Research Article

Scrum Language Use in a Software Engineering Firm: An Exploratory Study

—ERIN FRIESS

***Abstract*—Background:** *Although agile and Scrum have been important frameworks in software engineering for over a decade, little research has explored how teams use Scrum language within their sprints.* **Literature review:** *Most explorations of Scrum communication have been collected through self-reported means. These studies are inherently unable to explore how Scrum teams use Scrum-centric language in their meetings in ways that adhere or run counter to standard Scrum practice.* **Research questions:** *1. In what ways is Scrum reflected in the language used by team members in various sprint meetings? 2. What associations exist between the job title of team members and their use of Scrum language? 3. What does a discourse analysis reveal about the ways in which this team uses language to value and discount Scrum?* **Research methodology:** *For three sprints over 10 weeks, I recorded meetings of 27 Scrum team members. I transcribed these meetings, developed a codebook for assigning Scrum language categories, conducted an interrater reliability agreement on the data, completed a correspondence analysis on how Scrum language associates with meeting types and job titles, and conducted a discourse analysis to determine in what ways these teams value and discount Scrum.* **Results/discussion:** *Scrum language was found in all recorded meetings across all three sprints, with much language found in the planning meetings. Few associations existed between Scrum language and job title, suggesting that Scrum at this engineering firm is an egalitarian process. In addition, the discourse analysis revealed that this engineering firm valued User Story and Sprint Execution language while discounting Capacity and Story Pointing language.* **Conclusions:** *Although this group broadly adheres to Scrum practices about 68% of the time, this study finds that several current standard components of Scrum are routinely discounted. This exploratory study suggests that more research into the in-situ use of Scrum language in engineering workplaces is necessary to better inform engineering professionals about the communicative expectations of Scrum and to better enable engineering communication educators to prepare future engineers for Scrum realities.*

Index Terms*—Agile, discourse analysis, project management, Scrum.*

Agile has been a go-to process framework for software development for over 15 years, and Scrum, a specific agile process, has been the primary agile framework for more than a decade [1]–[4]. However, despite increased adoption of agile and Scrum, the communication patterns among Scrum team members within the workplace, including engineering workplaces, has been understudied [5]–[7]. Much of the discussion of Scrum centers on anecdotal experiences and prescriptions [8]–[10] or on self-reported data in the form of interviews, surveys, and focus groups [11]–[15]. In other words, little is known about how Scrum language is used *in situ* within workplace environments.

This lack of understanding has a handful of practical implications. First, Scrum is inherently a form of workplace teamwork. However, research suggests that although workers say that teamwork is important in the workplace, relatively few workers actually like to be involved in teamwork-based activities [16]–[18]. One study suggests that for agile teams, "the quality of teamwork is a major factor in improving team performance," and things as relatively mundane as misunderstandings pertaining to the group's activities, procedures, schedules, and goals could be factors in the quality of teamwork [19], [20, p. 282]. Non-parallel approaches to and misunderstandings of Scrum among team members could possibly derail what is otherwise a well-composed Scrum team. Therefore, a closer look at how team members of various job titles use Scrum language in their meetings may allow for a better understanding of how a group adheres to or deviates from Scrum principles and what those fluctuations may mean for Scrum implementation in practice. Furthermore, such a

Manuscript received August 31, 2018; revised December 19, 2018; accepted December 21, 2018. Date of current version May 17, 2019.
The author is with the Department of Technical Communication, University of North Texas, Denton, TX 76203 USA (email: Erin.Friess@unt.edu).
This paper has supplementary downloadable material at http://ieeexplore.ieee.org provided by the translators. The files consist of a Chinese translation of the abstract by Z. Xi (82 kB in size) and a Spanish translation of the abstract by R. Anido (78 kB in size).

IEEE 10.1109/TPC.2019.2911461

Practitioner Takeaway

- Although agile and Scrum have been important frameworks in software engineering for over a decade, research has not typically explored how teams use Scrum language within their sprints, and most has been based on self reports.
- Based on observation of three sprints over 10 weeks, this article finds that Scrum language was used in all recorded meetings, with much occurring in planning meetings.
- Discourse analysis revealed that this company valued User Story and Sprint Execution language while discounting Capacity and Story Pointing language.

study may enable to better prepare students for Scrum practice realities rather than Scrum myths.

Therefore, to identify what actually happens in Scrum environments, I observed and audio-recorded teams at a software engineering firm that had used Scrum methodologies for over a decade. Few studies have attempted to explore Scrum behaviors *in situ*, and this study addresses this gap. In addition, this study is exploratory in nature, in that its goal is not to necessarily provide generalizable outcomes but to "identify patterns . . . that one might not predict or assume," thus informing possible directions of future research [21, p. 301].

Literature Review

In this literature review, I begin with a broad overview of agile and Scrum and then discuss the prior communication research on these frameworks. I also overview the methods of analysis.

Overview of Agile and Scrum Agile is a framework or "project management ideology" that was developed in the 1990s as a "lightweight" software development alternative to the heavyweight "waterfall" approach of the 1980s [22, p. 114]. In waterfall approaches, each sequential aspect of a project was to be fully completed by specialists, such as designers or developers, before being handed off to the next specialist in line. Agile aimed at being much more responsive than the waterfall framework that required much "upfront planning and control" with limited changes to ensure the costs and schedule would remain intact [23, p. 4]. Instead of waiting to have projects handed off to them, agile relocated specialists working in specialty silos to

> multi-disciplinary and highly communicative teams that share their experiences and tacit knowledge in order to gain consensus regarding the solution. [24, p. 12]

In other words, agile proposed a shift away from a software methodology in which specialists acted individually to one in which specialists worked with a variety of other specialists throughout a project.

Scrum, a particular implementation of the agile principles, is a highly stylized agile approach to project management in terms of personnel, meetings, and tasks. I will address the components of Scrum most relevant to the current study here, and I encourage readers to explore Scrum more fully in the many resources dedicated to Scrum procedure [24]–[27].

Scrum proposes that work be "performed in short, timeboxed iterations, which usually range from a week to a calendar month in length" [25, p. 20]. During these timeboxed iterations, called sprints, "a cross-functional team does all of the work—such as designing, building, and testing—required to produce a completed working feature" [25, p. 16]. There are three roles for personnel on the Scrum team. The first is that of the team member, and there are usually 6–10 team members on each Scrum team, though there can be many more [19]. These specialists make up the cross-functional team. One team member is also the product owner, "the single voice of the stakeholder community" [25, p. 15]. Finally, the ScrumMaster is the "coach, facilitator, [and] impediment remover" who "provides process leadership" to the Scrum team [25, p. 16]. The ScrumMaster typically also serves another purpose on the team (i.e., he or she is also an architect or designer), and it is common for the ScrumMaster to change with each sprint.

The work of the team is conducted over a sprint with several hallmark meetings and tasks. The meetings are typically categorized as planning, standups, reviews, and retrospectives. In the planning meeting, the team identifies and prioritizes the issues (called “user stories”) that are to be accomplished by the end of the sprint (a process known as “grooming”). Furthermore, for each user story, a “definition of ‘done”’ is assigned by the team. Team members also discuss their capacity for work, and they assign story points to each user story that reflect the complexity of that user story. There is some flexibility in the setup of the meetings. In some teams (such as the one under review here), grooming and user-story development occur in the planning meeting, while the story pointing and capacity are discussed in a separate kickoff meeting. In the review meeting, all stakeholders are invited to see the team demonstrate how they have met the definitions of “done” for the user stories within the sprint. In the retrospective meeting, which is held at the conclusion of the sprint, the team members, in an attempt to improve the Scrum process, identify one or two strategic changes to implement during the next sprint.

During the sprint, team members hold a daily standup meeting in which each team member describes what they have accomplished since the last meeting, what they will do prior to the next standup, and obstacles that they are facing in pursuit of the definition of “done” for their assigned user stories. In some teams (like the one observed for this study), issues that extend beyond the scope of the standup are moved to a sprint maintenance meeting. Although a maintenance meeting is not a meeting specifically described in Scrum frameworks, it is a type of meeting often created by teams when issues from stand-up need further discussion [25].

Scrum is a highly structured process, and other resources can more thoroughly explain and examine Scrum implementation. For the purposes of this study, the sprint roles, the meeting types, and the task descriptions are the most critical components.

Agile and Scrum Communication Strategies Much research has been conducted on agile, broadly, and Scrum, specifically, with topics such as team composition [28]–[30], use and usefulness of agile and Scrum [31]–[34], and potential modifications to Scrum [28], [35]–[37] being well studied. However, systematic literature reviews of agile and Scrum have failed to demonstrate significant empirical research dedicated to communication within agile and Scrum environments [8]–[10], [38]. Much research about communication in Scrum is not about the content of the Scrum interactions but the barriers to team communication or the distribution of team members for communicative purposes [39]–[42].

In addition to the studies of barriers and distribution, some studies have approached the issue of communication within agile and Scrum frameworks. Conboy et al. found through semistructured interviews that a general lack of communication across team members resulted in team members disappointed in team interactions [43]. Dorairaj et al. also found through semistructured interviews that the lack of informal communication among team members perhaps negatively affected the formal meeting-based communication [44]. Downs et al. found through observational study and semistructured interviews that continuous communication enables stronger and more trustworthy collaborations [45]. Treude and Story investigated through observation and interviews how modifications to physical documents could improve overall Scrum communication [46].

However, none of the studies looked specifically at the language within the interactions among the team members. Only one study appears to assess the language used within Scrum sprints. McNely et al. found through observations, audio recordings, and interviews that the language interactions of team members, particularly through their self-mediations and coarticulations, enabled teams to pursue the accountability required by Scrum [47]. But even in this detailed longitudinal study, little attention is paid to the specific language the group used as a means to assess their Scrum practice. In other words, prior studies have assumed that the teams are, in fact, practicing Scrum because the teams themselves report to be practicing Scrum. Few studies have looked at how a team’s use of Scrum-specific language reflects upon the team’s acceptance of and dedication to Scrum [48].

Methods for Analyzing Talk Although agile and Scrum language have not previously been the object of study, there is a long history of assessing spoken communication both in and out of the workplace. Two rather complementary methods of analysis have long histories in assessing spoken communication [49]. First, discourse analysis is a qualitative method of analyzing language structure and meaning [49]. Discourse analysis allows for researchers to assess texts (including spoken exchanges) to ascertain how language is used by speakers and what those language choices reveal about the speakers' attitudes [50]. Discourse analysis has been used in workplace studies to explore power [51]; gender [52], [53]; politeness [51], [54], [55]; decision-making [54], [56], [57]; and leadership [58], among others.

Second, quantitative content analysis is a method that permits a rigorous assessment of "words, phrases, or in-text relationships" [59] and can allow for a "fixed, stable, and objectively verifiable" assessment of a corpus [49]. In a quantitative content analysis, a corpus is identified, the data are organized, a codebook is developed and tested for inter-rater reliability, and then the data are analyzed [59]. This rigorous approach allows for replicability of the assessment, which, in turn, allows for more longitudinal and distributed analyses across researchers. Content analysis has been used in business communication studies to explore the transfer of communication skills [60], the kinds of communication used in social media [61]–[63], and the development of professional identities [64], among others. As a whole, technical and professional communication hosts few quantitative content analysis studies that assess spoken exchanges [48].

One means of further assessing categorical data collected through content analysis is correspondence analysis, a method that "reveals patterns in complex data and provides output that can help researches interpret these patterns" [21], [65]. These patterns are useful to researchers who approach research from an exploratory rather than hypothesis-driven perspective. Correspondence analyses do not show causal relationships but indicate where relationship associations are possible. Although correspondence analysis has been used widely in a variety of disciplines, it has been used only limitedly in professional and technical communication [21], [65]–[69].

Therefore, this current study addresses the knowledge and methodological gaps by exploring the actual language used by teams involved in three Scrum sprints and assessing that language empirically in three ways. Learning more about how team members reflect Scrum activities in their language will enable a better understanding of how Scrum is employed in engineering workplaces. Specifically, this study asks the following questions:

> **RQ1.** In what ways is Scrum reflected in the language used by team members in various sprint meetings?
>
> **RQ2.** What associations exist between the job title of team members and their use of Scrum language?
>
> **RQ3.** What does a discourse analysis reveal about the ways in which this team uses language to value and discount Scrum?

METHODOLOGY

To identify how Scrum is used, valued, and discounted by team members within a sprint, I audio-recorded three sprints of a midsized US-based software engineering firm. According to the non-disclosure agreement (NDA) agreed upon by myself and the firm, I could audio-record and transcribe all meetings that did not include the firm's clients. Thus, I was not able to record the review meetings or several daily standup meetings that often included clients. Because I could not include all of the standup meetings in my dataset, I removed all standup meetings from this analysis.

Ideally, of course, these meetings would have been included in the dataset if the NDA had permitted, and such data would have provided a more thorough longitudinal perspective of the use of Scrum during the sprint. By adhering to the guidelines of the NDA, I removed approximately 64% of the total sprint meetings. However, stand-up meetings were strictly limited to 7 min. In addition, some stand-ups were shorter than those 7 min. Issues that pushed stand-ups to longer than 7 min were converted to maintenance meetings, and those maintenance meetings were all recorded because the clients never stayed for those meetings. Therefore, given the brevity of the stand-ups, the unrecorded meetings (and thus meetings not included in the dataset) consisted of less than 20% of the total meeting time the team members spent in meetings during the sprint. The need to omit some data to abide by the NDA is unfortunate; however, 80% of the data that remain allow for an in-depth first insight into how engineers use Scrum language in their workplace meetings. If we include language from the review meetings that was also not permitted in the dataset, about 75% of the data remain. Perhaps future researchers will be able to locate other real-world research sites that impose fewer recording restraints, which would thus eliminate this limitation faced by this study.

The three sprints were recorded over 10 weeks. Each sprint lasted 5 weeks, with Sprint 1 occurring first, followed by two concurrent sprints (Sprints 2a and 2b). The total number of hours that teams spent in the recorded meetings was about 15 h. The average meeting length was 69 min, with the longest meeting being 133 min and the shortest being 23 min. Although some teams at the firm work on new product development, the teams that I recorded work on feature development for existing software. Each team member signed an Institutional Review Board-approved consent form and completed a brief demographic survey prior to the recording of the meetings. The consent form stated that I would be audio recording the data, transcribing and anonymizing the data, and then coding the data based on the anonymized transcriptions. For Sprint 1, I was present in the room in which the ScrumMaster conducted the meeting. Also, for Sprint 1, I observed the stand-ups but did not record or take notes. For Sprints 2a and 2b, I called in to the meetings and informed all team members that I was on the call.

To assess how the teams used language related to the Scrum framework, I conducted a quantitative content analysis coupled with a correspondence analysis. I developed a codebook based on the Scrum activities described in two definitive resources for Scrum implementation [25], [26]. The following 10 codes for Scrum activities were included in the codebook:

- Capacity
- Commitment
- Sprint goal (defining “done”)
- Flow management
- Forecasting
- Grooming of backlog
- Sprint execution
- Story pointing
- User stories
- Velocity

The unit of investigation was clauses because spoken communication lacks traditional sentence structure. Furthermore, clauses, as relatively small units of measure, allowed for mutual exclusivity of the codes.

After coding approximately half of the dataset, I conducted an inter-rater reliability assessment. An independent rater and I both separately coded 10% of the dataset. The assessment yielded a Krippendorff’s alpha of 0.84, which indicates good agreement. After modifying the descriptions of the code based on areas of disagreement, we again coded a different 10% of the dataset. This second round of coding yielded a Krippendorff’s alpha of 0.90, which indicates very good agreement. The final codebook is available in Appendix I.

To determine how the teams valued or discounted Scrum language, I divided the meetings into subject sections when the conversations shifted in subject matter. Each meeting had between 5 and 33 subject sections. For each subject section, an independent rater and I marked each subject section as to whether the team members mostly adhered to the Scrum guidelines or mostly ignored or deviated from the guidelines. Every subject section was coded (i.e., no subject section was left blank). The assessment yielded a Krippendorff’s alpha of 0.92, which indicates very good agreement. I used these subject section markers to guide the discourse analysis on the ways that the Scrum-based language was valued and discounted by the groups [70]–[73].

RESULTS

I begin this results section with an overview of the descriptive statistics before exploring the research questions.

A total of 27 unique team members populated the three reviewed sprints. Sprint 1 had 16 members. Sprint 2a had 13 members with 7 members from Sprint 1. Sprint 2b had 14 members with 7 members from Sprint 1. The number of team members exceeds what is generally considered to be ideal [26], [27]. Two high-level members participated in all three sprints (project manager and product owner). On Sprints 1 and 2b, the same designer functioned as ScrumMaster. Sprint 2a also had a designer serve as ScrumMaster. In these three sprints, all developers, Quality Assurance (QA), and designers worked solely on one sprint at a time. All other positions worked across multiple sprints (though not all of those sprints were recorded for this study). The breakdown of participants by job title can be seen in Table I.

Although co-location is often desired, these teams, as is relatively common in Scrum environments, were geographically distributed [26], [74]. Although the meetings were technically held in a conference room at the firm’s US-based headquarters, fewer than half of the team members actually met face-to-face in the meeting room. One developer worked remotely 100% of the time from India, while another developer worked remotely from another US state 100% of the time. All other team members took part in the firm’s remote work policy, which

TABLE I
NUMBER OF TEAM MEMBERS WITH EACH JOB TITLE ON EACH SPRINT

	Sprint 1	Sprint 2a	Sprint 2b
QA	3	6	3
Developers	4	2	3
Architect	2	1	3
Designers	2	1	2
IT Engineer	1	1	0
Product Owner	1	1	1
Project Manager	1	1	1
Security Engineer	1	0	1
Technical Writer	1	0	0

enabled team members to work from home up to three days per week. Over 90% of the participants took advantage of this policy. Team members selected which days they would go to the office, often without supervisor approval. There was no pattern of who met face-to-face and who did not. Sometimes, team members who found themselves on several concurrent sprints would be at the headquarters but would phone into the meeting from their cubicles so that they could continue to work. During the period of my study, the percentage of team members meeting face-to-face in any given meeting ranged from 31% to 57%.

In addition, a total of seven team members were non-native English speakers. Nine identified their gender as female, and 18 identified their gender as male.

RQ1: In What Ways Is Scrum Reflected in the Language Used by Team Members in Various Sprint Meetings? The three sprints included a total of 19 recorded meetings (7, 6, and 6 in each sprint, respectively). The total number of clauses identified in all meetings exceeded 24,000. A little less than 18% of the clauses were coded as containing Scrum language (4438 clauses). A further breakdown of the clauses can be seen in Table II.

Although we might expect the language of the team to reflect the activities posited by Rubin and by Schwaber and Sutherland, that does not appear to be exactly the case [25], [26]. Indeed, the team never used language related to Velocity or Grooming of Backlog. This finding may be due to these issues being handled outside the framework of the meetings that I recorded. Indeed, typically the product owner decided on the sprint backlog after consulting with other product owners. Of the types of Scrum language that the team did use, 69.1% were related to only two categories: Sprint Execution (44.1%) and User Stories (25.0%). Less commonly used categories included language related to Capacity (10.3%), Story Pointing (7.5%), and Commitment (6.7%). Language related to Sprint Goals, Flow Management, and Forecasting were used less than 3% of the time when Scrum language was invoked.

A closer look at the meeting types suggests that each meeting type may have some Scrum-based hallmarks (see Table III). All eight Scrum categories used by the team were used in the planning meetings, and at least one category was not invoked in kickoff, maintenance, and retrospective meetings. This finding may suggest that the planning meetings cover more Scrum-based touchpoints than the other meetings. It may also suggest that kickoff, maintenance, and retrospective meetings are more narrowly focused, making the omission of certain categories (i.e., Forecasting, Story Pointing, User Stories) expected.

The most common Scrum-language category in every meeting was Sprint Execution, and Sprint Execution itself was most common in kick-off meetings. Kick-off meetings were also the most common meetings to locate discussions about Capacity and User Stories. Story pointing was never invoked outside planning and kickoff meetings. Planning meetings were the most common meeting to locate Story Pointing and Goal Definition. A correspondence analysis (see Appendix II) shows that despite these aggregate numbers, there are relatively few associations across meeting type and Scrum-language category. Most notably, kickoff meetings seem to be associated with Capacity, and maintenance meetings seem to be associated with Commitment. The few associations suggest that specific types of Scrum language are not necessarily reserved for specific types of meetings. Although certain language (such as Story Pointing) might be more likely to be found in certain kinds of meetings, the Scrum language used by this team is used relatively evenly throughout the meetings. Because part of the agile and Scrum process is to be self-reflective and modified for future sprints, it is not entirely surprising that most Scrum-language categories were used throughout all of the meetings.

This holistic assessment of the recorded meetings suggests that this groups leans into certain Scrum

TABLE II
DESCRIPTIVE STATISTICS OF THE CLAUSES IN MEETINGS

Type of Meeting (Number of Meetings)	Total Number of Clauses	Percentage of Clauses	Total Number of Scrum Clauses	Percentage of Scrum Clauses
Planning (5)	7544	30.4%	1550	34.9%
Kick-off (5)	9008	36.4%	1846	41.6%
Maintenance (6)	7002	28.3%	919	20.7%
Retrospective (3)	1224	4.9%	123	2.8%
Total (19)	24778	N/A	4438	N/A

TABLE III
TOTAL NUMBER OF CLAUSES OF EACH SCRUM-LANGUAGE CATEGORY PER MEETING TYPE

	Planning	Kickoff	Maintenance	Retrospective
Capacity	84	332	36	3
Commitment	69	95	133	0
Defining Goal	66	14	12	12
Flow Management	49	52	0	15
Forecasting	32	0	24	0
Sprint execution	622	784	480	81
Story Point	199	133	0	0
User Stories	429	443	239	0

categories in their talk, such as Sprint Execution and User Stories. It also suggests that other categories, such as Story Pointing and Goal Defining, are used less often on the whole and within particular meetings. Therefore, these data suggest that although Scrum-based language is not necessarily limited to particular meetings, some meeting-language associations emerge for this group.

RQ2: What Associations Exist Between the Job Title of Team Members and Their Use of Scrum Language? Team members were included in the sprint for some function related to their job title. The overall breakdown in terms of total clauses by job title per sprint can be seen in Table IV.

Furthermore, the breakdown of the number of clauses that contain Scrum language used by each job title can be seen in Table V.

Looking at Tables IV and V in tandem, a few take-aways appear related to overall Scrum use and job title. First, developers and designers combined to contribute approximately 64.9% of all clauses and 67.2% of the Scrum clauses in the sprints. This finding suggests that the developers and designers are the largest contributors of overall clauses and that their adherence to Scrum is needed for the group to maintain its Scrum-focus. In addition, the project manager (who was the same person for all three sprints) contributed the fewest overall clauses out of all the job titles (1.9%), but the project manager contributed the fourth-largest number of Scrum clauses at 9.1%. This finding may suggest that the project manager is highly focused on the standard aspects of Scrum and that other team members often discuss elements of the sprint that are not directly tied to the Scrum process. Those team members grouped as “other” (security engineer, IT engineer, and technical writer) contributed more overall clauses than the project manager (2.5%), but contributed the fewest Scrum clauses (0.6%). This finding suggests that perhaps these “other” positions are on the periphery of the Scrum process.

The data further suggest take-aways related to the specific type of Scrum use and job title. For example, Sprint Execution was the most commonly referenced category of Scrum by all the job titles except for the project manager (who referenced Capacity the most) and “other” (who referenced User Stories the most). The second most commonly referenced category for each job type varied more, but was one of only three of the Scrum categories. Developers, designers, and QA referenced User Stories; architects and product owners referenced Capacity; and project managers and “other” referenced Story Pointing as the second most common Scrum category. The least commonly referenced Scrum categories were Forecasting (for developers, designers, and architects) and Defining Goal (QA and project manager). Product owner and “other” had several categories that were never mentioned.

A correspondence analysis of job title and Scrum language showed potential associations between QA and Flow Management, product owner and

TABLE IV
TOTAL NUMBER OF CLAUSES PER SPRINT

	Sprint 1		Sprint 2a		Sprint 2b		Total Across All Three Sprints	
	Total	Avg.	Total	Avg.	Total	Avg.	Total	Avg.
Developer	2797 (4 team members)	699.25	1838 (2 team members)	919	3568 (3 team members)	1189.3	8203 (6 unique team members)	1367.2
Designer	3396 (2)	1698	2557 (1)	2557	1815 (2)	907.5	7768 (3)	2589.3
QA	801 (3)	267	1337 (6)	222.8	1644 (3)	548	3782 (10)	378.2
Architect	1148 (2)	574	432 (1)	432	574 (3)	191.3	2154 (3)	718
Product Owner	765 (1)	765	376 (1)	376	642 (1)	642	1783 (1)	1783
Project Manager	65 (1)	65	251 (1)	251	155 (1)	155	471 (1)	471
Other	240 (3)	80	223 (1)	223	154 (1)	154	617 (3)	205.7

TABLE V
SCRUM CLAUSE USE BY JOB TITLE

		Capacity	Commitment	Defining Goal	Flow Management	Forecasting	Sprint Execution	Story Point	User Story	Total Scrum Clauses
Developer (6 unique team members in all sprints)	Total	107	109	18	14	4	524	47	385	1208
	Avg.	17.8	18.2	3	2.3	0.7	87.3	7.8	64.2	201.3
Designer (3)	Total	49	145	65	54	23	718	147	572	1773
	Avg.	16.3	48.3	21.7	18	7.7	239.3	49	490	591
QA (10)	Total	28	7	0	22	11	200	35	48	351
	Avg.	2.8	0.7	0	2.2	1.1	20	3.5	4.8	35.1
Architect (3)	Total	40	8	14	12	0	380	15	33	502
	Avg.	1.3	2.7	4.7	4	0	126.7	5	11	167.3
Product Owner (1)	Total	69	0	7	0	3	92	0	0	171
	Avg.	69	0	7	0	3	92	0	0	171
Project Manager (1)	Total	162	23	0	14	15	48	80	63	408
	Avg.	162	23	0	14	15	48	80	63	408
Other (3)	Total	0	5	0	0	0	5	8	10	28
	Avg.	0	1.7	0	0	0	1.7	2.7	3.3	9.3

Capacity, and project manager and Capacity (see Appendix III). The lack of associations suggests that this group has a relatively egalitarian approach to Scrum in that no one person or job title has access to Scrum language that is off-limits. In this way, the group is reflecting Scrum's core "values of honesty, openness, courage, respect, focus, trust, empowerment, and collaboration" [25, p. 13].

To summarize, although team members with certain job titles spoke more about certain Scrum categories than others, most Scrum language categories were accessible to all team members. This finding suggests that the members of the multidisciplinary team have the ability to introduce and comment on nearly all aspects of the Scrum process; in other words, it does not appear that certain job titles necessarily shut other team members out of the conversation. However, this finding also suggests that all team members must be familiar with all components of Scrum to relate and reflect on those components in their discussion. Thus, it may potentially be useful for all team members regardless of title, either in onboarding or in their educational training, to learn more about how the Scrum process should be adhered to so that they may be prepared to adequately contribute to the discussion process.

RQ3: What Does A Discourse Analysis Reveal About the Ways in Which This Team Uses Language to Value and Discount Scrum? The incorporation of certain language into meeting discussion and the omission of other language suggests a value judgment. Language related to Velocity and Grooming of Backlog is completely absent from the meeting discussions. This finding suggests that these categories are potentially not important to the teams.

To further assess how the team values Scrum, I looked to the 68% of the subject sections in which the teams mostly adhered to Scrum practice. In these subject sections, the primary Scrum language was Sprint Execution and User Stories. The user stories themselves deviate somewhat from the standard "As a <User>, I want <Functionality> so I can <Action>" structure favored by most Scrum guides [25], [75]. Instead, the user story structure tended to drop the "User" component of the story and was more like the following from Sprint 2a, spoken by the designer ScrumMaster:

> Alright, so this is what we've got down for now: Add [editing functionality] in the [fulfillment page] because [client] said they want it in by end of quarter.

Despite this deviation from the standard user story structure, this group's language showed that they valued user stories. Specifically, their language suggests that they place value in a well-written (for their purposes) user story. For example, in Sprint 2a, the designer who was also the ScrumMaster said, "Okay, we got a lot figured out, but now we, it's not where we need it for dev yet." In Sprint 2b, a QA says,

> Umm, this is just a little, I think we could make this clearer or maybe we need to break it into two and save half for the next sprint, 'cause you know it might be too much or we might do too much.

These statements indicate that the team members place value and respect well-created user stories that consider the work that other team members either have done or will have to do in support of the user story. By encouraging the team to more fully develop and define the user stories, the designer and the QA representative are demonstrating the value that they place on well-crafted user stories. The product owner, half of the QA representatives, and all of the architects, developers, and designers made some kind of statement that indicated that a well-crafted user story is a pertinent issue for the success of the sprint.

Similarly, the same team members also indicated dedication to Sprint Execution. In several meetings, developers, architects, designers, and QA representatives carefully organized the flow management of the sprint and assigned tasks to each team member based on thoughtful considerations of their experience and their capacity. For example, at one point, the designer ScrumMaster said, "Let's think through this carefully .... If [developer] takes it up, when can we expect the turnaround from [QA]?". By actually saying "let's think through this carefully," the designer ScrumMaster is bringing direct and purposeful attention to the Sprint Execution issue at hand. In addition, a QA representative said, "We can get that back faster if Dev can, you know, maybe give us the first half?" Here, the QA representative directly indicates what can be done to improve the overall sprint execution for the team. Sprint Execution was not an aspect of the sprint to get through quickly; it was an aspect that for this group deserved careful study to ensure a successful sprint. The purposeful attention paid to the issues of the Sprint Execution along with the sheer amount of time dedicated to Sprint Execution indicates that this team values attention paid to Sprint Execution.

However, 32% of the subjects revealed language that suggests that the team members did not value some aspects of the Scrum process. For example, in the kickoff meeting of Sprint 2b, a developer indicated that one user story on tap for the sprint was already completed. The developer asked the product owner if the user story should be shifted to "completed," and the project manager said, "No, I need to, just leave it there so we can fill Capacity" (i.e., if the project manager removed the user story, then capacity would not be used entirely). Also in the Sprint 2b kickoff, the designer, after looking at the Capacity, humorously said to the project manager, "C'mon [project manager], do your math magic to make the numbers come out right in the burndown!" (i.e., make the numbers work so that the burndown chart won't show any unused Capacity). A developer immediately said, "Anything to make the numbers look right." And finally, in the Sprint 2a kickoff meeting, the project manager asked a QA representative, “Can I convince you to roll back [one of the task Capacity hours] to 8 instead of 10 so you’ll be in the green [i.e., so you’ll not go over Capacity]?” The QA representative responded, “Whatever you need to do.” These language choices suggest that Capacity is not important to the team members who had a job title other than project manager. Their language choices such as “whatever you need to do” and “anything to make the numbers look right” suggest that they care little about actually following the guidelines of Capacity; they seem to want to just move on to another Scrum category that more genuinely affects the outcomes of the sprint.

Similarly, this group seems to discount Story Pointing. For example, in assigning story points to one user story, a developer explained that because the user story involved a situation that the firm had never faced before, he assigned the user story a 20, an unusually high number for this group. The following exchange then took place:

> Project Manager: Wow! A 20? Are you sure? We’ve never assigned a 20 before.
>
> Developer: Well, you know, it’s just that, [client] wants us to add [a functionality] but doesn’t want us to use [a certain piece of software]. I’ve never done that. I don’t think like, you know, anyone here has ever tried anything like that. I’m not even sure it’s you know possible. Has anyone . . .
>
> Project Manager: But a 20? That just seems, it seems to me like that a 13 might be more reasonable.
>
> Architect: Are you sure [about a 13]? I mean, it’s going to be very difficult to even just get the environment [where the coding and testing can take place] where it can handle that.
>
> Project Manager: 20s are just very big red flag. Like big red flags. Are you like sure that a 13 won’t work?
>
> Developer: Sure. A 13 is fine.
>
> Project Manager: Awesome.

The project manager’s pushback on the story point of 20 was interesting given that a 20 is a legitimate number to assign to a user story (though, in the 10 weeks I recorded, 13 was, indeed, the highest number actually assigned to a user story). The developer relented to the project manager’s insistence with a less than persuasive “sure.” Although the statement by the developer indicated agreement, the tone and the context indicated that the developer perhaps simply saw an argument on story point assignment as an argument not worth pursuing.

In a different sprint, a developer wanted to assign a 13 to a particular user story.

> Project manager: Y’all, I just want to point out that, can you see this, that will give us . . . five 13s for this sprint. Can we like rethink that for a minute?
>
> Developer #1: Do we want to ya know like save [user story #2] for the next sprint?
>
> Project manager: No, ‘cause these are all a priority for [client] and we have got to get them going like yesterday.
>
> Developer #1: We can’t save one for next sprint?
>
> Project manager: Nope. Gotta get these turned around.
>
> Designer #1: Yeah, well I feel comfortable with 13s across the board.
>
> Project manager: I’d feel more comfortable if we could make ya know one maybe two of these, maybe [user story 1 and 5] an 8. What do y’all think of that?
>
> Developer #2: Sounds like it doesn’t matter what we think.
>
> Project manager: Thanks y’all.

Here, the project manager, despite clear disagreement from the designers and developers, pushes through the number that she wants for the user story regardless of the voiced concerns. The pursuit of a specific story point instead of the more appropriate story point for reasons outside the Scrum process suggests that story pointing is not valued by the team members.

I selected these examples because they are relatively easy to comprehend without much context. Every developer and designer as well as half the QA representatives used language akin to “whatever” when it came to Capacity and Story Pointing. This finding suggests that perhaps the mechanisms of Capacity, Burndown, and Story Pointing are simply not as important to them as they are to the product owners and project managers, who must show each team’s Capacity and Burndown to upper management. The language used by the team suggests that the numbers associated with Capacity and Story Pointing are relatively meaningless in their completion of the feature development: The task will take a certain number of hours to complete, regardless of the Capacity and Story Points assigned to the user story. Thus, it appears that although this team values certain aspects of Scrum, other aspects, particularly those related to Capacity and Story Pointing, are not valued and perhaps are viewed as boxes that must be checked off to pursue the “real” tasks of the sprint. These apparently different approaches to Scrum by the project manager and most of the other team members could potentially cause friction as the team tries to identify what aspects of Scrum are priorities for the team as a whole. Such friction and potential misalignment could possibly undermine the quality of the teamwork, which has been noted to be a major factor for team performance within agile frameworks [19], [20].

Discussion

The goal of exploratory research is to identify patterns that are not necessarily predicable to provide guidance for future research. In what follows, I will highlight areas of the results that lend themselves to more intensive investigations.

Planning Meetings are Critical for Scrum Although all meetings contained some Scrum language, much Scrum language, in terms of both overall quantity and number of categories covered, occurred in planning meetings. In each sprint, invocations of Scrum language decreased (both in overall count and in average per hour count) in meetings that followed the planning meeting. This finding may result from the fact that the planning meeting itself is tied to the specific structure of Scrum. Or it may be that as the sprint progresses, the group’s dedication to the hallmarks of Scrum wanes as the group actively focuses on achieving their definition of “done.” Because this is an exploratory study, these findings are not generalizable, but they do suggest that researchers should look closely at what happens in subsequent meetings to learn more about how meeting variation affects Scrum commitment.

Scrum Language Belongs to all Team Members Although designers and developers referenced Scrum hallmarks more often than their teammates, no one who participated in more than 2% of the sprint’s discussions (as measured by total clauses) was shut out of the Scrum process. Of the team members who actively engaged in the Scrum process, all job titles were able to invoke most Scrum categories; no Scrum category appeared to be explicitly off limits to certain job titles, nor were any categories exclusive to certain job titles. This exploratory study suggests that for this group, Scrum is the egalitarian project-management framework that it claims to be. Further research is needed to determine whether this egalitarianism exists in other groups as well as the degree to which group members perceive the group that they are part of to indeed be egalitarian.

Universality of Some Scrum Standard Elements Should be Re-Evaluated Although this group practices Scrum and values some aspects of Scrum (such as the User Story), this group appears to be merely going through the motions on other aspects of Scrum. Indeed, the language of the groups as seen in the discourse analysis seems to indicate that they view Capacity and Story Pointing as a farce. In the cases listed here, as well as in other cases, language such as “sure,” “whatever,” or even “it sounds like it doesn’t matter what we think” is used by the team members to indicate that they discount Capacity and Story Pointing because they themselves are being discounted in the process of determining Capacity and Story Pointing. In many ways, Capacity determination and Story Pointing do not entirely matter; the definition of “done” must be reached by the sprint’s end regardless of Capacity or Story Pointing. Therefore, the language used by the group seems to reinforce the notion that Capacity determination and Story Pointing are merely hoops that must be jumped through to complete the sprint. The potential harm here is—if the team members do not value a portion of the Scrum framework—at what point do they stop valuing the entire framework? More research is needed to expand this exploratory research to learn more about how team members feel about various parts of the Scrum process and how their feelings relate to the value of the framework.

Is This Scrum? This group professes to use Scrum. But in this exploratory study of their language, they seem to deviate from Scrum in some critical ways, such as discounting Capacity and Story Pointing, removing Grooming from the team setting, and avoiding Velocity altogether, along with other issues not directly reviewed here (such as longer sprint cycles and larger teams than recommended). Deviating from standard Scrum is not new, but at what point do these deviations from Scrum mean that the group is not performing Scrum at all [76]? Or perhaps more important, at what point do we consider that the deviations from standard Scrum mean that Scrum is not as helpful a framework as it has been purported to be? Indeed, at what point do the deviations suggest that a Kanban or a Scrumban framework (both of which do away with Capacity and Story Pointing estimates) might be more suitable [77]? More research is needed to learn more about the ways that groups modify Scrum to be functional and productive for their group. More research is needed to learn ways that groups value and discount alternative frameworks such as Kanban and Scrumban.

Limitations and Conclusions

This exploratory study, in an attempt to address a gap in the study of Scrum communication, empirically investigated the language used by members of Scrum teams within their sprint meetings. This study found Scrum language in all of the studied meetings, with Scrum language often clustered in the planning meetings early in the sprint. In addition, this study revealed that job title had little association with the type of Scrum language used, suggesting that this group used Scrum in egalitarian ways that fall in line with Scrum prescriptions. Finally, this study also showed that the group members valued User Stories while discounting Capacity determination and Story Pointing, facts that may suggest that this group is not conducting standard Scrum.

This study, although rigorous, is limited by the fact that only one firm was studied (though the three sprints were composed of unique teams). The data were meticulously collected in person over the course of 10 weeks, and they were collected only after all parties (including the author, the participants, management, and legal) agreed to the observations and recordings. To that end, about 20% of the meeting time and 67% of the total meetings (primarily short daily stand-ups) were removed from the dataset due to the agreements made with the engineering firm. This type of data collection is not easy, but it is desperately needed for the field of engineering communication to identify common behaviors and to establish best practices. An additional nuanced, recorded study that uses the codebook included here and replicates the study presented here would allow the field to understand which components of Scrum are mandatory for successful Scrum collaboration and which are optional. The collection of this knowledge would, in turn, allow formation of best practices. An additional, comparative studies using modifications of the codebook could explore organizations that use Scrum, Scrumban, and Kanban. Understanding the differences and similarities of these frameworks from a communicative perspective could help us learn how to better prepare engineering students to succeed in all three environments.

Finally, this study has shown that much is not known about communication within Scrum environments. Scrum has been the primary framework for technology teams for over a decade, yet the fields of professional and technical communication and engineering communication have largely been absent from the exploration of Scrum communication and collaboration. Indeed, only two articles about agile have been published in professional and technical communication journals, although only one has been published specifically on Scrum [22], [69], [78]. Meanwhile, the Society for Technical Communication professional magazine, *Intercom*, has published 13 articles on agile or Scrum in the same time period. These facts are not surprising given the division between technical communication practitioners and technical communication academics when it comes to research priorities [79]–[81].

This study is a first step to learn more about how team members use Scrum language to communicate, but more research utilizing empirical self-reported and non-self-reported data is needed to better understand agile and Scrum practices and their role in engineering workplaces. Continued research in the areas of agile and Scrum communication could better enable educators to prepare students for the realities of Scrum in the engineering workplace as well as provide best practices of Scrum communication and avenues of self-diagnosis to practicing engineering professionals.

Appendix I
Scrum Codebook

1. **Capacity:** Language that references the resources (typically time) available to perform work. Example: “Guys, I’m worried about my capacity here. I’m working on this sprint and on the other sprint, so I don’t want us to get caught up at the end.”
2. **Commitment:** Language that references a team member’s willingness to bind him or herself to a course of action. Example: “Yeah, I think that [user story] is reasonable. I can get it done by the demo.”

3. **Sprint Goal (defining "done"):** Language that embodies what the team will accomplish during the sprint. Language related to defining "done" for user stories is part of the sprint goal. Example: "Alright, so do we want to have this be QA testing-ready or done with QA testing?"
4. **Flow management:** Language related to the smooth steady movement or work and lack of idle of work. Example: "Let's think about the refinement. Is that gonna leave us empty until regression?"
5. **Forecasting:** Language related to predictions or estimation of events whose actual outcomes have not yet been observed. Example: "So the last sprint we managed to get through three of these. These look similar, so you know, I'm going to say they are the same."
6. **Grooming of Backlog:** Language related to the prioritization of backlog items. Example: None from sample.
7. **Sprint Execution:** Language related to the actual work of the Scrum team during the Sprint to accomplish the tasks need to complete each user story. Typically, this involves the negotiation of "who will do what." Example: "[Developer #1], you've done this before [on another project], so you take the lead here, and [Developer #2] can back you up."
8. **Story Pointing:** Language related to assessing a user story in story points (based on the Fibonacci sequence). Example: "There's a lot of unknowns, you know, here, so I'm going to give it an 8."
9. **User Stories:** Language related to the creation of user stories, which are items taken from the product backlog and then refined. Example: "I think that's kinda a lot to put in that one story."
10. **Velocity:** Language related to the rate at which work is completed. Example: None from sample.

Appendix II
Correspondence Analysis of Meeting Type and Scrum Language Category

Total inertia: 0.107

$p < 0.0001$

Although the inertia for the association between meeting type and Scrum language is somewhat low, the p-value indicates a significant association among the variables.

TABLE VI
Contributions of Scrum Language

Category	F1	F2
Capacity	0.667	0.025
Commitment	0.001	0.442
Defining Goal	0.106	0.029
Flow Management	0.009	0.054
Forecasting	0.093	0.083
Sprint Execution	0.002	0.043
Story Point	0.084	0.322
User Story	0.038	0.003

Table VI shows that Capacity, Commitment, and Story Pointing contribute significantly to the inertia of the analysis.

Table VII shows that planning, kickoff, and maintenance contribute significantly to the inertia of the analysis.

See Fig. 1 for a graphic representation of the possible associations within these data.

TABLE VII
Contributions of Meeting Type

Category	F1	F2
Planning	0.454	0.101
Kickoff	0.519	0.006
Maintenance	0.028	0.893
Retrospective	0.000	0.000

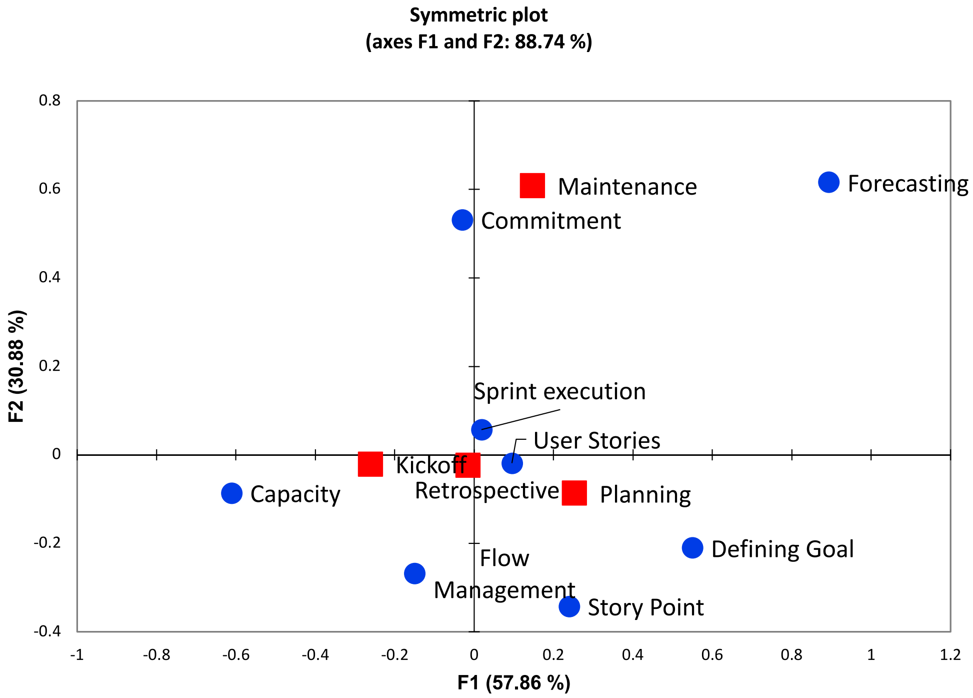

Fig. 1. Based on Tables VI and VII, this biplot suggests that Commitment and maintenance may have an association. In addition, Capacity and kickoff may have an association.

Appendix III
Correspondence Analysis of Job Title and Scrum Language Category

Total inertia: 0.304

$p < 0.0001$

Although the inertia for the association between meeting type and Scrum language is moderate, the *p*-value indicates a significant association among the variables.

TABLE VIII
Contributions of Scrum Language

Category	F1	F2
Capacity	0.007	0.754
Commitment	0.009	0.005
Defining Goal	0.002	0.037
Flow Management	0.863	0.044
Forecasting	0.007	0.018
Sprint Execution	0.031	0.001
Story Point	0.019	0.007
User Story	0.062	0.133

Table VIII shows that Capacity, Flow Management, and User Stories contribute significantly to the inertia of the analysis.

Table IX shows that QA, product owner, and project manager contribute significantly to the intertia of the analysis.

See Fig. 2 for a graphic representation of the possible associations within these data.

TABLE IX
Contributions of Job Title

Category	F1	F2
Architect	0.012	0.066
Product Owner	0.011	0.157
Developer	0.075	0.006
Designer	0.119	0.124
QA Representative	0.782	0.008
Project Manager	0.000	0.638

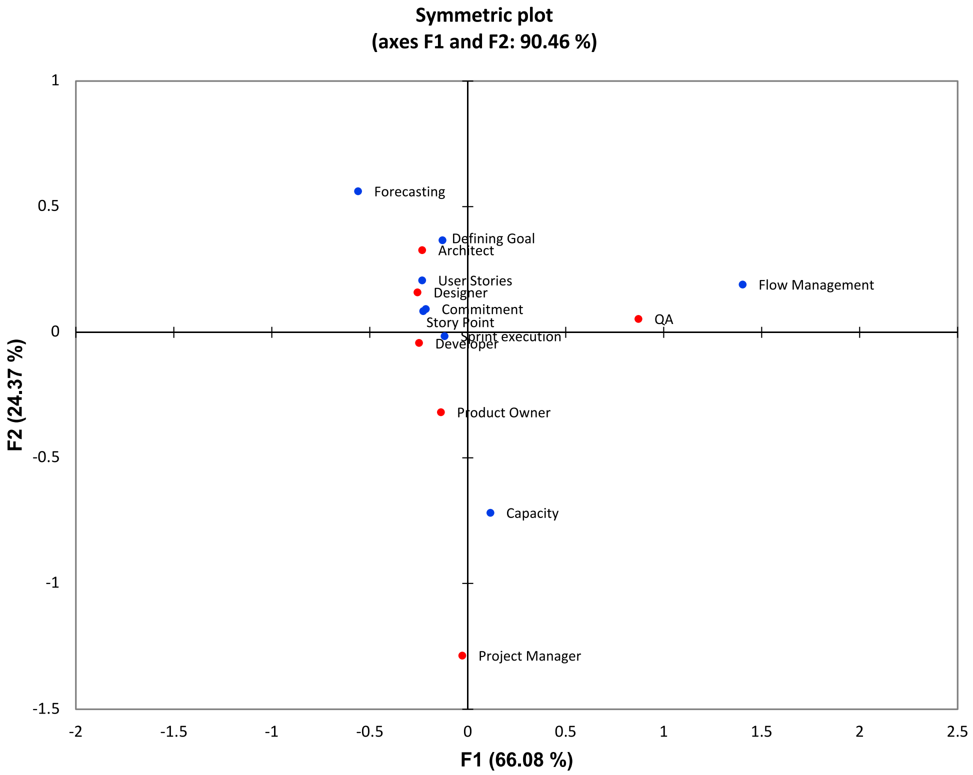

Fig. 2. Based on Tables VIII and IX, this biplot suggests that QA and Flow Management, product owner and Capacity, and project manager and Capacity may have associations.

References

[1] *Manifesto for Agile Software Development*, 2001. [Online]. Available: https://agilemanifesto.org/

[2] S. Nerur, R. Mahapatra, and G. Mangalaraj, “Challenges of migrating to agile methodologies,” *Commun. ACM*, vol. 48, no. 5, pp. 72–78, 2005.

[3] S. Sharma and N. Hasteer, “A comprehensive study on state of Scrum development,” in *Proc. Int. Conf. Comput. Commun. Autom.*, 2016, pp. 867–872.

[4] State of Scrum 2017-2018: Scaling and Agile Transformation, 2017. [Online]. Available: https://www.scrumalliance.org/

[5] E. C. Conforto, F. Salum, D. C. Amaral, S. L. da Silva, and L. F. M. de Almeida, “Can agile project management be adopted by industries other than software development?” *Project Manage. J.*, vol. 45, no. 3, pp. 21–34, 2014.

[6] B. Waldock, *Being Agile in Business: Discover Faster, Smarter, Leaner Ways to Work*. London, UK: Pearson, 2015.

[7] M. Korkala and P. Abrahamsson, “Communication in distributed agile development: A case study,” in *Proc. 33rd EUROMICRO Conf. Softw. Eng. Adv. Appl.*, 2007, pp. 203–210.

[8] K. Dikert, M. Paasivaara, and C. Lassenius, “Challenges and success factors for large-scale agile transformations: A systematic literature review,” *J. Syst. Softw.*, vol. 119, pp. 87–108, 2016.

[9] M. Hummel, “State-of-the-art: A systematic literature review on agile information systems development,” in *Proc. 47th Hawaii Int. Conf. Syst. Sci.*, 2014, pp. 4712–4721.

[10] I. Inayat, S. S. Salim, S. Marczak, M. Daneva, and S. Shamshirband, “A systematic literature review on agile requirements engineering practices and challenges,” *Comput. Human Behav.*, vol. 51, pp. 915–929, 2015.

[11] T. Dingsøyr and Y. Lindsjørn, “Team performance in agile development teams: Findings from 18 focus groups,” in *Proc. Int. Conf. Agile Softw. Develop.*, 2013, pp. 46–60.

[12] L. Lagerberg, T. Skude, P. Emanuelsson, K. Sandahl, and D. Stahl, “The impact of agile principles and practices on large-scale software development projects: A multiple-case study of two projects at ericsson,” in *Proc. ACM/IEEE Int. Symp. Empirical Softw. Eng. Meas.*, 2013, pp. 348–356.

[13] L. Pries-Heje and J. Pries-Heje, “Why Scrum works: A case study from an agile distributed project in Denmark and India,” in *Proc. Agile Conf.*, 2011, pp. 20–28.

[14] S. Ryan and R. V. O'Connor, "Acquiring and sharing tacit knowledge in software development teams: An empirical study," *Inf. Softw. Technol.*, vol. 55, no. 9, pp. 1614–1624, 2013.
[15] V. Stray, N. B. Moe, and G. R. Bergersen, "Are daily stand-up meetings valuable? A survey of developers in software teams," in *Proc. Int. Conf. Agile Softw. Develop.*, 2017, pp. 274–281.
[16] Univ. Phoenix. (2013). University of Phoenix survey reveals nearly seven-in-ten workers have been part of dysfunctional teams. [Online]. Available: https://www.phoenix.edu/news/releases/2013/01/university-of-phoenix-survey-reveals-nearly-seven-in-ten-workers-have-been-part-of-dysfunctional-teams.html
[17] L. Gren, R. Torkar, and R. Feldt, "Group maturity and agility, Are they connected?–A survey study," in *Proc. 41st Euromicro Conf. Softw. Eng. Adv. Appl.,* 2015, pp. 1–8.
[18] L. Gren, R. Torkar, and R. Feldt, "Group development and group maturity when building agile teams: A qualitative and quantitative investigation at eight large companies," *J. Syst. Softw.*, vol. 124, pp. 104–119, 2017.
[19] Y. Lindsjørn, G. R. Bergersen, T. Dingsøyr, and D. I. Sjøberg, "Teamwork quality and team performance: Exploring differences between small and large agile projects," in *Proc. Int. Conf. Agile Softw. Develop.*, 2018, pp. 267–274.
[20] Y. Lindsjørn, D. I. Sjøberg, T. Dingsøyr, G. R. Bergersen, and T. Dybå, "Teamwork quality and project success in software development: A survey of agile development teams," *J. Syst. Softw.*, vol. 122, pp. 274–286, 2016.
[21] C. Lam, "Correspondence analysis: A statistical technique ripe for technical and professional communication researchers," *IEEE Trans. Prof. Commun.*, vol. 59, no. 3, pp. 299–310, Sep. 2016.
[22] R. Pope-Ruark, "Introducing agile project management strategies in technical and professional communication courses," *J. Bus. Tech. Commun.*, vol. 29, no. 1, pp. 112–133, 2015.
[23] C. G. Cobb, *Making Sense of Agile Project Management: Balancing Control and Agility*. Hoboken, NJ, USA: Wiley, 2011.
[24] A. Moran, *Managing Agile: Strategy, Implementation, Organization, and People*. Cham, Switzerland: Springer, 2015.
[25] K. S. Rubin, *Essential Scrum: A Practical Guide to the Most Popular Agile Process*. Upper Saddle River, NJ, USA: Addison-Wesley, 2012.
[26] K. Schwaber and J. Sutherland. (2017). The scrum guide: The definitive guide to scrum: The rules of the game. [Online]. Available: https://www.scrum.org/resources/scrum-guide
[27] C. Sims and H. L. Johnson, *Scrum: A Breathtakingly Brief and Agile Introduction*. San Francisco, CA, USA: Dymaxicon, 2012.
[28] J. Noll, M. A. Razzak, and S. Beecham, "Motivation and autonomy in global software development: An empirical study," in *Proc. 21st Int. Conf. Eval. Assessment Softw. Eng.*, 2017, pp. 394–399.
[29] J. Sutherland, A. Viktorov, J. Blount, and N. Puntikov, "Distributed scrum: Agile project management with outsourced development teams," in *Proc. 40th Annu. Hawaii Int. Conf. Syst. Sci.*, 2007, pp. 274a.
[30] G. Wagenaar, S. Overbeek, and R. Helms, "Competencies outside agile teams' borders: The extended Scrum team," in *Proc. Softw. Eng. Workshop*, 2016, pp. 291–298.
[31] A. Khosravi, T. J. Gandomani, and H. Fahimian, "Introduction of scrum in an elite team: A case study," *J. Softw.*, vol. 12, no. 3, pp. 173–180, 2017.
[32] O. Salo and P. Abrahamsson, "Agile methods in European embedded software development organisations: A survey on the actual use and usefulness of extreme programming and Scrum," *IET Softw.*, vol. 2, no. 1, pp. 58–64, 2008.
[33] B. L. Romano and A. D. Da Silva, "Project management using the Scrum agile method: A case study within a small enterprise," in *Proc. 12th Int. Conf. Inf. Technol.-New Gen.*, 2015, pp. 774–776.
[34] D. West, T. Grant, M. Gerush, and D. D'silva, "Agile development: Mainstream adoption has changed agility," *Forrester Res.*, vol. 2, no. 1, p. 41, 2010.
[35] M. Hron and N. Obwegeser, "Scrum in practice: An overview of Scrum adaptations," in *Proc. 51st Hawaii Int. Conf. Syst. Sci.*, 2018, pp. 4496–4505.
[36] P. Kruchten, "Contextualizing agile software development," *J. Softw., Evol. Process*, vol. 25, no. 4, pp. 351–361, 2013.
[37] K. Vlaanderen, S. Jansen, S. Brinkkemper, and E. Jaspers, "The agile requirements refinery: Applying SCRUM principles to software product management," *Inf. Softw. Technol.*, vol. 53, no. 1, pp. 58–70, 2011.
[38] R. Vallon, B. J. da Silva Estácio, R. Prikladnicki, and T. Grechenig, "Systematic literature review on agile practices in global software development," *Inf. Softw. Technol.*, vol. 96, pp. 161–180, 2018.
[39] Y. I. Alzoubi, A. Q. Gill, and A. Al-Ani, "Empirical studies of geographically distributed agile development communication challenges: A systematic review," *Inf. Manage.*, vol. 53, no. 1, pp. 22–37, 2016.
[40] S. Ghobadi and L. Mathiassen, "Perceived barriers to effective knowledge sharing in agile software teams," *Inf. Syst. J.*, vol. 26, no. 2, pp. 95–125, 2016.
[41] M. Hummel, C. Rosenkranz, and R. Holten, "The role of communication in agile systems development," *Bus. Inf. Syst. Eng.*, vol. 5, no. 5, pp. 343–355, 2013.
[42] M. Pikkarainen, J. Haikara, O. Salo, P. Abrahamsson, and J. Still, "The impact of agile practices on communication in software development," *Empirical Softw. Eng.*, vol. 13, no. 3, pp. 303–337, 2008.
[43] K. Conboy, S. Coyle, X. Wang, and M. Pikkarainen, "People over process: Key people challenges in agile development," *IEEE Softw.*, vol. 28, no. 4, pp. 48–57, Jul./Aug. 2011.
[44] S. Dorairaj, J. Noble, and P. Malik, "Effective communication in distributed Agile software development teams," in *Proc. Int. Conf. Agile Softw. Develop.*, 2011, pp. 102–116.
[45] J. Downs, J. Hosking, and B. Plimmer, "Status communication in agile software teams: A case study," in *Proc. 5th Int. Conf. Softw. Eng. Adv.*, 2010, pp. 82–87.

[46] C. Treude and M.-A. Storey, "Work item tagging: Communicating concerns in collaborative software development," *IEEE Trans. Softw. Eng.*, vol. 38, no. 1, pp. 19–34, Jan./Feb. 2012.
[47] B. J. McNely, P. Gestwicki, A. Burke, and B. Gelms, "Articulating everyday actions: An activity theoretical approach to scrum," in *Proc. 30th ACM Int. Conf. Des. Commun.*, 2012, pp. 95–104.
[48] E. Friess, "Filling to Capacity": An exploratory study of project management language in agile scrum teams," *Tech. Commun.*, vol. 65, no. 2, pp. 169–180, 2018.
[49] T. Feltham-King and C. Macleod, "How content analysis may complement and extend the insights of discourse analysis: An example of research on constructions of abortion in South African newspapers 1978–2005," *Int. J. Qual. Meth.*, vol. 15, no. 1, pp. 1–9, 2016.
[50] B. Johnstone, *Discourse Analysis*. Hoboken, NJ, USA: Wiley, 2018.
[51] J. Holmes and M. Stubbe, *Power and Politeness in the Workplace: A Sociolinguistic Analysis of Talk at Work*. London, UK: Routledge, 2015.
[52] J. Baxter, *Speaking Out: The Female Voice in Public Contexts*. New York, NY, USA: Springer, 2016.
[53] J. Baxter, "Resolving a gender and language problem in women's leadership: Consultancy research in workplace discourse," *Discourse Commun.*, vol. 11, no. 2, pp. 141–159, 2017.
[54] E. Friess, "Politeness, time constraints, and collaboration in decision-making meetings: A case study," *Tech. Commun. Quart.*, vol. 20, no. 2, pp. 114–138, 2011.
[55] E. Friess, ""Bring the Newbie Into the Fold": Politeness strategies of newcomers and existing group members within workplace meetings," *Tech. Commun. Quart.*, vol. 22, no. 4, pp. 304–322, 2013.
[56] K. Halvorsen and S. Sarangi, "Team decision-making in workplace meetings: The interplay of activity roles and discourse roles," *J. Pragmatics*, vol. 76, pp. 1–14, 2015.
[57] B. Du-Babcock and A. C. Chan, "Negotiating consensus in simulated decision-making meetings without designated chairs: A study of participants' discourse roles," *Discourse Commun*, vol. 12, pp. 497–516, 2018.
[58] M. McGrath, "Safety education: culture, leadership and learning in the workplace: A critical discourse analysis," Ph.D. dissertation, Dept. Curriculum Instruction, Univ. Victoria, Victoria, BC, Canada, 2018.
[59] R. K. Boettger and L. A. Palmer, "Quantitative content analysis: Its use in technical communication," *IEEE Trans. Prof. Commun.*, vol. 53, no. 4, pp. 346–357, Dec. 2010.
[60] A. Liénard et al., "Transfer of communication skills to the workplace during clinical rounds: Impact of a program for residents," *PloS One*, vol. 5, no. 8, 2010, Art. no. e12426.
[61] R. D. Waters and J. M. Williams, "Squawking, tweeting, cooing, and hooting: Analyzing the communication patterns of government agencies on Twitter," *J. Public Affairs*, vol. 11, no. 4, pp. 353–363, 2011.
[62] T. McCorkindale, "Can you see the writing on my wall? A content analysis of the Fortune 50's Facebook social networking sites," *Public Rel. J.*, vol. 4, no. 3, pp. 1–13, 2010.
[63] C. Baehr and K. Alex-Brown, "Assessing the value of corporate blogs: A social capital perspective," *IEEE Trans. Prof. Commun.*, vol. 53, no. 4, pp. 358–369, Dec. 2010.
[64] C. Baehr, "Complexities in hybridization: Professional identities and relationships in technical communication," *Tech. Commun.*, vol. 62, no. 2, pp. 104–117, 2015.
[65] R. K. Boettger and C. Lam, "An overview of experimental and quasi-experimental research in technical communication journals (1992–2011)," *IEEE Trans. Prof. Commun.*, vol. 56, no. 4, pp. 272–293, Dec. 2013.
[66] M. A. Hannah and C. Lam, "Patterns of dissemination: Examining and documenting practitioner knowledge sharing practices on blogs," *Tech. Commun.*, vol. 63, no. 4, pp. 328–345, 2016.
[67] C. Lam, "Where did we come from and where are we going? Examining authorship characteristics in technical communication research," *IEEE Trans. Prof. Commun.*, vol. 57, no. 4, pp. 266–285, Dec. 2014.
[68] C. Lam and M. A. Hannah, "The social help desk: Examining how Twitter is used as a technical support tool," *Commun. Design Quart.*, vol. 4, no. 2, pp. 37–51, 2017.
[69] E. Friess, ""Filling to Capacity": An exploratory study of project management language in agile scrum teams," *Tech. Commun.*, vol. 65, no. 2, pp. 169–180, 2018.
[70] J. P. Gee, *How to do Discourse Analysis: A Toolkit*. New York, NY, USA: Routledge, 2010.
[71] J. Wolfe, "How technical communication textbooks fail engineering students," *Tech. Commun. Quart.*, vol. 18, no. 4, pp. 351–375, 2009.
[72] L. D. McNair, M. C. Paretti, and M. Davitt, "Towards a pedagogy of relational space and trust: Analyzing distributed collaboration using discourse and speech act analysis," *IEEE Trans. Prof. Commun.*, vol. 53, no. 3, pp. 233–248, Sep. 2010.
[73] E. Davidson and E. Vaast, "Tech talk: An investigation of blogging in technology innovation discourse," *IEEE Trans. Prof. Commun.*, vol. 52, no. 1, pp. 40–60, Mar. 2009.
[74] M. Ganis, S. Surdek, and E. Woodward, "Daily scrums in a distributed environment," in *Proc. Conf. Center Adv. Stud. Collaborative Res.*, 2009, pp. 349–349.
[75] A. DiPasquale. *A Beginner's Guide to the User Story Template: The Who, What, and Why of Requirements*, 2016. [Online]. Available: https://www.scrumalliance.org/community/articles/2016/october/a-beginner%E2%80%99s-guide-to-the-user-story-template?feed=articles
[76] P. Diebold, J.-P. Ostberg, S. Wagner, and U. Zendler, "What do practitioners vary in using scrum?" in *Proc. Int. Conf. Agile Softw. Develop.*, 2015, pp. 40–51.
[77] S. Pahuja, *What is Scrumban?* Corryton, TN, USA: Agile Alliance, 2015.
[78] D. Dayton and C. Barnum, "The impact of Agile on user-centered design," *Tech. Commun.*, vol. 56, no. 3, pp. 219–234, 2009.
[79] C. R. Lanier, "Toward understanding important workplace issues for technical communicators," *Tech. Commun.*, vol. 65, no. 1, pp. 66–84, 2018.

[80] K. St. Amant and L. Meloncon, “Addressing the incommensurable: A research-based perspective for considering issues of power and legitimacy in the field,” *J. Tech. Writing Commun.*, vol. 46, no. 3, pp. 267–283, 2016.
[81] R. K. Boettger and E. Friess, “Academics are from Mars, practitioners are from Venus: Analyzing content alignment within technical communication forums,” *Tech. Commun.*, vol. 63, no. 4, pp. 314–327, 2016.

Erin Friess received the Ph.D. degree in Rhetoric from Carnegie Mellon University, Pittsburgh, PA, USA. She is an Associate Professor of Technical Communication with the University of North Texas in Denton, Denton, TX, USA. Her research investigates language and decision-making in the workplace, the differing research needs of academics and practitioners, and curricular and pedagogical best practices for technical communication educators.