Various Shades of Teaching Agile

Necmettin Ozkan[1,3][a], Sevval Bal[2][b] and Mehmet Şahin Gök[3][c]
[1]R&D and Coordination Department, Architecht Information Systems, Istanbul, Turkey
[2]Information Systems Engineering Department, Sakarya University, Sakarya, Turkey
[3]Department of Business, Gebze Technical University, Kocaeli, Turkey

Keywords: Curriculum, Lecture, Education, University, Higher Education, Students, Agility, Scrum, Kanban.

Abstract: In parallel with the increasing demands for Agile in industry and academia, many lecturers have started teaching Agile Software Development in various programs. Teaching Agile at universities has both constraints, challenges and opportunities faced by both students and lecturers. Agile courses have been taught at universities by using different approaches that can mainly be divided into two categories: Teaching Agile in an agile way and teaching Agile in a conventional way. As the name calls for it, Agile should be taught in an agile way which is a challenging and still developing subject. Despite significance of Agile and Agile teaching, there is a lack of theoretical and comprehensive studies on Agile teaching and learning in an agile way. The existing literature seems to be more focused on practical and limited contexts as "case studies". In this study, we recommend and present various and agile ways to teach Agile by providing decision-tree-like paths with their reasonings for a course design. We aim to enlighten educators who are interested in teaching Agile within a higher education course while designing their courses.

1 INTRODUCTION

The penetration of Agile methods to varying domains including professional software development and various organization scales has been increasing. The increasing demand for Agile development in industry and academia has reinforced the need for teaching and learning agile approaches starting from university courses to get students ready in advance for their professional lives. Consequently, many lecturers have started teaching Agile Software Development in various programs (Hazzan, and Dubinsky, 2007). Some others prefer replacing traditional teaching methods with agile approaches while teaching various subjects such as construction, logistics, chemistry, and so on. They choose this because of the similarity and convergence between contemporary teaching techniques and agile principles. Generation Z is not satisfied with the traditional teaching techniques as they think that they "push" predefined content into their minds by a "holder of knowledge". They would rather prefer more interactive, mutual, dynamic, enjoyable, and pull-based versions of teaching that allow students to discover and develop their unique learning journey shaped based on their own needs. The students want to go beyond solely being in the role of absorber and to become active players in their classes. Parallel to this need, self-directed and project-based learning, problem-solving, teamwork, interpersonal and social skills, and leadership are becoming more crucial in teaching (Ozkan et. al, 2022). It is not surprising to see that such and similar contemporary methods are common among agile principles.

Consequently, agile approaches and techniques are being used in the modern education context more and more (Otero et al., 2020), and teaching Agile product/application/project/software development at university level has started to become more popular (Kropp and Meier, 2013). Many researchers have started to share their experiences of teaching Agile (Masood, 2018). Despite the positive picture and numerous benefits of Agile in education (Masood et. al, 2018), agile approaches in education are still underused (Otero et al., 2020).

[a] https://orcid.org/0000-0001-9876-8728
[b] https://orcid.org/0000-0002-1969-6517
[c] https://orcid.org/0000-0003-4072-2641

Ozkan, N., Bal, S. and Gök, M.
Various Shades of Teaching Agile.
DOI: 10.5220/0011997000003464
In *Proceedings of the 18th International Conference on Evaluation of Novel Approaches to Software Engineering (ENASE 2023)*, pages 709-717
ISBN: 978-989-758-647-7; ISSN: 2184-4895

Agile in education can mainly be categorized into two different branches; (1) Agile techniques can be used to teach various domains such as software development, construction, logistics, chemistry, and so on. (2) Lectures can focus on teaching the Agile approaches including the fundamentals, values, principles, practices, and tools either in an agile way or not. There are no unified but varying methods for teaching Agile (Matthies, Kowark, and Uflacker, 2016). Then, we can mainly divide teaching the Agile approaches into two sub-categories: (2.1) Teaching Agile in an agile way and (2.2) Teaching Agile in a conventional way.

Agile Software Development courses have been taught at universities by using different approaches including mixing traditional lecturing approaches with laboratories, reading literature on Agile, incorporating games, workshops, and interactive exercises in courses (Masood et. al, 2018; Werner, 2012) and in conventional ways such as learning from textbooks even though the students today may not prefer it. As the name calls for, Agile should be taught in an agile way (Devedzic and Milenkovic, 2011) which is a challenging and still developing matter.

The challenges faced in professional software development projects with the Agile approaches are also found in the domain of student learning. Teaching Agile at universities also involves various unique challenges and constraints such as adaptations to fit the context of education (Masood et. al, 2018). Complexity (introduction of entirely new concepts), under-defined problems (participants are not familiar with problem space), time-boxed development with frequent team meetings, and inevitable change (applying new knowledge) are among others (Mahnic, 2015). In addition, the short time-frames within semesters and regular lectures, the availability of external stakeholders (Schneider et. al, 2020), providing a realistic environment within the nature of academia, limited availability and commitment of information technology professionals (Linos et. al 2020) force the students and lecturers to find innovative and proper solutions to these challenges.

Teaching Agile can also have unique advantages for practitioners. The course taught for longer period of time, during one semester, can have more advantages over the short-term teaching in the sector in terms of affecting and penetrating the learners more deeply and properly. Additionally, teaching at universities can provide openness to trying different methods and learning cycle experiments. It is more possible that the students can have a purer and more meaningful aim than practitioners while learning Agile. Students can also have a more homogeneous level of knowledge at the initial stage of the courses; thus, their learning progress can be seen more clearly. In this regard, the students in the context of a particular university and students in the context of another university can resemble each other and the rate of transition from one context to another can be high. This case may pave the way for the practitioners in the field of education to make use of other practitioners' experiences.

Despite such significance and impact, there is a lack of theoretical and comprehensive studies on Agile teaching and learning. The existing literature seems to be more focused on practical and limited contexts as "case studies". In this study, we recommend and present varying teaching Agile ways by providing decision-tree-like paths in Figure 1 with their reasonings for a course design to enlighten educators who are interested in teaching Agile within a higher education course. In providing the recommendations, we will focus from time to time on possible constraints and opportunities faced by both students and lecturers.

2 THE COURSE DESIGN

In this section, we will cover possible and prominent manners related to the design of teaching Agile lectures. The possible options are not limited to the ones presented here, and those expressed in this paper do not aim to draw a line between the right and wrong approaches. It only aims to inspire and convey relevant and valuable knowledge to the people who teach such courses.

2.1 Principles

Madhuri and Goteti (2018) state that knowledge construction, collaboration among students, flexible and specialized curriculum, and building competencies are valuable in the Agile teaching context. Based on these values, the Agile approaches should be shaped according to the emergent needs of the students. In particular, syllabuses can dynamically and iteratively be formed with student feedback. In addition, focusing on the students and interactions, building their knowledge, collaborating with teachers and other students, and responding to changes in the course design can be considered by the lecturers while structuring their courses. Based on these principles and the Agile Manifesto, the following principles can guide Agile teaching. There can be different and additional principles, but the main idea is that the development of software, system, ideas, or teaching approaches can have common grounds to borrow from each other as they all exhibit complex system characteristics.

- P1: Students, as the customers of the classes, should be satisfied through the early and continuous delivery of valuable knowledge.
- P2: Lecturers should welcome changes in course content and syllabuses, even in the later times of the courses. For the sake of students' competitive advantage, teaching should benefit from change.
- P3: Delivering feedback to students and getting feedback from them within shorter timescales is essential.
- P4: Students, lecturers, and stakeholders should work together throughout the course.
- P5: Support and motivation should be provided to students in order to build and foster their learning environments.
- P6: The most efficient and effective method of conveying information to and within the class is face-to-face conversation.
- P7: Learning is the primary measure of progress.
- P8: Agile teaching should promote a sustainable pace.
- P9: Continuous attention to the technical excellence of teaching platforms enhances learning.
- P10: Simplicity is essential in teaching Agile.
- P11: Self-organizing teams should be encouraged among students.
- P12: Students and lecturers should reflect on how to become more effective, then tune and adjust their behaviour accordingly.

2.2 Theory

Every practice (should) bear(s) a theory behind it. For this reason, an Agile lecture may touch the theoretical side of Agile, before starting directly with the practice side. Thus, it is not possible to avoid the theoretical explanations of Agile. The main point of the theoretical part is about where it should start and end and how much it should be. However, there is no clear line about how much theoretical and practical lecture should be offered to the students, but it seems that both types should be included.

In practice, by the approaches adopted by many practitioners for training, Agile is regarded as a subject that does not go beyond the Agile Manifesto in terms of the starting point of the mental journey of /for Agile. In terms of origin, rather than starting with the Agile Manifesto that is commonly used to introduce Agile principles, the point should be taken from a wider and more extensive perspective to provide a better understanding of the agility concept (Ozkan, Gök, and Köse, 2020).

For a better understanding of the agility concept, lecturers should consider introducing what agile and agility mean, why we need them, and what their origins are. An agile mindset should be properly established before introducing any Agile methods, techniques, and rituals or before directly diving into a specific Agile method.

Agility is a phenomenon that has been in life, at least since the birth of living creatures. The living creatures are born with the pure agility capabilities they need by nature. Organizations are simply trying to bring the capabilities of this phenomenon into their fields. For this reason, the theory and philosophy of agility should go back to its origin, to its untouched (unspoiled) version in real life. Real-life covers and illuminates the need for agility and the nature and characteristics of complexity. It is a kind reminder that it is possible to reach a definition of agility that is purer, stronger, and more obvious in this way. Thus, The Agile course should start at least from this point (of view).

After anticipating the need for teaching an accurate Agile theory and that the Agile mindset, values, and principles are crucial in teaching Agile, we may face an issue; they are difficult to teach (Kropp and Meier, 2013). Maybe, this is why they are in most cases completely neglected in current course programs (Kropp and Meier, 2013). The issue is that without them, the practitioners can be deficient in how to be agile, their belief in being agile may stay weak and the lectures may become similar to Agile training excessively produced by/for the industry. Consequently, we have to face this reality; this subject should be taught properly to the students.

After the conceptual introduction to the agility phenomenon, human-made Agile artefacts, mostly known as Agile methods, take place to provide agility to organizations in concrete manners. When it comes to Agile methods to teach, they are varying options including Scrum, Kanban, Extreme Programming, and so on. At this point, teaching multiple methods instead of a single one can have many benefits. While the spectrum of view with a single method is limited to the relevant method only, it can be possible to go beyond such limits with a second method to enrich the perspectives of students. Thus, it can be possible to compare and combine multiple methods when possible and needed to get more a powerful one and/or when realizing context limitations with a particular method. Integrating multiple methods into the curriculum may also provide relevant university courses to stay more relevant and up-to-date

(Matthies, 2018). Specifically speaking regarding the method selection, Scrum, the most widely used and known one, can be preferred to support the students in their professional life in which Scrum is widely used. Already, most Agile teaching courses use and focus on Scrum (Matthies, 2018). In addition to Scrum, Kanban can be introduced to the students, as Kanban and combinations of it with Scrum have been increasingly used according to reports (Matthies, 2018). In terms of timing, as Kanban is rooted back to Lean principles rather than Agile principles, it may be better to introduce Kanban to the students after they gain experience with Scrum (Mahnic, 2015) which is closer to the basics of Agile approaches.

Agile can be praised a lot and negative aspects of Agile can be omitted in the classes (Kropp and Meier, 2013). In addition, to prepare the students for the "dark side of the coin" (Janes and Succi, 2012), the lecture can be followed by a caution stage about commercialized versions of Agile in the industry evangelized with economical concerns resulting in misleading concepts about agility (Ozkan and Gök, 2021) and about problems with the manifesto (Ozkan, 2019) in particular.

2.3 Reinforcement: Practice

Teaching is for learning (of students) (in relevance to P1, P7). The learning process has its unique way for each learner. The opportunity for each student to experience a development process, assess outcomes and iteratively apply their gained knowledge positively impacts learning success (The Joint Task Force on Computing Curricula, 2013). In this regard, rather than considering students solely as absorbers and teachers the only holders of knowledge (Barrows, and Tamblyn, 1980), the Agile teaching methods should let students create their learning journeys by putting the students' needs and engagement at the centre (Ozkan et. al, 2022) (in relevance to P1 and P5).

To build this individual and unique knowledge for each student, interactive learning can be helpful. With interactive learning facilitated by lecturers to accompany students' learning process, students can be engaged with higher motivation for a longer period of time. This deep interaction and long-time duration pave the way for the development of teamwork, interpersonal, leadership and social skills, self-directed learning, and problem-solving that are crucial in teaching (Fernandes, Dinis-Carvalho and Ferreira-Oliveira, 2021), as well as to realize what the Agile principles provide. Specifically speaking, project-based learning, gamification, laboratory environments to practice the theoretical lecturers especially when coding accompanied by the principles of Extreme Programming is executed, doing knowledge contests and evaluating students' level of knowledge, online platforms providing valuable content according to the needs of individual students, and debating between student groups (as in the case of Martin, Anslow, and Johnson (2017)) can be used to support students to create their individual learning process.

As an interactive learning technique, project-based learning and educational games may play crucial roles. Among the reinforcement methods, the most frequently used and prominent ones are already projects and applying gamification activities. Even though there are some challenges in terms of managing time limits in the classrooms, by playing educational games, engaging, interactive and relatively quick to learn and play, all students become pretty much at same level regarding their skills, allowing their equal participation and active contributions (Paasivaara, et. al, 2014). To refresh the students' theoretical knowledge, various gamification practices can be applied. Lego Scrum simulation exercises and creating objects from paper sheets during several sprints are among the prominent ones (Matthies, Kowark and Uflacker, 2016; Paasivaara, et. al, 2014) and are regarded positively by the students (Paasivaara, et. al, 2014).

In this study, we will give a special and dedicated place only for Scrum-project-based learning due to the space limitations.

2.3.1 Project-Based Learning

Structuring the Project-Based Learning

As Bruegge et al. (2009) point out, to penetrate and internalize the Agile values, a theoretical lecture alone is not enough. Mahnič (2015, b), Devedzic and Milenkovic (2011), and Kropp and Meier (2013) highlight the need for Agile approaches to be taught using practical projects. Real projects allow students to work on solving interesting and relevant real-world problems (Schneider et. al, 2020; Linos et. al, 2020). A real-world project with failures and successes motivates students to collaborate, explore and take responsibility (Feliciano, Storey, and Zagalsky, 2016).

Project ideas should be realistic and exhibit sufficient complexity and details to the students (Schneider et. al, 2020). The students should be allowed to form their problem and solution spaces in their projects. They can use Design Thinking methods in this step. Project ideas, solution designs, product backlogs, team formations, Sprint events, and team communication manners could differ across the teams whereas some other results including report templates, number of sprints, duration of sprints, expected delivery types from Sprints, having supervision meetings, and project assessment criteria can be common across the teams to synchronize the learning and teaching process with a sustainable pace (relevant to P8).

Some students appreciate working in self-organizing ways, while others may prefer to have more structural formations (Matthies, Kowark and Uflacker, 2016). However, self-organizing teams are located at the core of Agile principles and are crucial for Agile projects to be successful. To experience team dynamics and learn effective team management (Linos et. al, 2020), lecturers should allow students to self-organize in their projects (in relevance to P11) because self-organizing team formations support team members with a team spirit in a positive way (Matthies, Kowark and Uflacker, 2016).

Self-organizing team formations have some challenges independent from application domains; whether they are student or professional projects. For instance, uneven task distribution is an ongoing challenge in self-organizing teams (Matthies, 2018). In the context of student projects in education that are run in relatively short terms, self-organizing team members can face a high level of challenges, especially in the early stages of the projects. These challenges are likely to increase as the team members are not likely to have sufficient experience in the Agile approaches (assuming that all students start learning Agile within the same class). Classical team structures should not be an alternative for the Agile teams, and the student teams should be allowed to experience self-organizing somehow.

Customers play a crucial role in which they specify needs and acceptance criteria, review increments, and pivot the whole product in the desired direction. Such a role cannot be achieved by mimicking someone else. Rather than instructors or people from the course playing an artificial customer role, real customers may provide real challenges, feedback, and reactions to real solutions developed by the students. Having customers with real problems paves the way for establishing the products in more effective and efficient ways.

In this case, reaching such real customers when needed occurs as a new challenge. To make connecting to real customers easier, the students may be encouraged to contact their close environments that can be reached and accessed easily, such as their family members, other students, or their fellow citizens. Independent from the location of a university, thanks to the digital capabilities of this era, such profiles can be reached and involved in the projects. In this way, the students can apply quasi-real-world projects with such customer profiles. In some cases, including Masood, Hoda and Blincoe (2018), the students deal with real problems in various fields and work as a real Agile team to develop real solutions. This option can be more challenging in terms of finding and establishing such opportunities for the students than the former one. The last option can be to have projects which do not involve any real customers. In this case, someone can pretend to be a customer. However, playing any role, especially the customer role, can decrease the effectiveness of the projects.

In some cases, including the course by Mahnič (2010), we see that Product Owner (PO) and Scrum Master (SM) roles were assigned to the teaching team. Or, PO and SM roles can be performed by students (Matthies, 2018). Van Hout and Gootjes (2015) found the SM role being played by students has a positive effect in having greater sense of responsibility for outcomes and the self-organization of the teams. We argue that PO and SM roles can be more suitable to be included within the student teams. Having Scrum roles played by students, instead of a teaching team, does not at least harm collaboration in the teams (Matthies, Kowark and Uflacker, 2016). Such inclusion can support to be a compact Scrum team including PO, SM, and developers together. The teams may jointly select their PO and SM roles among the team members, with additional attention due to the inherent complexity and importance of these roles (Matthies, Kowark and Uflacker, 2016; Paasivaara et. al, 2014).

Special care should be taken of keeping the number of members of the Scrum teams within the recommended limits; not more than nine people including SM and PO, and not less than the number to handle the project to be done. The degree of closeness of the students in a team can be an advantage or disadvantage to the sense of "reality"; some students may treat their close friends differently/unprofessionally in work-related matters in the projects, because of that differentiating work-related-matters from friendship in the case of close friendships is not likely to be possible among some students.

When it comes to team formation period, there should be enough time before starting to the project. Thus, prospective teams can have more time to prepare for their projects and they can have an opportunity for team building with a shared aim, understanding, and synergy toward their projects (Ozkan et. al, 2022). There should also be some preparation steps for teams before their first Sprint as well. These steps can be used for setting up the teams, determining their products, preparing their product backlogs, and, like in Sprint 0, preparing the general infrastructure and architectural designs for the products. Thus, such steps can be performed in large and flexible time slots, before the first Sprint, which does not fully correspond to the Scrum Sprint structure. For instance, Linos et. al (2020) suggest locating a separate initial Sprint 0 for such cases. One of the things to remember is that the students do not have ready-to-start products, business units or customers to prepare product ideas at the first stage, rather, they are supposed to handle the preliminary preparations stages for their products.

Review of Projects

Giving the teams the authority to self-organize comes with some challenges. The new-born self-organizing student Agile teams may easily fall into mistakes and moreover they may not be able to recognize their mistakes. The student teams should be supported with intensive coaching, especially to align them with the theoretical lectures. In applications and adaptations of their processes according to their needs and contexts, intensive coaching and feedback are vital to learn from their mistakes, especially if the team cannot self-correct itself after a while (Matthies, Kowark and Uflacker, 2016).

To provide this consultancy, each or some Scrum events may be performed in the presence of a tutor and/or lecturer to answer questions and give advice (Matthies, Kowark and Uflacker, 2016). In addition, having students who have Scrum knowledge beforehand can help get support in this regard. As such an accompaniment requires a considerable increase in the amount of time spent on supervision and support of the teams (Matthies, Kowark and Uflacker, 2016), external support outside of the class can also be sought. Linos et. al (2020) exhibit a successful case for the professionals who voluntarily and regularly engaged with the student Scrum teams throughout their course. As reported, even though this practice has limitations such as the time commitment of the professionals, it also brought some benefits to stakeholders such as receiving some gratification, getting first-hand experience in teaching, learning, mentoring, leadership, revisiting fundamental concepts and bringing some ideas back to their office to improve their processes.

In absence of the support from the industry, it may be possible for the lecturers to accompany the online sessions of Scrum teams. Reviewing the projects during these additional online sessions outside of class hours may create available spaces and time.

The projects can be evaluated only by students, lecturers or by all participants including students, lecturers, and other stakeholders to reach a fully comprehensive evaluation. The evaluations made by students only may danger a fair evaluation due to the friendships between them. The evaluation scores should be assigned team-based according to the agile principles.

2.4 Means of Communication

Well-functioning Agile teams are advised to work together at the same place as it is a key standard for the Agile teams (Gren, Torkar, and Feldt2017). Meanwhile, the case of individually dispersed teams has become common for many organizations, especially after Sars-Cov-2 Pandemic and the shift to working from home. However, remote work brings several new challenges for the Agile teams, especially in terms of in-person interactions, such as a lack of face-to-face communication (Ozkan, Erdil, and Gök, 2022). Even though many tools support interactions efficiently, they are still not as effective as face-to-face conversation (Mancl, Fraser, 2020). Non-verbal communication that carries a lot more expressiveness like facial expressions, gestures, posture, proximity, tone of voice, pitch, etc. compared to verbal communication has been lost in this context (Mehrabian, 2017). Therefore, we strongly recommend conducting Agile teaching classes in a face-to-face manner (relevant to P6). Hybrid methods can be preferred when there are difficulties in accessing the lectures, for example, if a guest participant cannot attend them.

The classes can be supported with instant messaging channels and online tools like Trello to keep the course content topics as a Product Backlog List and survey tools that collect opinions and feedback anonymously, statistically and when needed. Online meeting tools can be used to conduct meetings for students and to easily integrate mentors/coaches from distance into their meetings (in relevance to P3, P4, and P9).

2.5 Feedback Loops

Having a continuous improvement spirit at its heart, Agile teaching should have a mechanism to improve the teaching methods and approaches (in relevance to P12). The data can be collected at varying parts of the classes with various manners to identify and make

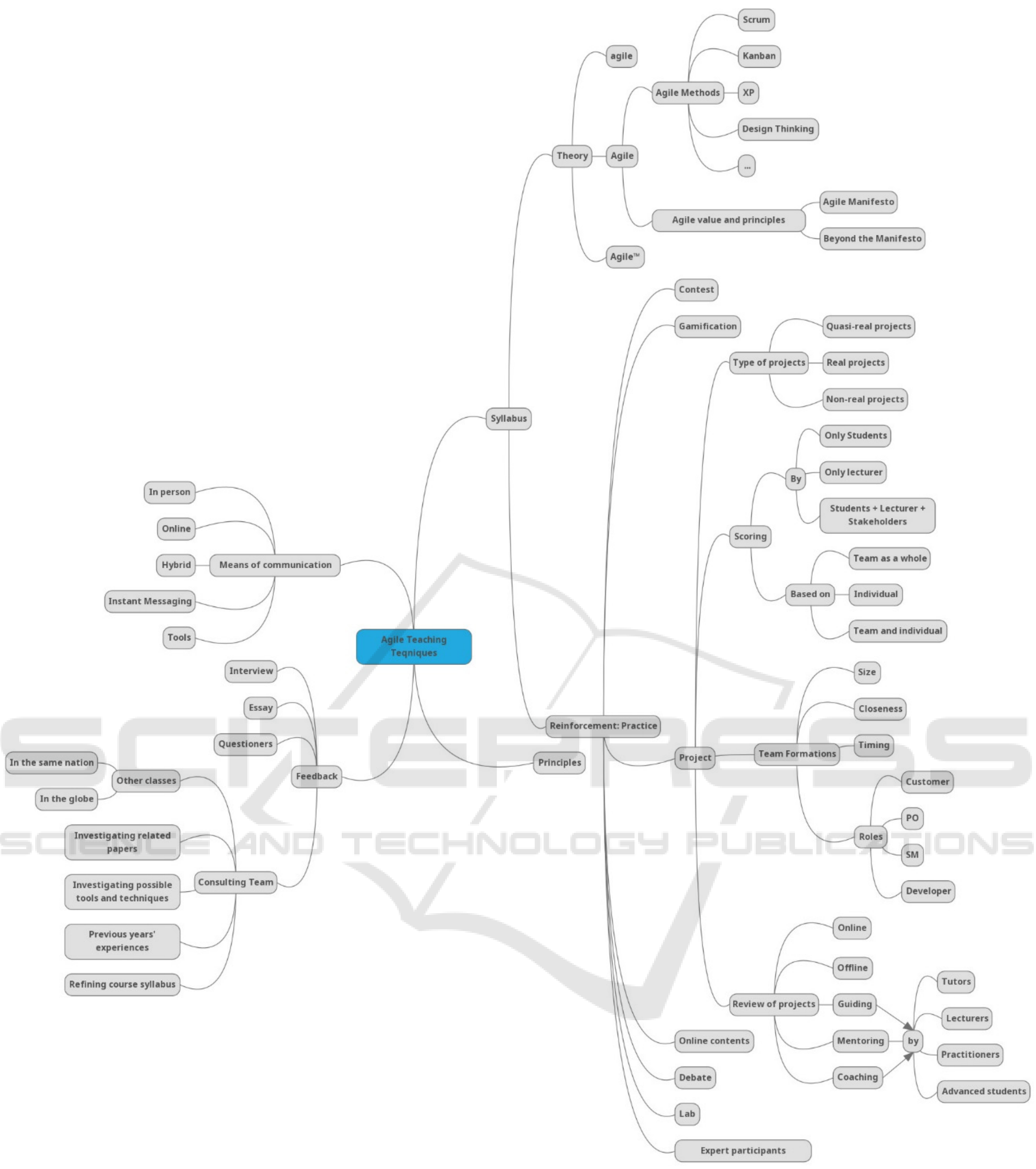

Figure 1: Agile Teaching Ways in a Decision-tree-like Path.

any necessary adjustments based on feedback from students (in relevance to P3).

Two types of surveys have been mainly observed by the authors of this paper: First one is to get feedback about key aspects of Agile practices. What we have seen about this type is that the students do not have sound references or baselines to compare the Agile practices with others. At best, the results from such surveys provide inputs to the body of knowledge with minor or no significant differences from current studies; mostly, the majority of the students point out a positive attitude towards the Agile practices. The second type is about Agile teaching and students' perceptions about and satisfaction with the Agile courses. These types of surveys appear to be efficient in improving such classes, especially when their results are shared with other practitioners.

Lecturers may ask students to write a learning diary (Paasivaara et. al, 2014) and/or essays (Masood, Hoda and Blincoe, 2018) about each teaching exercise. In addition, instructors can keep a personal journal during the classes with observations and challenges faced (Linos et. al, 2020).

Constructing a consulting team of students and the lecturer can be one another option in order to shape the lectures and to get feedback from the students. This team can focus on improving the lecture by refining the course syllabus and investigating the best and good practices from other similar and relevant lectures in both the context of the same country and the globe. Investigating related papers, possible tools and techniques to use, and evaluating previous years' experiences of the same lecture, if any, could be some other activities of this dedicated team. By getting and integrating fresh and accurate feedback on the way, the lecturer can make changes in course content and syllabus, even in further weeks of the courses to harness such changes for the sake of students' competitive advantage (in relevance to P2).

3 CONCLUSIONS

Agile methods and approaches continue reinforcing and expanding its place in today's complex world and business processes. The systems are being developed and take place in a complex world with highly vague environments including the concept of human being, who has high complexity. The agility phenomenon, which aims to cope with such complexity, has entered many large and small organizations in one way or another. Learning the agility phenomenon, which is becoming de-facto standard for dealing with complex world and processes, also contains complexity. The fact that the concept of agility is abstract and new, the area it wants to address is complex, and the level of human contact in its learning is high can be counted among the factors that increase the challenge in its teaching. Despite these challenges, it is inevitable that the agility phenomenon will become a standard in many sectors, and that it will be commonly used in the university education as well. Classical teaching methods are getting old and losing their effectiveness and the need to teach agility as a subject at the "early" stages of life has started to come to the surface more clearly.

There are diverse applications of Agile teaching and learning mentioned in the literature. However, they are mostly case studies, posing a lack of theoretical and comprehensive dimensions on the subject to guide educators. It makes it difficult for a lecturer who wants to design such a course to choose suitable methods according to needs among many options. In this study, considering these basic needs, we discussed multiple Agile teaching methods, emphasized the significance of some of them, and discussed them in detail. Thus, during the teaching of this subject, this paper aims to inspire and guide those who design, operate, and take such courses.

REFERENCES

Barrows, H. S., & Tamblyn, R. M. (1980). Problem-based learning: An approach to medical education (Vol. 1). Springer Publishing Company.

Bruegge, B., Reiss, M., & Schiller, J. (2009, April). Agile principles in academic education: A case study. *In 2009 Sixth International Conference on Information Technology: New Generations* (pp. 1684-1686). IEEE.

Devedžić, V. (2010). Teaching agile software development: A case study. *IEEE transactions on Education*, 54(2), 273-278.

Feliciano, J., Storey, M. A., & Zagalsky, A. (2016, May). Student experiences using GitHub in software engineering courses: a case study. *In 2016 IEEE/ACM 38th international conference on software engineering companion (ICSE-C)* (pp. 422-431). IEEE.

Fernandes, S., Dinis-Carvalho, J., & Ferreira-Oliveira, A. T. (2021). Improving the performance of student teams in project-based learning with scrum. *Education Sciences*, 11(8), 444.

Gren, L., Torkar, R., & Feldt, R. (2017). Group development and group maturity when building agile teams: A qualitative and quantitative investigation at eight large companies. Journal of Systems and Software, 124, 104-119.

Hazzan, O., & Dubinsky, Y. (2007). Why software engineering programs should teach agile software development. *ACM SIGSOFT Software Engineering* Notes, 32(2), 1-3.

Janes, A. A., & Succi, G. (2012, October). The dark side of agile software development. In Proceedings of the ACM international symposium on New ideas, new paradigms, and reflections on programming and software (pp. 215-228).

Joint Task Force on Computing Curricula. (2013). Computer Science Curricula 2013. ACM/Association for Computing Machinery.

Kropp, M., & Meier, A. (2013, May). Teaching agile software development at university level: Values, management, and craftsmanship. *In 2013 26th International Conference on Software Engineering Education and Training (CSEE&T)* (pp. 179-188). IEEE.

Linos, P. K., Rybarczyk, R., & Partenheimer, N. (2020, October). Involving IT professionals in Scrum student teams: An empirical study on the impact of students' learning. In 2020 *IEEE Frontiers in Education Conference (FIE)* (pp. 1-9). IEEE.

Linos, P. K., Rybarczyk, R., & Partenheimer, N. (2020, October). Involving IT professionals in Scrum student teams: An empirical study on the impact of students' learning. *In 2020 IEEE Frontiers in Education Conference* (FIE) (pp. 1-9). IEEE.

Madhuri, G. V., & Goteti, L. P. (2018, November). Adopting agile values in engineering education. *In 2018 IEEE 6th International Conference on MOOCs, Innovation and Technology in Education (MITE)* (pp. 103-106). IEEE.

Mahnic, V. (2010). Teaching Scrum through team-project work: Students' perceptions and teacher's observations. *International Journal of Engineering Education*, 26(1), 96.

Mahnič, V. (2015). From Scrum to Kanban: introducing lean principles to a software engineering capstone course. *Inter. J. of Engng. Educ*, 31(4), 1106-1116.

Mahnič, V. (2015b). Scrum in software engineering courses: an outline of the literature. Global Journal of Engineering Education, 17(2), 77-83.

Mancl, D., & Fraser, S. D. (2020, September). COVID-19's Influence on the Future of Agile. In Agile Processes in Software Engineering and Extreme Programming–Workshops: XP 2020 Workshops, Copenhagen, Denmark, June 8–12, 2020, Revised Selected Papers (pp. 309-316). Cham: Springer International Publishing.

Martin, A., Anslow, C., & Johnson, D. (2017). Teaching agile methods to software engineering professionals: 10 years, 1000 release plans. In Agile Processes in Software Engineering and Extreme Programming: 18th International Conference, XP 2017, Cologne, Germany, May 22-26, 2017, Proceedings 18 (pp. 151-166). Springer International Publishing.

Masood, Z., Hoda, R., & Blincoe, K. (2018). Adapting agile practices in university contexts. *Journal of Systems and Software*, 144, 501-510.

Matthies, C. (2018, June). Scrum2kanban: integrating kanban and scrum in a university software engineering capstone course. *In Proceedings of the 2nd International Workshop on Software Engineering Education for Millennials* (pp. 48-55).

Matthies, C., Kowark, T., & Uflacker, M. (2016, June). Teaching agile the agile way—employing self-organizing teams in a university software engineering course. *In 2016 ASEE International Forum*.

Mehrabian, A. (2017). Nonverbal communication. Routledge.

Naik, N. (2014, October). A comparative evaluation of game-based learning: Digital or non-digital games?. In *European Conference on Games Based Learning* (Vol. 2, p. 437). Academic Conferences International Limited.

Otero, T. F., Barwaldt, R., Topin, L. O., Menezes, S. V., Torres, M. J. R., & de Castro Freitas, A. L. (2020, October). Agile methodologies at an educational context: a systematic review. *In 2020 IEEE Frontiers in Education Conference (FIE)* (pp. 1-5). IEEE.

Ozkan, N. (2019, November). Imperfections underlying the manifesto for agile software development. In 2019 1st International Informatics and Software Engineering Conference (UBMYK) (pp. 1-6). IEEE.

Ozkan, N., & Gök, M. S. (2021). Towards the End of Agile: Owing to Common Misconceptions in the Minds of Agile Creators. In ICSOFT (pp. 224-232).

Ozkan, N., Gök, M. Ş., & Köse, B. Ö. (2020, September). Towards a better understanding of agile mindset by using principles of agile methods. *In 2020 15th Conference on Computer Science and Information Systems (FedCSIS*) (pp. 721-730). IEEE.

Ozkan, N., Erdil, O., & Gök, M. Ş. (2022, January). Agile teams working from home during the covid-19 pandemic: A literature review on new advantages and challenges. In Lean and Agile Software Development: 6th International Conference, LASD 2022, Virtual Event, January 22, 2022, Proceedings (pp. 38-60). Cham: Springer International Publishing.

Ozkan, N., Özcan-Top, Ö., Bal, S., & Gök, M. Ş. (2022, December). Teaching Agile in an Agile Way: A Case from the First Iteration in a University. *In 2022 3rd International Informatics and Software Engineering Conference (IISEC)* (pp. 1-6). IEEE.

Paasivaara, M., Heikkilä, V., Lassenius, C., & Toivola, T. (2014, May). Teaching students scrum using LEGO blocks. *In Companion Proceedings of the 36th International Conference on Software Engineering* (pp. 382-391).

Schneider, J. G., Eklund, P. W., Lee, K., Chen, F., Cain, A., & Abdelrazek, M. (2020, October). Adopting industry agile practices in large-scale capstone education. *In 2020 IEEE/ACM 42nd International Conference on Software Engineering: Software Engineering Education and Training (ICSE-SEET)* (pp. 119-129). IEEE.

Van Hout, M., & Gootjes, G. (2015). SCREAM! An integrated approach for multidisciplinary design teams in Higher Education. *Edulearn15 Proceedings*, 2157-2167.

Werner, L., Arcamone, D., & Ross, B. (2012). Using Scrum in a quarter-length undergraduate software engineering course. *Journal of Computing Sciences in Colleges*, 27(4), 140-150.