Identifying Struggling Teams in Software Engineering Courses Through Weekly Surveys

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 increasingly a core aspect of professional software engineering and most undergraduate computer science curricula. At NC State University, we teach communication and project-management skills explicitly through a junior-level software engineering course. However, some students may have a dysfunctional team experience that imperils their ability to learn these skills. Identifying these teams during a team project is important so the teaching staff can intervene early and hopefully alleviate the issues.

We propose a weekly reflection survey to help the course teaching staff proactively identify teams that may not be on track to learn the course outcomes. The questions on the survey focus on team communication and collaboration over the previous week. We evaluate our survey on two semesters of the undergraduate software engineering course by comparing teams with poor end-of-project grades or peer evaluations against teams flagged on a weekly basis through the surveys. We find that the survey can identify most teams that later struggled on the project, typically by the half-way mark of the project, and thus may provide instructors with an actionable early-warning about struggling teams. Furthermore, a majority of students (64.4%) found the survey to be a helpful tool for keeping their team on track. Finally, we discuss future work for improving the survey and engaging with student teams.

CCS CONCEPTS

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

KEYWORDS

software engineering education, team projects, peer evaluations, surveys, course interventions

ACM Reference Format:
Kai Presler-Marshall, Sarah Heckman, and Kathryn T. Stolee. 2022. Identifying Struggling Teams in Software Engineering Courses Through Weekly Surveys. In *Proceedings of the 53rd ACM Technical Symposium on Computer Science Education V. 1 (SIGCSE 2022), March 3–5, 2022, Providence, RI, USA.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3478431.3499367

SIGCSE 2022, March 3–5, 2022, Providence, RI, USA.

ACM ISBN 978-1-4503-9070-5/22/03…$15.00
https://doi.org/10.1145/3478431.3499367

1 INTRODUCTION

Computer science education is increasingly focused on team-based learning, where students work collaboratively to achieve the learning goals for a course [1]. Such an approach provides an educational environment that more closely resembles professional software engineering workplaces [25].

A team's success depends upon many factors, including good-faith participation, equitable contribution, and effective communication [30]. When a team lacks these characteristics, the impact on the team can be negative: students are left frustrated by underperforming teammates or confused about the team's progress [17]. Teams that are unable to communicate effectively will be at a disadvantage throughout the project [9]. Furthermore, poor teaming experiences can harm a sense of engagement or belonging with the Computer Science community, particularly among historically under-represented groups [7]. Consequently, while grade adjustments can be made at the end of the project, a better approach is to identify and mitigate problems before the project ends.

In this work, we seek to understand whether weekly collaboration reflection surveys can effectively identify struggling teams in our software engineering course. In particular, we seek to understand whether we can observe several commonly-observed issues: (1) students whose work quantity or quality does not rise to the expected standards, (2) teams where members are uncertain about what they should be accomplishing, and (3) teams where members are not attending meetings or following expected communications patterns [3, 31]. While peer evaluations can identify students who fail to contribute adequately to the success of their team, or who go above and beyond to ensure success [5], these tend to focus on evaluating individual teammates rather than the dynamic of the entire group. By contrast, our survey focuses on whole-team collaboration, as a team is more than the sum of its members. Finally, taking time to reflect on what is working and not working is a key component of self-regulated learning and may promote better learning outcomes [19].

We frame our work around the following research questions:

- **RQ1:** *Can weekly reflection surveys identify software engineering teams in need of instructor assistance?*
- **RQ2:** *Can weekly reflection surveys identify software engineering teams that need assistance sufficiently early?*
- **RQ3:** *Can weekly reflection surveys help support a better experience for software engineering teams?*

These questions evaluate the utility of a *weekly Team Collaboration Reflection Survey (TCRS)* we developed, based on prior work in teaming and collaboration [6, 18, 23, 27]. We administered the survey on a weekly basis to students in the junior-level software

engineering course at NC State University. We then analysed the results of the TCRS, comparing against grades and peer evaluations [4] to understand its effectiveness.

Our results show the survey is effective at identifying struggling teams, and by the halfway mark of the project in most cases. A majority of students reported the TCRS helped keep them on track.

Our contributions are as follows:

- A weekly team collaboration reflection survey (TCRS) suitable for undergraduate software engineering courses, and an approach for flagging struggling teams based upon it.
- A demonstration that the TCRS is capable of identifying teams that later face difficulties, and can in most cases do so by the halfway mark of the project.
- A blueprint for an intervention that may be helpful for engaging with teams that are struggling.

2 RELATED WORK

Professional software engineering is a team-based activity, often drawing together a team of diverse and distributed members [12, 24, 25]. To better prepare students, most software engineering classes include some form of collaborative learning [11, 28, 32].

However, group-based software engineering education comes with the risk of team challenges. Oakley et al. [17] report that many students struggle to form effective teams, as they lack the communication, project management, and conflict resolution skills necessary to succeed. They note that poor team experiences may leave students worse-off than simply working on their own. Outside of software engineering, Tucker and Reynolds [31] report that about 40% of teams in a project-based course are characterised by "conflict and selfish ambition" and fail to work together effectively.

Iacob and Faily [11] and Marques [14] report that dysfunction is a risk in team projects within software engineering education, where a lack of engagement or poor communication can hamper both individual and team outcomes. They report that bringing in software engineering professionals to serve as mentors to teams improves both learning outcomes and student satisfaction. However, such an approach requires significant time investments in recruiting willing and capable mentors and from the mentors themselves. We attempt to solve the same problems through a lightweight approach, where the TAs serve as more junior mentors, and the TCRS provides guidance on which teams particularly need help.

Existing literature in self- and peer-assessment (SAPA) has studied techniques and tools [10] to encourage students to reflect on their contributions and their teammates' contributions. Most educators agree that peer evaluations are necessary so that students don't receive equal grades for unequal effort [26], but acknowledge several difficulties: with limited face-to-face contact with students while they are working, peer evaluations can be one student's word against another [16]. Additionally, peer evaluations may be subject to bias: there is some evidence to suggest that gender and race impact scores students receive, and that white males may receive better scores for meeting stereotypes about what a "leader" should look like [8]. Our survey allows students to speak candidly and confidentially about what is working and what is not; the teaching staff can then cross-reference this with project data from a course version control system to decide how to engage with teams.

Prior work in education has used surveys to solicit feedback from students on their experiences with project-based learning. Owens [18] and Mendo-Lazaro et al. [15] discuss using surveys to understand what undergraduate students perceive as the main advantages and disadvantages of team-based learning and the challenges that they face working in such environments. Burdett [6] proposes interventions "through monitoring and arbitration" to help resolve issues teams are facing; in this work we attempt a realisation of this proposition.

3 BACKGROUND

At NC State University, a research-intensive university in the southeastern 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, such as 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 team-based projects that each span several weeks. The first project, an *onboarding project* (OBP), introduces the process expectations and technology stack. The second, a larger *team project* (TP), asks students to complete a more comprehensive project with a larger team. The OBP is completed in teams of two or three students; the TP in teams of five or six. Projects are broken into iterations, each typically lasting one week, that cover different learning objectives: requirements and planning, design, testing, and implementation. Students are evaluated in five categories: technical deliverables (including both code and technical documents); technical processes; project management; team collaboration; and peer review. Each of these high-level grade categories includes both team and individually graded components. 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 seek to supplement the peer evaluations with a more informal metric for identifying struggling teams.

The software engineering course at NC State University typically has between 120 and 160 students a semester, led by one PhD professor and three to five teaching assistants (TAs). Consequently, the student:teaching staff ratio is typically between 30:1 to 25:1. This necessitates light-weight approaches for detecting struggling teams. To support the project-based learning, the course features weekly lab sessions; led by the TAs, the labs provide time for teams to review technical deliverables from the previous week and plan tasks for the next week. Lab sessions are typically conducted in-person, but due to the university's response to the COVID-19 situation were conducted online via Zoom from Spring 2020 through Spring 2021.

With the backing of the DELTA Center, a Teaching Technology group at NC State University that offers support for educational technologies and course redesigns, we introduced a weekly reflection survey (TCRS) into both class projects. The TCRS, discussed in further detail in Section 4.2, provide students an opportunity to reflect on how their project is going, and gives the teaching staff regular, *in-situ* feedback on how teams are working together. Surveys have been administered sporadically, with inconsistent followup, from Fall 2017 to Fall 2019. We have recently revised the

Table 1: A summary of the participants involved in our study across the Fall 2020 and Spring 2021 semesters. *OBP* and *TP* are the two projects in our course, as discussed in Section 3.

	Fall 2020	Spring 2021
Students	120	162
IRB Opt-Outs	2	5
Teams - OBP	42	57
Teams - TP	21	28
Struggling Teams - OBP	9	13
Struggling Teams - TP	8	8

survey, and, starting with Fall 2020, administered the survey every week to enable drawing more reliable conclusions.

4 STUDY DESIGN

In our study, we deployed the TCRS to each student weekly throughout both course projects. This section describes the survey design, participants, analysis, and an intervention to help struggling teams.

4.1 Participants

We ran our study in Fall 2020 and Spring 2021, using the class described in Section 3. Of the 120 students in the course in Fall 2020, two students declined to let us analyse their data for research purposes. As data within a team cannot be separated for our purposes, their entire teams were excluded from further analysis. The Spring 2021 semester had 162 students enrolled; five students opted out of letting us analyse their data. A summary of the participants from each semester is shown in Table 1. The final two rows of the table describe the number of teams that we identified as "struggling" in each semester, as described in Section 4.3.

4.2 Surveys

The survey was originally developed by the DELTA Center at NC State University using prior work in teaming and collaboration [6, 18, 23, 27]. The survey was revised prior to Fall 2020 to add additional questions on what students were working on and to focus on key questions from prior work on identifying struggling teams. The questions on the survey are shown in Figure 1. The survey was deployed through Qualtrics, and all questions other than **Q14** were mandatory. Weekly response rates were approximately 90%; for example, for the TP in Spring 2021 response rates ranged from 87% to 93%, averaging 91% across the project.

To support a repeatable way of flagging teams, we needed a way to quantify TCRS responses. To do this, we broke down the survey into "positive questions" (ones where we would expect a successful team to answer either "*Agree*" or "*Strongly Agree*", such as **Q7**) and "negative questions" (where we would expect a successful team to answer either "*Disagree*" or "*Strongly Disagree*") and assigned numerical scores to the Likert scale responses. For positive questions, an answer of "Strongly Agree" was assigned a score of 4, "Agree" a score of 2, and so on down to -4 for "Strongly Disagree". For negative questions, this scale was reversed, with "Strongly Disagree" receiving a score of 4. The process was repeated for each question, and the scores for each were summed. Questions that did not fall into either the positive or negative categories, such as asking students what they had accomplished over the week, were excluded from the scoring process. Any survey where the overall score was 0 or less, indicating that the student felt that more things were going wrong than right, was flagged as indicating issues.

4.3 Observed Struggle Oracle

The surveys are intended to flag, or predict, teams that are struggling. To determine if the survey correctly identifies such teams, we need an oracle, which we formed from two metrics:

Low Project Grades: Most teams typically do well on the Team Project, with approximately 90% scoring an A or B. The Onboarding Project has more variability in project grades. In both cases, we use a cutoff of one and a half standard deviations below the mean project grade to identify struggling teams.

Peer Evaluations: Students evaluate themselves and their peers, rating each member between *1 (Infrequently)* to *6 (Above and Beyond)* on metrics such as their contributions and timeliness. The OBP has one peer evaluation, completed at the end of the project; the TP has two: one at the halfway mark and one at the end of the project. Each student received the average of the scores from their teammates (and, for the TP, averaged across both peer evaluations). Students who scored at least one and a half standard deviations below the class average were identified as struggling, and their team was selected for analysis.

These metrics have been used as measures of success in team-based learning environments [22, 29]. If either metric flags a team or team member, we consider that an indication of struggle; we refer to these as teams with *observed struggling* or *observed struggling behaviour*. The number of teams with observed struggling behaviour can be seen in the last two rows of Table 1.

To verify the oracle formed using these metrics, we cross-checked the team classifications with another data source, the end-of-project reflections (for the OBP, done through a semi-open-ended Google Form; for the TP, done through a three-page written document). The reflections asked students to consider the entire project and how they and their team had worked to meet their goals. As an open-ended task, this gave students more opportunity to explain team dynamics, and let us verify the teams with observed struggling. To confirm our observations, we read through the end-of-project reflections submitted by each member of the eight teams that were observed to struggle on the TP in Fall 2020. For seven of the eight teams, at least three members (out of the five or six members of the team) mentioned issues such as the team falling behind on deliverables or communication difficulties. On the eighth team, one member received poor peer evaluations, but there were no issues reported in the final reflections. To contrast this against the rest of the class and establish a baseline, we randomly selected five other teams with no observed struggle and read through their reflections. Students on one of the teams reported in their reflections that they faced communication issues as the project progressed; no issues were reported by any other students. Consequently, we consider the metrics of grades and peer evaluations to be reasonably accurate, if imperfect, for identifying teams struggling during the project.

Weekly tasks questions, answered with checkboxes in response to *This week I have:*
Q1: ☐ Designed a usecase (or a portion of one) ☐ Fixed a bug in the system ☐ Implemented a usecase (or a portion of one) ☐ Written black-box tests ☐ Written automated tests ☐ Other: _____
Q2: ☐ Completed all my assigned tasks ☐ Completed some of my assigned tasks ☐ Asked a teammate for help completing my tasks ☐ Helped a teammate complete a portion of their tasks
Q3: ☐ Met live with my team ☐ Participated in checkins with my team ☐ Opened a pull request and asked my team for feedback on my code ☐ Asked my team for feedback on my non-code work ☐ Reviewed technical artifacts for my teammates

Planning questions, answered with a five-point Likert scale: ○ **Much less** ○ **Less** ○ **About as much as** ○ **More** ○ **Much more**
Q4: This week, I have gotten done __ than I think I should have
Q5: This week, my team overall has gotten done __ than I think we should have
Q6: Next week, I intend to get done __ than I did this week

Collaboration satisfaction questions, answered with a five-point Likert scale: ○ **Strongly disagree** ○ **Disagree** ○ **Neither agree nor disagree** ○ **Agree** ○ **Strongly agree**
Q7: This week, I knew what I needed to get done
Q8: Overall, I think that everyone has been contributing adequately to the success of the project
Q9: In our team we relied on each other to get the job done
Q10: Team members kept information to themselves that should be shared with others
Q11: I am satisfied with the performance of my team
Q12: We have completed the tasks this week in a way we all agreed upon

Miscellaneous questions:
Q13: My progress this week has been impeded by:
☐ Difficulties with technologies or course materials ☐ Demands of other classes ☐ Other personal responsibilities or distractions ☐ Teammates who didn't complete their responsibilities ☐ Communication difficulties with my teammates ☐ Difficulty scheduling tasks so that I wasn't waiting for my team to complete their work ☐ Other: _____ ☐ None

Q14: How do you feel about your team's collaboration process in this project?

Figure 1: Team Collaboration Reflection Survey

4.4 Intervention

To assist struggling teams, we developed a checklist intervention with sample questions to ask teams. Based on prior work [11], the checklist focuses on getting students to articulate what specifically they are responsible for, how they have been meeting and collaborating with their teams, and helping them schedule tasks to allow for concurrent work. The TAs used this checklist during the lab sessions; additionally, they were encouraged to follow up with teams via email to help hold members accountable to their plans.

In Spring 2021, we conducted a structured experiment where struggling teams in half of the lab sections received the targeted follow-up intervention (experimental labs) and struggling teams in the other half of the lab sections did not (control labs). We then measured the impact of the intervention by comparing end-of-project grades and peer evaluations between the groups.

5 RESULTS

Here, we present results on the efficacy of the weekly TCRS at identifying teams in need of assistance (RQ1), identifying them early into the project (RQ2), and impacts on student success (RQ3).

Table 2: Success of surveys (*TCRS*) at predicting team struggle relative to struggle observed (*observed struggle*, see Section 4.3). Percentages are relative to the number of teams in total for that project in each semester.

		OBP - F20 TCRS		TP - F20 TCRS	
		Flagged	Not Flagged	Flagged	Not Flagged
Observed Struggle	Yes	8 (19.0%)	1 (2.4%)	7 (33.3%)	1 (4.8%)
	No	10 (23.8%)	23 (54.8%)	3 (14.3%)	10 (47.6%)

(a) Results for Fall 2020

		OBP - S21 TCRS		TP - S21 TCRS	
		Flagged	Not Flagged	Flagged	Not Flagged
Observed Struggle	Yes	11 (19.3%)	2 (3.5%)	8 (28.6%)	0 (0%)
	No	14 (24.6%)	30 (52.6%)	8 (28.6%)	12 (42.9%)

(b) Results for Spring 2021

5.1 RQ1: Identifying Struggling Teams

First, we sought to understand whether weekly reflection surveys can successfully identify software engineering teams in need of instructor assistance.

We find that identifying and reporting struggling relies on the entire team for accurate results. For example, for the OBP in Fall 2020, nine students (on eight teams) were flagged for receiving a peer evaluation at least one and a half standard deviations below the class average. The nine students submitted a total of 47 TCRS responses over the course of the project. Twelve responses across five flagged students (and four distinct teams) identified that the team was facing challenges. Consequently, only four of the nine OBP teams we observed struggling[1] were flagged through the TCRS responses of their most unproductive members. Factoring in TCRS submissions from the team increased this to eight of nine teams, which suggests that reporting works best as a whole-team effort.

A breakdown of the teams that were identified through the TCRS and a comparison to teams with observed struggle is shown in Table 2. Each sub-table shows the results for one semester; for example, Table 2a shows the results from Fall 2020. Within each subtable, the left side shows results for the OBP, the right side, for the TP. For example, the left half of Table 2a shows that for the OBP, 8 teams were flagged by the TCRS and had observed struggle. One team had observed struggle that was not flagged by the TCRS, 10 teams were flagged by the TCRS but had no observed struggle, and 23 teams were neither flagged nor had observed struggle. The right half of Table 2a shows results for the TP. Results for Spring 2021 are presented similarly in Table 2b, where we see every single team with observed struggle on the TP was flagged by the TCRS. In total, across two projects and two semesters, the TCRS flagged 34 of 38 teams, or 89.5%, with observed struggle.

[1]The final team with observed struggle was flagged solely through poor grades, not peer evaluations.

There is an inherent tradeoff between precision and recall: if the TCRS flags more teams, it will increase the recall (the number of struggling teams that the TCRS detects). However, this will come at the cost of lower precision (more teams flagged with no observed struggling). Given our use case, we prefer a survey that has high recall over one with high precision. As the cost of engaging with a team is low, rather than miss teams truly in need of instructor assistance, we prefer to engage with more teams that potentially don't need the help. That said, it is possible that the TCRS-flagged teams actually do need help, but their struggles did not translate to poor grades or poor peer evaluations. We discuss plans to probe this further in Section 6.4. Consequently, while many teams are flagged that do not ultimately demonstrate struggling outcomes – 35 over the course of two projects and semesters, giving a precision of 49.3% – the tradeoff suits the circumstances. The recall, by contrast, is much better – 89.5%. In Section 6.1 we discuss the struggling teams that were not flagged and measures to support similar teams.

RQ1: *The TCRS manages to identify most teams, 89.5% across two projects and two semesters, that exhibit observed struggling behaviour at the end of the project.*

5.2 RQ2: Identifying Teams Early

If the TCRS only reveals issues during the last week of a six-week project, it is unlikely that it will be of any practical use to the teaching staff. We seek to determine if the TCRS can identify struggling teams *sufficiently early*, defined as the halfway mark of the project. To answer RQ2, we find the first occurrence of a TCRS response indicating a problem for each team with observed struggle.

We present results for when teams were identified through the TCRS in Table 3. Table 3a presents results for the Onboarding Project; Table 3b presents results from the Team Project. Each column tracks a one-week iteration within each project, and the rows the semesters where the TCRS was used. The final two columns represent the number of teams with observed struggle that were not detected (**ND**) by the TCRS at any point during the project, and the percentage flagged by the halfway mark (**H?**). The final row summarises teams flagged in each half of the project. For example, the first row in Table 3a shows that in Fall 2020, one team was first flagged during Week 0 and five teams during Week 1. Ultimately, seven of the nine teams in Fall 2020, or 78%, were flagged by the TCRS by the halfway mark. Across both semesters, 14 of 22 teams, or 63.6%, were flagged by the halfway mark. The percentages are based on the teams with observed struggle, representing the oracle.

On the whole, the TCRS does a compelling job, identifying 28 of the 38 teams, or 73.7%, by the halfway marks of their respective projects. There is some difference between projects: 63.6% of teams on the OBP were identified by the halfway mark, compared to 87.5% of teams on the TP. This may be because the first several weeks of the OBP are spent on tasks the students find comparatively easy – requirements analysis, wireframing, and writing system tests – and consequently when implementation tasks start to pick up for Week 4, the workload increases and team dynamics can become strained.

If we move our goalposts one week later, the detection recall for the S21-OBP goes from 53.8% to 76.9%. For the scope of this project, the teaching staff would still have two weeks to help the teams improve. This suggests that the details of the project impact early detection. The TP involves more difficult tasks comparatively early on, which may make collaboration difficulties surface earlier, as seen in the final column of Table 3b.

Table 3: The first week that each team with *Observed Struggle* was flagged through the TCRS. Each column header represents a one-week iteration in the respective project. Teams in the *ND* column were not detected through the TCRS. The *H?* column shows the percentage of teams flagged by the TCRS by the halfway mark of the project.

	W0	W1	W2	W3	W4	W5	W6	ND	H?
F20	1	5	1	-	-	1	-	1	78%
S21	-	2	-	5	2	2	-	2	54%
Total	14/22 (63.6%)				8/22 (36.4%)				

(a) Onboarding Project

	W0	W1	W2	W3	W4	W5	ND	H?
F20	1	1	4	-	1	-	1	86%
S21	2	4	2	-	-	-	-	100%
Total	14/16 (87.5%)			2/16 (12.5%)				

(b) Team Project

RQ2: *The TCRS identifies a majority of struggling teams – 53.8% to 100%, depending on project and semester – by the halfway mark of the project.*

5.3 RQ3: Survey Impact on Team Success

In this section, we evaluate whether the TCRS has a positive impact on software engineering teams. We consider two factors: 1) does engaging with flagged teams improve their grades, and 2) do students find the TCRS useful for self-reflection or staying on track. We found that there was no improvement in students' grades, but a majority of students (64.4%) found the TCRS helpful.

As discussed in Section 4.4, we conducted an intervention in Spring 2021 where students in half of the labs received followups from the lab TA and students in the other labs did not. To understand the impact, we conducted unpaired Mann-Whitney U tests between the grades received by students in the control group and students in the experimental group, and found there was no statistically significant improvement in either end-of-project grades or peer evaluation grades ($p > .1$ for both metrics and projects).

Students found the TCRS a useful tool for self-reflection or keeping their team on track. Starting with Fall 2020, we added a question to the end-of-project reflection for the TP asking students whether the TCRS "*helped keep you and your team on track*". To complement the intervention, we read the reflections submitted by each student in Spring 2021. In total, we received 118 responses that explicitly mentioned the TCRS. Of the 118 students, 76, or 64.4%, believed that it was a useful tool for self-reflection or keeping them or their team on track. For example, students mentioned that the TCRS "*help[ed] keep our team on track*" or "*it forced us to demonstrate what we've done*", suggesting that it may be useful for getting students to reflect on their teaming experience. Of the remaining students who did not find it helpful, some said the project was going well and "*we*

rarely had any communication issues to reflect on". Others remarked that even though issues were brought up, a member of their team remained intransigent and the situation did not improve. However, most students appreciated the value of the TCRS. In Section 6.4 we discuss plans to encourage and support self-reflection.

***RQ3:** Most (64.4%) students believe the TCRS helps keep their team on track or provides a positive chance for self-reflection.*

6 DISCUSSION

6.1 TCRS Success

We observed four false negatives across our two projects and semesters: teams with observed struggle that were not flagged by the TCRS. Three of these teams were from the OBP, and two of these teams had only two students (as opposed to the typical teams of three); we posit this puts the students in a more difficult position as there is one fewer person to contribute to the team's tasks. In the future, we will make teams of three or four students. For the final team, on the TP, we read through the end-of-project reflections submitted by everyone on the team as well as their weekly TCRS responses. Issues were mentioned in the final reflection, as well as in some of the open-ended comments in the TCRS, but not the main Likert-scale questions. We have incorporated natural language processing [2] into our flagging process to alert us to these issues.

6.2 Facilitating TA Engagement with Teams

As discussed in Section 4.4, we conducted a targeted intervention in Spring 2021 to have TAs engage with struggling teams. Several times during the semester, we also checked in with the TAs to see if they were using the checklist and how the discussions with teams in lab were going. Anecdotally, the TAs mostly reported that teams said things were "fine" and were hesitant to discuss issues. It is possible that the TAs need more training in crucial conversations [21] so that they can more effectively discuss challenging dynamics with teams. Additionally, explicitly tracking when issues came up week after week could help the TAs take an increasingly hands-on approach for helping teams overcome their issues. Finally, if TAs are more forceful in reminding students that there are consequences for non-participation, it may encourage recalcitrant students to engage with team and the project.

6.3 Threats to Validity

Conclusion: In this work, we use project grades and peer evaluations as a proxy for *observed struggle* to identify teams that are in need of assistance from the teaching staff. While we can use both of these measures objectively, we have observed that they may not capture the true picture of what difficulties a team is facing. Indeed, when we read through end-of-project reflections, we found a team that received fine grades and no concerning peer evaluations, but two members still reported that they were facing issues communicating and collaborating effectively. Work remains to be done in finding an oracle of team distress that is objective and accurate.

Construct: While we observe that many students were willing to reveal issues in their teams to the teaching staff, this was not universally the case. One student reported in their final reflection: "*I did not do this [mention a struggling teammate] however, because I did not want to create tension*". Prior work suggests that women and students from historically underrepresented minorities may be less assertive [13, 20], and consequently potentially less comfortable alerting the teaching staff of perceived issues. Further work to detect team issues that can complement self-reporting is necessary.

Internal: In Fall 2020, we merely deployed the TCRS each week to get a baseline observation for how capable it is at identifying struggling teams. Consequently, we did not look at the responses until the end of the semester, and no followups were performed based on them. Several students remarked in their end-of-project reflections that they wished there were consequences for issues identified, or that there had been more prompt followup. It is possible that students, frustrated that they were not getting any followup, started taking the TCRS less seriously as the semester progressed.

External: This study was conducted in the context of one course at one university over two semesters. While we received promising results, replication needs to be performed to validate the use of a survey for flagging teams and promoting self-reflection.

6.4 Future Work

We have identified several promising avenues for future work.

Students mentioned in their end-of-project reflections that they found the TCRS useful for weekly self-reflection and staying on top of tasks that needed to be completed (Section 5.3). We intend to probe this further, and conduct follow-up interviews with students at the end of a future semester to understand the TCRS' use as a self-reflection tool and how to further improve it.

As mentioned in Section 6.3, a fundamental limitation of the TCRS is that it requires students to be willing to share the issues they believe their team is facing with the teaching staff. While our results suggest many students are willing to do so, this places a burden on students that may be particularly unwelcome for women and students from underrepresented groups. Consequently, future work that focuses on identifying successful and unsuccessful patterns from version control systems can give early warning signs of team struggle in a way that does not require students to self-report the issues. We intend to tackle this next as a way to complement the TCRS for detecting and overcoming team struggle.

7 CONCLUSION

In this work, we have designed a weekly reflection survey for identifying struggling teams in undergraduate software engineering courses. By matching survey results against project grades, we have found that the survey can flag teams with observed struggle in most cases (with an overall success rate of 89.5% across two projects and two semesters), and typically can do so early enough in the project that the course teaching staff may be able to intervene and help the team perform better. We devised an intervention to try and foster discussion in struggling teams and identify a plan for overcoming their collaborative difficulties. Our intervention did not result in any grade improvements, yet most (64.4%) students nonetheless reported that the surveys helped keep them on track and provided a chance for weekly self-reflection. We are planning improvements to the survey and the course in light of these findings to offer better support for teams.

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 for allowing us to use their data for analysis.

REFERENCES

[1] 2020. Criteria for Accrediting Engineering Programs, 2021 – 2022. https://www.abet.org/accreditation/accreditation-criteria/criteria-for-accrediting-computing-programs-2021-2022/

[2] 2021. NLTK: Sample usage for sentiment. https://www.nltk.org/howto/sentiment.html

[3] Neda Abbasi, Anthony Mills, and Richard Tucker. 2017. *Conflict Resolution in Student Teams: An Exploration in the Context of Design Education.* https://doi.org/10.4018/978-1-5225-0726-0.ch005

[4] Chie Adachi, Joanna Hong-Meng Tai, and Phillip Dawson. 2018. Academics' perceptions of the benefits and challenges of self and peer assessment in higher education. *Assessment & Evaluation in Higher Education* 43, 2 (2018), 294–306. https://doi.org/10.1080/02602938.2017.1339775 arXiv:https://doi.org/10.1080/02602938.2017.1339775

[5] Stephane Brutus and Magda B. L. Donia. 2010. Improving the Effectiveness of Students in Groups With a Centralized Peer Evaluation System. *Academy of Management Learning & Education* 9, 4 (2010), 652–662. http://www.jstor.org/stable/25782052

[6] Jane Burdett. 2003. Making Groups Work: University Students' Perceptions.

[7] Pearl Chen, Anthony Hernandez, and Jane Dong. 2015. Impact of Collaborative Project-Based Learning on Self-Efficacy of Urban Minority Students in Engineering. *Journal of Urban Learning, Teaching, and Research* 11 (2015), 26–39.

[8] 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 arXiv:https://doi.org/10.1080/02602938.2013.867477

[9] Siva Dorairaj, James Noble, and Petra Malik. 2012. Understanding Team Dynamics in Distributed Agile Software Development. In *Agile Processes in Software Engineering and Extreme Programming*, Claes Wohlin (Ed.). Springer Berlin Heidelberg, Berlin, Heidelberg, 47–61.

[10] Jan Fermelis and Richard Tucker. 2008. Self and Peer Assessment: Development of an online tool for team assignments in business communication and architecture. (01 2008).

[11] 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

[12] 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

[13] Campbell Leaper and Rachael D. Robnett. 2011. Women Are More Likely Than Men to Use Tentative Language, Aren't They? A Meta-Analysis Testing for Gender Differences and Moderators. *Psychology of Women Quarterly* 35, 1 (2011), 129–142. https://doi.org/10.1177/0361684310392728 arXiv:https://doi.org/10.1177/0361684310392728

[14] 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

[15] Santiago Mendo-Lázaro, María I. Polo-del Río, Damián Iglesias-Gallego, Elena Felipe-Castaño, and Benito León-del Barco. 2017. Construction and Validation of a Measurement Instrument for Attitudes towards Teamwork. *Frontiers in Psychology* 8 (2017), 1009. https://doi.org/10.3389/fpsyg.2017.01009

[16] Micah Gideon Modell. 2013. Iterating Alone over a Method and Tool to Facilitate Equitable Assessment of Group Work. *International Journal of Designs for Learning* 4, 1 (Jun. 2013). https://doi.org/10.14434/ijdl.v4i1.3283

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

[18] Jan P. Owens. 2015. Student Satisfaction with Group Work: Perceptions and Attitudes. In *Proceedings of the 2007 Academy of Marketing Science (AMS) Annual Conference*, Dheeraj Sharma and Shaheen Borna (Eds.). Springer International Publishing, Cham, 67–73.

[19] 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

[20] James B. Parham, Carmen C. Lewis, Cherie E. Fretwell, John G. Irwin, and Martie R. Schrimsher. 2015. Influences on assertiveness: gender, national culture, and ethnicity. *Journal of Management Development* 34, 4 (01 Jan 2015), 421–439. https://doi.org/10.1108/JMD-09-2013-0113

[21] 1946 Patterson, Kerry and ProQuest (Firm). 2012. *Crucial conversations : tools for talking when stakes are high.* New York ; London : McGraw-Hill, [2012], New York.

[22] Beatriz Pérez and Ángel L. Rubio. 2020. A Project-Based Learning Approach for Enhancing Learning Skills and Motivation in Software Engineering. 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, 309–315. https://doi.org/10.1145/3328778.3366891

[23] Elizabeth Pfaff and Patricia Huddleston. 2003. Does It Matter if I Hate Teamwork? What Impacts Student Attitudes toward Teamwork. *Journal of Marketing Education - J Market Educ* 25 (04 2003), 37–45. https://doi.org/10.1177/0273475302250571

[24] 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

[25] 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

[26] Chris Roberts, Christine Jorm, Stacey Gentilcore, and Jim Crossley. 2017. Peer assessment of professional behaviours in problem-based learning groups. *Medical Education* 51, 4 (2017), 390–400. https://doi.org/10.1111/medu.13151 arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1111/medu.13151

[27] Aleksandra Rudawska. 2017. Students' Team Project Experiences and Their Attitudes Towards Teamwork. *Journal of Management and Business Administration. Central Europe* 25 (01 2017), 78–97. https://doi.org/10.7206/jmba.ce.2450-7814.190

[28] 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

[29] Jirarat Sitthiworachart and Mike Joy. 2004. Effective Peer Assessment for Learning Computer Programming. In *Proceedings of the 9th Annual SIGCSE Conference on Innovation and Technology in Computer Science Education* (Leeds, United Kingdom) *(ITiCSE '04)*. Association for Computing Machinery, New York, NY, USA, 122–126. https://doi.org/10.1145/1007996.1008030

[30] 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

[31] 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

[32] 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