What Makes Team[s] Work? A Study of Team Characteristics in Software Engineering Projects

Kai Presler-Marshall
North Carolina State University
Raleigh, NC, United States
kpresle@ncsu.edu

Sarah Heckman
North Carolina State University
Raleigh, NC, United States
sarah_heckman@ncsu.edu

Kathryn T. Stolee
North Carolina State University
Raleigh, NC, United States
ktstolee@ncsu.edu

ABSTRACT

Teaming is a core component in practically all professional software engineering careers, and as such, is a key skill taught in many undergraduate Computer Science programs. However, not all teams manage to work together effectively, and in education, this can deprive some students of successful teaming experiences. In this work, we seek to gain insights into the characteristics of successful and unsuccessful undergraduate student teams in a software engineering course. We conduct semi-structured interviews with 18 students who have recently completed a team-based software engineering course to understand how they worked together, what challenges they faced, and how they tried to overcome these challenges. Our results show that common problems include communicating, setting and holding to deadlines, and effectively identifying tasks and their relative difficulty. Additionally, we find that self-reflection on what is working and not working or external motivators such as grades help some, but not all, teams overcome these challenges. Finally, we conclude with recommendations for educators on successful behaviours to steer teams towards, and recommendations for researchers on future work to better understand challenges that teams face.

CCS CONCEPTS

• **Applied computing** → **Collaborative learning**; • **Software and its engineering** → **Programming teams**.

KEYWORDS

software engineering teams, team challenges, team dysfunction, team characteristics, student teams

ACM Reference Format:
Kai Presler-Marshall, Sarah Heckman, and Kathryn T. Stolee. 2022. What Makes Team[s] Work? A Study of Team Characteristics in Software Engineering Projects. In *Proceedings of the 2022 ACM Conference on International Computing Education Research V.1 (ICER 2022), August 7–11, 2022, Lugano and Virtual Event, Switzerland.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3501385.3543980

ICER 2022, August 7–11, 2022, Lugano and Virtual Event, Switzerland

ACM ISBN 978-1-4503-9194-8/22/08…$15.00
https://doi.org/10.1145/3501385.3543980

1 INTRODUCTION

Professional software engineering is, almost without exception, a team-based activity, drawing together diverse teams to solve large problems. To help prepare students for professional practice, teaming is taught in many computer science programs and is also a skill assessed by ABET accreditors [3].[1]

We focus on student teams, and we observe that in software engineering, not all teams manage to work together effectively. Some students may have a sufficiently dysfunctional team experience that they are not able to learn key skills of how to manage and run a multi-member team. Students regularly complain about *freeriders*, or team members who fail to contribute equitably to the project, resulting in more work and stress for everyone else [9, 18]. Peer evaluations may be able to discourage freeriding [18, 49], but are not a general-purpose tool for addressing all teaming challenges. Indeed, while teams may be hampered by the explicit non-participation of one of their members, they may also be frustrated by a general sense of confusion and disorganisation that negatively impacts the entire team [35]. However, the precise details of the challenges that software engineering teams face have been under studied, which limits educators' ability to help teams overcome them.

In this work, we look beyond issues of non-participation, and seek to understand what makes teams work. We do so by identifying transient and persistent challenges faced by software engineering teams and attempts to overcome them. Additionally, we identify the characteristics of successful teams, which may serve as a model that educators can encourage students to adopt. We focus our efforts around the following research questions:

- **RQ1**: What team-related difficulties do students face on software engineering teams?
- **RQ2**: Why are some teams able to overcome the issues that they face, while others are unable to do so?
- **RQ3**: What support do students want from the course teaching staff for overcoming collaborative difficulties?
- **RQ4**: What are the characteristics of successful student software engineering teams?

We answered these research questions by conducting one-on-one interviews with students who have recently completed a team-based undergraduate software engineering course. These questions aimed to understand their experiences, successes, challenges, and how they tried to overcome these challenges.

Our results show that while some teams manage to work together successfully throughout the project, communication issues and poor time management caused challenges that other teams

[1] ABET is an organisation that accredits undergraduate and master's programs in applied sciences, engineering, and computing to ensure their rigor. This process is voluntary, but widespread [1].

struggled to overcome. Additionally, we find that self-reflection, a critical component of self-regulated learning [36], helps some teams overcome challenges, but is not capable of motivating recalcitrant teammates.

Our contributions are as follows:

- A discussion of the characteristics of software engineering student teams that worked well together, and teams that struggled to work effectively.
- A discussion of the steps students attempted to overcome challenges.
- A discussion of how teaching staff can help struggling teams.
- Data suggesting that teams may face more collaborative challenges than was previously understood, calling for researchers to better understand issues teams face.

To the best of our knowledge, this is the first paper to study the characteristics of software engineering student teams from the inside, discussing with students to understand their experiences and challenges. All prior work we are aware of looks at external factors of team success, such as team grades [24, 31, 42], peer evaluations [24, 42], whether projects could be deployed [31], or similar metrics [14]. By contrast, our one-on-one interviews with students provide novel insights into what teams do, and, from their perspective, what challenges were faced. This gives us a far richer perspective on how teams function, and lets us demonstrate empirically that software engineering teams function according to educational theory and many educators' intuitions.

2 RELATED WORK

Practically all professional software engineers work in teams, bringing together a diverse set of skills to enable engineering the software systems of the modern world [29, 44, 46]. While software teams have long been geographically distributed [29], the COVID-19 situation has accelerated this trend, with more developers opting for fully remote work [32]. Prior work suggests that remote work accentuates the challenges developers face working together [6, 34], making it crucial that developers enter the workforce with teaming experience. As a result, most software engineering classes include some form of team-based learning, from pair-programming efforts to longer-running, many-member teams [23, 24, 27, 37, 48, 53].

However, despite that teaming is a key learning outcome [3], some students have a dysfunctional team experience that imperils their ability to learn teaming skills. Prior work has demonstrated that up to 40% of teams in project-based courses are characterised by "internal strife" and fail to work together effectively [50], often caused by a lack of communication or effective project management [24, 35]. Iacob and Faily [24] report that dysfunction is a risk in student software engineering teams, where low engagement or poor communication can hamper individual and team outcomes. They identify that there may be team dysfunction, but do not study its causes. Marques [31] proposes having a "monitor" conduct weekly meetings with teams of software engineering students, observing them work and providing feedback on the overall team function and contributions of each member. They report mentored teams produced higher-quality software, and performed substantially better on their final project, but provide little elaboration on the details of the challenges that students faced in either case. Prior work by Presler-Marshall et al. [42] investigates the use of a *team collaboration reflection survey (TCRS)* for identifying struggling teams. They report that the TCRS can identify struggling teams, but provide little insight into the types of challenges teams faced. Maguire et al. [30] discuss how to train the mentors required by several of these approaches.

Computer science education researchers have considered student teams primarily by focusing on externally-visible characteristics, such as grades, peer evaluations, or version control system (VCS) commit history. Iacob and Faily [24] and Marques [31] focus on end-of-project grades as a measure of team success. Meanwhile, Gitinabard et al. [17] focus on VCS data to identify how small teams collaborated. Most similar to our work, Dzvonyar et al. [14] explore team forming and success in software engineering. They discuss considerations when forming teams for a project-based course, and survey teams at the end of a software engineering course, asking questions about team synergy and any challenges the team faced. They report that team synergy was generally high, but two teams struggled with low motivation and performed poorly. However, they do not discuss how teams themselves operated and how this may have impacted any challenges faced. Finally, Berglund [7] conducts an in-depth study of an upper-level networking course. They consider how teams distribute leadership responsibilities and whether they function as a cohesive whole. However, their work considers only two aspects of how teams function, and the course context is very different from software engineering courses, where teaming is a primary learning outcome.

Prior work has considered the characteristics of student teams more broadly in engineering education. Borrego et al. [9] present a comprehensive literature review of teaming in engineering education, and consider the learning outcomes and "negative behaviours" commonly associated with them. They show that teaming is widespread in engineering education, particularly in introductory courses and senior-level capstone courses. They report that social loafing, or *freeriding*, is the primary form of dysfunction faced by teams. They counter that freeriding can be reduced by having projects that are sufficiently complex that each student has a unique role [25, 26], and that academically unbalanced teams (those featuring both high and low performing students) are at the greatest risk of freeriding [40]. Beyond the issue of freeriding, they also consider how to promote teaming environments that lead to positive educational outcomes. They report that interpersonal conflicts between members of the team leads to "*reduc[ed] productivity and satisfaction*" [12] but that disagreements over how to solve tasks can help students consider a broader range of possible solutions and thus improve outcomes [11]. This finding has been echoed by other work [21]. More recently, Walsh et al. [52] consider impacts on team dynamics in engineering education during the COVID-19 situation. They report that teams experienced many of the same challenges that we observed, including issues with time management and timeliness, communication difficulties, forming effective relationships, and burnout and a lack of motivation. Finally, Pazos and Magpili [39] propose interviewing engineering students to understand how technology can support better teaming.

Team-based learning (TBL) is a learner-centred pedagogy, where students direct their own learning under the guidance of an instructor who serves as an "expert facilitator" [22]. TBL is grounded in

constructivist theory, which argues that students cannot merely absorb information passively, but must actively discover it. This theory says that learning is done through dialogue rather than a dissemination of facts. Prior work has demonstrated that this is typically a more effective pedagogy and results in better learning [5, 22]. For these reasons, the software engineering course we study in this work uses TBL extensively. Despite the benefits, researchers have recognised that TBL is not an educational panacea. Successful teamwork depends upon regular communication, particularly when members work asynchronously [16]. Additionally, teams must be capable of conflict resolution, which requires both identifying and resolving challenges [38, 43]. In order to help students navigate these challenges, many educators include team forming activities [20, 41, 45], self-and-peer assessment [13, 49], or discussions of teaming theory (such as Tuckman's model of teaming [8], discussed in Raferty [43] and Hansen [19]). In this work, we use Tuckman's model for characterising where teams faced challenges. Tuckman argued that teams progress through four stages: *forming*, as the members of the team meet each other but largely act independently, *storming*, as conflicts and disagreements arise between members, *norming*, where conflicts are resolved and the team starts to function cohesively, and finally *performing*, where members are engaged and the team works together effectively. Tuckman noted that some teams may skip the *storming* stage entirely, while others may face intense "storms" they never overcome. Later, Tuckman and Jensen added a fifth stage, *adjourning*, where the group disbands upon the completion of their tasks [51].

3 BACKGROUND

At NC State University, a large, research-focused university in the United States, undergraduate Computer Science students are required to take a Software Engineering course, typically during their third year. The course covers fundamentals in software engineering, including how to design, implement, and test a medium-sized object-oriented system; how to write requirements; and how to appropriately break down a project into manageable components, all in the context of two multi-week team-based projects. The first project, an *onboarding project* (OBP), introduces the process expectations and technology stack. The second, a larger *team project* (TP), tasks students with a more comprehensive project with a larger team. The OBP is completed in teams of three or four students; the TP in teams of five or six. Prior to team formation, the teaching staff distributes a Google Form to students, inviting them to fill out who they *would* or *would not* like to work with. The teaching staff makes a best-effort to build teams that satisfy these preferences. Avoidance requests are always satisfied, and students will usually get at least one, if not more, of the teammates that they request to work with. Projects are broken into iterations, each typically lasting one week, that cover different learning objectives: requirements and planning, design, testing, and implementation. Over the course of the projects, students are evaluated in five categories: technical deliverables (including both code and technical documents); technical processes; project management; team collaboration; and peer review. At the end of the project, the course teaching staff reviews peer evaluations and contributions to determine whether individual adjustments are needed (positive or negative). We focus here on the *team project*; a larger team provides more interesting dynamics, and a more recent project is easier for students to remember.

The *team project* features several aspects designed to promote positive teaming outcomes. The first iteration features a team forming activity based on prior work [20, 41, 45]. The teaming activity is facilitated by the course teaching staff, and encourages students to reflect on collaborative experiences in prior classes to identify the characteristics of successful teams. The activity features questions to facilitate discussion on what each student wants to learn, how they want to run their teams (including how they want to meet and communicate out of lab, how they want to resolve conflicts, and team roles), and timeliness expectations. Students are encouraged to establish individual feature-based roles to focus on specific tasks, as well as an overall team lead role. They are also encouraged to split their team approximately in half into two subteams to work in parallel, and establish leadership on each subteam. Teams are required to establish a real-time communication approach (i.e. something to supplement email) and produce a written document reflecting their discussion and the rules they have established, but are otherwise free to establish rules as they see fit. All members of the team are expected to sign the rules.

The Software Engineering course typically has between 120 and 200 students a semester, led by one PhD professor and three to five teaching assistants (TAs). To support team-based learning, the course features lab sessions each Thursday led by the TAs. Labs have 20-25 students each and provide time for teams to review work from the previous week and plan tasks for the next week. To ensure teams are prepared for the Thursday labs, weekly tasks are due Wednesday nights. Lab sections are run synchronously and while they have typically been run by a single TA, starting in Fall 2021, they are run by pairs of TAs. Due to the COVID-19 situation, labs were run online from Spring 2020 to Spring 2021, but have returned to in-person in Fall 2021. However, in keeping with safety protocols, students with a COVID exposure were asked to join their team by Zoom instead of attending physically.

In prior work [42], we introduced a *team collaboration reflection survey (TCRS)* to the class projects. Administered weekly through Qualtrics, the TCRS is mandatory and asks students to reflect on their contributions and how their team collaborated. The TCRS is capable of identifying a large majority of teams that ultimately perform poorly (with the team as a whole receiving a poor project grade, or one or more members receiving a low peer evaluation grade). In this paper, the TCRS is used as a tool as we seek insights into the challenges that teams face.

All authors of this paper are regular members of the teaching staff for the studied course. The first author is head TA for the course; the third author was the instructor of record in the Fall 2021 semester we studied. The second author is a regular instructor for the course, but was not a member of the teaching staff in Fall 2021.

4 METHODOLOGY

In this section, we discuss how we improved the TCRS, classified teams based on their project experience, recruited potential participants, and conducted and analysed interviews.

4.1 TCRS Improvements

The TCRS features an open-ended question asking students to reflect on their project experiences over the past week. As suggested in prior work [42], for the Fall 2021 semester we introduced natural language processing using VADER [4] as an additional way to identify struggling teams from this response. VADER is a sentiment analysis tool, and produces a 3-tuple of (positive, negative, and neutral scores) representing the sentiments detected in a piece of text. However, in our context, rather than individual sentiment scores we need to answer "*Is this TCRS response describing a problem the team is facing?*" Thus, we need a binary classifier that combines together the individual sentiment scores to determine if a comment is predominantly negative (that is, describing a problem, which the teaching staff would like to know about) or not (describing instead that the team is working well, or effectively saying nothing at all).

To construct and evaluate a classifier, we built a labeled dataset. We read through the 579 open-ended responses on the TCRS from Spring 2021, and manually labeled each one as expressing a predominantly positive sentiment, a predominantly negative sentiment, or no sentiment. This gave us a dataset of 437 positive comments, 93 negative comments, and 49 neutral comments. We then ran VADER on each comment, and built a binary classifier from the positive, negative, and neutral scores it produced. As in prior work, we prefer a high recall (a large majority of negative comments correctly labeled) over high precision, so we tuned our classifier until the recall exceeded 90%, which gave a precision of approximately 55%.[2] While the precision is relatively low, the classifier successfully narrows down approximately 110 comments submitted each week to no more than ten negative comments for the teaching staff to review, and a quick manual inspection lets us discard comments that were incorrectly labeled as negative.

4.2 Team Classification

In prior work [42], we used a two-part grades-based oracle for identifying struggling teams: low project grades and peer evaluation grades. In both cases, a threshold of 1.5 standard deviations below the class average was used; a team was flagged through the oracle if the overall team grade or any member's peer evaluation grade was below the threshold.

We adopted this same model, with the improvements discussed in Section 4.1, and then compared teams flagged through the oracle against teams flagged through the TCRS. The TCRS is due each week as part of project tasks, and was analysed to identify struggling teams. Therefore, while the oracle represents team struggle at the end of the project, the TCRS represents a metered lens into struggle throughout. We cross-referenced teams flagged through the oracle to teams flagged through the TCRS, splitting the 24 teams in the course into four distinct groups:

- **Group 1**, *eight teams*: Teams that were not flagged through the grades-based oracle, and were flagged ≤ 1 time through the TCRS. These are teams that ultimately did well, and any issues faced appeared to be transient.
- **Group 2**, *seven teams*: Teams that were flagged through the grades-based oracle, and were flagged ≥ 2 times through the TCRS. These are teams that ultimately received poor grades, and issues were seen consistently through the TCRS.
- **Group 3**, *three teams*: Teams that were flagged through the grades-based oracle, and were flagged ≤ 1 time through the TCRS. These are teams that ultimately received poor grades, but issues showed up at most briefly through the TCRS.
- **Group 4**, *six teams*: Teams that were not flagged through the grades-based oracle, but were flagged ≥ 2 times through the TCRS. These are teams that appeared to struggle during the project itself, but the issues did not manifest themselves in low grades at the end.

[2] These figures represent training error, rather than test error; the skew of our dataset towards positive comments means there is an insufficient number of negative comments for a typical training/test split.

Table 1: The number of students contacted, and who participated in interviews, from each of the groups studied. Also shown is the number of teams represented in our interviews.

	Students Contacted	Students Interviewed	Teams Represented
Group 1	17	8	2
Group 2	15	3	2
Group 3	16	4	3
Group 4	10	3	2
Total	58	18	9

4.3 Recruitment Process

From each group, we randomly selected three teams for analysis, with the exception of Group 4, where we made an administrative error and only selected two teams. We then sent individual recruitment emails to each member of the selected teams, inviting students to discuss their project experiences. Recruitment emails were sent in January 2022, and interviews were conducted in late January, approximately two months after the conclusion of the project. We did not ask students to hide their participation in the study from their teammates.

This study received IRB approval. Participation was voluntary, and students were not compensated for participating. Willing students signed up for an interview timeslot from a provided calendar; we then sent them Zoom information and a consent form. Students were asked to sign and return the consent form before their interview slot. Every student who signed up participated in an interview. As shown in Table 1, 18 of the 58 students we invited participated, for a response rate of 31%. The 18 interviews represent nine of the eleven teams we contacted.

4.4 Interview Process

To ensure that all students were asked the same core set of questions, we prepared a semi-structured interview outline, shown in Figure 1. Students were free to direct the conversation, so questions were not always asked in the same order or with exactly the same wording, but we asked the same core questions in each interview. As discussed in Section 4.5, after the first three interviews, we added questions on teams' communication and leadership approaches (Q5 and Q6). All interviews were conducted by the first author.

Collaboration:

Q1: Could you tell me about your collaboration experience on the TP?

Q2: Could you tell me one thing about your collaboration experience that you think worked out well on the TP and one thing that could have been improved upon?

Q3: Is there anything that you would definitely do again, and anything that you would definitely change?

Team Formation:

Q4: At the start of the project we took a day for team forming, setting goals and rules with the team. Did you find this helpful at establishing a common plan for the project? If not, is there anything that you think could have been done to improve things? If it worked, what did you find most helpful?

Communication and Leadership:

Q5: Could you tell me about how your team communicated outside of lab, and how, if at all, you met up together?

Q6: Could you describe your team's leadership approach?

If the team was flagged through the TCRS and grades-based oracle:

Q7: Did you find the TCRS & followup from TAs in lab to be helpful? Did it help you and your teammates do a better job splitting up tasks, communicating among yourselves, or ensuring that work got done and according to your standards?

Q8: Is there anything that you would have liked us to do differently based on the issues we observed?

Q9: Would you have liked us to bring up the issues we observed more directly, and been more explicit about telling people on the team what to do and requiring followups? Or would you prefer having the teaching staff bring up issues but leaving it more hands-off on how to resolve them?

Q10: Would you have liked us to mention we saw issues from the TCRS, or in a different way, such as through Github contributions?

Q11: Is there anything that you wish had been done differently on your team in regards to how we responded?

Q12: More broadly, is there anything you wish *you* had done differently?

If the team was flagged through the TCRS, but not the grades-based oracle:

Q13: We saw from some of the TCRS responses (remind student of context) that there were some issues your team faced, but your team did well in the end. If you remember, could you talk a bit more about what was going on? What do you think helped your team overcome issues like these?

Q14: Were there any other issues that you or your team experienced that we didn't see here?

Q15: Would you have liked any help from the teaching staff to help overcome them, and if so, what sort of help?

If the team was not flagged by the TCRS or grades-based oracle:

Q16: Reading through the TCRS responses you (and your team) submitted each week, we didn't notice any issues that needed to be addressed. However, we realise that these don't necessarily capture the entire story. Thinking back over the course of the project, were there any issues that you encountered communicating or collaborating with your team?

Q16a: ***(if yes)*** Would you have wanted help from the teaching staff? What questions could we have asked that would have helped uncover them?

Q16b: ***(if yes)*** What sort of followup would you want us to take?

Reflections on self-reflection:

Q17: Did you find the TCRS helpful for self-reflection during the project?

Q17a: ***(if yes)*** What questions did you think were particularly helpful? Are there any changes we could have made to help make them better?

Q17b: ***(if no)*** Can you think about what we could have done to make this more useful to you?

Figure 1: Interview Outline

4.5 Analysis

To analyse interviews, we followed a grounded theory approach [10], transcribing and performing preliminary analysis concurrently with interviewing. To reduce bias, we replaced all student names and pronouns with gender-neutral pseudonyms. As suggested by Saldaña [47], we began with structural coding [33], identifying a preliminary set of codes and categories from the interview script and common themes. During this process, we added Q5 and Q6 to the interview script, shown in Figure 1. To supplement our initial set of codes and categories, we followed an open coding approach [28], letting insights from each interview guide our analysis, and revisiting prior interviews to see if and how each newly-discovered topic was discussed.

On five of nine teams represented in the interviews, we interviewed two or more members. Consequently, we compared interviews within a team for consistency and contradiction. As expected, we found students emphasised and discussed different aspects of their experiences, but we found no contradictions between different members of the same team. We verified claims where possible, checking Git logs to confirm comments on the timeliness and distribution of labour. In no cases did we find information substantially different than what students told us.

As we compared student responses, we found that the nine teams could be arranged into four distinct categories, depending on whether they faced collaborative challenges and how effectively they overcame them. We use Tuckman's model of teaming [8], as discussed in Section 2, to characterise team experiences:

- **Category I: Ineffective Collaboration**. Teams that faced a substantial collaborative issue they were unable to overcome. These teams never successfully made it past the *storming* stage of Tuckman's model. We name these three teams *Alpha, Bravo* and *Charlie*.
- **Category II: Partially Ineffective Collaboration**. Teams that faced a substantial collaborative issue which they were able to partially, but not fully, overcome. These teams struggled to move through the *storming* stage of Tuckman's model, and while they made more progress than the teams in *Category I*, they faced conflict until the end of the project. We name these two teams *Delta* and *Echo*.
- **Category III: Effective Collaboration with Issues**. Teams that faced a collaborative issue which they were able to fully overcome. These teams lingered in the *forming* stage of Tuckman's model. We name these two teams *Foxtrot* and *Golf*. Foxtrot lingered in the *forming* stage for approximately two weeks, and *Golf* for a bit over one week.
- **Category IV: Effective Collaboration**. These teams never faced acknowledged collaborative issues. They progressed through the forming, norming, and performing stages without difficulties, with little sign that they faced a *storming* stage at all. We name these two teams *Hotel* and *India*.

To answer **RQ1**, we focused on the challenges faced by **Categories I-III**. To answer **RQ2**, we focused on **Categories I-III**, looking at how they tried to overcome these challenges and comparing the fully successful attempts in **Category III** with partially successful attempts in **Category II** and unsuccessful attempts in **Category I**. To answer **RQ3**, we focus on students from all categories, soliciting feedback on steps that were or could be taken by the teaching staff to help address similar issues. Finally, to answer **RQ4**, we focus on what students from all categories described as the successful attributes of their teams, paying particular attention to teams from **Category III** and **Category IV**.

5 RESULTS

Here, we present the results on challenges teams face (RQ1), teams' ability to overcome them (RQ2), support for teams from the teaching staff (RQ3) and the characteristics of successful teams (RQ4).

5.1 RQ1: Challenges Faced

First, we sought to identify the challenges that impeded effective teamwork. We focus here primarily on answers to Q1-3 and Q5-15 from our interview script.

5.1.1 Communication Difficulties. We find that poor communication underlies most team issues. Two teams from Category I and all four teams from Categories II and III reported that, at least for part of the project, their teams did not communicate effectively, leaving them unclear about their current progress. Adrian, on Team Alpha, described the communication difficulties on their team succinctly: "*I would text things in the chat, and there would be radio silence*". For some teams, communicating effectively was even a challenge within lab. On Team Bravo, Finnegan said that "*the first few labs it was like almost like silent...and I don't really know what's going on, [or] what are they doing*". Meanwhile, Spencer, from Team Delta, reported that their teammates were "*a little ashamed that they hadn't started*" and would not respond until the very last moment, once they had actually started. This lack of communication meant that "*a lot of times we didn't know if [individual tasks] were done or were going to be done before lab*". As we discuss in Section 5.2, Team Bravo managed to partially overcome these issues, but communication deficiencies impeded many teams.

We also find that the communication platform students used and how they used it impacted their communication efforts. Of the nine teams, eight chose Discord, a popular channel-based, text-and voice chat program [2]. Only one team, Alpha, used anything else: SMS text messages, which Adrian described as an "*awful*" solution that "*discouraged*" necessary conversations. However, despite that all remaining teams used Discord, some used it more successfully than others. Page, from Team Foxtrot, reported that their team had a "*big group*" channel for the entire team. However, rather than using this, or something else that would be visible to the team, Page and the other member on their subteam communicated via direct messages (DMs). They described DMs as the "*obvious*" approach, but later reported that "*[I] no idea what was going on on the second subteam because I didn't talk to any of them*". In Section 5.4 we discuss how more successful teams used Discord to communicate.

Finally, we observe that language barriers can contribute to communication challenges. We interviewed two teams with an ESL[3] student; one of them cited this as a major challenge. In Section 5.2 we discuss how Team Bravo partially overcame this issue. Meanwhile, Parker, from Team Golf, reported that their team experienced a minor language barrier with one student, but that "*we adjusted to it and we did okay*" and it did not impede their work.

5.1.2 Time Management. The second most prevalent issue that teams faced is one of time management and accountability. Two of the three teams from Category I and both teams from Category II reported that they had issues getting work done on time, with a tendency to wait until the last moment. Spencer, on Team Delta, explained that "*we all kind of were pushing what we need to do back*". They reported that the team often would not *start* tasks until after the Wednesday deadline, completing tasks between the deadline and the start of their lab session the next day: "*we fudged that a little bit...not really doing [our work] until like the next day, Thursday...and that didn't work*". When technical issues arose, it put their team in an untenable position, since "*you know how it is, there's not really time to figure it out*". Emery, from Team Charlie, also reported issues with procrastination: "*nobody would start it early...everything was done at the last possible minute*". Ultimately, Team Delta and Team Alpha managed to get their projects done, although Adrian reported that it took "*an all-nighter that night it was due*". However, on Team Charlie, Emery reported that "*we didn't have time to fix [several broken pieces]*".

Exactly *why* teams struggled to get work done on time varied from team to team. On Team Delta, Spencer reported that "*the team doesn't keep you accountable*" and at one point on the project, "*other than Kennedy, nobody else really cared*". Spencer admitted that their team was "*more concerned about the grade than actually learning*". More pressure from the TAs may have been necessary. Emery was more optimistic about their team's process, and said that if their approach of "*just make sure you do it before the deadline*"

[3] English as a second language

was replaced with "*days with [the] team to work on things*" they may have worked more effectively.

5.1.3 Task Planning. Our results also suggest that teams struggle with task planning and organisation, which tended to exacerbate other problems. Spencer, from Team Delta, reported that "*we didn't really have a good execution plan*". They wished that their team had been "*more specific about the [meeting] agenda*". Meanwhile, Adrian, on Team Alpha, reported that while their team managed to figure out *what* needed to happen, "*those [time] estimates that we had were just completely wrong*" and resulted in an "*unequal distribution*" of labour. Additionally, Adrian reported that "*we didn't really think about* " task dependencies, which meant that they were often blocked because "*some things were dependent on other things and you didn't necessarily know that*". Incorrect time estimates caused troubles for Hayden and Team Echo as well. In their case, during the final iteration Nuru was assigned a "*process [that] took way less time than we thought*". Nuru then went to work at their day job, and was unable to help their teammates, who were "*helping with the process that took a lot longer*". While Hayden said that "*overall I don't blame [Nuru]*" because "*[they] did all work that we assigned*", this created difficulties for everyone else.

5.1.4 Other Issues. We find some evidence that team leadership strategies also impact team success. Adrian, on Team Alpha, reported that their teammate Casey established themself as team lead. Adrian reported that Casey largely refused to delegate tasks, and would go and "*[change] my code for me...and edit the things I've done*". Adrian described Casey's leadership style as "*abrasive*". Other teams had varying degrees of success with their leadership approaches. Hayden, from Team Echo, acknowledged that the decentralised leadership approach of their team "*usually doesn't work out*" but said "*it worked out perfectly*". However, Team Echo only managed to pull together a working project through a crunch at the end, and we question if things worked as smoothly as Hayden described. Meanwhile, on Team Golf, Parker reported their initial experience was very chaotic. They described their first two team meetings as "*torture*", saying that "*some people were throwing out ideas, some people were just kind of silent*". Parker reported that their experience after establishing a team leader was much smoother.

Three teams also faced issues with mental health challenges. On Team Delta, Spencer reported that one team member "*sorta just disappeared*" and stopped attending class or participating in the team chat; attempts from the team and the teaching staff to contact them were unsuccessful. Emery, on Team Charlie, faced a similar issue with their teammate Alex, who the teaching staff was also unable to contact. On Team Bravo, Landon said they were struggling from "*burnout*". They described their performance as "*hot and cold*" and said that sometimes they were engaged with the team, and sometimes they "*[were] not able to perform*". We asked Landon about being referred to the counseling center for mental health support; they "*probably would not have taken advantage of it as I should have*" and "*would have denied*" help. We suspect that these issues were exacerbated by the COVID-19 situation.

On Team Echo, Hayden mentioned an issue unique to their team: two members of the team had jobs outside of school. Hayden reported that in the first couple weeks of the project, they "*[were] actually a bit hesitant*" because the team members with jobs "*never responded...until very late at night*". As discussed in Section 5.2, Team Echo was able to partially overcome this issue, but combined with task planning issues (see Section 5.1.3) it still presented a challenge.

> ***RQ1:*** *We find that teams struggle with communication, setting and keeping to deadlines, and task planning. Some teams also faced challenges with leadership and burnout.*

5.2 RQ2: Overcoming Challenges

In this section, we discuss the two teams that completely overcame their challenges, the two teams that partially overcame them, and the three teams that faced challenges they were unable to overcome. We focus here primarily on answers to Q1-3, Q12-13, and Q17 from our interview script.

5.2.1 Successful Attempts. We find that two teams completely overcame their challenges by addressing deficiencies in communication and leadership. On Team Foxtrot, Page reported that in the first half of the project, they had "*no idea*" what the other subteam was doing, because they "*didn't talk to them*". The turning point was when the team "*completely missed*" one weekly task. Page explained the grade "*really hit us*" and the team realised "*we really need to start [talking]*" to stay on top of tasks and "*make sure [a bad grade] is not a trend*". They remarked that while this oversight "*could have turned into a blame game very easily*" their team "*handled it very gracefully*". Everyone realised that "*no one was told to do this*" and consequently, responsibility fell on the team. Ultimately, Team Foxtrot overcame this issue through pair programming, which facilitated communication between the subteams. Additionally, the team "*talked actually quite a bit with*" a TA to make sure they "*were 100% prepared*". Team Foxtrot faced no further issues and worked together effectively henceforth.

Team Golf managed to overcome their issues just as effectively. Parker described the first two team meetings as "*torture*" as the team meandered aimlessly. Parker explained that when they sat down to complete the TCRS, they "*put [their] thoughts about that week together*" and realised the team needed a plan. Parker credits self-reflection for identifying the problem, and thought without it they "*could have just been really disappointed and demotivated*". To address this, their team instituted a "*rotating team leader*" who could steer discussions. Parker said that as their team figured out the project, they used the team leader less, but that it was still "*nice having it there...as a safety button*" if needed.

5.2.2 Partially Successful Attempts. Team Delta originally faced severe issues on several fronts: poor communication, last-minute work, and next to no collaboration. Spencer reported that they tried, largely unsuccessfully, to organise the team, "*volunteering to be in the library*" and asking for progress updates. They described the inflection point for their team as the week that they got sick. Spencer said "*when I wasn't there they had to step up*" and this "*kick-started*" the other members of the team into participating. Additionally, they remarked that "*we realised how there's a lot of work left to do*" and not much time. While Team Delta never heard from the missing member, Spencer said that "*once we started working*" their team started making some progress, although they conceded the team remained

more focused on "*[the] grade than actually learning*". Ultimately, their desire for a better grade encouraged better collaboration.

Team Echo partially overcame their issues by working around the schedules of the members with jobs. Hayden said that after a couple of meetings in the library when "*these two people were just MIA*" their team scheduled meetings for "*weekends and evenings*" to accommodate everyone. Ultimately, while Team Echo struggled with time estimates until the end, they made progress working together.

5.2.3 Unsuccessful Attempts. While four teams managed to overcome many of the challenges they faced, three more did not. Team Alpha was ultimately unable to overcome Casey's "*abrasive*" leadership. Adrian said while everyone privately agreed that Casey was behaving unreasonably, they were "*scared or hesitant*" to call out the problem, only to "*be deflected, and be gas lit*" by the member causing it. Adrian self-described as "*not a very confrontational person*" and preferred to suffer through the problem rather than speak up. Meanwhile, Team Charlie had one member of the team drop the course, leaving more work for everyone else. As the team never established effective leadership or held to deadlines, they struggled with many tasks until "*the hour before [they were] due*". As we discuss in Section 5.3, Team Charlie may have needed external accountability.

Finally, Team Bravo was in a unique position. Finnegan reported that three members of their team "*were very involved, and very willing to like [do] anything*". They said "*it was productive for us three*" and the team identified tasks, Finnegan delegated them, and they shared progress. However, Finnegan explained that tasks assigned to the remaining two members "*wouldn't get done...and would put us in [a] bad position*". Finnegan conjectured this was partially due to a language barrier, as Glenn "*had a hard time understanding us*". Ultimately, Glenn collaborated with Max, who came from a similar cultural background. Landon also struggled to participate effectively on Team Bravo. Finnegan reported that Landon missed every out-of-lab meeting. Landon acknowledged being "*aware of the problem*" but was "*not in a position*" to solve it. There was no clear way the team could have overcome this challenge.

> ***RQ2:*** *When everyone on a team is making an effort to participate, reflecting on what is working and what is not working can be enough for teams to figure out what they need to do differently. Other teams may need the pressure of an impending deadline, or external motivation from the course teaching staff, to encourage everyone to contribute. Finally, abrasive leadership and mental health challenges posed insurmountable barriers for other teams.*

5.3 RQ3: Support from Teaching Staff

In this section, we discuss how the course teaching staff can help teams work more effectively, by focusing on answers to Q4, Q8-11, and Q13-16 from our interview script.

In every lab, TAs meet with each team to check on their progress and offer guidance. However, as students do most of their work independently outside of lab, it is often difficult for TAs to identify team dynamics and whether everyone is doing their part and the team is working effectively together. Consequently, in prior work [42] we assembled a checklist-based intervention to supplement the TCRS, providing questions for TAs to ask teams on communication, collaboration, and project management. Additionally, the intervention encouraged TAs to conduct mid-week email checkins with struggling teams to help hold members accountable. Prior work found no improvement in grades from this intervention. In this work, we aim to identify why and how to fix it.

As part of our interviews, we asked students how we, as members of the course teaching staff, could help them overcome collaboration issues. For teams that faced no issues, we asked how we could help them overcome hypothetical issues similar to those we observed.

5.3.1 TA Interventions. Prior work demonstrated that this checklist-based intervention for helping teams overcome challenges was ineffective. Comments from students on struggling teams overwhelming tell the same story: they want more help, and more active engagement, from members of the teaching staff. On Team Charlie, Emery requested "*more of a guiding hand*" than a "*passing comment*". They explained that "*[no]body really noticed or paid attention to*" comments from TAs. On Team Alpha, Adrian expressed a similar sentiment. They said that group projects "*typically expect people to be confrontational about problems*", which they were not comfortable doing, remarking that "*it's easier to do things yourself than try to explain to [TAs]*" what is happening. Adrian requested that TAs "*be a facilitator*" because if "*a person in authority*" brings up problems, the team will be less likely to "*not just say it's all OK*". When we asked Blair, from Team Hotel, how TAs could help with a hypothetical team challenge, they commented "*not a lot of people would want to directly confront someone*" and suggested that TAs take a more active role.

We received slightly more conflicting requests from students on other teams. On Team Delta, Spencer requested that TAs "*step in earlier*" and let teams know when their grade "*might be affected in the future*". By contrast, on Team Hotel, Carson would prefer to let teams "*try and resolve it amongst [ourselves] for a week*" before the teaching staff intervenes. Educational theory supports Carson's suggestion, arguing that it is important for teams to *try* to overcome their challenges before getting help. Also on Team Hotel, Blair made a similar comment, saying "*if it's recurring and it's a problem*" then TAs should get involved, but they would like the team to try first.

In Fall 2021, a student remarked in their end-of-project reflection "*When we had a major contribution issue, I reflected on that in the [TCRS] and TAs were able to intervene*". This student clearly appreciated knowing their reflections were used to foster discussions. However, we recognise that some students may feel they are being "called out" for what was said. Thus, we asked participants in this study if they would prefer we mention the TCRS, so that they know we are acting on them, or would they prefer that issues be brought up more generally, such as in the context of Github contributions. Spencer, on Team Delta, acknowledged both sides. They appreciated knowing that "*y'all take them seriously...you're actually reading them*", but said that if issues were brought up through the survey, they would "*immediately try and figure out who is doing this, who said this*". Consequently, they would prefer for us to "*depersonalise*" the comments. They said that it would be "*good*" to make a note of the reflections, but make it "*not the main reason*" or focus of the conversation. Eilian, on Team Hotel, described the tradeoff similarly. They said that "*it's really awkward*" for the TAs

to say that "*according to the feedback some people weren't doing their part*". They suggested that TAs start by asking students what they have contributed and how they have collaborated, and then follow up with more probing questions and discussion if the answers did not appear to match what was in the TCRS. Eilian suggested that we could "*still acknowledge [the TCRS]*" but preferred that it would not be the primary focus. While Emery, on Team Charlie, said they "*wouldn't mind*" if comments were brought up through the TCRS, the overall consensus is students prefer for comments to be brought up without it.

5.3.2 Team Formation. Some students suggested changes to the team forming activity discussed in Section 3. Parker, on Team Golf, acknowledged that the team forming activity was "*a good idea*" but said it was "*really hard*" to do effectively on the first day of the project when "*you don't really know the people and you don't really know the project*". They said it would be helpful to take time in lab on the second week of the project to review the team rules and goals and identify "*are [these] still working?*". To encourage everyone to read project materials ahead of time and make the project easier to discuss, Hayden, from Team Echo, said that "*A small quiz...would have helped*". This suggestion was echoed by Sawyer and Corey, of Team Golf.

Spencer, on Team Delta, suggested encouraging members to share "*our specific strengths and weaknesses*", acknowledging that while they discussed "*our technical skills*" they never discussed "*how we worked, or if we were bad at getting started on things early*". They hesitated to call out teammates for not getting things done because "*you never want the first impression to be like 'Hey guys, you're not doing your crap'*" and expressed optimism that further team forming would help. On Team Foxtrot, Page said their team was formed from two smaller groups from the prior project: "*three kids already knew each other, and then I had someone from my previous group which was really nice*". They said this made the team forming exercises "*really painless*", but acknowledged they "*didn't talk to any*" of the new team members for the first half of the project. They suggested "*swapping it [up] could have been more efficient*" as a way to introduce everyone. One of their teammates, Jamie, similarly said activities to "*break the ice between subteams, but without forcing them to cooperate*" may foster teamwork.

> ***RQ3:*** *We find that most students want a more active role from the course teaching staff, using their position of authority to bring up issues and then guide teams to a solution, holding members accountable as necessary. Students also suggested improvements to the team formation activity as a way for everyone to get to know each other more quickly.*

5.4 RQ4: Characteristics of Successful Teams

In this section, we discuss the characteristics of successful teams, by focusing on responses to Q1-6 and Q16-17 from our interview script.

In many ways, the characteristics of successful teams were largely the opposite of the teams that struggled the most. Carson, on Team Hotel, said that their group "*did really well communicating*" throughout. Blair, also on Team Hotel similarly said they "*communicated really well using Discord*". Also on Team Hotel, Emerson said that with multiple Discord channels "*we could sort of compartmentalise different discussions*". On Team Golf, Jesse echoed this, saying that "*we weren't all trying to talk in the same channel*" but "*we could still see what the other [sub]team was doing*". On Team India, Riley explained everyone discussed the tasks they were working on and "*brought up and talked about*" any disagreements. On Team Hotel, Eilian said that they would "*just straight up tell them [team members] like 'stop'*" if they were distracting meetings.

In addition to regularly communicating, we find that the most successful teams also worked on tasks together. On Team Hotel, Carson said that members on their team would "*hop in the voice chat real quick [when] we needed help on this or that*". Eilian, also on Team Hotel, explained that their teammates were "*really attentive*" and would "*come on Discord...until 8 or 9 [PM]*" if someone got stuck. On Team Foxtrot, Page said that a prior internship gave them experience with some of the technologies the project used. To get their teammate up to speed, they would "*almost strictly work in pair programming at first*", and remarked that "*I think that was actually really good...I....was able to teach [them]*". On Team Golf, Sawyer said that their teammates served as effective mentors, and would be "*like a guide to me*". They credited this relationship for helping them learn a new technology that they struggled with on the previous project.

We also find that successful teams held themselves accountable. Emerson, on Team Hotel, said they would hold scrum-style meetings once a week, and "*post updates in the chat on certain days*". Because of this, "*no one put their work to the last minute*". They said that the "*implicit shame*" of showing up to meetings unprepared ensured that everyone did their work. On Team India, Riley explained they "*would have a meeting every Monday afternoon*" to discuss progress and come up with a plan for "*anything that needed to be done*".

We also find that successful teams invested heavily in task planning. Sam explained that Team India would "*split up each thing and just estimate it...we tried to split every task [so they're] pretty small*". After estimating times, they would create a "*wheel [of] fortune thing with all of our names*" and assign tasks at random, ensuring that each person had "*the same amount of hours*" of work to complete. Sam said that this process was "*so much fun*" and ensured that everyone was engaged with the process. Team Golf took a more conventional but equally involved approach to task planning. Parker explained their team "*took every task [for the week]...and put all of them on a [white]board*", at which point each member of the team "*took turns grabbing what we wanted to do*". They said this approach was "*great for learning*" because everyone got a chance to do "*a little bit of everything*", but conceded that "*with our stochastic approach, it was a lot harder to coordinate*" dependencies. Jesse echoed that this planning approach worked well; they said that after the team planned out everything, Team Golf would photograph the plan and share it in Discord, at which point "*everybody knew what needed to be done*". Team Hotel used technology to facilitate their task planning. Carson explained that "*we utilised Github Projects*[4] *and the [Github] Issues like religiously*" and said this worked effectively.

We find that successful teams considered team leadership, but carried it out in a largely decentralised manner. On Team Hotel,

[4] A Kanban-board style task tracking system

Emerson served as initial team lead, delegating tasks and ensuring that everyone knew what to do. Over time, however, as the team "*got familiar with what worked for us*" leadership became a shared effort. On Team India, Sam explained "*a tendency to take leadership roles*", but said their role was more "*get[ting] conversations going and ideas moving*" than directing people. Meanwhile, Riley, also on Team India, described their leadership approach as decentralised, explaining that "*we all served as like passive leadership*". On Team Golf, Parker described how their team instituted a leader role after an initial rudderless week where the meetings were disorganised to the point of being "*torture*". Jesse, also on Team Golf, explained that the team leader's role was "*just to keep the meetings on track*" and figure out "*what needed to be talked about...and make sure that's what was talked about*". They said this approach "*worked really well*" for their team.

> ***RQ4:*** *Successful teams communicate regularly, sharing their current progress with the team. Additionally, members of these teams regularly collaborated on their tasks. We also see that these teams plan out tasks carefully and then hold themselves accountable. These teams consider leadership roles, but largely function in a decentralised manner, trusting each person to ask for help as needed.*

6 DISCUSSION

In this section, we discuss the significance of our results and how they compare to our prior work (Section 6.1), consider whether team formation impacts team success (Section 6.2), discuss possible improvements to team forming (Section 6.3), and finally discuss threats to validity (Section 6.4).

6.1 Significance of Results

We observe that none of the characteristics of successful and unsuccessful teams, as discussed in Section 5, come as a surprise. Prior work in teaming, as discussed in Section 2, has identified that student teams are at risk of dysfunction [14, 24, 35, 42, 50], particularly due to challenges with communication and project management. However, to the best of our knowledge, this is the first work to study student teamwork in software engineering from the perspective of the teams themselves, rather than external factors. Thus, we offer the first concrete evidence that the student-identified challenges faced in software engineering teams are consistent with broader pedagogy.

Our results here suggest that teams may face more challenges than grades alone reveal. All members of Team Alpha ultimately received an A on the project, but the team was nonetheless flagged several times through the TCRS for having collaboration issues. In this case the team did not work together as well as the grades indicated. This suggests that collaborative challenges may be greater than previously understood. Given the four distinct stages Tuckman [8] argues teams progress through, the six weeks of the project may simply not be long enough for all teams to overcome their challenges and work effectively.

6.2 Teammate Requests as a Predictor for Team Success

As discussed in Section 3, teams are formed by the teaching staff with student input. We sought to understand whether there was any relationship between team formation and success. We did so by looking at how each team was formed, and classifying it as A) at random, B) around one group of students who requested each other, or C) around two or more groups of students who requested each other. For example, Team Alpha falls into Group B, as four students mutually requested each other and they were matched with one additional student. By contrast, Team Hotel was formed from a group of three students who requested each other, a group of two students who requested each other, and a student with no requests, so Hotel falls into Group C. We find the formation of the team has little bearing on how effectively it worked. Carson, on Team Hotel, was the only member of the team who did not have another "buddy" that they had requested, but reported being "*grateful to have a really good group*". By contrast, on Team Alpha, the conflict that Adrian reported was between four people who mutually requested each other. Likewise, on Team Echo, the conflict was between two people who mutually requested each other. On Team Golf, which faced and overcame a collaborative challenge that was not due to any member in particular, three members mutually requested each other, but peer evaluations and comments in interviews showed that all six members were happy together. Overall, we find no clear relationship between how a team was formed and its success.

We note that although the teams studied in this work were enrolled in a synchronous, in-person class, this class followed a year and a half of online classes. Prior work has argued that students struggle to form effective relationships when working online [52], which are necessary for establishing the trust that supports positive teaming outcomes [9]. It is possible that students with a more normal educational trajectory would have established relationships that support more effective teaming.

6.3 Student-Suggested Improvements

Most of the project changes suggested by students offer a clear pedagogical improvement. The suggestion that we add a small quiz to encourage students to read project materials is used in flipped classrooms [54], and compelling students to prepare for the team forming activity may ward off the disorganisation that Parker described as "*torture*". Page, on Team Foxtrot, suggested that we make the final question on the TCRS, *How do you feel about your team's collaboration process in this project?*, mandatory, which may encourage further self-reflection. Hayden, on Team Echo, suggested that we allow teams to submit a redacted chat log demonstrating peer review in place of the review in lab, which can be accomplished with a rubric to ensure equitable grading [15]. Indeed, if this encourages students to engage in more peer review outside of lab, it would have clear benefits by promoting more timely and collaborative work. We have found both of these behaviours are associated with the most successful teams, and encouraging students to engage in them is advantageous.

Currently, both project deliverables and the TCRS are due Wednesday night, to prepare for labs on Thursday. Emery, on Team Charlie, said that they would "*put in the survey what I would expect people to*

submit", but their teammates would often not follow through, and thus their responses did not accurately represent team progress. We recognise that as a *reflection* survey, it may make more sense to have it due after other tasks, but unfortunately with one lab per week, this is not feasible. Emery suggested that we could "*allow for like, a second survey after lab*" if students had any followup comments. Combining this with a dashboard for viewing team challenges over time could help the teaching staff track persistent issues.

6.4 Threats to Validity

In this section, we discuss threats to validity, using categories suggested by Wohlin et al. [55].

Construct: When we ask students what worked effectively for their teams, and what they struggled with, they may interpret success differently than we do, focusing on what actions led them to a higher grade rather than better learning. We asked clarifying questions to focus them on collaborative behaviour when their answers did not match the questions.

Responses are subject to hypothesis guessing, particularly as interviews were conducted by a member of the course teaching staff. However, participants were forthcoming about both the strengths and weaknesses of their teams, suggesting a willingness to discuss their experiences frankly. Additionally, as discussed in Section 4.5, we compared responses from teammates, and checked what we could on Github, and found no misrepresentations.

Internal: In this work, we describe the characteristics of struggling teams and teams that worked together effectively. Students reported that when they took the steps towards behaviour we see associated with successful teams, they did better. This suggests a causal relationship.

The responses are subject to recall bias, as there was two months between the conclusion of the project and the interviews. We cross-referenced answers from students on the same team, and checked what information we could against Github, and found no misrepresentations. Not all information, such as how students met outside of lab, could be externally verified, so we must rely on what students said. Aside from what we were told by students, we do not have information on their other obligations, such as how many classes they are taking or whether they have day jobs.

External: This study was conducted with 18 students from one course and one semester at one university. We caution that these findings may not be broadly representative of team-based software engineering projects, and encourage replication with students from different courses.

Reliability: The interviews for this study were conducted by the first author, and the data analyses primarily by them. However, after minor changes following the first several interviews, our interview script remained unchanged, and all subsequent interviewees were asked the same questions. This step improves reliability as all interviewees were asked a consistent set of questions.

7 CONCLUSION & FUTURE WORK

In this work, we have studied how students on software engineering teams work together. We have revealed that students face issues communicating, establishing and keeping to deadlines, and estimating the difficulty of tasks. Some teams are able to overcome these issues, partially or completely, by reflecting on what is working and not working, or through external motivators such as grades. However, mental health challenges and intransigent teammates remain a challenge, suggesting that instructors need to do more to offer support for struggling teammates and encourage better behaviour. Additionally, we discuss the characteristics of successful teams, and report that these teams stay in regular communication, using Discord to facilitate asynchronous discussion and holding meetings to work on tasks together as a team. Members on these teams hold each other accountable and support each other. We consider suggestions that students offer on how to provide more effective feedback and guidance.

This work has identified behaviours associated with struggling teams and ones associated with successful teams. We encourage future work to identify whether an intervention can steer teams towards these latter behaviours. Additionally, as our study is limited in scope, we suggest a replication of this work with a course where students work in the same teams all semester to see both what challenges teams face and whether they are more successful at overcoming them when they are working together for sixteen weeks as opposed to six.

ACKNOWLEDGMENTS

This work was supported in part by NSF SHF grants #1749936 and #1525173. We would like to thank the students of NC State University's Software Engineering course who graciously participated in our interviews.

REFERENCES

[1] [n.d.]. About ABET | ABET. https://www.abet.org/about-abet/
[2] [n.d.]. Discord | Your Place to Talk and Hang Out. https://discord.com/
[3] 2020. Criteria for Accrediting Engineering Programs, 2021 – 2022. https://www.abet.org/accreditation/accreditation-criteria/criteria-for-accrediting-computing-programs-2021-2022/
[4] 2021. NLTK: Sample usage for sentiment. https://www.nltk.org/howto/sentiment.html
[5] Mehmet Ayaz and Hanifi şekerci. 2015. The Effects of the Constructivist Learning Approach on Student's Academic Achievement: A Meta-Analysis Study. *Turkish Online Journal of Educational Technology* 14 (10 2015).
[6] Lingfeng Bao, Tao Li, Xin Xia, Kaiyu Zhu, Hui Li, and Xiaohu Yang. 2020. How does Working from Home Affect Developer Productivity? - A Case Study of Baidu During COVID-19 Pandemic. *ArXiv* abs/2005.13167 (2020).
[7] Anders Berglund. 2005. *Learning computer systems in a distributed project course : The what, why, how and where.* Ph.D. Dissertation.
[8] Denise A. Bonebright. 2010. 40 years of storming: a historical review of Tuckman's model of small group development. *Human Resource Development International* 13, 1 (2010), 111–120.
[9] Maura Borrego, Jennifer Karlin, Lisa D. McNair, and Kacey Beddoes. 2013. Team Effectiveness Theory from Industrial and Organizational Psychology Applied to Engineering Student Project Teams: A Research Review. *Journal of Engineering Education* 102, 4 (2013), 472–512. https://doi.org/10.1002/jee.20023 arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1002/jee.20023
[10] Kathy Charmaz. 2014. *Constructing grounded theory.* sage.
[11] Carsten KW De Dreu and Laurie R Weingart. 2003. Task versus relationship conflict, team performance, and team member satisfaction: a meta-analysis. *Journal of applied Psychology* 88, 4 (2003), 741.
[12] Frank RC De Wit, Lindred L Greer, and Karen A Jehn. 2012. The paradox of intragroup conflict: a meta-analysis. *Journal of applied psychology* 97, 2 (2012), 360.
[13] Molly Dingel and Wei Wei. 2014. Influences on peer evaluation in a group project: an exploration of leadership, demographics and course performance. *Assessment & Evaluation in Higher Education* 39, 6 (2014), 729–742. https://doi.org/10.1080/02602938.2013.867477
[14] Dora Dzvonyar, Lukas Alperowitz, Dominic Henze, and Bernd Bruegge. 2018. Team Composition in Software Engineering Project Courses. In *2018 IEEE/ACM International Workshop on Software Engineering Education for Millennials (SEEM).* 16–23.

[15] J. Feldman. 2018. *Grading for Equity: What It Is, Why It Matters, and How It Can Transform Schools and Classrooms*. SAGE Publications. https://books.google.com/books?id=gxhoDwAAQBAJ

[16] Lucy L. Gilson, M. Travis Maynard, and Erich B. Bergiel. 2013. Virtual Team Effectiveness: An Experiential Activity. *Small Group Research* 44, 4 (2013), 412–427. https://doi.org/10.1177/1046496413488216 arXiv:https://doi.org/10.1177/1046496413488216

[17] Niki Gitinabard, Ruth Okoilu, Yiqiao Xu, Sarah Heckman, Tiffany Barnes, and Collin F. Lynch. 2020. Student Teamwork on Programming Projects. What can GitHub logs show us?. In *Proceedings of the 13th International Conference on Educational Data Mining, EDM 2020, Fully virtual conference, July 10-13, 2020*, Anna N. Rafferty, Jacob Whitehill, Cristóbal Romero, and Violetta Cavalli-Sforza (Eds.). International Educational Data Mining Society. https://educationaldatamining.org/files/conferences/EDM2020/papers/paper_153.pdf

[18] David Hall and Simone Buzwell. 2013. The problem of free-riding in group projects: Looking beyond social loafing as reason for non-contribution. *Active Learning in Higher Education* 14, 1 (2013), 37–49. https://doi.org/10.1177/1469787412467123

[19] Randall S. Hansen. 2006. Benefits and Problems With Student Teams: Suggestions for Improving Team Projects. *Journal of Education for Business* 82, 1 (2006), 11–19. https://doi.org/10.3200/JOEB.82.1.11-19

[20] Alan Hogarth. 2008. Introducing a collaborative technology strategy for higher education students: Recommendations and the way forward. *Education and Information Technologies* 13 (09 2008), 259–273. https://doi.org/10.1007/s10639-008-9064-x

[21] Lu Hong and Scott E. Page. 2004. Groups of diverse problem solvers can outperform groups of high-ability problem solvers. *Proceedings of the National Academy of Sciences* 101, 46 (2004), 16385–16389. https://doi.org/10.1073/pnas.0403723101 arXiv:https://www.pnas.org/content/101/46/16385.full.pdf

[22] Patricia Hrynchak and Helen Batty. 2012. The educational theory basis of team-based learning. *Medical Teacher* 34, 10 (2012), 796–801. https://doi.org/10.3109/0142159X.2012.687120 arXiv:https://doi.org/10.3109/0142159X.2012.687120 PMID: 22646301.

[23] Christopher Hundhausen, Adam Carter, Phillip Conrad, Ahsun Tariq, and Olusola Adesope. 2021. *Evaluating Commit, Issue and Product Quality in Team Software Development Projects*. Association for Computing Machinery, New York, NY, USA, 108–114. https://doi.org/10.1145/3408877.3432362

[24] Claudia Iacob and Shamal Faily. 2020. The Impact of Undergraduate Mentorship on Student Satisfaction and Engagement, Teamwork Performance, and Team Dysfunction in a Software Engineering Group Project. In *Proceedings of the 51st ACM Technical Symposium on Computer Science Education* (Portland, OR, USA) *(SIGCSE '20)*. Association for Computing Machinery, New York, NY, USA, 128–134. https://doi.org/10.1145/3328778.3366835

[25] Steven J Karau and Kipling D Williams. 1993. Social loafing: A meta-analytic review and theoretical integration. *Journal of personality and social psychology* 65, 4 (1993), 681.

[26] Steven J Karau and Kipling D Williams. 1995. Social loafing: Research findings, implications, and future directions. *Current Directions in Psychological Science* 4, 5 (1995), 134–140.

[27] Jayden Khakurel and Jari Porras. 2020. The Effect of Real-World Capstone Project in an Acquisition of Soft Skills among Software Engineering Students. In *2020 IEEE 32nd Conference on Software Engineering Education and Training (CSEE T)*. 1–9. https://doi.org/10.1109/CSEET49119.2020.9206201

[28] Shahedul Huq Khandkar. 2009. Open coding. *University of Calgary* 23 (2009), 2009.

[29] P. Layzell, O. P. Brereton, and A. French. 2000. Supporting collaboration in distributed software engineering teams. In *Proceedings Seventh Asia-Pacific Software Engineering Conference. APSEC 2000*. 38–45. https://doi.org/10.1109/APSEC.2000.896681

[30] Joseph Maguire, Nathalie Sheridan, Steve Draper, and Quintin Cutts. 2019. Mentoring Mentors in Cooperative Software Engineering Education Programmes. In *Proceedings of the 2019 ACM Conference on International Computing Education Research* (Toronto ON, Canada) *(ICER '19)*. Association for Computing Machinery, New York, NY, USA, 307. https://doi.org/10.1145/3291279.3341205

[31] Maíra Rejane Marques. 2016. Monitoring: An Intervention to Improve Team Results in Software Engineering Education. In *Proceedings of the 47th ACM Technical Symposium on Computing Science Education* (Memphis, Tennessee, USA) *(SIGCSE '16)*. Association for Computing Machinery, New York, NY, USA, 724. https://doi.org/10.1145/2839509.2851054

[32] Courtney Miller, Paige Rodeghero, Margaret-Anne Storey, Denae Ford, and Thomas Zimmermann. 2021. *How Was Your Weekend? Software Development Teams Working From Home During COVID-19*. IEEE Press, 624–636. https://doi.org/10.1109/ICSE43902.2021.00064

[33] Emily Namey, Greg Guest, Lucy Thairu, and Laura Johnson. 2008. *Data reduction techniques for large qualitative data sets*. 137–162.

[34] Derrick J. Neufeld and Yulin Fang. 2005. Individual, social and situational determinants of telecommuter productivity. *Information & Management* 42, 7 (2005), 1037–1049. https://doi.org/10.1016/j.im.2004.12.001

[35] Barbara Oakley, Rebecca Brent, Richard Felder, and Imad Elhajj. 2004. Turning student groups into effective teams. *Journal of Student Centered Learning* 2 (01 2004).

[36] Ernesto Panadero. 2017. A Review of Self-regulated Learning: Six Models and Four Directions for Research. *Frontiers in Psychology* 8 (2017), 422. https://doi.org/10.3389/fpsyg.2017.00422

[37] Reza M. Parizi, Paola Spoletini, and Amritraj Singh. 2018. Measuring Team Members' Contributions in Software Engineering Projects using Git-driven Technology. In *2018 IEEE Frontiers in Education Conference (FIE)*. 1–5. https://doi.org/10.1109/FIE.2018.8658983

[38] Trena M. Paulus, Brian Horvitz, and Min Shi. 2006. 'Isn't It Just Like Our Situation?' Engagement and Learning in an Online Story-Based Environment. *Educational Technology Research and Development* 54, 4 (01 Aug 2006), 355–385. https://doi.org/10.1007/s11423-006-9604-2

[39] Pilar Pazos and Nina Magpili. 2015. Facilitating team Processes in virtual team projects through a web-based collaboration tool and instructional scaffolds. In *2015 ASEE Annual Conference & Exposition*. 26–754.

[40] Vreda Pieterse and Lisa Thompson. 2010. Academic alignment to reduce the presence of 'social loafers' and 'diligent isolates' in student teams. *Teaching in Higher Education* 15, 4 (2010), 355–367.

[41] Rodley Pineda and Linda Lerner. 2006. Goal attainment, satisfaction and learning from teamwork. *Team Performance Management* 12 (07 2006), 182–191. https://doi.org/10.1108/13527590610687938

[42] Kai Presler-Marshall, Sarah Heckman, and Kathryn Stolee. 2022. Identifying Struggling Teams in Software Engineering Courses Through Weekly Surveys. *Proceedings of the 53rd ACM Technical Symposium on Computer Science Education* (2022), 7 pages. https://doi.org/10.1145/1122445.1122456

[43] Patricia D. Rafferty. 2013. Group Work in the MBA Classroom: Improving Pedagogical Practice and Maximizing Positive Outcomes With Part-Time MBA Students. *Journal of Management Education* 37, 5 (2013), 623–650. https://doi.org/10.1177/1052562912458644 arXiv:https://doi.org/10.1177/1052562912458644

[44] Frederike Ramin, Christoph Matthies, and Ralf Teusner. 2020. More than Code: Contributions in Scrum Software Engineering Teams. In *Proceedings of the IEEE/ACM 42nd International Conference on Software Engineering Workshops* (Seoul, Republic of Korea) *(ICSEW'20)*. Association for Computing Machinery, New York, NY, USA, 137–140. https://doi.org/10.1145/3387940.3392241

[45] Tammy Rapp and John Mathieu. 2007. Evaluating an Individually Self-Administered Generic Teamwork Skills Training Program Across Time and Levels. *Small Group Research - SMALL GROUP RES* 38 (08 2007), 532–555. https://doi.org/10.1177/1046496407300479

[46] Ita Richardson, Valentine Casey, Fergal McCaffery, John Burton, and Sarah Beecham. 2012. A Process Framework for Global Software Engineering Teams. *Information and Software Technology* 54, 11 (2012), 1175 – 1191. https://doi.org/10.1016/j.infsof.2012.05.002

[47] Johnny Saldaña. 2009. *The coding manual for qualitative researchers*. London ; Thousand Oaks, Calif. : Sage, 2009., London. Includes bibliographical references (pages [210]-219) and index.

[48] J. E. Sims-Knight, R. L. Upchurch, T. A. Powers, S. Haden, and R. Topciu. 2002. Teams in software engineering education. In *32nd Annual Frontiers in Education*, Vol. 3. S3G–S3G. https://doi.org/10.1109/FIE.2002.1158712

[49] Anya Tafliovich, Andrew Petersen, and Jennifer Campbell. 2016. Evaluating Student Teams: Do Educators Know What Students Think?. In *Proceedings of the 47th ACM Technical Symposium on Computing Science Education* (Memphis, Tennessee, USA) *(SIGCSE '16)*. Association for Computing Machinery, New York, NY, USA, 181–186. https://doi.org/10.1145/2839509.2844647

[50] Richard Tucker and Catherine Reynolds. 2006. The Impact of Teaching Models, Group Structures and Assessment Modes on Cooperative Learning in the Student Design Studio. *Journal for Education in the Built Environment* 1, 2 (2006), 39–56. https://doi.org/10.11120/jebe.2006.01020039 arXiv:https://doi.org/10.11120/jebe.2006.01020039

[51] Bruce W. Tuckman and Mary Ann C. Jensen. 1977. Stages of Small-Group Development Revisited. *Group & Organization Studies* 2, 4 (1977), 419–427. https://doi.org/10.1177/105960117700200404 arXiv:https://doi.org/10.1177/105960117700200404

[52] Alexis Rae Walsh, Sarah E Norris, Nathaniel Blalock, Daniel Patrick Mountain, and Courtney June Faber. 2021. Exploring the Team Dynamics of Undergraduate Engineering Virtual Teams During the Rapid Transition Online Due to COVID-19. In *2021 ASEE Virtual Annual Conference Content Access*.

[53] L. A. Williams and R. R. Kessler. 2000. The effects of "pair-pressure" and "pair-learning" on software engineering education. In *Thirteenth Conference on Software Engineering Education and Training*. 59–65. https://doi.org/10.1109/CSEE.2000.827023

[54] Stephanie Gray Wilson. 2013. The Flipped Class: A Method to Address the Challenges of an Undergraduate Statistics Course. *Teaching of Psychology* 40, 3 (2013), 193–199. https://doi.org/10.1177/0098628313487461 arXiv:https://doi.org/10.1177/0098628313487461

[55] Claes Wohlin, Per Runeson, Martin Höst, Magnus C. Ohlsson, Björn Regnell, and Anders Wesslén. 2012. *Experimentation in Software Engineering*. Berlin, Heidelberg : Springer Berlin Heidelberg : Imprint: Springer, 2012., Berlin, Heidelberg.