

Create your own agile methodology for your research and development team

Enikő Ilyés

Eötvös Loránd University
in Budapest

sny. Pázmány Péter 1/C, 1117 Budapest, Hungary
Email: ilyese@inf.elte.hu

Abstract—Agile methodologies conquer space beyond their industrial use. For applying them in a situation other than classic software development, one should first assess the features of this specific environment. As a next step, elements from various well-known agile methodologies (roles, events, products) can be considered as building blocks. These elements combined in a new way, adopted appropriately result in a specific, own agile methodology. In this study, we present a list of aspects that one should consider when creating a specific agile methodology for a R&D team. Our own agile methodology created for the txtUML R&D team from the Faculty of Informatics of Eötvös Loránd University is built along this list of aspects. Known and new agile elements were included in this specific methodology and are explained in this article in detail. The txtUML R&D methodology has been used with satisfaction since 2018, as evidenced by backward surveys.

I. INTRODUCTION

THE introduction of agile methodologies has been of great interest in recent years in Central Europe [9], [10], [11]. They are used not only in software engineering, but also in other fields - such as economics-, and not only within the industry, but in research and development as well [1], [2], [3], [4], [6]. Naturally, every time agile methodologies are applied in a new area, the original agile framework needs to be adapted to the specifics of the new field.

In 2018 we decided to develop a specific agile methodology to manage the work of a research and development team at the Eötvös Loránd University. In the research and development teams of our university teachers, research staff and students work together. Students can be PhD students, master's students or undergraduates. The diversity of the team members influences the frequency of meetings. The fluctuation of team members results in different levels of knowledge, differences in the ability to work independently, as well as varied levels of motivation. Therefore, many aspects of collaboration need to be addressed in order to ensure the efficient operation of these teams.

In the second chapter of this paper we summarize some examples of agile methodology used within the research and development teams of other universities. In the third chapter, we present a list of aspects we used and recommend to be taken into consideration before creating a new methodology

for a research and development team. In the fourth chapter we present our own methodology, along with the related feedback. Finally, the fifth chapter summarizes the results.

II. AGILE METHODS IN R&D TEAMS

If we look at the characteristics of agile methodologies, we can see that they fit properly with the nature of research and development teams. Maik Sayfert, who has more than 10 years real project experience with agile software development, declares: “When considering research as an area of high uncertainty and open results, it smells like agile methods are perfectly suited. From all methods I personally can imagine Scrum with short/mid sized iterations it the best match, as Kanban (flow based) is more suited to connect several disciplines/departments and XP is more technically driven.” [15]

Jeff Sutherland, the inventor and Co-Creator of Scrum reports in 2016: “Many of the leading research labs in the U.S. use Scrum. The one I have worked with most often is the John’s Hopkins Applied Physics Lab, the leading Naval research lab. Their research plan is their backlog. They map it out like an AI tree. Time boxing the research stories gets them done twice as fast. And the quality of the research is much higher with daily meetings.” [16]

Some articles present concrete cases using agile methodology to manage a research and development team.

There are several cases where the introduction of agile methodology is related to the will of an industry partner, who uses agile methodologies from earlier. For the agile coordination of the research team and the industrial partner, traditional roles may be modified or extended. For example H. Sharp and co-authors in [1] reports such a case: “an important adaptation was the inversion of the semantic of the product owner, for in our context he is a member of the lab allocated at the client. This change was made because of the difficulties associated with having a client in the lab.” Some research teams have developed new agile methodologies for improving collaboration with industrial partner, such as the the Agile Research Network collaboration model. The creators of this model found the following key challenges in this kind of collaboration: timeliness, relevance, rigour, access. Based on this recognition they build the ARN model, which consists of the following sections: Collaboration Kick-off, Investigation of the focus area, Implementation, Evaluation. [2]

The research project was supported by the European Union and co-financed by the European Social Fund (EFOP-3.6.3-VEKOP-16-2017-00002).

The introduction of agile methodologies in research and development teams are not necessarily motivated by cooperation with industry but rather by coping with other challenges, such as distributed teams. In such cases, new roles other than the original Scrum roles are introduced (eg. Unit Coordinator, Research unit, etc.). Events are mostly online, such as: Daily Scrum is done via instant messaging. [3]

The inventors of the SCORE agile method for research and development teams addressed another challenge: “Working with and mentoring Ph.D. students is the central activity in running an academic research group, with two broad goals: (1) to collaboratively produce high-quality research results, and (2) to help students to become independent researchers capable of working at research labs or academic institutions. At first, we followed a simple, fairly typical approach to mentoring: we met once or twice per week with each student in roughly half-hour or hour-long slots. Unfortunately, as the number students grew from two or three each to six or seven each, and as our outside commitments steadily increased, our simple approach reached its limits.” [4]

SCORE methodology uses elements such as status meetings (daily Scrum), on demand technical meetings, weekly reading group, weekly lunch; The core idea is to keep status separate from research. “The Scrum meeting is for status, and the on-demand meetings are for solving unforeseen problems. Keeping the two activities separate allows them to be undertaken more efficiently.” [4]

Based on the idea of “the last step of having a successful team is to build a supportive environment.” the authors of [5] proposed an approach for planning research projects, considering the use of 2-type fuzzy numbers for research project planning based on Scrum.

Many positive experiences were drafted regarding using agile methods in research and development teams. Some examples: “using Scrum contributed to the education of team members that were interns (students) and/or autonomous professionals (freelancers)” [6] “The involvement and commitment of members of the team with the results increased (...). We also realized that team members were motivated and open to changes in work.” [1] “Everything is visible to everyone, team communication improves, a culture is created where everyone expects the project to succeed.” [3] “Students say they are more productive, more enthusiastic about research, and have better interactions with other students and with their adviser. Students reported that there is now a real sense of community in the group that was never there before.” [4] “Though SCORE is conceptually simple, its benefits to us have been significant. (...) when students are struggling, it often only takes a day or two to realize something is not right, and to begin to address it. Our time is spent far more effectively.” [4]

The text-based feedbacks sound good, but unfortunately it is still a challenge to find a metric to evaluate formally the gain achieved by using agile methodologies in R&D teams [1]. Another claim is that university students do generally not have the maturity to understand the agile practices and their consequences. “The suitability of agile methods in education

is thus an ongoing debate.” [7]

III. ASPECTS TO BE ANALYZED WHILE CREATING A SPECIFIC AGILE METHOD FOR A R&D TEAM

Based on literature and our experiences, we have gathered a list of aspects to take into consideration if someone wants to create an appropriate agile methodology for their research and development team.

- 1) Objectives of the research and development team: As we have mentioned, a research and development team may have several goals with varying degrees of emphasis. The team’s primary goal may be to develop a new product, but another goal may be to involve students more effectively in the research, or even to recruit researchers and teachers. These goals need to be clarified, prioritized and then tailored to the research team’s workstyle.
- 2) Proportion of research and development tasks: The new methodology should sufficiently support team members in both research and development tasks.
- 3) Type of members: The group may include undergraduate, master's and PhD students, as well as research staff and teachers. These individuals have varying levels of prior knowledge, may differ in terms of their ability to work independently, and may be able to devote different amounts of time to the R&D team's work.
- 4) Motivation of team members: Team members participate in the R&D work with different motivations. Some students volunteer on the team to get to know the world of research, while others have more specific goals, such as writing their thesis. There are also students who work for university credits and others to receive a scholarship. Research staff is mostly motivated by their interest in the topic of research and possible results. Teachers are primarily motivated by recruiting teachers and researchers. When developing a unique methodology, all these motivations need to be taken into account to create an engaged team.
- 5) Fluctuation of team members: It is important to consider turnover within a team. For example, while an undergraduate student stays on the team for an average of half a year, a PhD student is likely to remain a team member for several years. New team members have to be integrated into the team as early as possible, and keeping existing knowledge within the team when a significant member leaves is also very important.
- 6) Number of team members: The number of people who have to work together has always been a very important aspect of managing a team. Large groups can benefit from a greater knowledge base, but communication can become more difficult. The efficiency of events and different roles assigned is highly influenced by the number of people on the team.
- 7) Frequency of meetings: As students, teachers and researchers perform many different tasks at the university during working hours, it is not easy to organize team meetings. The team's new working methodology will be

strongly influenced by the possible number and timing of meetings.

- 8) Relation between tasks: It is recommended to take into consideration how the potential research and development tasks are built on each other and how connected they are to each other. For example, if some tasks have the same topic, it is recommended that team members taking on these responsibilities work together and share their experience. Perhaps the creation of subgroups along task topics would be effective, including the appointment of a subgroup leader.
- 9) Documentation obligations: The level of detail needed in documenting the group's activities is an important factor. For example, if a company financially supports the research and development team, they may have to meet specific documentation requirements. Another purpose of detailed documentation could be to attenuate the effects of high turnover among team members; the group needs to ensure that the knowledge of its members remains in the team even after they leave. On the other hand, if the team does not need documentation for any purpose, creating a detailed record may be a waste of the members's valuable time.
- 10) External partners: Beyond the members of the research team, there may be a sponsor, a customer or a partner whose expectations they need to meet, such as performing a certain task by a certain deadline. Working with an external partner requires special attention: Who is the partner? What deadlines have they set for the team? How can the team achieve the fulfillment of the partner's requirements without sacrificing its own goals?

Please note, that the list is not exclusive. For example, there is a growing phenomenon that the members of the research team are not co-located, which poses additional challenges while creating a new methodology

IV. OUR SPECIFIC AGILE METHOD

A. The *txtUML* R&D team

The Faculty of Informatics at the Eötvös Loránd University currently has several functioning research and development labs. The Model Driven Development Research Lab has been operating since 2014, with a project called *txtUML* as one of its primary undertakings. A brief description of the software is: "The name *txtUML* stands for textual, executable and translatable UML. It is an open source project with the goal to make model driven development easier." (For a more detailed description see the website of *txtUML* [12]. Before we have created the agile methodology for the *txtUML* research team, I attended their meetings for half a year. I considered that the work of the team was not efficiently organized. The meetings consisted of telling everyone what they were doing last week and what problems they encountered. When someone came in, he/she talked about his/hers problems for a long time. One or two people who understood the topic discussed technical details, while others waited quietly. This took 2 hours, so the

meetings were long and demotivating. Students who joined the team from the beginning of the semester could not involve in the discussions till the middle of the semester. There was no time allocated for them to learn from the experienced students during the regular meetings. This would have been a great necessity and possibility, since the students in the team had different level of understanding of the subject, and there were also undergraduates, masters and doctoral students in the team. Many types of motivation were present in the team thanks to the many types of students - some of them joined the research group as a course named "Software Technology Lab.", others wrote their theses, etc. I saw that the methodology does not take into account the presence of different motivations and, as a result, there is not enough commitment to work. The research team was supported by a tender that required weekly logging and fulfillment of predefined goals. I considered that this methodology did not help students enough to fulfill this expectation. As a result of my observations I collected all the important factors which made the actual method work ineffectively. This list was the basis of the list I presented in the previous chapter.

In 2018 we decided to introduce an agile methodology to manage the *txtUML* team's work more efficiently. Taking into consideration the particularities of our team, designing our own agile methodology using elements from other agile methods as building blocks and inspiration seemed to be the best solution.

The particularities of our team were:

- 1) Objectives of the research and development team: The development of *txtUML* software and the involvement of students in the world of research are equally important objectives for us.
- 2) Proportion of research and development tasks: We predominantly perform development tasks (80%), with research tasks being less frequent (20%).
- 3) Type of members: Our team is currently comprised of 9 undergraduate students, 1 PhD student and 1 teacher. The individuals have different levels of prior knowledge regarding *txtUML*: three senior students who have been working on our team for more than a year; three students who joined the team six months ago and three completely new students.
- 4) Motivation of team members: The 9 undergraduate students gain 4 credits/semester for their work on the team, while 4 of the students also receive a scholarship as a result of participating in this research. The PhD student improves her PhD work with the help of this team, and the one teacher's primary motivation is the recruitment of new teachers and researchers.
- 5) Fluctuation of team members: 3 students are expected to leave the team at the end of the semester, while others may stay for 2 more semesters.
- 6) Number of team members: 10
- 7) Frequency of meetings: A 2-hour meeting/week is attainable.
- 8) Relation between tasks: The research and development

tasks of our team can be categorized in four groups. The following terms are used to refer to their topic: language front-end, visualization, C ++ export and model testing.

- 9) Documentation obligations: Due to the scholarship some student receive in relation to their work with txtUML, we must log the progress of the team members on a weekly basis. As three senior students leave the team at the end of the semester, we need to ensure that their knowledge is “saved” (documented) before they leave the group.
- 10) External partners: Currently, our only external partner is the scholarship program. At the beginning of each semester, we have to make commitments and present out progress at the end of the semester.

As a member of the txtUML team with a “team coach” role, I design a new agile method for the txtUML R&D team based on the team's characteristics. The methodology is presented below.

B. txtUML's agile methodology

In the following, I will describe the agile methodology designed for the txtUML team and used by it since February 2018. I will group the elements of the methodology along roles, events and artifacts. Regarding each element, I will clarify the purpose for which it was introduced and between {} signs I will specify which aspects from the list “Aspects to be analyzed while creating a specific agile method for a R&D team” (Chapter 4.) are mostly related to the mission of that specific element.

1) *Roles*: There are five different roles in the txtUML team: project leader, Scrum master, technical leader, subgroup leader, developer.

Project leader: Represents the goals of the research team towards the university leadership and applications. He is the main contact person of the team, holds up the results of the team (for example, in the case of “University Open Day”, “Researchers’ Night” events). He has a word in any decision-making. His role provides a solid framework and direction for the research team. He participates on group meetings and is also easily accessible between them. { 1, 4, 9, 10 }

Scrum master: The name of the role was inspired by the Scrum methodology. Similarly, the responsibility of this person is to protect the team’s operational values. This can be achieved by initiating and coordinating of various events (for example, initiating and coordinating weekly Scrum for effective discussions; training the members to effectively report their blockers, etc.) and leading by example. The Scrum master participates on group meetings. She monitors the situation of each team member and the dynamics of the whole group. She does not have software development tasks. { 3, 4, 5, 6, 9 }

Technical leader: Has a comprehensive view of the software and is also familiar with many details. Coordinates team members’ work from a technical point of view. All important technical decisions must be agreed with him. It also has a mentoring role: to pass on the knowledge and experience of the previous developers. He is primarily available at group

meetings, but often responds to questions through online communication as well. { 1, 2, 3, 4, 5, 6, 7 }

Subgroup leader: The research and development tasks of our team can be distinguished in four groups. The following terms are used to refer to their topic: language front-end (3 members); visualization (2 members); C ++ export (2 members); model testing (2 members). Students working on the same topics form a subgroup led by a senior student who has the most knowledge and experience in that specific field. This student designates smaller research and development issues to subgroup team members and gives developers mentoring during meetings and through online communication. Usually he/she performs development tasks as well. During meetings he/she organizes discussions on its own topic to transfer its knowledge and presents the results of its subgroup to the entire research group. { 1, 3, 4, 5, 6, 7, 8 }

Developer: As a first step in joining the team, every new member gets acquainted with txtUML from the “txtUML user” perspective. Than he/she selects the area in which he wishes to contribute to the research and development work and thus becomes a member of a subgroup. With the help of the subgroup leader he/she chooses an appropriate task and contributes to the expansion and refinement of the txtUML software. Developers also participate on group meetings where they learn from others and pass on their experience as well. { 1, 3, 4, 5 }

Remark: It may be strange that the role of the Product Owner is not strongly emphasized, though it is a key role in the Scrum methodology for example. According to my observation, in the university environment, the role of Product Owner is often less powerful, perhaps because the role of the customer is not as definable as in the industrial environment. Most often, the teacher fills the role of the customer. For him/her, however, the functionality of the product is not the only focus, but there are also pedagogical goals. In our case, the role of the Project leader was as close as possible to the Product Owner: he pointed out the development directions; the developers, subgroup leaders talked about the functions to be realized and their acceptance criteria with him.

2) *Events*: The members of the txtUML team meet weekly in a two-hour session. Events related to the agile methodology are integrated into this meeting. In the intervening period everyone works individually on their own task and uses the slack ([13]) tool for online consultation, if needed.

Our four major agile events are: preparation, weekly routine, retrospective, demonstration.

Preparation: The preparation phase takes the first 2-3 meetings of the semester. The project leader, technical leader and subgroup leaders discuss the main research and development directions of the semester. The new team members get to know the system, the other developers indicate what specific tasks they would like to take. The Scrum master presents the values of the group's operation and the methods that the group will use during the semester to support putting these values into reality (Example: Values: effective team discussions, sustaining motivation, transfer of knowledge; One

method used to support these values: weekly Scrum) {1, 2, 3, 4, 5, 6, 8}

Weekly routine: After the preparation phase come the weekly meetings with the following structure: short news, weekly Scrum, topic of the week, discussion with the whole team, discussion in small groups.

In the *news section*, after the greeting of team members, the Scrum master announces to the community if someone is going to miss the meeting or is going to be late; gives a brief reflection on the weekly reports submitted by the team members; notifies the group about the short news related to the R&D lab (For example: “We were invited to attend the University Open Day”). The *news section* grabs the focus of the team members and is an effective way to spread out the most important information regarding the team. {3, 6, 7, 10}

The *weekly Scrum* is a ritual with a timebox of 15 minutes. Technical leader, subgroup leaders and developers are required to participate. Each of them answers these three questions briefly: “What have I done since the last meeting?”; “What do I plan to do until the next meeting?”; “Is there anything that blocks me?”. There is a temptation to go into technical details that are important to one or two team members. It is important to avoid this, because if the other members aren't involved in that subject, they will become very bored and demotivated. If an important topic arises during the weekly Scrum the key words of that topic are recorded by the Scrum master and put on the table after the *weekly Scrum*, in the second half of the *weekly routine (discussion with the whole team, discussion in small groups)*. However, the weekly Scrum can include sharing of short tips with each other, as everyone is attentive to it. A positive side effect of the *weekly Scrum* is that as the team members speak out loudly about their rhythm of progress, they motivate themselves and each other as well. {1, 4, 6, 7}

The *topic of the week* is a 10-15 minutes period during which the Scrum master presents and trains a value of the team's operation. During the first half of 2018, there were topics such as: identifying individual motivation; identifying a common goal; understanding shared responsibility; formulating weekly reports in an effective way; estimating work left; etc. Discussion of such topics help new and old members to learn the values and methods that enable them to realize a true teamwork rather than working side by side. {1, 3, 4, 5, 6, 7, 10}

The *discussion with the whole team* and *discussion in small groups* phase provides an opportunity to discuss issues that have arisen during the weekly Scrum, but their extraction has been postponed to effectively assess the overall group situation anteriorly. For example, if a more serious technical decision must be made (that affects the overall structure of the product), the most involved members will present the question to be discussed and then everyone will argue about it. More of this kind of questions can come up at a weekly meeting, or not even one. If only a few people, or perhaps only one developer-and-subgroup leader-pair are involved in a question, then they separate and discuss it between themselves. An example of this kind of “pair cases” is when a developer gets stuck in his

own job and asks the subgroup leader for help; or is about to finish his/her task and asks for a new one. The *discussion section* provides an excellent opportunity for team members to share their experiences and develop their knowledge of a topic. {1, 2, 3, 4, 5, 6, 7, 8}

Retrospective: During the *retrospective* the team members can reflect on the quality of the team-work and formulate directions and methods that are likely to have a positive impact on it in the next iterations. In the first half of 2018, we held a retrospective during the seventh meeting of the semester. The retrospective was coordinated by the Scrum master, who handed out different-colored, small-sized paper sheets to each team member. One color stand for positive, the other color for negative feedback. On each sheet of paper, a key word or thought could be written by the team members in an anonymous way. The recollected sheets of paper were organized along the “negative / positive” and “technique / methodology / team” axes by the Scrum master on the board. She placed on top of each other the sheets of paper representing the same idea. The heaps appearing in this way warned: it would be worthwhile to deal with those feedbacks urgently, since they affect many team members. After reviewing all the feedbacks, a discussion took place during which the team formulated ideas for improving their team-work. A concrete example: more people have indicated that they perceive some administrative tasks as redundant - writing weekly reports, using GitHub [14], and participating on the weekly Scrum. During the meeting, we managed to clarify that each of them has separate goals. {1, 3, 4, 5, 6, 8}

Demonstration: The *demonstration* is the closing event of the semester, during which all members of the research and development lab summarize their work realized during the semester and present their results. Guests may also be invited to attend the event. The structure of the meeting is: the project leader presents the main research and development directions emphasized during the semester, the Scrum master gives information about the methodology used to organize the team work, and the developers, together with the subgroup leaders, demonstrate the implemented new functions. On July 2, 2018, this year's first demonstration took place, which was also a demonstration of the txtUML 0.7.0 release. On this occasion, old team members also joined us. Cake and champagne enhanced to the festive atmosphere, which aimed to emphasize the common success as motivation for upcoming team work. {1, 2, 3, 4, 5, 6, 10}

3) *Artifacts:* The primary product of the group's work is the developed code base, which is available on gitHub [14].

On the same website, you can find the tasks related to the software product, which can be considered as a *product backlog (under the issues tab)*. Developers, subgroup leaders, and technical leaders can add or select from these - of course, according to the priorities fixed by the Project leader. {1, 2, 6, 8, 9, 10}

Each of the subgroups has an *issue board* (also on the above-mentioned website, under the projects tab) that displays the current tasks of that subgroup. They include columns

like: 'to do'; 'in development'; 'under testing'; 'pull request'; 'done'. The board therefore provides a comprehensive picture of the current status of the subgroup's work. {1, 6, 7, 8, 9, 10}

In addition, there is an internal website, a form, used to collect information about the weekly progress of the members in a background table. It is called *weekly report*. The following data can be given on its surface regarding to a task: contributors name; type of the task (for example: research, development, mentoring, administration, etc.); a brief description of the task; a detailed description of the task (if the short description is not sufficient); associated GitHub task code (if any); lesson to be learned (if any). Developers must fill out the form every week before the meeting, for every task they worked on since the last meeting. In connection with the *weekly report*, the members of the team repeatedly expressed their disapproval. Nonetheless, we kept the weekly report because of its positive effect: since it precedes the *weekly Scrum*, it helps the group members to give a more focused report during the *weekly Scrum*; produces data series suitable for research and application accounting; can help in catching up for the team members that will join the team later on; collects data for self-reflection, logs personal performance; is a practice that develops the skills needed for group work (collaboration, organization, communication) and adapts to the administrative requirements of a general workplace. {1, 2, 5, 6, 7, 8, 9, 10}

4) *Bonus: Agile training*: In the second semester of 2018 I myself, as the agile coach of the team, organized an extra team event, called Agile training. The purpose of this was to consolidate agile values, strengthen team spirit, invent together new methods that improve team work.

The topics of the agile training were: getting know each other better - build trust, build relationships; clarifying the purpose and responsibilities of the different team roles; (re)defining the values of our team - taking inspiration from Scrum values and Google's research about effective team work; identifying the main motivation of each team member and the whole team (for that specific semester, regarding our research).

C. Results and feedback

During the first semester of 2018, the methodology described above was applied for 18 weeks. Our observations were: weekly meetings were more efficient and dynamic than they were before; the number of communication interactions among team members has increased; the consciousness of group responsibility began to develop; regular administration of the tasks began to become accepted; individual and team goals related to the R&D lab have become clear.

At the review meeting (7th week), team members gave similar feedback: Teamwork was highlighted as a positive experience, even the collaboration between the subgroups. Shorter and useful meetings have been acknowledged. They were very positive about the senior students's competence, which, combined with the good meeting structure provided many learning opportunities for the new team members. The methodology used to provide the transfer of knowledge through group structure (hierarchical roles) and weekly routine

(mainly weekly Scrum and the discussion phase). Satisfaction with the effectiveness of the communication was also highlighted by more people. Three of the students reported that the collection and clear formulation of goals had a positive effect on them.

Negative feedback came primarily regarding the weekly report, which was indicated to be redundant. There was also a complaint regarding "over-formalizing the process". Our reaction was to give more time for these elements to prove their positive impact. If later reviews also highlight their demotivating effect, they can be omitted or modified following a common decision.

At week 17, team members filled out questionnaires regarding the whole semester. The questionnaires were filled by 7 people, i.e. 77% of the team. Some questions asked for a number as an answer, from a scale of 1 to 5, where "1" meant "the least", and "5" meant "the most". The question most relevant from the perspective of the methodology was: "To what extent do you think the value achieved during the semester is related to the project's management?" The average of the responses was 3.71. Another interesting question was "In what a measure did the values of the agile methodology prevail during the team work?" The average was a bit lower: 3.41. We don't consider these values very low, but we want to achieve a higher rating.

There were other aspects as well for that the rates received seemed to be considerably high. These could reflect the effect of the new agile methodology. These results are summarized in triplets below (question - average - possible explanation).

- To what extent was the task realized by you appropriate for you? - 4.28 - The *discussion in small groups* and the mentoring role of the subgroup-leaders helped team members to choose and succeed with a task appropriate for them.
- How satisfying was the number and quality of feedback you received regarding your work? - 4.57 - The *weekly Scrum*, the *discussion with the whole team* and *discussion in small groups*, the mentoring attitude of the subgroup leaders and the technical leader, provided space and possibility for giving individuals feedback.
- To what measure did the group work as a team? - 4.14 - The common reception of *news* at the beginning of the weekly meetings (for example, a joint congratulation to a team member success), the involvement of everyone in the *weekly Scrum*, the *topic of the week* helped to strengthen team spirit. The mentoring attitude of the technical leader and the subgroup leaders could also contribute to a sense of belonging to the team.
- How much did you enjoy being part of the team? - 4.71 - The *weekly routine* and the proper design of the roles helped to give every individual in the team enough attention and to work smoothly together.

According to some students, the best experiences were: "teamwork, development, encouraging each other"; "We have worked together on an interesting task." Concerning the "biggest challenges", we have noticed that team members who

had development and mentoring tasks as well struggled to allocate time for both of them. In the category of “What would you change?”, the fear of losing senior students (who complete their university studies) appeared, which was a warning to us that we should pay more attention to the transferring of the project knowledge. It appeared a need for deeper understanding of the methodology. (As a response to this need the above presented Agile training was introduced in the second semester of 2018, presented in chapter 5.)

For the results, please note that the research is still in an initial phase. The members of the group filled out the questionnaires, but since there are few of them, the feedback comes from a small number of people. Another lack of the research is, that team speed is not measured yet. Further iterations of the research are expected to improve regarding this aspects.

V. SUMMARY

In order to manage the work of a software engineering research and development team operating at the university, we can get inspired by the agile methods used in the software development industry. There are some examples of how their use in a classical software development course has succeeded. However, if we want to use them in a research and development team, we have to keep in mind some characteristics like: the objectives of the research and development team, proportion of research and development tasks, type of members (student, teacher, researcher), documentation obligations, etc. A list of this aspects to be analyzed while creating a specific agile method for a R&D team is a result of this research.

Along these aspects, we developed an agile methodology for the txtUML research and development team. The *project leader's* guiding role assures well defined directions regarding the research, while the *Scrum master* assists for teamwork to run smoothly. *Technical leaders - subgroup leaders - developers* create a hierarchical chain targeting the efficient flow of the knowledge. Preparations, news section, topic of the week, and retrospective events are the key to raise awareness of team work values. *Weekly Scrum, product backlog, tables, and weekly reports* aim to enhance transparency and thus efficiency and motivation. The *discussion with the whole team* and *discussion in small groups* events assure knowledge transfer.

Regarding the usage of txtUML's agile methodology the feedback highlights: the communication between the team members has become more efficient; goals are clearer; the atmosphere is pleasant; teamwork is more effective; new members develop rapidly through the flow of the project knowledge. Feedback also highlighted further development opportunities, such as the introduction of on agile training for deepening agile values. All in all, we are satisfied with our methodology and we want to continue to “contribute to creating values for a group of people with passion and creativity, with the help of agile leadership” [8].

Mike Cohn, the famous Scrum trainer, the co-founder of the Scrum Alliance states “I hope we see an end to methodology

wars; Scrum vs. Kanban, SAFe vs. LeSS, Disciplined Agile, Enterprise Agile and every other scaling framework. Instead of arguing about methodologies, we need to focus more on agile as a large set of practices, some of which work well in combination.” [9]. We think that the approach presented in this article is a good example of how we can develop a suitable agile methodology for a team with specific characteristics by combining agile elements creatively, based on the analyses of some previously fixed aspects.

ACKNOWLEDGMENT

I am grateful to all members of the txtUML R&D team who have welcomed all my ideas openly and helped to develop our own agile method.

The research project was supported by the European Union and co-financed by the European Social Fund (EFOP-3.6.3-VEKOP-16-2017-00002).

REFERENCES

- [1] H. Sharp, L. Plonkas, K. Taylor and P. Gregory, “Overcoming challenges in collaboration between research and practice: the agile research network” *SER&IPs'14*, Hyderabad, India, 2014, pp. 10–13. doi:10.1145/2593850.2593859
- [2] L. Barroca, H. Sharp, D. Salah, K. Taylor and P. Gregory, “Bridging the gap between research and agile practice: an evolutionary model” *Springer*, 2015. doi: 10.1007/s13198-015-0355-5
- [3] M. Marchesi, K. Mannaro, S. Uras and M. Locci, “Distributed Scrum in Research Project Management” *Agile Processes in Software Engineering and Extreme Programming, 8th International Conference*, Como, Italy, 2007. doi: 10.1007/978-3-540-73101-6_45
- [4] M. Hicks and J. S. Foster, “Adapting Scrum to Managing a Research Group” *Department of Computer Science Technical Report #CS-TR-4966*, 2010.
- [5] A. Klaus-Rosinska, J. Schneider and Vivian Bull, “Research Project Planning Based on SCRUM Framework and Type-2 Fuzzy Numbers” *ISAT 2018*, Nysa, Poland, 2018, pp. 381–391. doi: 10.1007/978-3-319-99993-7_34
- [6] I. R. Lima, T. de C. Freire and H. A. X. Costa, “Adapting and Using Scrum in a Software Research and Development Laboratory” *Salesian Journal on Information Systems*, vol. 9, 2012, pp. 16–23.
- [7] T. Dingsøyr, T. Dybå and P. Abrahamsson, “A Preliminary Roadmap for Empirical Research on Agile Software Development” *AGILE 2008*, Toronto, Canada, 2008. doi: 10.1109/Agile.2008.50
- [8] K. Graßer and R. Freisler, “Agil und erfolgreich führen” *managerSeminare Verlags GmbH*, 2017.
- [9] A. Przybyłek, M. Olszewski, “Adopting collaborative games into Open Kanban” *Federated Conference on Computer Science and Information Systems (FedCSIS'16)*, Gdansk, Poland, 2016. doi: 10.15439/2016F509
- [10] A. Przybyłek, D. Kotecka, “Making agile retrospectives more awesome” *Federated Conference on Computer Science and Information Systems (FedCSIS'17)*, Prague, Czech Republic, 2017. doi: 10.15439/2017F423
- [11] A. Przybyłek, M. Zakrzewski, “Adopting Collaborative Games into Agile Requirements Engineering” *13th International Conference on Evaluation of Novel Approaches to Software Engineering (ENASE'18)*, Funchal, Madeira, Portugal, 2018. doi: 10.5220/0006681900540064
- [12] Website of txtUML project, <http://txtuml.inf.elte.hu/wiki/doku.php>
- [13] Website of Slack communication tool, <https://slack.com/intl/en-hu/>
- [14] txtUML project on gitHub, <https://github.com/ELTE-Soft/txtUML/>
- [15] Quora, <https://www.quora.com/Do-you-know-of-research-teams-adapting-Agile-frameworks-scrum-kanban-XP-etc-and-or-Design-Thinking-techniques-for-managing-research-projects>
- [16] ProjectManagement.com: [Applying-Scrum-to-Research-Projects](https://www.projectmanagement.com/blog-post/43810/Applying-Scrum-to-Research-Projects), <https://www.projectmanagement.com/blog-post/43810/Applying-Scrum-to-Research-Projects>
- [17] ProjectManagement.com: [What-should-Scrum-look-like-in-2019-](https://www.projectmanagement.com/blog-post/50228/What-should-Scrum-look-like-in-2019-), <https://www.projectmanagement.com/blog-post/50228/What-should-Scrum-look-like-in-2019->