Skills Development Through Agile Capstone Projects

Evangeli Boti[1], Vyron Damasiotis[1], and Panos Fitsilis[1,2(✉)]

[1] University of Thessaly, Gaiopolis Campus, 41500 Larissa, Greece
{eboti, damasiotis, fitsilis}@uth.gr
[2] School of Science and Technology, Hellenic Open University, Parodos Aristotelous 18, 26335 Patras, Greece

Abstract. Agile development, which has been accepted by many organizations in the area of management and software engineering in the last two decades, nowadays, tends to become an emerging teaching and learning methodology in higher education. A great number of educational institutions are offering courses in programming and software engineering using agile methods, setting aside the traditional teaching. This paper attempts to point out the impact of agile methodology in skills' development on university students. Its aim is to explain Scum's application in university students of a computer science program in a capstone project. It tries to identify the role of agile methods in improving students' transversal skills such as communication, collaboration, team cohesion, team self-organization and autonomy, problem-solving, creativity, and generally project planning skills as well as the need for training in agile methods. The research is based on a survey concerning a capstone project implemented by students of Hellenic Open University (HOU). Results indicate that implementation of agile methods can benefit project team members and help them develop both their transversal skills and team working characteristics.

Keywords: Agile methodology · Higher education · Transversal skills

1 Introduction

During the last years, a new teaching and learning methodology based on Agile methodology tends to enjoy acceptance by plenty of educational organizations in higher education. Agile methods are widely accepted in software engineering industry as traditional processes in software engineering and information systems development cannot cope with the increased complexity, constant changes and challenges that occur in modern software development. Project stakeholders are required to work as a team in order to deliver higher quality projects in less time [3]. With the great impact and the growing implementation of agile methodologies such as eXtreme Programming and their processes such as Scrum and Kanban, graduates can gain the knowledge and skills needed to be productive and successful in their future professional activity [40]. According to Lang [20], Agile Learning is "the application of the processes and principles of agile software development to the context of learning". The use of agile methodologies can help them to communicate and collaborate better in order to

G. Succi et al. (Eds.): ICFSE 2021, CCIS 1523, pp. 97–112, 2021.
https://doi.org/10.1007/978-3-030-93135-3_7

overcome the challenges they face in their student project teams [15, 43]. Relevant researches has shown a great impact of Project-based learning to computer science, software engineering and information systems students [9, 24, 38] and teaching agile methods continues to gain prominence for information systems projects [39]. Also, in software engineering, agile methodologies, as they include continuous communication and iterative development, can promote knowledge management [21]. Educating students in a communicative environment through agile development is a key goal for the instructors in computer science programs in higher education [19]. It also constitutes a more dynamic and effective approach to teamwork/project management compared to traditional management [25]. For these reasons this research implemented a new didactic approach based on Agile Methodology by engaging students of the computer science program from HOU in a capstone project, using the Scrum method, as part of their software engineering/programming curriculum.

In the next section the relative literature of agile concepts in education in general and of Scrum methodology in particular, as well as their impact in team members skills are presented. In continuous the research questions set and the research methodology followed are described in detail. Next, the survey and its results are analyzed both quantitatively and qualitatively. Finally, conclusions are drawn in the final section.

2 Theoretical Background

2.1 Agile Mindset

Although agile software development has been around over the last decade, implementation of agile methodologies in a learning and teaching context is in its infancy, but it has drawn a lot of attention in educational and research conferences. Researchers emphasize that software engineers need to develop not only technical skills but also a set of soft skills that are required to make their work more efficient. They also emphasize the same for the field of education arguing that students have to apply agile methods to their projects because theoretical lectures are not enough for their personal development [5]. Traditional teaching and learning produce graduates with skills and knowledge but don't provide a context for applying those skills. In contrary, agile methods provide active learning by having students to work in a way that makes them to face real problems like they will do on their future jobs [19].

Agile mindset includes a set of methods and procedures that try to engage students with real world experiences in an iterative team-based approach which focuses on team communication and interaction. It diffuses the values and principles from Agile Manifesto [4]. According to it, agile means iterative, quick response to changes, interactive, incremental development and quick feedback. Quick and continuous feedback is among the most important principles in Agile, as the enablers learn from the previous iterations and improve every next iteration, detecting their mistakes early and fixing them as soon as they can. Creative problem solving is also encouraged by continuous feedback through sprint retrospective in the Scrum Framework [19]. Feedback from all stakeholders leads to continuous improvement for any organization [17]. In Agile there is no concept of best practices. More and better practices will

emerge through continuous iterations [17]. There is no need to follow a specific plan but always shape the plan according to the changes that occur.

The Agile principles and values in the context of learning, as Peha's [33] version says are:

- "individuals and interactions over processes and tools
- meaningful learning over the measurement of learning
- stakeholder collaboration over constant negotiation
- responding to change over following a plan" (Peha [33]:23; [30]).

Agile methodologies include quite a few software development and project management processes, such as Scrum, eXtreme Programming (XP), Kanban, Dynamic Systems Development Method (DSDM) and others. The most popular agile approach to software development is Scrum.

2.2 Scrum Framework

Scrum is a framework for developing, delivering, and sustaining complex products. Applying as an alternative educational methodology for team work increases students' perception of learning by allowing self-managing of their time and resources for meaningful active learning [27]. It consists of short project cycles, called "sprints". The team members have to deliver a designed, built, tested, reviewed and useful product iteratively and incrementally through continuous feedback. If students understand Scrum rules they will be able to overcome any occurred obstacles, and to tackle management and teamwork as real professionals. All these imply an increased need for soft skills, as success depends on effective and efficient communication and fast execution [10].

2.3 Skills Development Through Agile Methodologies

According to a variety of researches, soft skills are as important as hard skills [32]. All required soft skills are described in the conference "Supporting Key Competence Development: Learning approaches and environments in school education" about Education and Training from European Commission which took place in November 2019 at Brussels, Belgium, in which it is highlighted that key competences are best developed in systems, which promote and use a variety of learning approaches and environments, support their teachers and assess and validate key competences [29]. The implementation of an agile approach in the context of learning contributes to personal and team development. Further, according to the Association of American Colleges & Universities, the majority of hiring managers believes that graduates cannot succeed in entry-level positions because of their poor soft skills, locating the problem within higher education (Association of American Colleges & Universities 2019 [2]), emphasizing that graduates are not able to apply learning in real-world settings. That's why employers try to hire graduates with soft skills in order to contribute to today's economy [7]. Because of the social dimension of agile practices, skills such as collaboration, communication, teamworking, self-managing, trust and transparency are being developed.

An empirical study about an implementation of eXtreme Programming (XP) in a laboratory course, focused on disseminating knowledge through collaborative practices such as pair programming, has shown that there is a huge potential to enhance the development of collaborative skills. Because the fact that it relies upon sharing knowledge and information, leads to the development of organizations [18], it could help teachers and students share technical knowledge through their teamworking. In pair programming students as partners discuss and work on the given problem while sharing their experience and knowledge [47]. Thus, agile learning becomes a self-regulated learning process because the students learn through iterative cycles and continuous knowledge exchange among them according to their needs. Traditional teaching cannot arouse students' interests, which results in less active engagement in learning. Therefore, teaching students by embracing changes with agile practices, is appropriate both for students' needs and also for improving teaching quality [45].

In an adaptation of a Scrum process has been found that interpersonal communication has been developed. The regular and early meetings helped students improve their communication. It, also, helped them to develop problem solving skills, and creative solution skills through complex adaptive systems methods, leading them to continuous improvement addressing the issue and improving the confidence among the team members [25]. The adoption of Scrum can be very helpful for the student learning processes, empowering their teamworking [14]. It increased students' commitment allowing them to self-reflect on their performance in the Sprint Retrospective meetings [35]. Continuous communication through iterations contributes to mutual learning and increasing progress among team members [45].

Universities are required to develop students' skills needed to have a successful hands-on experience in a software engineering environment close to the real world [31]. Thus, in another case study on a capstone project in a Scrum-based training it has been found that students maximize their performance and develop their problem-solving skills [36].

During a research project where agile-method learning tools were used, project participants, by employing their personal abilities, developed their skills in cooperation and creativity. This happened as the project kept them engaged itself, because of the common understanding about the goals which have been set [26].

According to Hof et al. [16], collaboration and communication are key to successful agile software development through agile values as respect, openness, transparency and trust in a multiweek Scrum simulation project, the evaluation showed that students enjoyed more fun, and the collaboration in the team.

Agile working promotes a psychologically safe working environment. Agile meetings lead students to improve communication, and increase a shared sense of responsibility and respect among them [25]. Because of the fact that the agile approach has no strict rules but is based on useful principles, it seems to suit perfect skill development [5].

3 Research Methodology

The aim of this research is to identify if teaching agile can improve transversal skills such as communication skills, collaborative skills, and team cohesion, team self-organization and autonomy, problem-solving, creativity, and generally project planning skills.

More specifically, in this research we would like to examine the following research questions:

RQ1: Which are the skills that are developed during the implementation of a project using agile methods?
RQ2: Is necessary team members training in agile methods before participating in an agile project?
RQ3: Are the agile team organization and team roles easily comprehended and what are the challenges faced by students?
RQ4: In which way is the agile software development process affected by factors such as a) the team's geographical distribution, b) task switching between project work and other activities, c) lack of team members' commitment?

To achieve the objectives set a survey was conducted with students of Hellenic Open University (HOU). In particular, students of the computer science program of studies were asked to take part in a survey with regard to the skills they developed after the implementation of the capstone project using Scrum methodology. During their third year of study, these students were taught agile methods by engaging themselves in a capstone project as part of their software engineering/programming curriculum. HOU students are mainly distance learning students. Students of two separate years of studies were asked to participate in order to be able to see differences that arise with modality. Specifically, the survey included all students that attended the module during academic years 2019–2020 and 2020–2021. Students were organized into groups. Each group had between 20 to 30 students leading to 8 to 9 student groups per academic year. Finally, students asked to form their own capstone project team using the "self-selection method" that consisted of 3 to 4 persons. The "self-selection" method is a method that gives full responsibility to the students in contrast to random selection or alphabetical methods. Further, it allows the formation of excellent and/or poor performing teams, in contrast with a random selection that produces usually "average" performing teams [1].

In this capstone project, students were given a product backlog for developing a java application and they were asked to implement this application using agile practices. The product backlog contained approximately ten user stories. The user stories described a software system that contained a simple graphical user interface, the use of a restful API, storage and retrieval of data to a relational database, the creation of an XLM file, and the creation of threads.

The teaching team was directing the students to use the SCRUM method. The duration of the capstone project lasted roughly two months, giving them sufficient time to run up to three sprints. However, the duration of each sprint was at the discretion of the team. Teams usually consisted of three to four members, and they were self-

organized, meaning selection of members of the team and as well the assignments of the roles to members of the team. A member of the teaching team was assigned to the position of product owner, while another member of the team was assigned to the role of scrum master. Each team was required to submit reports on a variety of software engineering topics, such as system design, requirements prioritization, sprint backlog, and risk management.

More specifically students were asked as a deliverable of their capstone project to provide the following:

1. Calculation of the required effort per requirement documentation in detail how to calculate the effort. It was suggested to use the planning poker method [23].
2. Calculation of the priorities of the receivables documenting in detail the way of calculating the priority. More specifically, for prioritization, the teaching team recommended the priority poker method, which is a variant of planning poker, where participants, instead of choosing effort estimates for each user story, choose priorities. The method can be ap-plied using "T-Shirt Sizes" cards with the assumption that XXS (eXtra-eXtra-Small) corresponds to the lowest priority and XXL (eXtra-eXtra-Large) to the highest [37].
3. The sprint backlog as it was formed after each iteration.
4. For each of the user stories of the product backlog, the acceptance criteria.
5. The assignments of responsibilities to the team members in each repetition (sprint).
6. The implementation time per user story/per sprint compared to what was planned, as well as the burnt down chart
7. A retrospective review report.

As an output of the retrospective report students were asked to evaluate several factors that relate to the difficulties they faced either concerning the project technicalities or about teamwork. As well, they were asked to report on how they would improve their work taking into account the acquired experience. These reports were used as a qualitative input for this research. More specifically, students were asked to report on:

1. the problems the team encountered during the implementation of the capstone project,
2. the deviations of the actual implementation from the sprint planning,
3. the usefulness of the online collaboration tool that was used to assist the team in their daily work
4. the risks that occurred during the project, and their response to them,
5. to propose changes in the work practices for a next subsequent project, and on
6. the knowledge, competences, and skills acquired.

This research had two parts. The quantitative part was conducted with the development and administration of a questionnaire and the qualitative part which was based on the views as they were expressed by students at the retrospective review report.

3.1 Quantitative Research

The research took place by uploading the questionnaire online using Google Forms and sending it out to the selected students. Before sending out the questionnaire, it was tested internally with a pilot group to ensure its validity as in its structure, language and questions asked. During the pilot research, it was confirmed that the questions were easily comprehensible, engaging, and offered answers relative to the scope of the research. The questionnaire was sent out to 239 current (the academic year 2020–2021) and 238 from previous academic year students (2019–2020) at the Hellenic Open University (https://www.eap.gr/en/). All of the students had or are currently attending a computer science course at the university that included agile software engineering. The students had 30 days to answer and of the total 477 students, 115 replied to the questionnaire, a percentage of 24.1%.

The questionnaire that was distributed had three sections:

- The first section included questions regarding the level of familiarity of the former students with the agile concepts both in university courses as well as in their work environment and some personal information about them.
- The second section consisted of questions regarding the perceived improvement of students' communication skills after the implementation of the project. This section included questions such as, if the students' ability to communicate with other project stakeholders and to understand customer needs or requirements has improved. It was asked if the use of scrum practices and roles helped organize their work and overall improved the communication.
- The third section of the questionnaire focused on the efficiency of agile methods and their use to improve collaboration and overall team cohesion. Students were asked to evaluate the active participation of each member of the team in the planning of the project goals in each sprint, if sprint retrospectives helped to identify the weaknesses of the team and each member individually, or to improve team members' commitment, etc.
- The next section was focused on how agile methods affect team self-organization and group autonomy. Students were asked to evaluate if Scrum affected their ability to create their own sustainable work pace, how the ownership of the decision making affected project execution etc.
- The fifth section was related to problem-solving and creativity skills. The questions in this section were evaluating if agile methods can affect the ability of the team members to solve problems and to be more creative.
- Finally, the sixth section of this questionnaire was related to planning skills and if agile methods affect the ability of team members to manage time, to evaluate the effort estimations required per sprint or user story, etc.

3.2 Qualitative Research

The questions asked in the retrospective review report were the basis of the qualitative analysis. The retrospective review reports were similar to interview questions. Usually, in a semi structured interview, a set of 5–7 topics is formed and the respondents are prompted to talk about, while the interviewer is taking notes [46].

Further, these reports were considered as student feedback. Student feedback is used systematically by all universities to assess the quality or to assess if the learning outcomes were met.

Considering that this was a group assignment where teams had between 3 to 4 team members and the fact that some students have not submitted their assignment at all, and some assignments were incomplete in this section, a total number of 109 assignment reports were evaluated.

In the literature, it is suggested that one effective way to analyze qualitative input is the use of Computer Aided Qualitative Data Analysis Software programs (CAQDAS) such as NVivo and SPSS, etc. Such tools implement learning analytics algorithms that enable researchers to discover insights by qualitatively analyzing student feedback [13]. However, the student input language that was Greek did not allow us to use such tools.

For the above reason, an inductive research approach was used. When the inductive approach is used, research findings emerge from the raw data. Usually, the topics that emerge are the most frequent, dominant, or significant themes [22, 44]. Consequently, interpretation of themes is done by comparison. The inductive approach is flexible since it is not strongly guided such as grounded theory, narrative research, or case study.

When the inductive approach is used [44] five are the main steps:

1. studying the raw data which in our case were the retrospective reports,
2. identification text that is related to the specific research questions
3. labelling text to relate the text with the initial research categories and addition of new emerging categories,
4. reducing overlapping categories, and
5. creating a model consisting of the major most influential categories.

The data collected were statistically analyzed with descriptive statistics for summarizing the findings.

4 Research Analysis

4.1 Quantitative Survey Analysis

In total 115 responses received during the survey. Initially participants were asked to declare if they had any prior experience in software development projects and assess this experience according to a 5-scale Likert scale ranging from “No experience” to “Excellent experience”. According to responses 47.8% had none or insignificant experience, 23.9% had moderate experience and 28.3 had very good or excellent experience in software projects (see Fig. 1).

Next, they were asked to evaluate the experience gained from their participation in the project using the same Likert scale. According to their responses, 7.9% declared that they still had none or insignificant experience, 34.2% declared that they now have moderate experience and 57.9% declared that they have very good or excellent experience in software projects (see Fig. 2).

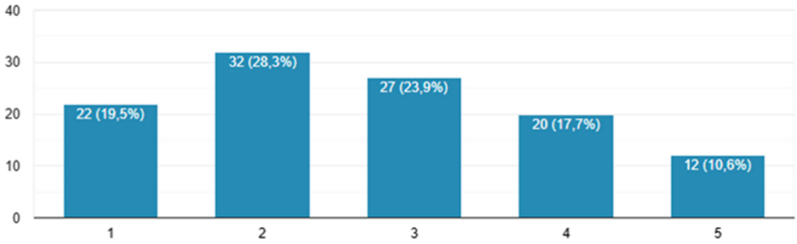

Fig. 1. Level of experience in software projects before participating in project

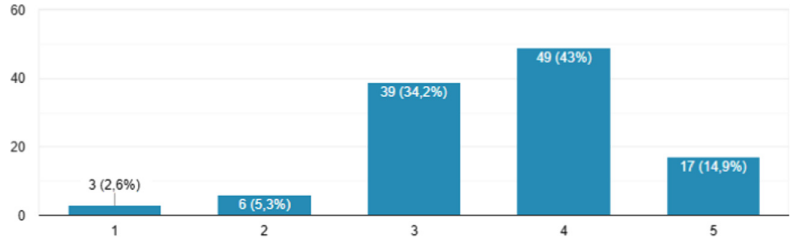

Fig. 2. Level of experience in software projects after participating in project

The project was conducted using Agile methods and specifically the SCRUM methodology. It is worth notice that only 14.9% of the participants had previous experience in agile methods while 85.1% of them had no previous experience in using Agile methods. The survey tried to identify if agile methods helped them to improve their transversal skills such as communication, collaboration, team working and the effectiveness of Scrum method in teamworking improvement.

Respondents could answer using a 5 point Likert scale in all questions. In all cases, more than 65% of the responders declared that their transversal skills were improved or very much improved, about 25% declared that had a moderate improvement and less than 10% declared that had no or insignificant improvement.

Specifically, 74.6% declared that they had improved or very much improved their personal communication skills, 21.9% had a moderate improvement and only the 8.8% said that had no or insignificant improvement. Regarding the effectiveness of Scrum in team level communication either through role assignment to team members or sprints more than 65% of participants declared that Scrum methodology significantly contributed to team communication improvement while less than 12% said that there was insignificant improvement. The rest declared that there was indeed an improvement although in moderate level. As that, it can be undoubtedly concluded that team members' communications skills can be improved by their participation in an agile (Scrum) project.

Participants were asked to evaluate the importance of team members participation in setting sprint backlog, the role of sprints and retrospections in revealing personal and team weaknesses, the contribution of sprints in team members commitment and willingness to concentrate both to personal and team success. Regarding the first two, more

than 85% of responders identify them as important or very much important (values 4 or 5 in Likert scale) while for the rest the same opinion is little higher than 70%. Responses that consider all the above as insignificant are lower than 12% in all cases. This sets a grounded base that agile methods can improve team collaboration and cohesion skills.

Considering team organization issues, respondents believe that team members should have autonomy and authority in making decisions about topics related to their work and members performance evaluation should be also made by the same team members and this approach can improve project progress. Specifically, 25.7% of responders believe that teams' self-autonomy and authority to make decisions have a moderate effect in project success, 45.1% have a significant effect and 26.5% have an extreme effect on it. Considering the topic of self-evaluation, 27.7% strongly agree with it, 33% agree with it, 30.4% neither agree or disagree and 8.9% disagree or totally disagree. According to these, agile methods foster team team-autonomy and self-evaluation skills.

Next, it was examined if agile methods and scrum specifically can improve transversal skills such as problem analysis, change management and providing feedback. Specifically, 16.7% of responders strongly agree that their change management capabilities were improved, 55.3% agree that there was an improvement, 20.2% neither agree or disagree and the rest do not believe that there was any improvement. Regarding improving problem analysis skills, about 70% of the participants believe that their capabilities to analyze problems, evaluate alternatives and merging different approaches and proposals were clearly improved (values 4 or 5 in Likert scale) due to requirements for repetitions and retrospectives set by Scrum methodology. As for their capabilities improvement in providing feedback, 51.8% of responders believe that there was indeed an improvement, 18.8% consider there was an extreme improvement, 17.9% believe that was a small improvement and 11.6% believe that essentially there was no improvement. These also consist of a solid base that the aforementioned skills can also benefit from agile methods application.

Finally, participants were asked to evaluate the effect of Scrum methodology and generally of agile methods in planning and management skills of team members. They were asked to evaluate to what degree Scrum structure and processes e.g. sprints, user stories, burndown charts, actions for task prioritization improve their time management skills. In all relative questions, at least 70% of the responders declared that their time management capabilities were significantly or very much improved, about 25% declared that they had a moderate improvement while only 5% responded that there was no or insignificant improvement. Considering the improvement in planning and effort estimation capabilities, about 55% of responders declared that they had more than a significant improvement, about 25% declared that had a moderate improvement and less than 10% declared that had no or insignificant improvement. As such, it can be concluded that planning and management skills can also be improved.

From data analysis and considering RQ1 can be concluded that the implementation of agile methods in a project can benefit project team members and help them develop and grow both their transversal skills and team working characteristics such as cohesion, collaboration, effective communication, clear role assignments and mutual understanding.

4.2 Qualitative Analysis of Students' Feedback

As it was mentioned in the research methodology section, students were asked to provide a critical assessment of their work and on the challenges that they faced during their work. More specifically, they were asked to report on the challenges they faced, the risks that occurred, and their response to them, and, also, on the possible changes in their work practices that they could recommend. Finally, they were asked to report on the knowledge, competences, and skills acquired during this project.

According to the feedback received by students the most frequent problems/challenges the teams encountered were the following:

Student initial education and information: Initially, students were not familiar with the agile principles and the culture behind these principles. Even though agile practices were easily comprehensible, the adoption of them was challenging for most of the teams. Some students mentioned that the agile methods were introduced quickly and thus the provided material and guidance was inadequate in the beginning. We can provide numerous examples of the challenges faced by students in their projects. For example, this lack of agile culture or agile experience was expressed with many different symptoms:

- Inability to deliver user stories according to the planning and keep up a constant delivery pace
- Inability to deliver working and tested software at the end of each day (daily build) or the end of the sprint
- Daily communication with the product owner
- Use of visual tools for reporting the progress of the team (the team were instructed to use a collaborative online tool (e.g. Trello)
- Etc.

Similar findings have been reported in other studies such as in Gandomani et al. [11] and Nuottila et al. [28]. The aforementioned provides a clear answer in RQ2 about the necessity of agile education prior project execution.

Team organization and roles. It was a mandatory project requirement for all student teams to use agile roles. However, there were problems in team organization and assignment of roles, since students could not comprehend in practice the responsibility of each role. A typical paradigm, and maybe the most common, for a software development team, is that team members are assuming a vertical role e.g. acting as analysts, as designers, as programmers, etc. This is not the case in agile software development that requires cross-lifecycle developers, developers that execute all steps of a software development process. This fact caused confusion and communication difficulties, especially during the first weeks of the project. Further, in many cases, the roles of scrum master and product owner were not clear to the students, since scrum masters were reluctant to contact product owners (a role assumed by members of the teaching team) asking for clarifications or decisions. Finally, there were cases where the scrum master was acting as a project manager since he/she was dictating to other team members assignments and deadlines. The project manager is a role that is not supposed to exist in an agile project. The existence of a project manager in agile projects that has

been revealed in this research indicates a gap between theory and practice that is in line with the findings of other studies [41].

Nevertheless, these problems were expectable to a certain extent since capstone projects are short, in duration, projects, which implies that the team has not sufficient time to be well organized. The above analysis provides a clear answer to RQ3 that the agile team organization and team roles are difficulty to comprehend, especially in the beginning of the implementation, adversely affecting project execution.

Student commitment: HOU students are part-time students attempting to combine in most of the cases work, personal life, and studies, which is directly linked with time constraints. As was expected, most teams reported that the available time was not sufficient. From this fact, we can conclude that agile methods are better applicable when team members are working full-time on a project. This is in line with agile lean principles where context switching is considered as a source of waste [34].

Student distributed location. According to Calefato and Ebert [6], "Today, software engineering is characterized by two strong trends: agile and distributed. Both together are increasingly demanding and challenging teams and projects due to lack of discipline, insufficient transparency, agile "ping-pong," and thus overheads and rework." This was the case for all these capstone projects since they were administered during the spring semester of the academic years 2019–2020 and 202–2021, academic years where the HOU used exclusively distance learning, due to the COVID-19 pandemic. According to literature [42] distributed agile teams suffer from some problems, such as inadequate communication, insufficient knowledge sharing, project and process management discrepancies, technical issues, etc. All the above symptoms were present and they were reported by the students and especially insufficient knowledge sharing, project and process management discrepancies.

User stories effort estimation. Planning poker is an agile software estimation technique that has two main prerequisites: a) software developers have experience from previous projects, so they can estimate the size of the user story and b) they are familiar with group decision-making techniques that require consensus. Both the above were important constraints that were faced by the students since in most cases they didn't have relevant experience. Reaching consensus on the size of user stories is the core principle of Planning Poker [23]. To overcome the challenge of reaching consensus, students used averaging which is not considered an accurate method [12]. Overall, the application of effort estimation didn't produce accurate results since most of the teams reported deviations of the actual implementation from the sprint planning.

For many of the students, it was the first time they worked as part of a team during their studies and especially in a project that had technical, tooling, methodological and organizational challenges. As it was reported:

- The successful implementation of the project improved their collaborative skills, teamwork skills along with their technical skills, and this was considered as one of the most positive outcomes of the project. In relation to the above outcome, we are quoting from one of their reports that "Finally, with the delivery of the project, we have the feeling of satisfaction for the successful completion of such a demanding project."

- Knowledge sharing was considered as an important benefit from this capstone project and according to some of them, knowledge sharing and distribution of work accelerated the overall work outcome. This is in line with personal efficiency improvement objectives "to do more work in less time".
- The fact that the teams were self-organized allowed the students to assume full responsibility for their work including the planning. This empowered them and improved their organizational and planning skills.
- The fact that students had to work in a distributed fashion enabled them to use modern collaborative tools and to improve their ability to work within a team remotely, which was considered quite important.

The aforementioned provides a clear answer to RQ4 that project execution can be negatively affected by factors such as geographical distribution, task switching between project work and other activities and lack of team members' commitment.

5 Conclusion

This paper presents an implementation of a Scrum training model in an agile learning environment for university students. The methodology was implemented in a capstone project, using the Scrum method, as part of their software engineering/programming curriculum. To evaluate the impact of the agile methodology implementation a survey and a retrospective review were completed by with the students and analyzed by data analysis.

The results of the analysis indicated a great level of student satisfaction with the agile method implementation. It also have been indicated a positive impact on their performance as well as they improved both their software development skills and their transversal skills such as communication, collaboration, team cohesion, team self-organization and autonomy, problem-solving, creativity, and generally project planning skills. They became very responsible for their work by using collaborative tools, improving their ability to work within a team.

Knowledge sharing was considered as an important benefit from this capstone project as agile methodologies including continuous communication and iterative development, can promote knowledge management as stated also by Levy et al. [21].

Although there were problems in team organization and assignment of roles causing confusion and communication difficulties, especially in the beginning of the project continuous communication through iterations contributed to mutual learning and increasing progress among team members as also stated by Yang et al. [45].

In conclusion, according to the above findings, it can be assumed that Agile methodology could be of interest to anyone planning to implement agile procedures to projects in a context of software engineering curriculum, as well as in other disciplines in higher education [8], in order students to develop not only technical skills but also a set of transversal skills that are required in today's economy.

References

1. Aller, B.M., Lyth, D.M., Mallak, L.A.: Capstone project team formation: mingling increases performance and motivation. Decis. Sci. J. Innov. Educ. **6**(2), 503–507 (2008)
2. Association of American Colleges and Universities: Fulfilling the American dream: liberal education and the future of work (2018). https://aacu.org/research/2018-future-of-work. Accessed 07 Sept 2021
3. Bica, D.A.B., da Silva, C.A.G.: Learning process of agile scrum methodology with Lego blocks in interactive academic games: viewpoint of students. IEEE Revista Iberoamericana de Tecnologias del Aprendizaje **15**(2), 95–104 (2020)
4. Beck, K., et al.: The Manifesto for Agile Software Development (2001). http://agilemanifesto.org/. Accessed 25 Aug 2021
5. Bruegge, B., Reiss, M., Schiller, J.: Agile principles in academic education: a case study. In: 2009 Sixth International Conference on Information Technology: New Generations, pp. 1684–1686. IEEE, April 2009
6. Calefato, F., Ebert, C.: Agile collaboration for distributed teams [software technology]. IEEE Softw. **36**(1), 72–78 (2019)
7. Coates, K.: The value of soft skills: preparing the next generation of workforce for future work (2020)
8. Cubric, M.: An agile method for teaching agile in business schools. Int. J. Manag. Educ. **11** (3), 119–131 (2013)
9. Ding, D., Yousef, M., Yue, X.: A case study for teaching students agile and scrum in capstone course. J. Comput. Sci. Coll. **32**(5), 95–101 (2017)
10. Fitsilis, P., Lekatos, A.: Teaching software project management using agile paradigm. In: Proceedings of the 21st Pan-Hellenic Conference on Informatics, pp. 1–6, September 2017
11. Gandomani, T.J., Zulzalil, H., Ghani, A.A., Sultan, A.B.M., Sharif, K.Y.: How human aspects impress Agile software development transition and adoption. Int. J. Softw. Eng. Appl. **8**(1), 129–148 (2014)
12. Gandomani, T.J., Faraji, H., Radnejad, M.: Planning Poker in cost estimation in Agile methods: averaging vs. consensus. In: 2019 5th Conference on Knowledge Based Engineering and Innovation (KBEI), pp. 066–071. IEEE, February 2019
13. Gottipati, S., Shankararaman, V., Gan, S.: A conceptual framework for analyzing students' feedback. In: 2017 IEEE Frontiers in Education Conference (FIE), pp. 1–8. IEEE, October 2017
14. Grimheden, M.E.: Can agile methods enhance mechatronics design education? Mechatronics **23**(8), 967–973 (2013)
15. Harding, L.M.: Students of a feather "flocked" together: a group assignment method for reducing freeriding and improving group and individual learning outcomes. J. Mark. Educ. **40**(2), 117–127 (2017)
16. Hof, S., Kropp, M., Landolt, M.: Use of gamification to teach agile values and collaboration: a multi-week scrum simulation project in an undergraduate software engineering course. In: Proceedings of the 2017 ACM Conference on Innovation and Technology in Computer Science Education, pp. 323–328, June 2017
17. Kamat, V., Sardessai, S.: Agile practices in higher education: a case study. In: 2012 Agile India, pp. 48–55. IEEE, February 2012
18. Kavitha, R.K., Ahmed, M.S.I.: Knowledge sharing through pair programming in learning environments: an empirical study. Educ. Inf. Technol. **20**(2), 319–333 (2013). https://doi.org/10.1007/s10639-013-9285-5

19. Kumar, S., Ureel, L.C., Wallace, C.: Agile communicators: cognitive apprenticeship to prepare students for communication-intensive software development. In: 2015 Agile Conference, pp. 71–75. IEEE, August 2015
20. Lang, G.: Agile learning: sprinting through the semester. Inf. Syst. Educ. J. **15**(3), 14 (2017)
21. Levy, M., Hadar, I., Aviv, I.: Agile-based education for teaching an agile requirements engineering methodology for knowledge management. Sustainability **13**(5), 2853 (2021)
22. Liu, L.: Using generic inductive approach in qualitative educational research: a case study analysis. J. Educ. Learn. **5**(2), 129–135 (2016)
23. Mahnič, V., Hovelja, T.: On using planning poker for estimating user stories. J. Syst. Softw. **85**(9), 2086–2095 (2012)
24. Mahnic, V.: A capstone course on agile software development using scrum. IEEE Trans. Educ. **55**(1), 99–106 (2012)
25. Marder, B., et al.: 'Going agile': exploring the use of project management tools in fostering psychological safety in group work within management discipline courses. Int. J. Manag. Educ. **19**(3), 100519 (2021)
26. Mihalik, J.: Agile approach in higher education-a collaborative research project report. Opus et Educatio **6**(4), 470–476 (2019)
27. Milićević, J.M., Filipović, F., Jezdović, I., Naumović, T., Radenković, M.: Scrum agile framework in e-business project management: an approach to teaching scrum. Eur. Proj. Manag. J. **9**(1), 52–60 (2019)
28. Nuottila, J., Aaltonen, K., Kujala, J.: Challenges of adopting agile methods in a public organization. Int. J. Inf. Syst. Proj. Manag. **4**(3), 65–85 (2016)
29. O'Shea, M., Frohlich Hougaard, K.: Supporting key competence development: learning approaches and environments in school education. Input Paper (2019)
30. Parsons, D., MacCallum, K.: Agile education, lean learning. In: Parsons, D., MacCallum, K. (eds.) Agile and Lean Concepts for Teaching and Learning, pp. 3–23. Springer, Singapore (2019). https://doi.org/10.1007/978-981-13-2751-3_1
31. Paasivaara, M., Lassenius, C., Damian, D., Raty, P., Schroter, A.: Teaching students global software engineering skills using distributed scrum. In: Proceedings of 35th International Conference on Software Engineering, May 2013, pp. 1128–1137 (2013)
32. Patacsil, F., Tablatin, C.: Exploring the importance of soft and hard skills as perceived by IT internship students and industry: a gap analysis. J. Technol. Sci. Educ. **7**, 347 (2017)
33. Peha, S.: Agile Schools: How Technology Saves Education (Just Not the Way We Thought It Would) (2011). https://www.infoq.com/articles/agile-schools-education/. Accessed 10 Sept 2021
34. Poppendieck, M., Poppendieck, T.D.: Implementing lean software development: from concept to cash. Softw. Qual. Prof. **9**(3), 45 (2007)
35. Rodriguez, G., Soria, Á., Campo, M.: Virtual scrum: a teaching aid to introduce undergraduate software engineering students to scrum. Comput. Appl. Eng. Educ. **23**(1), 147–156 (2015)
36. Rodríguez, G., Soria, Á., Campo, M.: Measuring the impact of agile coaching on students' performance. IEEE Trans. Educ. **59**(3), 202–209 (2016)
37. Sachdeva, V.: Requirements prioritization in agile: use of planning poker for maximizing return on investment. In: Latifi, S. (ed.) Information Technology – New Generations. AISC, vol. 558, pp. 403–409. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-54978-1_53
38. Saltz, J., Heckman, R.: Exploring which agile principles students internalize when using a Kanban process methodology. J. Inf. Syst. Educ. **31**(1), 51 (2020)
39. Schmitz, K.: A three cohort study of role-play instruction for agile project management. J. Inf. Syst. Educ. **29**(2), 93–103 (2018)

40. Sharp, J.H., Mitchell, A., Lang, G.: Agile teaching and learning in information systems education: an analysis and categorization of literature. J. Inf. Syst. Educ. **31**(4), 269–281 (2020)
41. Shastri, Y., Hoda, R., Amor, R.: The role of the project manager in agile software development projects. J. Syst. Softw. **173**, 110871 (2021)
42. Shrivastava, S.V.: Distributed agile software development: a review. arXiv preprint arXiv: 1006.1955 (2010)
43. Takai, S., Esterman, M.: Towards a better design team formation: a review of team effectiveness models and possible measurements of design-team inputs, processes, and outputs. In: ASME 2017 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference (2017)
44. Thomas, D.R.: A general inductive approach for analyzing qualitative evaluation data. Am. J. Eval. **27**(2), 237–246 (2006)
45. Yang, J., Zhang, X.L., Su, P.: Deep-learning-based agile teaching framework of software development courses in computer science education. Procedia Comput. Sci. **154**, 137–145 (2019)
46. Wilkinson, S., Joffe, H., Yardley, L.: Qualitative data collection: interviews and focus groups. In: Marks, D., Yardley, L. (eds.) Research Methods for Clinical and Health Psychology, pp. 39–55. SAGE Publications, London (2004)
47. Williams, L.A., Kessler, R.R.: All I ever needed to know about pair programming I learned in kindergarten. In: Communications of the ACM. Association for Computing Machinery (ACM), New York (2000)