

Learning Soft Skills through Distributed Software Development

Emily Laue Christensen
emic@itu.dk
IT University of Copenhagen
Copenhagen, Denmark

Maria Paasivaara
maria.paasivaara@lut.fi
LUT University
Lahti, Finland

ABSTRACT

The software industry needs universities to train developers to have besides the technical skills, also strong soft skills to collaborate in globally distributed software development projects. To develop these soft skills, we organized a distributed online software development project course, during which student Scrum teams of 5–8 members from five Belarusian universities worked in industrial projects for Danish customers. The course aimed to 1) teach students the Scrum framework and soft skills, such as teamwork and communication with international customers; and 2) to give Belarusian teachers ideas for organizing similar courses in the future. Based on 20 post-course semi-structured interviews with students and stakeholders, and the analysis of 24 student learning diaries we studied the learning outcomes and challenges related to soft skills. The main reported learning outcomes were: communication, methodical use of Scrum, problem solving, organizational/planning skills, teamwork, interpersonal skills, and time management.

CCS CONCEPTS

• **Social and professional topics** → **Software engineering education**; • **Software and its engineering** → *Agile software development*; • **Applied computing** → *Collaborative learning*.

KEYWORDS

Scrum, software engineering education, soft skills, global software engineering

ACM Reference Format:

Emily Laue Christensen and Maria Paasivaara. 2022. Learning Soft Skills through Distributed Software Development. In *Proceedings of the International Conference on Software and System Processes and International Conference on Global Software Engineering (ICSSP'22)*, May 20–22, 2022, Pittsburgh, PA, USA. ACM, New York, NY, USA, 11 pages. <https://doi.org/10.1145/3529320.3529331>

1 INTRODUCTION

Universities aim at providing education that is aligned with industry needs to better prepare students for the workplace. Several studies have found that novice software developers entering their first industry jobs lack some of the crucial skills. For example, Begel and Simon [6] found that many of the problems the novice software

developers faced at Microsoft were due to their lack of soft skills, such as social and communication skills. More recently, Capretz and Ahmed [8] argue that more emphasis should be put on developing soft skills in universities, as the importance of soft skills in the creation of software is often overlooked by educators and practitioners due to their technical background and expertise. Oguz and Oguz [20] provide an overview of the gap between the software industry and software engineering education, the causes of which include the students lack of soft skills, e.g. teamwork and interpersonal skills. Universities should follow the industry needs for the graduates, and match the courses accordingly.

The Erasmus+ project MaCICT (2018-2021)¹, aimed to modernize the master curriculum of ICT studies in five Belarus universities with support from European Union universities from Germany, Poland and Denmark. In addition, Belarus teachers participated in didactic training courses, and jointly the European Union partners and Belarus universities organized courses as examples of the newest teaching methods that respond to the industry needs. Belarus is an attractive outsourcing location for both European and global software development companies, as the technical skills of the developers are high, and the location is conveniently in Eastern Europe, while the labour costs are significantly lower than in the Central European countries. However, the soft skills of the developers graduating from the Belarusian universities do not match to the industry needs according to the surveys the universities conducted in the software industry as part of the project. Especially soft skills such as teamwork and collaboration with international customers, as well as language skills, were noticed to be too low. The future jobs of many of the Belarus software development students will be in globally distributed projects, where they will be working for international clients, using modern software development frameworks, such as Scrum, and using English as their working language. Therefore, we designed a globally distributed project course to teach these skills. The aim for designing and running this course was twofold: 1) to teach students from five Belarusian universities the Scrum framework and soft skills, and 2) to give Belarusian university teachers experiences and ideas for organizing similar courses.

In this paper, we first report how we arranged a global software engineering (GSE) project course for B.Sc and M.Sc level students in five Belarusian universities. This course differs from previously reported GSE courses especially regarding the course set-up, which is an *outsourcing scenario*: each student team consisted of members from 2–3 universities located in different cities, but in the same country, Belarus, while the industrial customers, the agile coaches and the main teaching personnel were located in Denmark. Thus, this set-up is similar to the outsourcing scenario that is typical in

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
ICSSP'22, May 20–22, 2022, Pittsburgh, PA, USA

© 2022 Association for Computing Machinery.
ACM ISBN 978-1-4503-9674-5/22/05...\$15.00
<https://doi.org/10.1145/3529320.3529331>

¹Modernisation of Master Curriculum in ICT for Enhancing Student Employability (MaCICT), for more information see the project web site: <https://macict.eu>

software companies in Belarus: developers located in Belarus and the customer in another country. In other similar GSE courses (e.g. [11] [10]) the students collaborating in the same teams are typically located in two or three different countries, while the customer, that in some cases is played by the teaching personnel, and in some by real customers, is located in the same country as at least part of the students. However, in our course we especially wanted to teach students to work in an outsourcing scenario with real international customers. In this way the course is similar to capstone project courses [13], which have been increasing in popularity. It differs from capstone project courses however, in that it is not placed at the end of a study programme.

Second, we report our findings on soft skills the students learned based on the analysis of 24 student learning diaries, eight semi-structured interviews with the students, and 12 semi-structured interviews with the industry Product Owners, agile coaches, and Belarus teachers. Moreover, we present our findings on the soft skills the students were particularly challenged with, as reported by the stakeholders. Several studies focus on the complexities while teaching globally distributed project courses, i.e. communication and information sharing issues [15] and challenges in teamwork and student motivation [3] and recommend various approaches to handle these challenges. This study differs from the previously reported studies on GSE courses as we analyzed *the student learning outcomes* especially related to the soft skills they learned. In their systematic literature review on GSE courses, Clear et al. [9] noticed that previous studies have not put enough emphasis on the learning outcomes. We found only a few studies reporting GSE course learning outcomes [23][21]. Therefore, by this study we contribute to filling in this gap.

We also report our findings on the practices that supported the learning of soft skills. Finally, we report our findings on the overall course experience, key takeaways and challenges based on the five interviews with the Belarus teachers.

2 RELATED WORK

2.1 Industry Needs

It is often said that universities teaching computer science concentrate too much on “hard” skills, such as new programming paradigms and development methodologies. However, the “soft” skills related to human factors in software engineering are not well supported by the courses. Therefore, students entering the job market need to learn those missing skills [6]. Begel and Simon [6] studied novice software developers at Microsoft. Their study revealed that novice developers found themselves in situations that differed from their university experience, e.g., working with a large team and with a legacy code base. The authors found many social and communication problems, which were mainly due to the anxieties of working in a different kind of situation than they were used to. Therefore, the authors recommend universities to arrange more authentic learning situations where students would be working with a large pre-existing codebase, and students would need to interact with more experienced colleagues who challenge them. Beecham et al. [5] surveyed three companies using GSE to identify which GSE practices were problematic and important to the companies. They compared those to the practices GSE education literature

reported and found that GSE educators fell short in teaching project management, and the interpersonal and leadership skills that are required to manage complex coordination and collaboration in GSE projects, i.e., soft skills.

2.2 Soft Skills

Soft skills can be interpreted in many different ways. In this study, soft skills are defined as “*the combination of the abilities, attitudes, habits, and personality traits that allow people to perform better in the workplace, complementing the technical skills required to do their jobs and influencing the way they behave and interact with others*” [18]. Matturro et al. [18] performed a systematic mapping study on soft skills in software engineering and identified 30 main categories of soft skills that are considered relevant to the practice of software engineering. These included e.g., communication, teamwork, analytical, organizational and interpersonal skills.

To shed light on the *non-cognitive abilities*, or soft skills needed for developers to excel in the software industry, Groeneveld and Jacobs [14] performed a Delphi study with world-wide industry experts from 11 different countries. The skills identified and ranked were classified into four main areas: communicative skills, collaborative skills, problem solving skills, and personal skills. Caeiro-Rodríguez et al. [7] and Idrus [16] investigated the most relevant soft skills for engineering and the students perception on their importance. They found that the students highly valued soft skills such as meta-cognitive skills (e.g. independent and autonomous learning, willingness to learn, critical and analytical thinking), intra-personal (e.g. initiative, self-discipline and planning) and problem solving skills.

Several studies report how soft skills are taught in universities, however, most studies do not analyze which skills students really learn during those courses. E.g., Oguz and Oguz [20] interviewed students and recent graduates to determine which skills were acquired during their education and in course projects, and some of the students reported that the university had provided them with opportunities to develop and improve soft skills. However, the particular skills learned were not specified. Richardson et al. [23] performed a study of student learning experiences in global software development projects. Here the students reported learning about the importance of regular team communication and interpersonal awareness, and team dynamics. Paasivaara et al. [21] also studied student learning outcomes in GSE and found that students learned how to communicate effectively and divide work efficiently.

We found only one study that aimed to analyze the learning outcomes regarding the soft skills specifically. Sousa and Rocha [27] presented game-based learning contexts for developing soft skills in project management. The students were asked to reflect several questions in the course forum. The reflections were analyzed by the teachers and showed that the students developed skills such as communication, planning, and prioritization [27].

2.3 Capstone Project Courses

Capstone project courses have become increasingly popular in universities. Dugan [13] surveyed around 200 computer science capstone project courses. According to the survey, the ten most often mentioned skills these courses aimed to teach were (by citation

frequency): design (24), requirements (22), groupwork (21), testing (19), writing (17), speaking (15), software process (14), project management (14), large system experience (14) and knowledge integration (10). Many of these are *soft skills*, such as groupwork, writing, and speaking. Bastarrica et al. [2] studied student learning in their capstone project course by asking students about their learning, both in the early phases of the projects and in the end. They found that the perceived relative value of soft skills grew towards the end of the project, while the value of the technical challenge dropped. They found that students felt that the relative difficulty of soft skills grows in comparison to that of the technical challenge. Thus, they concluded that students seem to be well prepared regarding technical skills in their previous courses, but not so prepared regarding the soft skills. During the projects they both learn those soft skills and realise the importance of the soft skills to the success of the projects.

The course we present in this study is similar to a capstone project course, as it is designed to prepare students for work in the software industry through practical experience. This course was not placed at the end of their study programme however, so it does not serve as a culminating academic and intellectual experience for the students.

2.4 Teaching Global Software Development

Literature has recognized many skills that are important to be taught to students learning GSE, e.g., communication between distributed team members [23][21], team dynamics [23], interpersonal skills [5] [23], and managing time [28]. All of these are soft skills [18]. There are many ways to teach students GSE skills, e.g., emulating the workplace by organizing cross-university, multi-site courses; engaging students contribute to open source projects, which would give students real-world experience without the overhead involved in cross-university courses; or using online simulations and games that do not require as much calendar time as the two other approaches [4]. However, while the cross-university, multi-site courses might be the most demanding to organize, they might be the most useful to the students, thus many papers have reported this format to teach GSE. A systematic literature review on GSE teaching [19] stresses that GSE teaching should be supported by practical experiences through which students can learn by doing.

Many of the distributed project courses use Scrum, e.g. [11, 21, 22, 25], as it is currently the most popular project framework in the industry. Clear et al. [9] performed a systematic literature review on GSE courses and found that a large number of universities have already successfully developed courses teaching GSE, and have developed ways to overcome the many challenges of teaching GSE. They noticed however that the studies put less focus on learning outcomes. This paper aims to contribute in filling the gap recognised by Clear et al. [9] by studying the student learning outcomes. Therefore, our main research question is: *Which soft skills did the students learn or improve?*

3 THE DISTRIBUTED PROJECT COURSE

In this section we describe the globally distributed software project course that we arranged during the fall 2020. The course was organized during the Covid-19 pandemic and therefore all activities

were undertaken purely online. We describe the learning goals of the course, the stakeholders and their roles in the projects, the teaching methods, and the software development process used.

3.1 Course Learning Goals

The main learning goals of the course were:

- (1) Getting hands-on experience of all phases of an industrial software development project
- (2) Applying Scrum and other work methods and tools in a project
- (3) Learning “soft skills” such as communication, teamwork, and time management
- (4) Learning global software engineering
- (5) Applying new technologies
- (6) Understanding the common challenges involved in commercial software development projects

The course learning goals were predefined in the course description, and prior to starting the course, we asked from all enrolled students their personal learning goals for the course, their technical skills, ideal work hours and experience with Scrum. Before the course all students had at least heard about Scrum, a few knew the basics and four had used Scrum in their job. To our surprise, the most often mentioned goal was strengthening the English skills, especially spoken English. Thus, even though English was not one of the official learning goals of the course, it certainly was something that students had the opportunity to learn, as all course events were in English, and communication with the Product Owners (PO), the agile coaches, the main teacher, and the teaching assistant happened purely in English.

3.2 Student Groups

In September 2020, 33 students registered to the course, 3–9 students per university, from five universities. We created four groups of 8–9 students. Each group consisted of 7–8 B.Sc. and M.Sc level students and a student Scrum Master (SM). The students were enrolled in various programmes, i.e. software development, programming, computer science, and computers, systems and networks. Three of the SMs were B.Sc students, and one was a M.Sc student. The groups were formed by the teacher and the teaching assistant, so that in each group there would be students from 2–3 universities and preferably around the same number of students from each site. This helped ensure that the groups were as balanced as possible. While forming the groups we took into account the students preferred working times and their preferred fellow group members. After the initial group formation, several students were permitted to switch groups upon request. Nine students dropped out of the course, for varying reasons, e.g. the unstable political situation in Belarus and language problems. One student joined shortly after the start.

The student groups carried out project planning sessions in the first week of the course. They were instructed to plan the allocation of the budgeted effort per student in each sprint, schedule ideal work days, decide on the main practices and tools for communication, discuss teambuilding activities, prepare a group CV and create a group agreement.

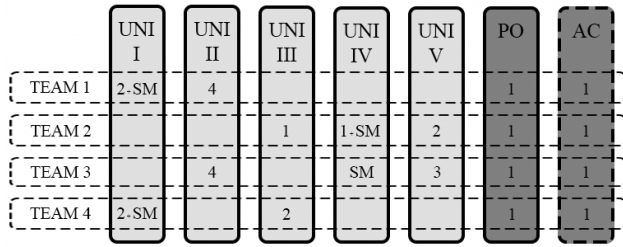


Figure 1: Scrum Teams

3.3 Clients and Topics

As customers we invited Danish companies, as the main university organising the course was from Denmark and had collaborated with the Danish companies on a similar course previously. We received five project proposals that the four student groups could choose from. The chosen projects were to be useful for the company proposing it and the project results were expected to be taken into use in the company after the project. Each company had to provide a PO, who would actively collaborate with the team weekly, and participate, e.g., in the Sprint Planning meetings and Sprint Reviews.

Companies prepared project proposals before the course start up and the student groups sent their group CVs to the companies they were interested in working with. During the second online lecture the POs gave short pitches on their projects, after which online interviews between the student groups and POs took place. In the end of this lecture session, both student groups and POs prioritised their partner choices and we ran a matchmaking software based on the stable marriage algorithm to match the projects and student groups. The four final teams of the course consisted of 5–8 students and one PO supported by an agile coach (AC), see Figure 1.

The four project topics that the student groups chose were: 1) a job centre dashboard for refugees in Denmark, that would allow efficient integration of refugees in the Danish labour market; 2) a prototype of a smart office platform for a global company; 3) a SaaS recruitment software, allowing for case based recruitment; and 4) a web-based tool to help students make the right choices about their studies and direct their studies towards their goals in life. The companies providing topics ranged from a small start-up company to a big global IT company.

3.4 Agile Coaches

Each team was assigned a professional agile coach from Denmark. These coaches worked either as coaches or SMs in the industry. All had previous experience in supporting student Scrum teams in Denmark as agile coaches. In this course they coached their Scrum team, and especially the SM, regarding Scrum and agile. In addition, the coaches organised two Scrum training workshops. The coaches were expected to use around 40 hours coaching their team and were paid by the project.

3.5 Course Personnel

The main organizer of the course was a Danish university that provided the main teacher and a teaching assistant, who had organized similar courses previously for the Danish students. They

Table 1: Course timeline

Phase	Events	Week
Group Planning	Group formation and Scrum workshop I	1
Release Planning	Project selection and DST Scrum simulation	2
Sprint 1	Scrum workshop II	3–4
Sprint 2	Course review I and experience exchange I	5–6
Sprint 3		7–8
Sprint 4	Course review II and experience exchange II	9–10
Sprint 5		11–12
Finishing	Course review III and final celebration	13

planned and organized this distributed, online course, based on the previous Danish course. While each participating Belarusian university provided at least one teacher, who selected the students, supported students locally when needed, participated in the course review meetings and finally, graded their students. Their main role was supporting their students and getting ideas from the course for their own teaching. Altogether six Belarusian university teachers participated.

3.6 Teaching Methods

This is a project-based course with only a few lectures and other events. The first lecture consisted of an introduction to the course and group formation. The project selection event took place in the second lecture, and two Scrum training workshops were also carried out during the course. The DST Scrum simulation, a playful digital simulation created in the game Don’t Starve Together [17] in combination with Trello [1], was carried out with each team, as well as one session with three students who had not been able to participate with their own teams and a SM who volunteered to help. Two experience exchange sessions were facilitated by the SMs, where the students exchanged experiences and advice across teams. In the three course review meetings the students presented their project progress to the course personnel, coaches and POs, and received feedback. In the final celebration event the students presented their projects to all course participants. An overview of the course timeline is presented in Table 1.

The written student deliveries consisted of three group submissions: a process overview, technical overview, and the slides used for review presentations. An individual learning diary was also submitted by each student. The learning diary is an iterative report, so the students submitted four times in total, and were graded on the final submission by the Belarus teachers. After the first three submissions, each student received individual feedback from the teaching assistant in the form of an assessment rubric and personalised comments on their written work. In the following deliveries the students were recommended to self-assess their work prior to submission according to the rubric, and personal comments were provided as feedback. The main tools for lectures and communication were Discord, Zoom, and email (see Table 2).

3.7 Software Development Process

All teams were required to use Scrum [26], and take into account the requirements and recommendations in the course instructions.

Table 2: Online tools for course lectures and communication

Discord
<ul style="list-style-type: none">- The main communication and information distribution channel.- Used by the course personnel, students, agile coaches, POs and other company stakeholders.- Used for the experience exchange sessions.- Private text channels for each team, the SMs, course personnel and coaches, and for the students.- Open voice channels for each team.
Zoom
<ul style="list-style-type: none">- Used for lectures, workshops, and the course review meetings.
Email
<ul style="list-style-type: none">- Used in the beginning to inform all stakeholders.- Used throughout the course to communicate with POs and Belarus teachers.
Other
<ul style="list-style-type: none">- The teams used a variety of backlog management tools and instant messaging applications, e.g., Trello, Jira, and Telegram.

If a particular course requirement fit poorly into the project, the team was permitted to propose changes. Each team was required to have five 2-week sprints that contained at most 36 hours of effort per student. Prior to the first sprint the teams carried out release planning, which involved getting to know the team members and the company, crafting a Product Vision and the initial Product Backlog, selecting and studying technologies, and planning the initial application of Scrum and other work methods and tools. In the following sprints the student were expected to deliver software increments.

Together with their PO, the students were required to create a Product Vision that briefly characterized the “why, what and for whom” aspects of the project and a Product Backlog which contained items that had a description, effort estimation, and were ordered according to priority. They were required to have a Sprint backlog for each Sprint that contained all identified tasks with a name/description and an effort estimate. Backlogs were managed in a dedicated online backlog management tool. The course also required that each team create a burnup chart for logging the estimates in the Product Backlog and tracking the work velocity. The teams were required to arrange Sprint Planning, Reviews and Retrospectives for each Sprint. The course recommended having a Daily Scrum meeting at least twice per week, as students typically spend 2 days per week on their project work.

4 METHODOLOGY

We wanted to understand learning outcomes and gather feedback on the skills of the students from the stakeholders and pose the following research question:

RQ: Which soft skills did the students learn or improve?

4.1 Data Collection

Our data consists of 24 student learning diaries and 20 semi-structured interviews with course participants or stakeholders.

In their learning diaries students were asked to list their learning goals for the whole project, a reflection on the DST Scrum simulation, learning goals for each sprint, the main points learned and contributions in each sprint, and final reflections on the project.

The learning diaries varied in length from 1,400–4,600 words, averaging 2,600 words each, with the SM students submitting the longest diaries as per the course requirements. All students were informed during the first course lecture and reminded again at the end of the course, that we would be analysing their learning diaries as part of the data collection for this study to better understand their learning, and that they had the opportunity to opt out without explanation, but none requested this.

The interviews took place in late December 2020 and early January 2021. Each interview took 17–63 minutes and was 37 minutes long on average. We asked students to volunteer for the interviews after the course exams. All four student SMs, and four student developers from three different teams volunteered. Moreover, all four agile coaches, three of the POs, and five Belarus teachers were interviewed. All interviewees were informed of the purpose of the interviews and the anonymity of the data collection, via either an announcement in the course Discord or email, and they provided written consent in their responses and again orally, just prior to the start of the interview. 18 interviews were carried out by the teaching assistant, who was in close communication with the course participants, while two interviews (a PO and a teacher) were carried out by the course teacher. All of the student interviews were carried out by the teaching assistant, who was not involved in the grading.

The main themes in the interviews² were: participant backgrounds, the most important learning outcomes, the most difficult problems encountered in the project, the best experiences on the course, the collaboration between students distributed in Belarus and POs and coaches in Denmark, support from the teachers in Belarus, ideas for improving the course further, and what participants would do differently if they could start the course again. The SM students were also asked about their participation in the SM Community of Practice, and the Belarus teachers received questions about student selection and grading. In addition, the teachers, POs and coaches were asked about their motivation to participate in the course, the students’ knowledge and learning, the suitability of the project topics, and the collaboration with the organizing Danish university. All interviews were audio recorded and transcribed by the teaching assistant.

4.2 Data Analysis

The qualitative data from the student learning diaries and interviews were thematically analysed in NVivo by first identifying and coding all statements that described learning outcomes, and creating an initial set of codes which defined the identified themes. The codes were then compared with a list of 30 soft skill categories relevant to the practice of software engineering [18].

Ten of the thirty soft skill categories identified by Matturro et al. [18] were not defined explicitly in their study, whereas for some skill categories they provided several overlapping definitions. Therefore, we first created definitions for the categories missing a definition, based on the soft skills grouped under those categories, as identified by Matturro et al. [18]. Second, we compiled some of the definitions provided by Matturro et al. [18] to minimise redundancy. E.g., Matturro et al. provided four definitions for their category *communication*, which were compiled into one brief definition. The

²Interview guides: <https://doi.org/10.6084/m9.figshare.17072117.v1>

Table 3: Soft skill codes and definitions in alphabetical order

Code	Definition	Reference [18]
Analytical skills	The ability to understand and explain each part of a whole. The ability to break a situation down into its component parts, recognize what needs to be done and plan a suitable course of action in a step-by-step way. The ability to think logically and analytically.	Compiled
Autonomy	The capacity to govern themselves by their own means. The ability to plan, carry out and complete work tasks independently, under minimal supervision.	Compiled
Change management	The ability to adapt and work effectively with different situations and face of change. The ability to accept and adapt to changes when carrying out tasks without showing resistance.	Compiled
Communication	The ability to communicate orally and written in a simple, concise, unambiguous, and easily understood way. The ability to convey information effectively so that it is well received and understood.	Compiled
Critical thinking	The ability to determine carefully and deliberately accepted, refutation or suspension of the trial about a particular piece of information.	Verbatim
Customer orientation	The ability to identify and meet the needs of its customers.	Verbatim
Decision making	The ability to judge alternatives and make appropriate and sensible decisions based on available information.	Compiled
Initiative	The ability to propose and/or take any action without the need for others to come to ask or say.	Compiled
Interpersonal skills	The ability to behave in ways that increase the probability of achieving the desired outcomes. The ability to deal with other people through social communication and interactions under favourable and inauspicious conditions.	Verbatim
Leadership	The ability to lead and supervise others.	Created from skills
Methodical	The ability to use a set of steps, neatly arranged, set by methods (techniques) to solve a particular issue or problem.	Verbatim
Motivation	Motivation to work.	Created from skills
Organizational/planning skills	The ability to assess, sort, prioritize and control the execution of tasks according to plan. The ability to efficiently manage various tasks without wasting resources. The ability to make people work efficiently.	Compiled
Problem solving	The ability to understand, articulate, and solve complex problems.	Compiled
Stress management	The ability to withstand stress without losing control. The ability to work calmly and efficiently, even under time pressure or occupational stress.	Created from skills
Team management	The ability to manage a team and maintain team cohesion.	Created from skills
Teamwork	The ability of an individual who is good at working closely with other people. The ability to work effectively in a team environment and contribute toward the desired goal. The ability to cooperate with other teammates during teamwork.	Verbatim
Time management	The ability to plan and schedule time realistically, and complete tasks accordingly in an organized manner.	Created from skills
Willingness to learn	Eagerness and motivation to learn. Willingness and ability to become acquainted with novel subjects and areas in a self-directed, active manner.	Created from skills

soft skill category list and the original definitions by Matturro et al.[18] and our modified and shorted definitions can be found from a web appendix³.

Next, the qualitative coding of student learning diaries and interviews was enhanced and partially re-coded using the 30 soft skill categories by Matturro et al.[18] and the final definitions which were copied verbatim, created, or compiled by us. Particular attention was paid to statements that may have been previously overlooked due to the English skills of the students. E.g. none of the students reported interpersonal skills specifically as a learning outcome, but the analysis revealed multiple statements that corresponded to the definition of that specific skill (Figure 2). All coded passages from the student interviews were also cross-checked against the same students’ coded learning diaries passages. If the same learning was reported in both, it was only recorded as one instance, while calculating which soft skills were mentioned by each specific student. The learning outcomes from the initial set of codes that were not categorized as soft skills, e.g., learning new programming languages, were excluded from the analysis.

The stakeholder perceptions of the students’ soft skills were explored through analysis of the interviews with the POs, agile coaches, and Belarus teachers. Here statements that described both learning outcomes and challenges related to the soft skill categories were identified and coded. Out of the 30 soft skill categories listed by Matturro et al.[18], 11 skills were not discussed by the students or stakeholders as either learning outcomes or challenges in our data. These skills were: conflict management, commitment/responsibility, flexibility, ethics, results orientation, innovation, presentation skills, creativity, negotiation skills, listening skills, and fast learner. The final set of 19 codes and definitions can be viewed in Table 3. The results of this analysis were compiled in excel data sheets and summarized in a graph (Figure 3).

All interviews and learning diaries were then analysed to explore how these learning outcomes been achieved and two main practices were identified as particularly important: 1) overcoming language barriers, and 2) team building activities. Finally, the interviews with the Belarus teachers were analysed again, to gain insight into their experiences and key takeaways from the course, including the challenges encountered. In addition, quotations from the learning diaries and interviews were chosen to illustrate the main findings. The entire coding process was carried out by the first author and

³All 30 soft skill categories and their definitions can be found here: <https://doi.org/10.6084/m9.figshare.18999875.v1>

Interview, student 4 (SM)	Codes
So of course it is a huge experience in teamwork because it is hard to work with some new people and make some relationships in the development with them. I think yes, this is the most important thing, kind of interesting and useful thing through the course that I have learnt.	<ul style="list-style-type: none"> • Teamwork • Interpersonal skills

Figure 2: Coded statement from interview

the identified themes in the statements were discussed and checked by both authors, to ensure that we agreed on their fit with the codes.

5 RESULTS

In this section we present results to our research questions.

5.1 Learning Outcomes and Challenges

Here we present the 17 soft skills learned by the students, based on the analysis of interviews with the stakeholders, and the analysis of the student learning diaries and interviews. We also present the soft skills the students were particularly challenged in, as reported by the stakeholders. Several of the soft skills the students were challenged in, were the same skills they learned. The last two soft skills presented in this section, analytical thinking and change management, were reported only as challenges, with no learning outcomes in those areas. The results are presented in Figure 3 and explained in the following sections.

Communication: 23/24 students and 8/12 stakeholders reported that communication was a learning outcome of the course. The students’ oral communication in particular improved greatly, as was demonstrated through active participation in discussions during meetings.

“During the formation of the sprint backlog and the prioritization and assessment of the issue, we began to understand the sprint and how to communicate correctly with each other and improved team communication.”
 – Learning diary, student 1 (developer)

“I think they understood they do not need to be ashamed or afraid to speak out.. they were just a bit too shy. But they opened up in the end and I think that was also a great learning from them that they can take forward.”
 – Product Owner, team 1

Two of the student’s also reported that they had learned to ask more clarifying questions, and three reported that their writing skills had improved. In addition, learning how to communicate with a PO from a real company was a key learning outcome that was expressed by twenty students.

“..during these two sprints, we began to communicate more closely with the product owner and realized how much it simplifies the work. After we have been pointed out the importance of the Definition of Done we began to seek the opinion of the Product Owner more often, send mockups for design approval, ask questions of interest to us.”
 – Learning diary, student 2 (developer)

Communication was also reported as one of the main challenges in the collaboration with the students by 8/12 stakeholders. Here oral communication specifically was discussed as a challenge by six stakeholders. While the students are quite good at written English, and the assessment of the learning diaries placed most students at good/excellent work, they initially struggled to express themselves orally, especially in meetings with the agile coaches and POs. This

ties closely to language barriers which influenced the communication, a point that was expressed by 9/12 stakeholders, and will be discussed later in this paper, in relation to the students’ English skills.

“The overall biggest challenge was definitely the language. It was really hard for them to understand me and the PO, and it was almost impossible to get the team speaking.”
 – Agile coach, team 1

Two stakeholders also reported specifically that the students were not initially asking them for help, while three stakeholders reported that the students were not reaching out to ask questions. Finally, one stakeholder reported that the students struggled with conveying information about the political situation in Belarus, which was affecting the project work.

Methodical: 20/24 students expressed that learning how to follow and use the Scrum framework was one of the main topics learned. While many of the student’s were initially unfamiliar with Scrum roles and events, they quickly learned how the allocation of roles and the steps outlined in the events contributed to effective collaboration in the team.

“Understanding and organizing Scrum within the team, initially, it was quite difficult and incomprehensible, with many new unfamiliar terms, words, actions. There was no clear understanding of why this was needed, and how it could help in the product development process. However, after quite clear and numerous activities carried out by the project, this understanding began to appear. With each lesson, I became more and more interested in this topic and began to understand the main aspects of Scrum.”
 – Learning diary, student 3 (developer)

This was also expressed by 6/12 stakeholders who reported that the students showed learning in their use of the Scrum framework.

“..it was still really great to see how they understood Scrum in the end and how they used it as well. We could see in the burndown chart as well, it looked very smooth. It was great, the last one at least. I think they really showed that now they understand that they don’t just have to, by the end of the Sprint, go update stuff on the board. They actually have to use the board as a benefit for them, during the daily work.”
 – Agile coach, team 1

Regarding challenges learning Scrum, 4/12 stakeholders reported that the students had struggled initially with learning the basics, i.e. what the events were, what they needed to do during the events, and why.

Problem solving: 15/24 students reported problem solving as a soft skill learned during the course. In particular, the students reported that participating in the Daily Scrums and Sprint Retrospectives, and actively communicating with the PO helped them learn how to best solve problems they encountered in the development work.

“During the first sprint, our team faced difficulties. For a more complete understanding of the project, it was necessary to obtain more complete information from the Product Owner, obtain application layouts and other information about the technical part of the project. This allowed me, as a Scrum Master, to gain the problem solving skills of the development team through communication with the Product Owner.”
 – Interview, student 4 (SM)

Organizational/Planning skills: 14/24 students reported that learning how to best organise their work was a key learning outcome. Specific skills mentioned by the students included task delegation and workload estimation. Organizational/Planning skills were also reported as a learning outcome by 1/12 stakeholders.

“It is very important to set tasks that are not too difficult, otherwise they will take too much time and are more difficult to understand, so it is worth breaking them down into smaller tasks.”
 – Learning diary, student 5 (developer)

Teamwork: 14/24 students felt their skills in teamwork improved throughout the course. Working on a distributed team project was a new experience for most of the students, so they

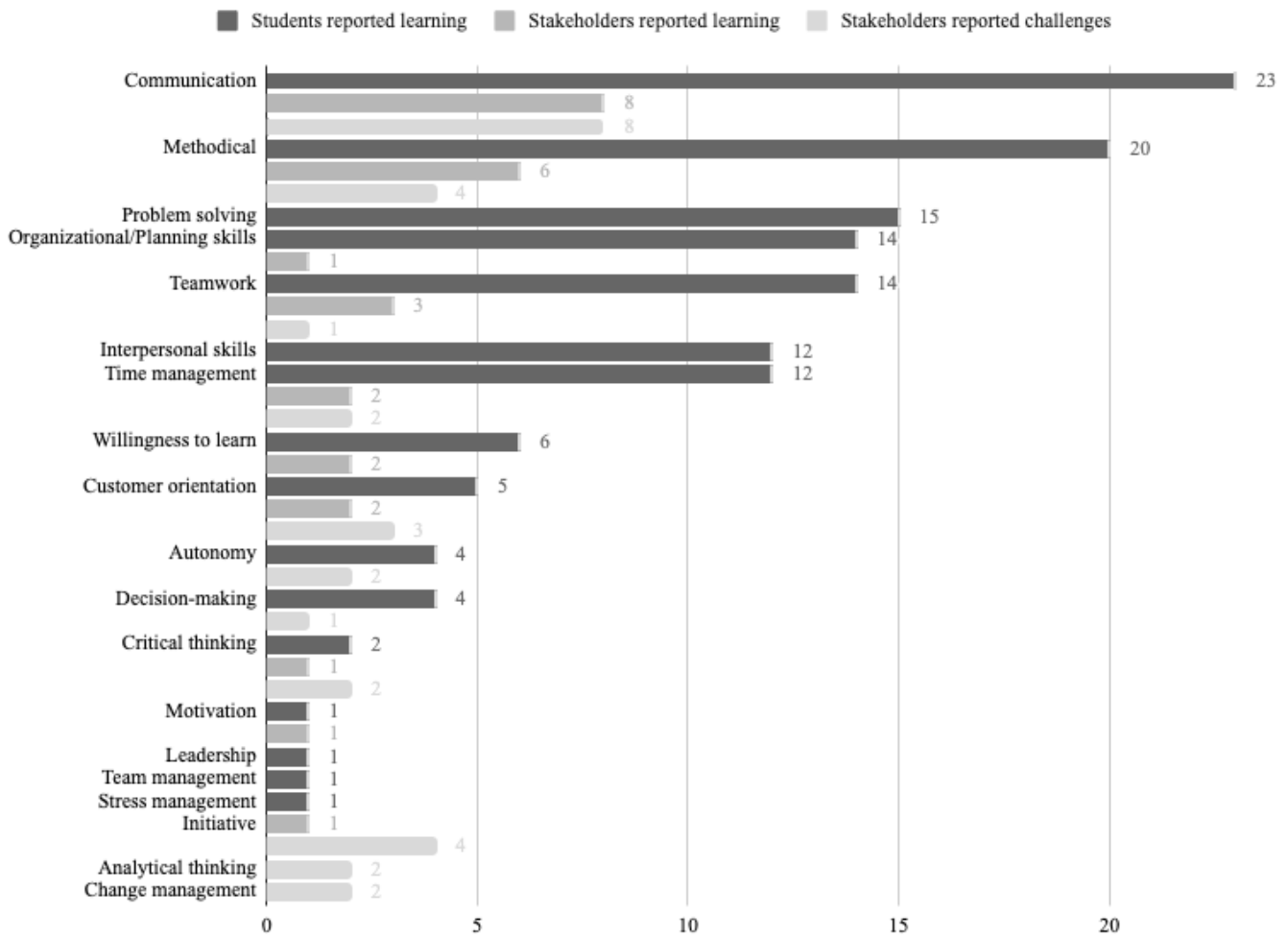


Figure 3: Learning outcomes and challenges

learned how to interact and cooperate through practice. This learning outcome was also reported by 3/12 stakeholders who noted definite improvement in the teamwork during the project.

“It seems to me that the most important thing is teamwork. This is because, in the current world, teamwork is everywhere and is highly valued. After all, one person cannot surpass the work of several. Before this, I had never worked in a team and the experience in this project will be invaluable.”
 – Learning diary, student 5 (developer)

Teamwork was only reported as a challenge by one stakeholder (1/12) who had expected that the students had more experience in this area.

“..there was challenges with the teamwork. I thought they were more used to it. And that was a very big barrier to break for them.”
 – Product Owner, team 1

Interpersonal skills: 12/24 students reported that their interpersonal skills had improved during the course. Here the students discussed how the projects provided an opportunity to build relationships and strengthen their personal and social skills. Although some of the students were initially nervous about working with team members who they were not well acquainted with, they overcame this quickly, and several of the students established friendships through the project work.

“I’ve learned and practiced a lot of social skills. There were some people in my team that were feeling uncomfortable and I was trying to be an example how to stay cool and not to feel angry or upset.. we had one incident when I with my teammate were really angry with each other. We just needed several minutes to come back in reality and apologized to each other.”
 – Interview, student 6 (developer)

Time management: 12/24 students and 2/12 stakeholders reported learning time management as a main learning outcome. Three students reported specifically that creating team calendars and coordinating team schedules was a new learning.

“It has always been a problem for me to plan my activities. I often overworked and sat on tasks for more than the allotted time. During these sprints I’ve learned to better track time.”
 – Interview, student 2 (developer)

Regarding challenges, coordinating schedules and conveying information about meetings initially presented a problem for the students according to 2/12 stakeholders. The students sometimes forgot to invite the PO to the Scrum events and other teamwork sessions, or tried to arrange work sessions with the PO on too short notice.

Willingness to learn: 6/24 students reported willingness to learn as a learning outcome. Willingness to learn from mistakes

specifically, e.g. forgetting to delegate or estimate tasks, was discussed by five of these students. This was also reported as a learning outcome by 2/12 stakeholders.

"By the end of the Sprint we understood that we made mistakes when did estimation of tasks. We estimated all the sub-tasks but forgot about estimating the whole task.. But we learned from our mistake, so it won't happen again."

— Interview, student 7 (developer)

"They really had some great discussions in the three final Sprints, especially the last two ones. They were really reflecting a lot, where in the first two Sprints they didn't reflect at all. When we had the first retrospective they were just like: okay this is an event we need to do because we're told so. But they really understood: we're actually learning from our own mistakes and our own successes." — Agile coach, team 1

Customer orientation: 5/12 students and 2/12 stakeholders reported customer orientation as a student learning outcome. The students learned the importance of providing value to the customer and how to best identify and meet their needs. Regarding challenges, 3/12 stakeholders discussed customer orientation as an area the students struggled with. Here they mentioned specifically that the product could have been improved further if the students had been more curious about the end users and delivering value to the customer, earlier in the project.

Autonomy and Decision-making: 4/24 students reported that they learned autonomy during the course, i.e. how to work independently and self-organise, and 4/24 students reported that their decision making skills had improved. Two of the coaches (2/12 stakeholders) reported challenges in autonomy, e.g. that they had to tell the students exactly what to do during meetings. In addition, one coach (1/12 stakeholder) reported that the students initially struggled with making decisions.

Critical thinking: 2/24 students and one stakeholder (1/12) reported that the students' critical thinking skills had improved during the course. The students learned how to think critically and challenge the requirements put forward by the Product Owner.

"During these two sprints I realized that decisions made by the Product Owner might not be the best or possible at all. For example, our Product Owner was encouraging us to use webflow for deployment of our project, and we were provided one development account for it. But after looking into it we found a few problems with it."

— Learning diary, student 8 (developer)

Two of the coaches (2/12 stakeholders) reported that the students initially lacked the skills in this area.

Motivation: One student (1/24) expressed motivation to pursue a career path as a programmer after participating in the course project. This student was initially considering switching to a different study program, but was motivated to continue studying software development through their participation in the course. This was also reported by one stakeholder (1/12), namely the PO from this students' Scrum team.

"I've started to want to be a programmer again, thanks to this project, although I had a lot of doubts about the career choice.. I find it the best outcome of these several months."

— Learning diary, student 9 (developer)

Leadership and Team management: One of the student developers (1/24) reported leadership skills as a learning outcome and discussed their interest in becoming a Scrum Master. In addition, one student SM (1/24) reported team management skills as a learning outcome.

Stress management: One student (1/24) reported stress management skills as a learning outcome.

"At first when we had piles of work waiting for us, we were shocked. If we were starting over again, we wouldn't be so stunned."

— Learning diary, student 10 (developer)

Initiative: Regarding initiative, only one coach (1/12 stakeholders) reported that the students improved in this area during the course, while all four coaches (4/12 stakeholders) reported that the students did not take initiative to reach out to them for guidance which proved to be a challenge.

Analytical thinking and Change management: Regarding analytical thinking and change management, none of the students or stakeholders reported learning in these skills. Two stakeholders (2/12) expressed specifically that it was a challenge to get the students to understand the topic and the bigger picture of what they were building. The students were very focused on small details, and it was a struggle to get them to understand that they should focus on what they were trying to accomplish during the current Sprint.

"..they (the students) had a tendency of wanting to complete one feature entirely or go into some very very narrow or specific details which did not matter at all. And then they had to get into this subject understanding that what they're doing, it doesn't have to be perfect, it doesn't have to be entirely complete, it just has to be better than what's in production already, and that was quite a struggle for them."

— Agile coach, team 4

This ties closely to managing changing requirements, which also presented a challenge for the students according to 2/12 stakeholders.

5.2 Supporting Practices

The two practices that were discussed by the most students and stakeholders as particularly important to support the learning of soft skills are discussed next.

Overcoming language barriers: English is widely used in the industry and proficiency in the language boosts the confidence of individuals and is instrumental in the development of soft skills, such as communication [24] and presentation [12]. 15/24 students and 3/12 stakeholders expressed that the English skills of the students had improved during the course.

"It was a good experience for learning and practising English. Every activity brings some positive issues in English, from teambuilding, like "self presentation", and joint movie watching, to reviews, retrospectives, sprint planning, the mock-up sending to PO, and learning diary writing."

— Learning diary, student 9 (developer)

Teambuilding activities: Regarding the students' interpersonal skills and the friendships and relationships that were established, teambuilding activities, both scheduled and spontaneous, were reported as beneficial. 7/24 students discussed the benefits of teambuilding activities and how these contributed to the teamwork and to closer relationships within the teams.

"We have started to conduct teambuildings more often. We realised that they strengthen our team spirit and we learn more about each other. In addition, these activities help us to distract a little from routine work and do something else. During teambuildings, we tell different stories, laugh together and do interesting surfing on the Internet. These are some of the best moments in our work."

— Learning diary, student 3 (developer)

5.3 Belarus Teacher Experiences

All five interviewed Belarus teachers stated that they were motivated to assist with the course because they wanted to examine the practices that European universities use to teach computer science students, and to compare how these courses are organized with how they are organised in their local regions and universities. A key point here is the collaboration with real companies, and with POs from other countries, which is not practised in universities in Belarus. According to the teachers, computer science students usually work on "toy projects", where the teaching assistants act

as the POs. One of the teachers was also very curious about the potential output of the students, as they had previously experienced that it was very difficult to achieve the goal of creating something valuable within the given time-frame. The teachers discussed this form of collaboration as a key takeaway from the course, along with the course organisation, teaching methods, e.g., strict deadlines, learning diaries for reflection, and the tools used, e.g. Discord, Kahoot, Mentimeter and DST Scrum simulation. The Belarus teachers were also planning to try out organizing similar courses on their own.

"I like the environment that you set up for this course and all of these channels for teams, some channels with course materials and announcements. It looks simple, nothing complicated to set up this environment. But it was done and it is actually cool, because everyone has some common room to communicate, and it works very well."
— Belarus teacher, team 3

One of the main challenges encountered in the course was the fact that nine students dropped out after the initial group formation. According to one of the teachers, this was a result of several factors combined, e.g. the political situation in Belarus and military recruiting. The technical infrastructure in Belarus also posed a challenge, and presented issues in the form of poor internet connections and a lack of hardware and software for programming and distributed work in general, such as video cameras and microphones.

6 DISCUSSION AND CONCLUSIONS

In this paper we presented how we organized a globally distributed course in an online environment between Belarus and Denmark. This course differed from previously reported GSE courses in several aspects: 1) the course aimed to teach soft skills and study the learning outcomes, 2) the course was organized in an outsourcing set-up: customers, teacher, coaches from one country, students from another but from different universities, 3) the course was taught fully online utilizing different technologies (e.g. Discord) and techniques (e.g. DST Scrum simulation). The main idea of the course was to give Belarus university teachers an example of a course they can run to teach their students, in particular, soft skills that are needed in the industry. Figure 3 that summarizes the main learning outcomes as reported by the students shows that the course succeeded well in teaching the skills we aimed to, especially the soft skills. Among the learning outcomes, the students reported communication, teamwork and problem solving as learned in the course. Our definitions of these skills aligns with the soft skill areas identified by Groeneveld and Jacobs [14], namely communicative, collaborative, and problem solving skills, which they cite as three of the skills needed for developers to excel in the software industry. The students also improved their English skills which will contribute to their future employability, as proficiency in the English language is often required in the industry [20].

The challenges the teams faced were mainly quite usual challenges students face in this kind of projects and from which students learn. However, three of the challenges surprised us: many students dropping out, the infrastructure challenges and how difficult especially spoken English was for many of the students. Finally, we were positively surprised by two aspects: how well the course worked in an online environment and how well students worked in distributed teams that had members from 2–3 universities. The online environment of the course might have supported the latter fact, as

it made all students in the team more equal participants, when the major part of all communication happened in the online platforms.

Limitations: We could not interview all students participating in the course, as participating in an interview was voluntary. We asked everyone to volunteer and eight students, i.e., one third of the students, agreed. However, learning diaries of all students who completed the course were analysed. Regarding teachers, coaches, and POs, almost all agreed to be interviewed. Only one PO and one teacher did not volunteer, thus our data covered the viewpoints of these roles quite well. The teaching assistant carried out all interviews, except two, as we wanted a person that all knew well to do the interviews, so that they would feel comfortable and answer the questions as openly as possible. However, that could have affected the answers, as the interviewees might have said slightly more positive things to a person they knew well and who had been organizing the course, than if they had been interviewed by an outsider. Finally, only one person performed each interview, which might put some bias as to how the questions were asked, and especially how further elaborations in semi-structured interviews were asked. We aimed to mitigate this by planning the questions in detail jointly in the research team (teacher and teaching assistant).

Future research: As the main learning goals of this course were soft skills, we studied how students reported learning them both by analysing their learning diaries and by interviewing them. In the future, we encourage teachers to investigate the learning of soft skills. Often students emphasize the learning of technical skills, and might not even realize all soft skills that they learn or like. Bastarrica et al. [2] had noticed students might not understand the value of soft skills, nor how difficult they are to master. Thus, we suggest emphasizing already in the beginning of the course the value and effort to learn soft skills. The possession of soft skills is difficult to objectively measure, which makes measuring the learning outcomes challenging and grading courses that teach these skills difficult. In this paper we presented one way of measuring the learning of soft skills, by using learning diaries and analysing their content. Based on this analysis we could measure more concretely which soft skills students learned and how they learned them. We encourage future studies to create more methods and ways to measure the real learning outcomes of courses teaching soft skills.

ACKNOWLEDGMENTS

We would like to thank all the participants for making this study possible, especially all the students, Product Owners, agile coaches and the Belarus teachers. Finally, we are grateful to the reviewers for their insightful comments and feedback, which has helped us to improve the paper.

REFERENCES

- [1] Atlassian. 2011. *Trello [Software]*. <https://trello.com/>
- [2] Maria Cecilia Bastarrica, Daniel Perovich, and Maira Marques Samary. 2017. What can students get from a software engineering capstone course?. In *Proceedings of the 39th International Conference on Software Engineering: Software Engineering and Education Track*. IEEE Press, 137–145.
- [3] Sarah Beecham, Tony Clear, John Barr, Mats Daniels, Michael Oudshoorn, and John Noll. 2017. Preparing tomorrow's software engineers for work in a global environment. *IEEE Software* 34, 1 (2017), 9–12.
- [4] Sarah Beecham, Tony Clear, Daniela Damian, John Barr, John Noll, and Walt Scacchi. 2017. How best to teach global software engineering? Educators are divided. *IEEE Software* 1 (2017), 16–19.

- [5] Sarah Beecham, John Noll, and Tony Clear. 2017. Do We Teach the Right Thing? A Comparison of GSE Education and Practice. In *2017 IEEE 12th International Conference on Global Software Engineering (ICGSE)*. 11–20. <https://doi.org/10.1109/ICGSE.2017.8>
- [6] Andrew Begel and Beth Simon. 2008. Novice software developers, all over again. In *Proceedings of the fourth international workshop on computing education research*. 3–14.
- [7] Manuel Caeiro-Rodríguez, Mario Manso-Vázquez, Fernando A Mikic-Fonte, Martín Llamas-Nistal, Manuel J Fernández-Iglesias, Hariklia Tsalapatas, Olivier Heidmann, Carlos Vaz De Carvalho, Triinu Jesmin, Jaanus Terasmaa, et al. 2021. Teaching Soft Skills in Engineering Education: An European Perspective. *IEEE Access* 9 (2021), 29222–29242.
- [8] Luiz F. Capretz and Faheem Ahmed. 2018. A Call to Promote Soft Skills in Software Engineering. *Psychology and cognitive sciences—Open Journal*, Vol. 4, No. 1, 2018.
- [9] Tony Clear, Sarah Beecham, John Barr, Mats Daniels, Roger McDermott, Michael Oudshoorn, Airina Savickaite, and John Noll. 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 (Vilnius, Lithuania) (ITICSE-WGR '15)*. Association for Computing Machinery, New York, NY, USA, 1–39. <https://doi.org/10.1145/2858796.2858797>
- [10] Ivica Crnković, Ivana Bosnić, and Mario Žagar. 2012. Ten tips to succeed in global software engineering education. In *2012 34th International Conference on Software Engineering (ICSE)*. IEEE, 1225–1234.
- [11] Daniela Damian, Casper Lassenius, Maria Paasivaara, Arber Borici, and Adrian Schröter. 2012. Teaching a globally distributed project course using Scrum practices. In *Collaborative Teaching of Globally Distributed Software Development Workshop (CTGDSD), 2012*. IEEE, 30–34.
- [12] Richa Singh Dubey and Vijayshri Tiwari. 2020. Operationalisation of soft skill attributes and determining the existing gap in novice ICT professionals. *International Journal of Information Management* 50 (2020), 375–386.
- [13] Robert F Dugan Jr. 2011. A survey of computer science capstone course literature. *Computer Science Education* 21, 3 (2011), 201–267.
- [14] Wouter Groeneveld, Hans Jacobs, Joost Vennekens, and Kris Aerts. 2020. Non-cognitive abilities of exceptional software engineers: a Delphi study. In *Proceedings of the 51st ACM Technical Symposium on Computer Science Education*. 1096–1102.
- [15] Rune Hjelsvold and Deepti Mishra. 2019. Exploring and Expanding GSE Education with Open Source Software Development. *ACM Transactions on Computing Education (TOCE)* 19, 2 (2019), 1–23.
- [16] Hairuzila Idrus. 2014. Developing well-rounded graduates through integration of soft skills in the teaching of engineering courses. In *2014 IEEE Frontiers in Education Conference (FIE) Proceedings*. IEEE, 1–9.
- [17] Klei Entertainment. 2013. *Don't Starve Together [Video Game]*. https://store.steampowered.com/app/322330/Dont_Starve_Together/
- [18] Gerardo Matturro, Florencia Raschetti, and Carina Fontán. 2019. A systematic mapping study on soft skills in software engineering. *JUCS—Journal of Universal Computer Science* 25 (2019), 16.
- [19] Miguel J. Monasor, Aurora Vizcaíno, Mario Piattini, and Ismael Caballero. 2010. Preparing students and engineers for global software development: a systematic review. In *2010 5th IEEE International Conference on Global Software Engineering*. IEEE, 177–186.
- [20] Damla Oguz and Kaya Oguz. 2019. Perspectives on the gap between the software industry and the software engineering education. *IEEE Access* 7 (2019), 117527–117543.
- [21] Maria Paasivaara, Kelly Blincoe, Casper Lassenius, Daniela Damian, Jyoti Sheoran, Francis Harrison, Prashant Chhabra, Aminah Yussuf, and Veikko Isotalo. 2015. Learning global agile software engineering using same-site and cross-site teams. In *2015 IEEE/ACM 37th IEEE International Conference on Software Engineering*, Vol. 2. IEEE, 285–294.
- [22] Maria Paasivaara, Casper Lassenius, Daniela Damian, Petteri Rätty, and Adrian Schröter. 2013. Teaching students global software engineering skills using distributed scrum. In *2013 35th International Conference on Software Engineering (ICSE)*. IEEE, 1128–1137.
- [23] Ita Richardson, Sarah Moore, Daniel Paulish, Valentine Casey, and Dolores Zage. 2007. Globalizing Software Development in the Local Classroom. In *20th Conference on Software Engineering Education Training (CSEET'07)*. 64–71.
- [24] Marc J Riemer. 2007. Communication skills for the 21st century engineer. *Global J. of Engng. Educ* 11, 1 (2007), 89–100.
- [25] Christelle Scharff. 2011. Guiding global software development projects using Scrum and Agile with quality assurance. In *2011 24th IEEE-CS Conference on Software Engineering Education and Training (CSEE&T)*. IEEE, 274–283.
- [26] Ken Schwaber and Jeff Sutherland. 2017. *The Scrum Guide: The Definitive Guide to Scrum: The Rules of the Game*. (2011). Available: scrumguides.org (2017).
- [27] Maria José Sousa and Álvaro Rocha. 2017. Game based learning contexts for soft skills development. In *World Conference on Information Systems and Technologies*. Springer, 931–940.
- [28] Kathleen Swigger, Robert Brazile, Fatma Cemile Serce, George Dafoulas, Ferda Nur Alpaslan, and Victor Lopez. 2010. The challenges of teaching students how to work in global software teams. In *2010 IEEE Transforming Engineering Education: Creating Interdisciplinary Skills for Complex Global Environments*. IEEE, 1–30.