Investigating Student Insight in Software Engineering Team Projects

Simona Motogna[a], Dan Mircea Suciu[b] and Arthur-Jozsef Molnar[c]
Department of Computer Science, Babeş Bolyai University, M. Kogalniceanu 1, Cluj Napoca, Romania

Keywords: Software Engineering Education, Soft Skills, Teamwork, Project, Empirical Study.

Abstract: Preparing software engineering students for industry jobs is a difficult task. Employers often ask for a combination of technical and soft skills. In this paper, we describe an exploratory study focused on 47 teams of 3^{rd} year Computer Science students who were each involved in the development of a medium sized software application. We employed open coding on free-text student feedback and focused on key areas of working process, planning, challenges and lessons learned. We evaluated how student perception aligned with feedback received from mentors with industry experience, as well as with the assessment from teaching staff experienced in software engineering. We open-sourced a replication package that allows reusing, repeating or extending our investigation. We showed that the course objectives were reached, that soft skills remain an important component of team projects as well as identified several action points that can further improve education in software engineering.

1 INTRODUCTION

There exists a general demand from the industry that students should possess professional skills when graduating from Universities, and in consequence, there exists a constant demand on university curricula to try to accommodate such skills in the learning objectives of their courses. On one hand we discuss about technical skills that are needed to develop real life, large and complex software systems, and on the other hand about soft skills, from which teamwork is the essential one in the software engineering (SE) domain (Ahmed et al., 2013). As a consequence, project-based coursework during which students have the opportunity to acquire such professional skills are becoming a must in undergraduate curricula.

This paper describes an exploratory study resulting from the teaching experience at the Babeş-Bolyai University in implementing a project-based course with three particular characteristics: real life projects, industry mentors and soft skills workshops. The course was designed for third year students. They are expected to already have a strong technical background in different programming languages, frameworks, development environments and methodologies. To a certain degree however, they lack the experience of contributing to large scale projects and that of longer-term collaborative work. Such courses are good examples in which a strong cooperation between universities and local companies can bring significant improvements, as they provide students with valuable work experience.

In our study we carry out a detailed investigation of the degree to which students acquired technical abilities and soft skills by assessing the results of their evaluation, the feedback received from mentors working in the industry and survey responses from the students themselves. We combined external project evaluation, mentor evaluation and opinions gathered at the end of the semester with an in-depth self assessment of student teams regarding their challenges and lessons learned with regards to professional skills.

The main goal of the study is to obtain a deeper understanding of the student perception on the activities and learned abilities. The exploratory observations conducted enabled us to draw qualitative conclusions. We set up and carried out our exploratory study according to existing best practices (Runeson and Höst, 2009), and reflect this in the structure of our paper. We published the raw data to enable replicating or extending our study (Motogna et al., 2021). Section 2 provides the context in which our study was carried out. We present the main objective of our approach in Section 4, describe the data collection process within Section 5 and study results in Section 7. Threats to validity were identified and briefly detailed

[a] https://orcid.org/0000-0002-8208-6949
[b] https://orcid.org/0000-0002-5958-419X
[c] https://orcid.org/0000-0002-4113-2953

Motogna, S., Suciu, D. and Molnar, A.
Investigating Student Insight in Software Engineering Team Projects.
DOI: 10.5220/0010478803620371
In *Proceedings of the 16th International Conference on Evaluation of Novel Approaches to Software Engineering (ENASE 2021)*, pages 362-371
ISBN: 978-989-758-508-1

in Section 6, while the final section was reserved for conclusions and an overview of future work planned.

2 KEY ELEMENTS OF SE PROJECTS

As part of the third year *Team Project* course, students in *Computer Science* and *Mathematics and Computer Science* educational tracks are required to work in teams of 8-12 students and develop a software application of their choice, during the 14-week first semester taking place from October to January.

Starting from 2017, with the purpose to foster collaboration between team members and to improve the quality of the delivered product, we integrated experienced mentors from the local software industry. Each team is assigned a mentor who helps them organize, collaborate and follow the development methodology selected by the students at project onset. Whenever it is needed, the mentor also plays the role of the customer, in order to give the team a better perspective of how a final user would see their solution. Usually, each mentor is assigned to coordinate two or three teams of students, from a total between 46 - 50 teams. The exact number of teams depends on student enrollment and organization.

Examples of software application themes usually selected for implementation include internship management tools, learning management systems, educational quizzes, e-commerce solutions, and systems for the management of personal finances.

In parallel with their activities regarding project development, in order to improve soft skills, enrolled students attended four different half-day workshops where communication, presentation skills, Agile-driven project management and entrepreneurship were discussed.

At the end of the 14-week semester, each team presents the result of their work in a 15-minute live demonstration session. This is carried out in front of a faculty committee composed of professors specialized in software engineering, who assign a grade to each of the teams. Within each team, student grades are decided by the mentors, based on each team member's contribution and such that the average of all team member's grade fits the grade assigned to the software application developed by them.

The distinctive characteristics of the course are:

Project based: the project to be developed represents the kernel of the course and also the main evaluation criterion.

Real-life: the project solves a real-life problem, and real-life project management is simulated as the team is supervised by a mentor, who is an experienced professional from partner software companies.

Teamwork: students organize the team based on their choice and work in teams for the whole period of the semester.

Evidence based: all performed activities correspond to set milestones and must be documented.

Freedom of Choice: students are encouraged and have to decide both the software methodology and the technology stack, including programming languages, application architecture, employed frameworks and tools for project management and version control.

Learning objectives are focused on practical competencies acquired by students, which include the knowledge and skills necessary to implement and follow through a software project management process, adapting the life cycle of a software project to an Agile context and improving team communication and collaboration skills.

The data used in this study corresponds to academic year 2019-2020, when 492 students, forming 47 teams and 15 mentors were involved. The semester was unaffected by the subsequent pandemic-related restrictions.

3 RELATED WORK

A systematic literature review of papers presenting experiences with student global SE projects resulted in identification of overall challenges of such courses and some recommendations to solve them. One important conclusion that has been addressed by our approach is that ”integrating cultural training, conducting teamwork exercises to build trust, and instructor monitoring of team communication are all examples of techniques that have been used successfully by educators according to our review” (Clear et al., 2015).

A comprehensive study on student perception about group projects (Iacob and Faily, 2019) provided answers to three important research questions related to student expectations about teamwork, issues related to the actual process of project development and finally whether results matched the expectations. Other results worth mentioning targeted collaborative activities of students (Troussas et al., 2020) and analysis of members activities within team projects (Parizi et al., 2018).

There have been several contributions reporting experiences with team projects and capstone projects. They include (Bastarrica et al., 2017), who investigated a capstone course focusing on soft skills and agile practices, and (Holmes et al., 2018), who used open coding focusing on projects, tools and mentors.

Even if several strategies targeting capstone projects may be successfully applied to other project based courses, there exist different learning objectives and outcomes for capstones, respectively team projects.

Team projects have also been the object of study for other contributions, such as (Raibulet and Fontana, 2018), (Delgado et al., 2017) reporting experiences at different universities: University of Milano - Bicocca , respectively University Nacional of Colombia, where student project curricula evolved over the years based on collected feedback.

As an experimental study that relies on student perception, we believe our work can contribute to the overall understanding of such courses. The distinctive characteristic of our approach lies in the cross sectional analysis of student perception using two independent assessments (teaching staff evaluation and mentor feedback) together with the open coding interpretation of student survey responses. The differential feature of our course lies in including the workshop activities for developing soft skills.

4 OBJECTIVES

We structured the steps of our study according to current best practices (Runeson and Höst, 2009) and formulated the main objective of our work using the goal question metric approach (Solingen Rini and Rombach, 2002) as a "*qualitative investigation on the student perception regarding the challenges and learning outcomes of collaborative project work*". We operationalized our main objective in the form of the following four perspectives:

Q_1**:** *How did students perceive project planning and the working process?* Focus is set on two crucial activities taking place within the initial phase of software projects. Research and practice have confirmed the importance of adequate project planning, which remains rife with issues known both from anecdotal and case report research (Serrador, 2012; Posten, 1985; Zwikael and Globerson, 2006). Next, we aimed to investigate the working process as a separate concept, as for many students it was the first opportunity to experiment a collaborative environment.

Q_2**:** *What were the students' main challenges?* We investigated both technical and soft skills-related challenges, as they were perceived by the students themselves. In answering this question we attempted to identify the existence of any common technological challenges and to characterize their occurrence and severity. We also aimed to investigate the degree to which students could identify and pinpoint non-technical challenges that we grouped under the general umbrella of soft skills. Existing literature has already shown their role (Fowler, 2020) and importance in the workplace (Ahmed et al., 2013), so we believe that more accurate student perception of these issues can create a long-term industry impact.

Q_3**:** *What were the main lessons learned from the students' perspective?* We analyzed student feedback provided after the group projects were completed and evaluated. We drew parallels with the answers for Q_1 and Q_2 and examined the relation between working process, challenges and lessons learned.

Q_4**:** *To what extent did student perception match the overall assessment of their project?* We placed findings from student feedback into the context provided by the assessment from the teaching staff committee, which was made up of professors with relevant experience in software engineering and product development.

5 METHODOLOGY

Data collection was carried out as part of the proposed learning activities of the *Team Project* coursework. Once technical project work was completed, each team was asked to respond to a survey (one response per team) consisting of a questionnaire as part of which they provided the team name, a short summary of the project topic and a detailed description of their perspective during and after project completion. All fields were free text and students were encouraged to provide detailed descriptions. Mentor feedback was solicited in the form of a separate questionnaire. We published the questionnaires, together with student and mentor responses in the form of a data replication package (Motogna et al., 2021). Finally, projects were evaluated, with teaching staff providing team grades that were refined by mentors according to individual participant effort and involvement.

5.1 Student Surveys

At project completion, teams were asked to respond to a survey consisting of four questions: 1) What was the working process? 2) How was the project planned? 3) What challenges did you encounter? and 4) What were the lessons learned during this project?

We decided to apply Grounded theory (Glaser and Strauss, 1967) as a qualitative analysis method, respectively open coding (Corbin and Strauss, 2014) for determining conceptual codes collected from the data itself, and then merging them in more general categories. Such a method is recommended in case of surveys in which the same questions are asked and

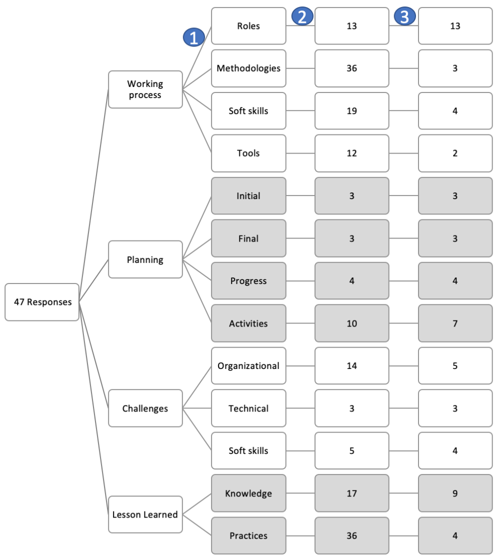

Figure 1: Open coding process: the steps of the process appear as rounded indexes, and the numbers in rectangles represent the count of initial topics, respectively merged topics.

the responses are collected in the form of free text.

Figure 1 describes the steps of the coding process. The 47 enrolled teams provided answers in the form of free text structured according to the above mentioned perspectives. For each category, two independent coders performed the initial step of open coding (Step 1 from Figure 1) during which they established the labels and then populated them with topics (Step 2). The third independent coder proposed the merging process that was later debated and mutually agreed upon during discussion by all three coders (Step 3). This resulted in the list of topics to be analyzed for each perspective. The numbers in the figure represent the count of initial topics (result of Step 2), then the number of merged topics (result of Step 3). The following tables present in detail the labels and topics accompanied by suggestive quotes for each perspective. For each label we introduced an abbreviation, in order to associate the team answer to the questionnaire rubric. As an example, answer T12.WR represents the answer given by team 12 (indices assigned randomly for anonymization) to the Working Process section, labeled with Roles.

5.2 Mentor Feedback

Professionals fulfilling the mentor role were solicited to fill in a questionnaire at the end of the course, after the student evaluation phase. The questionnaire was focused on student performance, typical issues encountered in student team projects as well as main lessons learned (Iacob and Faily, 2019).

5.3 Project Evaluation

Each software application was assigned a grade between 1 and 10, based on the evaluation of the implementation documentation and a 15-minute presentation that included a live demonstration. The evaluation was done considering the following five evaluation criteria and deliverables: application design complexity and suitability, degree of the main functionalities coverage, quality of user experience (intuitive, consistent and uniform), code structure and documentation.

6 THREATS TO VALIDITY

We organized our exploratory study according to existing best practices (Runeson and Höst, 2009). Questionnaires were set up and sent to respondents after course completion, as previously described. One of the present paper's authors was directly involved within the course, while all three have previous expertise in software engineering, both from an academic as well as industrial perspective. The major steps carried out for the study regarded processing student responses using open coding, analyzing the results and carrying out the cross sectional analysis.

Internal Threats. Were addressed by involving researchers who are both experienced teachers as well as practitioners of software development. Questionnaires were designed and validated within the team. We considered the *open coding* phase to be the most prone one to threats, so work was done in pairs, with the third person having a validation role, and conducting a final agreement overview.

While our study's findings are in line with existing literature (Ville and Daniels, 2019; Delgado et al., 2017; Ahmed et al., 2013), we are aware that circumstances regarding course curricula, team size and organization, as well as means and moment of reporting can all act as **external threats**. In order to address them, we used previous results from literature to establish questionnaire structure and contents, as well as the four key areas that represent the focus of our research. In addition, one of the study authors had no previous involvement with the course structure, in order to control bias (Ville and Daniels, 2019).

Furthermore, we focused on carrying out a qualitative evaluation, where open coding and cross sectional analysis in particular were employed in order to strike a balance between eliminating questionnaire bias and capturing the essence of issues reported by students.

Table 1: Labels for "Planning".

Label	Topics	Examples
Initial planning (was followed/how much) (label: PI)	slightly followed, somewhat followed, almost followed	*"we covered a proportion of 85% from the initial plan", "Initial planning was followed only by a third of the team"*
Final result (% of final implementation (label: PF)	good (over 75%), medium (≈ 50%), weak	*" we achieved in the end what we had intended at the beginning", "some features were not fully implemented"*
Process progress (label: PP)	slow start accelerated finish, significant adjustments, continuous process, good start slow finish	*"the larger chunk of the work was done during the very last weeks of the time that was left", "we adjusted the initial planning and managed to reach a realistic plan"*
Planning related activities (label: PA)	technology choice, task oriented, role oriented, sprint oriented, feature oriented, component oriented, adapt	*"we quickly decided upon the technologies", we created and choose the tasks each of us would do", "planning was carried out according to the role of the actors"*

7 RESULTS AND DISCUSSION

Q_1: *How did students perceive project planning and the working process?* The planning section of the survey intended to assess how students constructed the initial plan, what means they used to follow the plan, which activities were dedicated to following the plan and in the end, to have an overview of the deviation from the initial plan. Given that students used free text for responses, we realized when interpreting this section of the survey that almost all teams considered important to state to what extent they followed the initial plan, respectively in the end to what extent their implementation was complete. We found that only some of the teams described their progress (17 teams) or which planning activities they adopted (33 teams), as shown in Table 1.

Process Progress. The onward evolution of the process was reported by 17 teams, and the open coding processing of the responses identified four categories: 10 teams increased their working pace when the deadline was approaching (*"we started working on the project a bit later than what we expected"*[1]*, T27.PP)*, four teams considered that the development pace was constant (*"the process has been smooth, with the tasks being executed in a good enough time", T45.PP*), two teams made significant modifications to initial plans (*"the initial planning was changed a lot, we made a lot of refactoring", T9.PP)* and one team decreased the development rate near the end of the project: *"the rate has decreased ongoing, deadlines have not been respected any more", T38.PP*.

Planning Related Activities. As revealed by student responses, in most of the cases activities were oriented on sprints, tasks or features/functionalities. A number of 11 teams used sprint-driven planning (*"we aimed to have at the end of each sprint a working version of the software", T28.PA, "At the beginning of the sprints we had planning sessions", T18.PA*), seven teams made task oriented planning (*"we created and choose the tasks each of us would have to do"*, T5.PA), while six teams had feature oriented planning: *"Though we didn't succeed in implementing all the features we talked about at the beginning", T32.PA*.

Table 2 summarizes the labels, topics and examples corresponding to the working process perspective. Four main concepts were inferred as a result of applying open coding in this part of the survey: the roles assigned to team members, the applied development methodology, soft skills as part of the working process and tools that were used by the teams.

Roles. 26 teams stated that they divided the team into backend and frontend subteams (*"the tasks were divided... into backend and frontend tasks", T4.WR, "to allow parallel development on backend and frontend", T38.WR*), 10 teams had an assigned role for testing (*"members responsible with testing"*, T30.WR), while six teams acknowledged the role of the Scrum master: *"each subteam had its own Scrum master", T36.WR*. A set of other roles have been mentioned, either functional or non-functional (such as IT support, documentation, team leader).

Methodologies. Agile methodologies were adopted by most teams, as 15 teams mentioned Agile, 10 others mentioned Scrum, respectively 5 mentioned Kanban, and 2 teams used feature driven development. A careful investigation revealed that in some cases, even

[1]Text in italics represents unmodified snippets from student responses.

Table 2: Labels for "Working process".

Label	Topics	Examples
Roles (identified roles in the team) (label: WR)	backend/frontend, project manager, product owner, requirements engineer, devops, testing, IT support, technical leader, documentation, reviewer, team leader, scrum master, AI	*"X took responsibility for the backend", "groups that worked on different tasks: mobile application, backend development, testing, documentation."*
Methodologies (label: WM)	*Agile*, including: Scrum, Kanban, Feature driven development (with several related activities: sprints, stand-ups, voting, planning poker, user stories), *upfront design, task management*	*"Apply AGILE methodology technique (Scrum board)", "we split our available time for development in sprints", "We went with a Kanban way"*
Soft skills (label: WS)	communication, collaboration, mentoring, learning	*"weekly meetings", "Provide a proper feedback", "co-location", "we allocated time to learn new technologies"*
Tools (label: WT)	source control tools and task management tools	*"Github", "GitLab", "BitBucket", "Discord"*

Table 3: Labels for "Challenges".

Label	Topics	Examples
Organizational (label: CO)	time management, task management, teamwork, collaboration, technology choice	*"synchronizing as a team and working together", "decide which tools and frameworks to use", "finish everything before the deadline"*
Technical (label: CT)	technical skills, effective collaboration, over-engineering	*"learn to use the technologies", "how to combine what we worked on", "over-engineering in some places"*
Soft skills (label: CS)	involvement, communication, collaboration, teamwork	*"Being able to motivate everyone to work consciously and constantly", "it was way harder for everyone to keep in touch with everyone ", "organizing and coordinating"*

if the students declared to have an agile approach, their responses actually described a waterfall methodology. Upfront design was adopted by four teams: *"the first couple of weeks was mostly dedicated to the engineering of the layers and architecture"*, T5.WM.

Soft Skills. Given that teamwork was one of the learning objectives of the course and as students benefited from two workshops targeting soft skills, we found it reassuring that the students included soft skill aspects in their working process. As desired, communication-related terms appeared most frequently, in 17 responses, such as: meetings (*"we planned weekly meetings", T15.WS*), co-location (*"we strive to find a common space to serve as office", T10.WS*, negociation (*"tried to do what was best for us as a team, which basically meant we compromised a lot", T2.WS*) or feedback (*"Provide a proper feedback to every teammate",T8.WS*).

We appreciated the desire of teams to learn, as training was labeled in seven team answers: *"first ticket was associated to different learning activities", T17.WS*, and mentoring (*"We scheduled workshops, where some of our more experienced colleagues held training sessions for the rest of the team", T14.WS*). One team was identified as having dysfunctional teamwork: *"I'll tried... but the team mates mostly ignored me", T43.WS*.

We also identified a number of 8 teams that exposed self-organizing procedures in their working process: *"brainstorming", "efficient organization", "interact", T37.WS*.

Tools. Teams used several tools as part of project implementation. They can be classified in tools for source control (such as Github, GitLab, Bitbucket) and task management (Trello and Discord).

Conclusion for Q1: Students acquired experience in project planning, processes and teamwork as a soft skill
Action Point: Improve constant working

$\boldsymbol{Q_2}$**:***What were the students' main challenges?* The challenges that students faced and tried to overcome can be grouped into three main types: organizational, technical and related to soft skills, as shown in Table 3. In order to have the full picture, we incorporated mentor feedback, illustrated in Table 4 into our analysis.

Organizational Challenges. 16 teams considered task management to be the hardest challenge (*"Challenge: create and detail tasks to be easily understood", T2.CO*), respectively time management (*"one challenge was intense work in last days to finish*

Table 4: Mentor feedback on 10 common issues in team projects. Scores recorded on a 1 (very poor) to 5 (excellent) scale, including standard deviation (σ).

Challenge	Mean	σ
Lack of engagement from team members	3.62	1.30
Student task knowledge and assumption	3.50	0.53
Miscommunication within the team	3.25	1.03
Problems with time management	3.25	0.88
Team leader dictates what team members do	2.87	1.24
Lack of technical expertise	2.87	1.12
Dividing the work among team members	2.50	1.41
Work is carried out by one or two team members	2.37	1.3
Conflicts among team members	2.37	1.4
No one wants to act as team leader	2.12	1.45

project", T10.CO). Summarizing, either tasks or time have put pressure on project development for approximately 68% of the teams. Eight teams faced technology choice as the main challenge: *"main issues was to decide which tools and frameworks to use", T14.CO*.

Technical Challenges. More than half of the teams (53%) dealt with technical problems, in some part due to lack of technical knowledge (*"a more messy challenge was learning new frameworks and libraries on the go", T2.CT*), or due to incorrect project management (*"Our challenge arose when, after starting our implementation, we frequently needed to change the DB schema and relations, thus indirectly forcing everyone to do the same updates"*, T28.CT).

Soft Skills Challenges. The extent of each team member's involvement was the main problem for 10 of the teams: *"Being able to motivate everyone to work consciously and constantly", T5.CS*. Five teams were confronted with communication problems: *"biggest challenges that we encountered were communication with the team members, ... and the communication with the mentor", T18.CS*.

One team declared that they had no challenges worth mentioning, and that the ones that appeared were easily solved, while two other teams (T12, T24) did not complete this part of the survey.

Mentor Feedback. Table 4 shows the mentor's answers regarding some of the most common challenges in the team projects (Iacob and Faily, 2019). The most significant highlighted challenge was the lack of engagement from team members, for which mentors reported a mean score of 3.62, with 3 out of 8 respondents attributing it the maximum value on a 5-point scale. Another significant issue regarded communication and the assumption of project roles and tasks within teams. We note the low value of the standard deviation (σ=0.53), which shows agreement among the mentors. Mentor feedback was mostly positive regarding the existence of team conflicts, or for the presumed case of students not wanting to assume coordination responsibilities. In most cases, we observe a value of $\sigma \geq 1$, which shows that the prevalence of issues still varied among the teams.

We were also interested in a cross-sectional exploration of challenges as perceived by students versus mentors. In this regard, our answer to Q_2 shows alignment between student and mentor perception. On one hand, challenges were an important motivator of progress; we believe that agreement between student and mentor perceptions represents an important indicator of effective communication and team progress.

Conclusion for Q2: Students learned project management and improved technical skills
Action Point: Address lack of involvement

Q_3: *What were the main lessons learned from the students' perspective?* Student responses can be grouped into what they have learned, in the sense that what knowledge they have accumulated while working on the project, respectively "lessons learned" in terms of experiences and practices that they realized are important in project development, as described in Table 5. Like in the case of Q_2, we employed mentor feedback included within Table 4 to complete the perspective provided through student feedback.

Acquired Knowledge. Teamwork was mentioned by the largest number of teams (12), and this must be appreciated as it was one of established learning outcomes, and also the subject of one workshop. The teams identified several aspects of teamwork to be equally important, such as *"we all agreed that in the end we learned how to be a team"(T16.LA)*, *"teamwork makes the dream work"(T33.LA)*, or in even more detail: *"we saw how to work in a team, similar to a working place (responsibilities, targets, deadlines)"(T9.LA)*.

Technical skills were mentioned as acquired by a total of 9 teams: *"enhancing technical skills through knowledge sharing"(T24.LA)*, or *"working with technologies we have never used before and doing it successfully"(T32.LA)*. Six teams considered communication as the principal gain of the project: *"The project represented a valuable exercise in improving communication"(T45.LA)*, or *"we communicated a lot throughout all the development stages"(T32.LA)*.

Table 5: Labels for ”Lessons learned”.

Label	Topics	Examples
Acquired knowledge (label: LK)	teamwork, technical skills, communication, estimation, collaboration, code review, SCRUM, time management, version control system, task management	*”We learned ... teamwork and the ability to support and help each other”, ”We have learned to make constructive reviews and learn from the feedback we receive”, ”We learned mostly about the efficiency of the SCRUM methodology, about teamwork and time management”*
Good practices (label: LP)	communication, motivation, teamwork, project management	*”Communication is Key”, ”we have learnt is how hard it is to motivate people, and to manage a team of 9 people”, ”task management is crucial”*

The insights also revealed special attention dedicated to learning (*”learn by listen to team mates opinions”, T6.LA*), review (*”We have learned to make constructive reviews and learn from the feedback we receive”, T11.LA*) or adaptation (*”adapt to others' needs”, T15.LA, ”adapt to several ways of problem solving”, T38.LA*).

Good Practices. 16 teams considered that communication was the most important practice in the team project: *”Communication is Key - is what probably most of us discovered more and more during our collaboration”, T2.LP*, *”The most important lesson is that communication is essential in order to get things done”, T14.LP*. Six teams gave equal importance to practices related to motivation: (*”The most important lesson that we have learnt is how hard it is to motivate people”, T18.LP*) and task management: (*”During the implementation of our software solution, we were taught that ... task management is crucial”, T27.LP*).

We believe it is important to perform a *cross sectional analysis* in order to identify and characterize the relation between lessons learned, working process and challenges.

Lessons Learned vs. Working Process. Different correspondences were tried, and even while some of them did not result in a significant dependency between terms from the two sections of the survey, we have identified two important relations that can be summarized in the following:

- Out of the 16 teams that considered ”communication” as a good practice, 10 identified and used roles during their working process (label Roles in Table 2), which led us to the conclusion that roles enabled better communication between team members;
- Among the 16 teams for which ”communication is key” was mentioned as a good practice, only 3 mentioned communication activities in the description of the working process (label Soft skill in Table 2). So, in most cases the teams realized that they did not communicate efficiently, and that they would have performed better if they would have included communication activities in the process;

Lessons learned vs. Challenges. the cross sectional analysis performed on these two survey sections may identify whether students were able to overcome and find solutions to their challenges in terms of acquired knowledge and/or lessons learned. For the organizational challenges, the following dependencies have been observed:

- Out of the 16 teams for which ”task management” was the organizational challenge (see Table 3, label: Organizational) 8 stated ”communication is key”, 4 mentioned motivation, 3 stated collaboration, and 2 mentioned teamwork as good practice;
- Considering the 16 teams challenged by ”time managment”, 4 highlighted the importance of planning as part of project management;
- Teamwork was mentioned as a good practice by 13 teams, and from these only three have considered team related activities as challenges.

As 25 teams stated that the lack of technical skills represented their main technical challenge (label Technical in Table 3), we examined if progress was made by the end of the project: namely, seven of these teams (28%) acquired technical skills, and another nine teams (36%) formulated different good practices related to technical abilities. Summarizing, 64% of teams have extended their grasp of technical knowledge and practices.

A closer analysis of the challenges related to soft skills (label Soft skills in Table 3), revealed a number of 10 teams that dealt with ”lack of involvement” in the project. Three of these teams mentioned the good practice of ”difficult to motivate” (out of a total of six teams), meaning that only half of the teams managed to overcome this challenge. It made us conclude that keeping students motivated and conserving involvement of all members in the team remains an important challenge and one that must be considered for improvement.

Table 6: Mentor feedback of team progress in four key areas. Scores recorded on a 1 (very poor) to 5 (excellent) scale, including standard deviation (σ).

Progress Area	Mean Score	σ
Project Management	3.43	0.82
Software Development	4	0.75
Communication	3.87	0.99
Team work	3.87	0.99

Another observation was concluded on the 5 teams that nominated "communication" as a challenge, and four of them considered in the end that "communication" is a good practice, which shows that the objectives related to this soft skill have been reached to a great extent.

Mentor Feedback. As presented in Table 6, we summarize the mentor's feedback regarding key areas of improvement. In all four areas, we notice that the standard deviation $\sigma \approx 1$, illustrating differences within the mentors' own perceptions. Mentors reported progress in all areas, and we note that none of them reported the lowest possible score of 1 for any of the areas, which shows that mentor assessment pointed toward progress taking place.

We also noted that student and mentor perceptions were well aligned. Since we made the same observation in our analysis regarding challenges as part of Q_2, we can conclude that the student perception was validated using mentor feedback.

Conclusion for Q3: Main objectives of the course are reached: technical and soft skills;
Action Point: Most important skill is communication

Q_4**:** *To what extent did student perception match the overall assessment of their project?* All 47 student teams succeeded to develop a software product which was evaluated at the end of the 14-week semester.

We found that from all teams that addressed the extent of following the initial plan, one third declared that they applied consistent changes to the initial plan, another third made smaller changes while the last third followed the initial plan almost completely (label Initial Planning in Table 1).

We compared the students' assertions with the grades assigned to their projects by the teaching staff, and found a moderate positive correlation of 0.45. This could be related to the fact that in the beginning most teams were optimistic about their capacity to deliver a complex software product. Anyway, even if most of them had to adjust their plans, this did not necessarily affect the requested quality or complexity.

A number of 23 teams have made some conclusions about the final stage of their software product (label Final result in Table 1). Our open coding procedure identified three categories: we considered a 75% or over completion declared by 16 teams, around 50% by four teams, and three teams recognized as having weak results, in the sense that the application was functional but only a few features were implemented. We could not identify a relation between following the initial plan and the final result.

Regarding the students' perception on the final result and its relationship with the way teaching staff evaluated it, we found a strong positive correlation of 0.80. This result could be the consequence of the teams' good self-evaluation skills as well as good relative estimation skills, understanding where their project was placed in comparison with other projects. Nevertheless, a pessimistic perspective on the quality of their own product could have a negative impact on the live demonstration and could influence the evaluation.

Conclusion for Q4: All students completed the project
Action Point: While initial planning was not indicative of project success, students were able to correctly assess the result of their work.

8 CONCLUSIONS AND FUTURE WORK

The goal of this paper was to provide a better understanding of the students perception on the activities and learned abilities needed to develop a software product. We analyzed different outcomes such as student surveys, mentor feedback and the result of the final evaluation; we employed a variety of empirical methods such as open coding, cross sectional analysis and statistical evidence, and we assessed both technical abilities and soft skills acquired during the studied coursework. The cross analysis performed on challenges encountered by student teams and lessons learned by team members revealed that in most cases, students were able to overcome both technical and soft skills related challenges.

We designed our report in the form of four questions, and for each of them, our analysis has revealed a conclusion related to course objectives, and action points that represent aspects that can be improved in the future. Summarizing the four conclusions of our objectives, we can state that the course outcomes have been achieved, as all student teams managed to deliver a working product while acquiring essential professional skills; organized workshops were found to have a positive impact on soft skills.

The action points revealed that the teaching staff should search for incentives to address continuous participation and motivation of all students in the teams, should supervise project planning more thoroughly and should encourage constant and efficient communication.

For future work, we intend to extend our investigation to cover several consecutive academic years for a more accurate overview of the Team Project course evolution and to gain a deeper understanding of student perception on their overall performance. Furthermore, since the 2020-2021 academic year the *Team Project* course took place fully online, a comparative evaluation between onsite and online approaches is of interest in order to identify the main changes in student performance and perception.

REFERENCES

Ahmed, F., Capretz, L. F., Bouktif, S., and Campbell, P. (2013). Soft Skills and Software Development: A Reflection from Software Industry. *Journal of Information Processing and Management*, 4(3):171–191.

Bastarrica, M., Perovich, D., and Samary, M. (2017). What can students get from a software engineering capstone course? In *IEEE/ACM 39th ICSE-SEET*, pages 137–145.

Clear, T., Beecham, S., Barr, J., and o. (2015). Challenges and recommendations for the design and conduct of global software engineering courses: A systematic review. In *Proceedings of the 2015 ITiCSE on Working Group Reports*, page 1–39.

Corbin, J. and Strauss, A. (2014). *Basics of Qualitative Research. Techniques and Procedures for Developing Grounded Theory. 4th edition*. SAGE Publications.

Delgado, D., Velasco, A., Aponte, J., and Marcus, A. (2017). Evolving a project-based software engineering course: A case study. In *2017 IEEE 30th CSEE&T*, pages 77–86.

Fowler, M. (2020). On pair programming. https://martinfowler.com/ articles/on-pair-programming.html.

Glaser, B. and Strauss, A. (1967). *The Discovery of Grounded Theory: Strategies for Qualitative Research*. Mill Valley, CA: Sociology Press.

Holmes, R., Allen, M., and Craig, M. (2018). Dimensions of experientialism for software engineering education. In *Proceedings of the 40th ICSE-SEET*, page 31–39.

Iacob, C. and Faily, S. (2019). Exploring the gap between the student expectations and the reality of teamwork in undergraduate software engineering group projects. *Journal of Systems and Software*, 157:110393.

Motogna, S., Suciu, D. M., and Molnar, A.-J. (2021). Replication Data Set. 10.6084/m9.figshare.13636445.

Parizi, R., Spoletini, P., and Singh, A. (2018). Measuring team members' contributions in software engineering projects using git-driven technology. *2018 IEEE Frontiers in Education Conference (FIE)*, pages 1–5.

Posten, R. M. (1985). Preventing software requir. specification errors with ieee 830. *IEEE Software*, 1(2).

Raibulet, C. and Fontana, F. A. (2018). Collaborative and teamwork software development in an undergraduate software engineering course. *J. Syst. Softw.*, 144:409–422.

Runeson, P. and Höst, M. (2009). Guidelines for conducting and reporting case study research in software engineering. *Empirical Soft. Eng.*, 14(2):131–164.

Serrador, P. (2012). The importance of the planning phase to project success. In *PMI Global Congress 2012—North America*.

Solingen Rini, Basili Vic, C. G. and Rombach, D. (2002). *Goal Question Metric Approach*. Encyclopedia of Software Engineering, Wiley.

Troussas, C., Giannakas, F., Sgouropoulou, C., and Voyiatzis, I. (2020). Collaborative activities recommend. based on students collaborative learning styles using ann and wsm. *Interactive Learning Env.*, pages 1–14.

Ville, I. and Daniels, M. (2019). Searching for global employability: Can students capitalize on enabling learning environments? *ACM Trans. Comput. Educ.*, 19(2).

Zwikael, O. and Globerson, S. (2006). Benchmarking of project planning and success in selected industries. *Benchmarking: An Internat. Journal*, 13:688–700.