice. *Journal of Software: Evolution and Process* 33, 1 (2021), e2311.
[151] Frank van Boxmeer, Christiaan Verwijs, Robert de Bruin, Jacco Duel, and Martin Euwema. 2007. A direct measure of Morale in the Netherlands Armed Forces Morale Survey:'theoretical puzzle, empirical testing and validation. In *Internation Military Testing Association Symposium*.
[152] Marieke Van der Hoek, Sandra Groeneveld, and Ben Kuipers. 2018. Goal setting in teams: Goal clarity and team performance in the public sector. *Review of Public Personnel Administration* 38, 4 (2018), 472–493.
[153] Evelyn Van Kelle, Joost Visser, Aske Plaat, and Per van der Wijst. 2015. An empirical study into social success factors for agile software development. In *International Workshop on Cooperative and Human Aspects of Software Engineering*. IEEE, 77–80.
[154] Guus Van Waardenburg and Hans Van Vliet. 2013. When agile meets the enterprise. *Information and Software Technology* 55, 12 (2013), 2154–2171.
[155] VersionOne. 2021. *15th Annual State of Agile Report*. Technical Report. 10 pages.
[156] Christiaan Verwijs, Johannes Schartau, and Barry Overeem. 2021. *Zombie scrum survival guide: A Journey to Recovery*. Addison-Wesley.
[157] Ruth Wageman, J Richard Hackman, and Erin Lehman. 2005. Team diagnostic survey: Development of an instrument. *The Journal of Applied Behavioral Science* 41, 4 (2005), 373–398.
[158] Elizabeth Whitworth and Robert Biddle. 2007. The social nature of agile teams. In *Agile*. IEEE, 26–36.
[159] Laurie Williams, Gabe Brown, Adam Meltzer, and Nachiappan Nagappan. 2011. Scrum engineering practices: Experiences of three microsoft teams. In *International Symposium on Empirical Software Engineering and Measurement*. IEEE, 463–471.
[160] Robert K Yin. 1994. Case study research: Design and methods, applied social research. *Methods Series* 5 (1994).
[161] Raymond Young and Ernest Jordan. 2008. Top management support: Mantra or necessity? *International Journal of Project Management* 26, 7 (2008), 713–725.

# A APPENDIX

| 1st Order Observations & concepts | 2nd Order Themes | Aggregate Dimension |
|---|---|---|
| Stakeholders are (more) satisfied with the work done by the team | Stakeholder Satisfaction | Team Effectiveness |
| Stakeholders frequently complain about the quality of the product | | |
| Stakeholders give compliments to the team for a job well done | | |
| Team members feel motivated in their work | Team Morale | |
| Team members frequently complain, but there is no talk about how to solve things | | |
| Team members feel like they're in a hamster wheel (different day, same problems) | | |
| The team feels that iterations are artificial (i.e, "There's always a next Sprint" or "Lets just carry it over to the next Sprint") | Refinement | Responsiveness |
| Team members spend time during a Sprint to prepare work for the coming Sprints | | |
| Sprint Planning tends to drag on as teams try to discover what they need to create that Sprint | | |
| The Sprint Backlog contains only a handful of large Product Backlog Items | | |
| Sprints generally result in at least one releasable increment for this team | Release Frequency | |
| Acceptance testing is done after the sprint, and not during | | |
| Sprint Reviews are used to review an Increment that is running on an environment that is as close to production as possible (e.g, not a developers pc) | | |
| Sprint Reviews don't actually demonstrate working software (e.g, instead PowerPoint presentations are used, or the team simply talks about it) | | |
| The team frequently spends time cleaning up the Product Backlog by removing items that don't make sense anymore (e.g, in light of changed product vision, strategy) | Value Focus | Stakeholder Concern |
| Team members other than the Product Owner can reiterate the strategy behind the upcoming sprints | | |
| The team talks about how or why items on the Product Backlog are valuable to their stakeholders during various Scrum Events | | |
| The Backlog contains mostly technical tasks, and is difficult to connect to specific business needs by stakeholders | | |
| The team is familiar with the vision and strategy for the product they are working on | | |
| The work that a team does is aligned with other aspects of the business (support, sales, marketing, etc.) | Stakeholder Collaboration | |
| Except for the Product Owners, the members of a team don't generally speak with users, customers or other stakeholders | | |
| Product Owners involve customers/user/stakeholders in the prioritization of the backlog | | |
| Product Owners frequently meet with the customers or users using their software | | |
| The Backlog is not easily available to stakeholders, end-users and other teams | | |
| There is no interaction during Sprint Reviews between members of the team; people sit back and watch | | |
| Stakeholders have opportunities during Sprint Reviews to interact and review a working product or working features | Sprint Review Quality | |
| The team prepares for the Sprint Review by determining what they'd like feedback on and in what order | | |

| 1st Order Observations & concepts | 2nd Order Themes | Aggregate Dimension |
|---|---|---|
| The Sprint Review is attended by users, customers and other stakeholders | | |
| The Product Owner uses the Sprint Review to get feedback from stakeholders on the work that was done | | |
| Most Sprints have a Sprint Goal that explains how the work in a Sprint ties together | Shared Goals | |
| There is a roadmap of Sprint Goals for the upcoming sprints | | |
| The Scrum Master is distributing work and chairing the Planning and the Sprint Review | Self Management | Team Autonomy |
| The Product Owner has a mandate to make decisions about the work that the team spends time on | | |
| The team is responsible for planning its own work capacity | | |
| Teams are involved in, or have complete control over, decisions that affect their composition or work-method | | |
| Teams are slowed down by dependencies on external teams or individuals who need to perform tasks that they can't do themselves | | |
| Teams are given responsibility only for specific functional areas (e.g, development, testing, design) | | |
| Individual team members are made responsible (and rewarded) for specific specializations | Cross-Functionality | |
| Team members collaborate on tasks from the Sprint Backlog, regardless of their specializations | | |
| Team members believe they are solely responsible for their specialization (e.g, "I'm only here to code") | | |
| Team members actively keep each other up to date about what they are working on throughout the day | | |
| Teams are organized in terms of functional specializations (e.g, a 'continuous delivery team' or 'a design team') | | |
| The testing for a particular item on the Sprint Backlog is performed by someone other than the developer (in or outside the team) | | |
| Sprint Retrospectives always yield the same issues over extended periods | Sprint Retrospective Quality | Continuous Improvement |
| Sprint Retrospectives are used to dive into deeper problems, not just the superficial symptoms | | |
| The action items from Sprint Retrospectives are usually not implemented | | |
| A lot of bugs and issues are discovered during the Sprint Review, or quickly after | Concern for Quality | |
| Team members often talk about how to improve the quality of their work (e.g, during Sprint Retrospectives) | | |
| The Definition of Done doesn't define what high-quality looks like for a team | | |
| Team members are quick to shoot down ideas (e.g, "We've tried that before...", "That doesn't work here") | Psychological Safety | |
| Team members generally offer vague and unclear contributions to the Daily Scrum (e.g, "I've was busy yesterday, I will be busy today") | | |
| Team members make an effort to listen to each other before sharing their own view | | |
| Team members openly ask for help when they are stuck | | |
| When a mistake is made, people are not singled out or punished for it | | |
| Teams share their learnings and challenges with other teams in the organization | Shared Learning | |

| 1st Order Observations & concepts | 2nd Order Themes | Aggregate Dimension |
|---|---|---|
| Teams involve people from outside the team to work together on overcoming challenges they face | | |
| Team members express a desire to try new things or experiment with new technologies | Learning Environment | |
| Team members are motivated or encouraged to learn new things, read professional books, visit conferences or meetups | | |
| Management predominantly tells team members how to perform their work, rather than asking where they can support them | Management Support | Management Support |
| Management understands why this team works with Scrum | | |
| Management is available to help teams remove impediments | | |

Table 13. Data structure of observations and themes for the aggregate dimensions derived from case studies

Table 14. Results of Structural Equation Modeling. Fit indices of the model and standardized maximum likelihood estimates

| Model | Chi-Square | Df | GFI | CFI | RMSEA | SRMR | TLI | Notes |
|---|---|---|---|---|---|---|---|---|
| | | | *Measurement Models* | | | | | |
| MM1 | 3,307.80 | 898 | .93 | .957 | .037 | .038 | .949 | Base measurement model with only indicators and first-order latent factors (covaried) |
| MM2 | 3,169.55 | 895 | .933 | .96 | .036 | .038 | .951 | Added covar between PS2 and PS3. PS1 and PS4. VF2 and VF3 due to high MI (similar wording) |
| MM3 | 2,564.08 | 806 | .944 | .967 | .033 | .036 | .959 | Removed SH1 and PS5 due to many residual covariances with other items |
| MM4 | 2,324.47 | 763 | .934 | .97 | .032 | .028 | .963 | Removed RE1 due to low factor loading |
| | | | *Common Method Bias testing* | | | | | |
| CMB1 | 1,946.49 | 720 | - | .977 | .029 | .021 | - | CMB with only a CLF (note: covariance matrix is not positive definite. so a marker variable must be used instead) |
| CMB2 | 2,231.15 | 739 | - | .971 | .032 | .026 | - | CMB factor loads directly on all indicators (unconstrained) |
| CMB3 | 2,589.89 | 823 | - | .967 | .033 | .041 | - | CMB factor loads directly on all indicators (zero-constrained) |
| CMB4 | 2,524.85 | 822 | - | .968 | .032 | .033 | - | CMD factor loads directly on all indicators (constrained to same value to test even distribution of bias) |
| | | | *Path Models* | | | | | |
| PM1 | 2,757.81 | 710 | .932 | .959 | .038 | .035 | .952 | Base path model |

Table 15. Scrum Team Survey Questionnaire

| Aggregate Dimension | Scale | Question | Type |
|---|---|---|---|
| | Team | Where are the people that this team works for - like users and customers - mostly based? | Single choice |
| | Team | How many members does this team typically have, including a Scrum Master and Product Owner? | Single choice |
| | Team | I consider this team to be very experienced with Scrum. | Likert (1-7) |
| | Organisation | Which sector is this organisation mostly active in? | Single choice |
| | Organisation Size | What is the size of this organisation? | Single choice |
| Responsiveness | Refinement | The Sprint Backlog of this team usually contains many small items. | Likert (1-7) |
| | | During the Sprint, this team spends time to clarify work for the next couple of Sprints. | Likert (1-7) |
| | | During the Sprint, this team spends time breaking down work for coming Sprints. | Likert (1-7) |
| | Release Frequency | The majority of the Sprints of this team result in software that can be deployed to production. | Likert (1-7) |
| | | For this team, most of the Sprints result in an increment that can be released to users. | Likert (1-7) |
| Stakeholder Concern | Stakeholder Collaboration | Members of this team frequently meet with users or customers of what this team creates. | Likert (1-7) |
| | | People from this team often invite or visit people that use what this team works on. | Likert (1-7) |
| | | People in this team closely collaborate with users, customers and other stakeholders. | Likert (1-7) |
| | Shared Goals | This team generally has clear Sprint Goals. | Likert (1-7) |
| | | During Sprint Planning, this team formulates a clear goal for the Sprint. | Likert (1-7) |
| | Sprint Review Quality | The Product Owner of this team uses the Sprint Review to collect feedback from stakeholders. | Likert (1-7) |
| | | During Sprint Reviews, stakeholders frequently try out what this team has been working on during the Sprint. | Likert (1-7) |
| | Value Focus | The Product Owner of this team has a clear vision for the product. | Likert (1-7) |
| | | The Product Backlog of this team is ordered with a strategy in mind. | Likert (1-7) |
| | | Everyone in this team is familiar with the vision for the product. | Likert (1-7) |
| Continuous Improvement | Shared Learning | This team frequently works with other groups or teams to solve shared problems. | Likert (1-7) |
| | | Teams in this organization share what they learn with other teams. | Likert (1-7) |
| | | Members from this team frequently meet with other teams to identify improvements. | Likert (1-7) |
| | Learning Environment | In and around this team, people are given time to support learning. | Likert (1-7) |
| | | In and around this team, people are rewarded for learning. | Likert (1-7) |
| | Psychological Safety | In and around this team, people give open and honest feedback to each other. | Likert (1-7) |
| | | In and around this team, people listen to the others' views before speaking. | Likert (1-7) |
| | | In and around this team, whenever people state their view, they also ask what others think.. | Likert (1-7) |
| | | In and around this team, people openly discuss mistakes in order to learn from them. | Likert (1-7) |
| | | In and around this team, people help each other learn. | Likert (1-7) |
| | Quality | Members of this team have a shared understanding of what quality means to them. | Likert (1-7) |
| | | People in this team frequently talk about quality and how to improve it. | Likert (1-7) |
| | Sprint Retrospective Quality | The Sprint Retrospectives of this team generally result in at least one useful improvement. | Likert (1-7) |
| | | During Sprint Retrospectives, this team openly talks about improvements. | Likert (1-7) |
| | | This team uses Sprint Retrospectives to explore solutions for persistent challenges. | Likert (1-7) |
| Team Autonomy | Cross-Functionality | Most people in this team have the ability to solve the problems that come up in their work. | Likert (1-7) |
| | | Everyone in this team has more than enough training and experience for the kind of work they have to do. | Likert (1-7) |
| | Self-Management | This team has control over the scheduling of teamwork. | Likert (1-7) |
| | | This team is free to choose the method(s) to use in carrying out work. | Likert (1-7) |
| | | This team is able to choose the way to go about its work. | Likert (1-7) |
| Management Support | Management Support | People in a management position generally understand why this team works with Scrum. | Likert (1-7) |
| | | People in a management position help this team work with Scrum. | Likert (1-7) |
| Team Effectiveness | Stakeholder Satisfaction | Stakeholders frequently compliment this team with their work. | Likert (1-7) |
| | | Stakeholders are generally happy with the software this team delivers. | Likert (1-7) |
| | | Stakeholders are generally happy with how fast this team responds to their needs. | Likert (1-7) |
| | | Our stakeholders compliment us with the value that we deliver to them. | Likert (1-7) |
| | Team Morale | I am proud of the work that I do for this team. | Likert (1-7) |
| | | I am enthusiastic about the work that I do for this team. | Likert (1-7) |
| | | I find the work that I do for this team full of meaning and purpose. | Likert (1-7) |
| Control Variables | Social Desirability | I am always courteous even to people who are disagreeable. | Likert (1-7) |
| | | No matter who I'm talking to, I'm always a good listener. | Likert (1-7) |
| | | I never feel resentful when I don't get my way. | Likert (1-7) |
| | Covid 19 Control | The Covid19 pandemic, and working from home, has impacted my answers on this survey | Likert (1-7) |
| | | I would have provided different answers if the pandemic never happened. | Likert (1-7) |

Table 16. Results of Confirmatory Factor Analysis. Principal Components Analysis with Oblimin rotation and Kaiser normalization. Items and components are ordered by extraction. Factor loadings below .30 have been suppressed for readability.

| Item | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| PsychologicalSafety2 | .719 | | | | | | | | | | | | | | | | |
| PsychologicalSafety3 | .688 | | | | | | | | | | | | | | | | |
| PsychologicalSafety1 | .604 | | | | | | | | | | | | | | | | |
| PsychologicalSafety4 | .560 | | | | | | | | | | | | | | | | |
| StakeholderCollaboration1 | | .943 | | | | | | | | | | | | | | | |
| StakeholderCollaboration3 | | .792 | | | | | | | | | | | | | | | |
| StakeholderCollaboration2 | | .790 | | | | | | | | | | | | | | | |
| TeamMorale2 | | | .893 | | | | | | | | | | | | | | |
| TeamMorale2 | | | .865 | | | | | | | | | | | | | | |
| TeamMorale3 | | | .845 | | | | | | | | | | | | | | |
| SharedLearning1 | | | | -.809 | | | | | | | | | | | | | |
| SharedLearning3 | | | | -.761 | | | | | | | | | | | | | |
| SharedLearning2 | | | | -.719 | | | | | | | | | | | | | |
| ReleaseFrequency1 | | | | | .931 | | | | | | | | | | | | |
| ReleaseFrequency2 | | | | | .903 | | | | | | | | | | | | |
| SelfManagement2 | | | | | | .868 | | | | | | | | | | | |
| SelfManagement3 | | | | | | .787 | | | | | | | | | | | |
| SelfManagement1 | | | | | | .664 | | | | | | | | | | | |
| ManagementSupport2 | | | | | | | -.931 | | | | | | | | | | |
| ManagementSupport1 | | | | | | | -.926 | | | | | | | | | | |
| SocialDesirability2 | | | | | | | | .863 | | | | | | | | | |
| SocialDesirability1 | | | | | | | | .854 | | | | | | | | | |
| Refinement2 | | | | | | | | | -.917 | | | | | | | | |
| Refinement3 | | | | | | | | | -.893 | | | | | | | | |
| SharedGoals2 | | | | | | | | | | -.972 | | | | | | | |
| SharedGoals1 | | | | | | | | | | -.953 | | | | | | | |
| CrossFunctionality1 | | | | | | | | | | | -.867 | | | | | | |
| CrossFunctionality2 | | | | | | | | | | | -.772 | | | | | | |
| StakeholderHappiness2 | | | | | | | | | | | | -.840 | | | | | |
| StakeholderHappiness4 | | | | | | | | | | | | -.836 | | | | | |
| StakeholderHappiness3 | | | | | | | | | | | | -.791 | | | | | |
| SprintRetrospectiveQuality1 | | | | | | | | | | | | | .808 | | | | |
| SprintRetrospectiveQuality3 | | | | | | | | | | | | | .790 | | | | |
| SprintRetrospectiveQuality2 | | | | | | | | | | | | | .668 | | | | |
| ValueFocus1 | | | | | | | | | | | | | | -.803 | | | |
| ValueFocus2 | | | | | | | | | | | | | | -.724 | | | |
| ValueFocus3 | | | | | | | | | | | | | | -.592 | | | |
| SprintReviewQuality1 | | | | | | | | | | | | | | | .778 | | |
| SprintReviewQuality2 | | | | | | | | | | | | | | | .761 | | |
| LearningEnvironment2 | | | | | | | | | | | | | | | | .818 | |
| LearningEnvironment1 | | | | | | | | | | | | | | | | .727 | |
| ConcernForQuality2 | | | | | | | | | | | | | | | | | -.810 |
| ConcernForQuality1 | | | | | | | | | | | | | | | | | -.794 |

Table 17. Means, Standard Deviations, and Correlations (Pearson) for first-order latent factors. All correlations are significant at $p < 0.001$

| | Factor | Mean | SD | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | Learning Environment | 4.57 | 1.15 | | | | | | | | | | | | | | | | |
| 2 | Concern for Quality | 4.52 | 1.12 | 0.76 | | | | | | | | | | | | | | | |
| 3 | Shared Learning | 3.62 | 0.99 | 0.75 | 0.79 | | | | | | | | | | | | | | |
| 4 | Sprint Retrospective Quality | 4.93 | 1.22 | 0.73 | 0.75 | 0.63 | | | | | | | | | | | | | |
| 5 | Psychological Safety | 5.06 | 1.12 | 0.87 | 0.85 | 0.76 | 0.85 | | | | | | | | | | | | |
| 6 | Shared Goals | 4.30 | 1.61 | 0.50 | 0.55 | 0.53 | 0.56 | 0.57 | | | | | | | | | | | |
| 7 | Stakeholder Collaboration | 3.74 | 1.35 | 0.53 | 0.50 | 0.59 | 0.35 | 0.50 | 0.37 | | | | | | | | | | |
| 8 | Sprint Review Quality | 3.53 | 1.20 | 0.56 | 0.59 | 0.63 | 0.52 | 0.57 | 0.62 | 0.76 | | | | | | | | | |
| 9 | Value Focus | 4.76 | 1.26 | 0.68 | 0.71 | 0.66 | 0.64 | 0.70 | 0.64 | 0.59 | 0.74 | | | | | | | | |
| 10 | Cross-Functionality | 4.57 | 0.88 | 0.69 | 0.77 | 0.67 | 0.66 | 0.79 | 0.52 | 0.49 | 0.52 | 0.66 | | | | | | | |
| 11 | Self-Management | 5.20 | 1.10 | 0.73 | 0.61 | 0.56 | 0.64 | 0.73 | 0.48 | 0.44 | 0.45 | 0.59 | 0.82 | | | | | | |
| 12 | Refinement | 4.31 | 1.30 | 0.53 | 0.60 | 0.51 | 0.61 | 0.54 | 0.49 | 0.28 | 0.47 | 0.57 | 0.49 | 0.49 | | | | | |
| 13 | Release Frequency | 4.48 | 1.48 | 0.45 | 0.60 | 0.47 | 0.49 | 0.54 | 0.42 | 0.45 | 0.52 | 0.48 | 0.56 | 0.47 | 0.48 | | | | |
| 14 | Team Morale | 5.12 | 1.08 | 0.60 | 0.64 | 0.61 | 0.61 | 0.69 | 0.47 | 0.46 | 0.52 | 0.65 | 0.68 | 0.63 | 0.40 | 0.45 | | | |
| 15 | Stakeholder Satisfaction | 4.71 | 1.05 | 0.62 | 0.67 | 0.62 | 0.60 | 0.67 | 0.47 | 0.61 | 0.68 | 0.70 | 0.70 | 0.61 | 0.47 | 0.61 | 0.71 | | |
| 16 | Management Support | 4.63 | 1.29 | 0.68 | 0.58 | 0.59 | 0.50 | 0.59 | 0.44 | 0.45 | 0.58 | 0.66 | 0.57 | 0.60 | 0.44 | 0.41 | 0.56 | 0.60 | |
| | **Control variables** | | | | | | | | | | | | | | | | | | |
| 17 | Social Desirability | 4.37 | 0.69 | 0.55 | 0.54 | 0.53 | 0.54 | 0.64 | 0.41 | 0.36 | 0.39 | 0.49 | 0.63 | 0.58 | 0.36 | 0.37 | 0.67 | 0.58 | 0.50 |

Table 18. Parameter Estimates, Confidence Intervals, Standard Errors, Standardized Coefficients for Direct Effects and Indirect effects for hypotheses (statistically significant hypotheses at $p < 0.05$ are set in boldface), and Factor Loadings

| Parameter | Unstandardized | 95% CI | SE | *p* | Standardized |
|---|---|---|---|---|---|
| *Direct Effects* | | | | | |
| **H1: Responsiveness → Team Effectiveness** | **.235** | **(.111, .418)** | **.050** | **.007** | **.402** |
| **H2: Stakeholder Concern → Responsiveness** | **.400** | **(.265, .544)** | **.067** | **.002** | **.414** |
| **H3: Stakeholder Concern → Team Effectiveness** | **.291** | **(.138, .417)** | **.047** | **.015** | **.513** |
| **H4a: Continuous Improvement → Stakeholder Concern** | **.616** | **(.491, .749)** | **.061** | **.001** | **.545** |
| **H4b: Continuous Improvement → Responsiveness** | **.372** | **(.166, .582)** | **.082** | **.008** | **.340** |
| **H5a: Team Autonomy → Continuous Improvement** | **.710** | **(.61, .825)** | **.054** | **.001** | **.655** |
| H5b: Team Autonomy → Stakeholder Concern | .135 | (.011, .273) | .061 | .074 | .110 |
| **H5c: Team Autonomy → Responsiveness** | **.306** | **(.110, .513)** | **.081** | **.015** | **.258** |
| **H6a: Management Support → Team Autonomy** | **.251** | **(.208, .292)** | **.020** | **.001** | **.427** |
| **H6b: Management Support → Continuous Improvement** | **.107** | **(.067, .147)** | **.020** | **.001** | **.168** |
| **H6c: Management Support → Stakeholder Concern** | **.184** | **(.143, .226)** | **.021** | **.001** | **.255** |
| H6d: Management Support → Responsiveness | -.027 | (-.078, .029) | .028 | .466 | -.039 |
| *Indirect Effects* | | | | | |
| **H3: Stakeholder Concern → Responsiveness → Team Effectiveness** | **.094** | **(.044, .225)** | **.138** | **.004** | **.166** |
| **Continuous Improvement → Responsiveness → Team Effectiveness** | **.087** | **(.047, .152)** | **.092** | **.004** | **.137** |
| **Team Autonomy → Responsiveness → Team Effectiveness** | **.072** | **(.015, .175)** | **.051** | **.016** | **.104** |
| **Team Autonomy → Continuous Improvement → Responsiveness → Team Effectiveness** | **.062** | **(.034, .108)** | **.023** | **.003** | **.223** |
| *Factor Loadings* | | | | | |
| Self-Management → Team Autonomy | | (.682, .789) | | | .739 |
| Cross-Functionality → Team Autonomy | .882 | (.736, .848) | .050 | .001 | .791 |
| Learning Environment → Continuous Improvement | | (.706, .782) | | | .745 |
| Concern for Quality → Continuous Improvement | 1.016 | (.759, .825) | .052 | .001 | .792 |
| Shared Learning → Continuous Improvement | .786 | (.636, .709) | .046 | .001 | .672 |
| Sprint Retrospective Quality → Continuous Improvement | 1.044 | (.676, .744) | .054 | .001 | .713 |
| Psychological Safety → Continuous Improvement | 1.004 | (.824, .883) | .05 | .001 | .856 |
| Sprint Goals → Stakeholder Concern | | (.531, .605) | | | .571 |
| Stakeholder Collaboration → Stakeholder Concern | .780 | (.517, .597) | .054 | .001 | .558 |
| Sprint Review Quality → Stakeholder Concern | 1.124 | (.701, .782) | .062 | .001 | .741 |
| Value Focus → Stakeholder Concern | 1.089 | (.784, .847) | .053 | .001 | .818 |
| Refinement → Responsiveness | .840 | (.516, .626) | .055 | .001 | .570 |
| Release Frequency → Responsiveness | | (.525, .637) | | | .582 |
| Team Effectiveness → Stakeholder Happiness | 1.336 | (.715, .805) | .078 | .001 | .761 |
| Team Effectiveness → Team Morale | | (.542, .635) | | | .591 |